2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

An Iterative Approach for Heterogeneous Multi-Agent Route Planning with Temporal Logic Goals and Travel Duration Uncertainty

Kaier Liang, Gustavo A. Cardona, and Cristian-Ioan Vasile

***Abstract*— This paper introduces an iterative approach to multi-agent route planning under chance constraints. A heterogeneous team of agents with various capabilities is tasked with a Capability Temporal Logic (CaTL) mission, a fragment of Signal Temporal Logic. The agents' motion is modeled as a finite weighted graph, where the weights represent travel durations. Given the probability distribution over the durations of each edge's traversal, we want to find paths for all agents such that (a) the specification robustness is maximized, (b) travel time is minimized, and (c) the success probability is maximized. We tackle the problem using an iterative approach. In each stage, it selects edges' traversal duration and success probabilities and then solves a multi-agent route planning problem. We use an efficient Mixed-Integer Linear Programming (MILP) encoding for the latter. Our method provides a framework for agents to make informed decisions in choosing the most suitable edge attributes (travel durations and success probabilities) that consider agents' capabilities to perform tasks in the environment. The proposed iterative method leverages graph structure to generate a more efficient search space. The effectiveness of our method is demonstrated through simulated case studies where obtaining the optimal solution would otherwise be computationally expensive. Our approach efficiently explores the solution space, generating better solutions and improving the performance of multi-agent route planning with uncertain travel durations.**

I. Introduction

In recent years, heterogeneous multi-robot systems have gained significant interest. These systems capitalize on the combined capabilities of multiple robots that can communicate and collaborate to carry out complex tasks. Frequently, such tasks are spatially and temporally sensitive, encompassing activities like search and rescue, automated manufacturing, and precision agriculture [1]–[4].

Meanwhile, there have been increased applications for using temporal logic in robotics systems such as for robot path planning, control synthesis, and dynamic systems [5]–[12]. Linear Temporal Logic (LTL) can specify untimed tasks such as "moving to location A after reaching location B" [13]. For time-sensitive tasks, Signal Temporal Logic (STL) can provide reasoning for time specifications such as "Remaining in a region for 5 hours" [14], [15]. Additionally, Capability Temporal Logic (CaTL), a fragment of STL, can accommodate the spatial and temporal constraints for agents with different capabilities [16], [17].

In [17], the author employed CaTL to consider route planning for heterogeneous multi-agent systems with a probability of success traversal. The paper addressed environments with pre-defined (fixed) travel durations and success probability. However, in practice, the tasks that robots work on are often uncertain, not only from external factors such as weather but also from internal factors such as robot maneuverability. For example, an autonomous drone can change the motors' speed to adjust the velocity, while this speed adjustment can influence the traversal duration and the traversal success probability. All these usually unconsidered factors from low-level controllers can affect the tasks' satisfaction.

Moreover, depending on the system, the traversal duration time and success probability can be opposing factors. For instance, proceeding too aggressively can cause errors such as actuator saturation or inaccuracies that lead the system to an unsuccessful travel. Therefore, considering these two opposing quantities together and making the most suitable decisions can be computationally expensive when the number of decisions and options becomes intractable. In contrast, we propose a route planning problem with uncertain travel duration time. We consider that instead of being constant values, travel duration distributions are given, and bounds need to be synthesized that ensure good performance and success probabilities. To address this problem, we formulate it as a bi-level optimization problem and solve it using an iterative method to obtain the traversal scheme for agents. The iterative optimization method is often used to solve complex problems with large search spaces and non-linearity [18], [19]. The iterative method offers several advantages, including adaptability to different problems and scalability for convergence. In this paper, we adopt the simulated annealing algorithm as the framework [20], leveraging environmental information to refine the iterations and improve the search.

The main contributions of this work are, 1) We propose a heterogeneous multi-agent route planning problem to accommodate uncertainty for travel durations. 2) We pose a bi-level optimization model that overcomes the non-linearities generated by considering time durations of transitions as stochastic variables to optimize. 3) We propose an iterative algorithm approach to solve the bi-level optimization problem, where we optimize for the best selection of probability of success and traversal durations to maximize the satisfaction of the specification. 4) We accelerate the iterative method based on a) the simulated annealing algorithm that reduces the computation expense to find a sub-optimal solution compared to brute force search; b) We exploit the edge correlations based on the graph structure to guide and improve the iteration search space. 5) We show empirically that our iteration method can significantly increase the efficiency of the maximum iteration value and iteration steps.

II. Preliminaries and Notation

In this section, we introduce some of the notation used in the paper and define the environment and robot models in

Kaier Liang, Gustavo A. Cardona, and Cristian-Ioan Vasile are with the Mechanical Engineering and Mechanics Department at Lehigh University, PA, USA: {kal221, gcardona, cvasile}@lehigh.edu

the context of the mission specification framework Capability Temporal Logic (CaTL) [16].

Notation: Let $\mathbb{Z}$, $\mathbb{R}$, and $\mathbb{B}$ denote the sets of integer, real, and binary numbers sets, respectively. The set of integers and real numbers greater than a are $\mathbb{Z}_{\geq a}$ and $\mathbb{R}_{\geq a}$, respectively. For a set $\mathcal{S}$, $2^{\mathcal{S}}$ and $|\mathcal{S}|$ represent its power set and cardinality. We denote $x \sim \mathcal{X}$ to mean that the random variable x is sampled from the probability distribution $\mathcal{X}$. For $\mathcal{S} \subseteq \mathbb{R}$ and $\alpha \in \mathbb{R}$, we denote by $\alpha + \mathcal{S}$ the set $\{\alpha + x \mid x \in \mathcal{S}\}$. The empty set is denoted by $\varnothing$. Let $[a..b] = \mathbb{Z} \cap [a, b]$ denote the range of integers between a and b (inclusive). For a range $I = [a..b]$, we use $\underline{I} = a$ and $\bar{I} = b$. $\mathbf{1}_{(i,j)} \in \{0, 1\}$ is the indicator function which takes 0 if $i = j$ and 1 otherwise.

Environment and robot models: Consider a group of heterogeneous robots $\mathcal{J}$ operating within an environment abstracted as a transition system $Env = (\mathcal{Q}, \mathcal{E}, \mathcal{W}^{det}, AP, \mathcal{L})$, where $\mathcal{Q}$ is the set of states q representing locations of interest in the environment, $\mathcal{E} \subseteq \mathcal{Q} \times \mathcal{Q}$ captures the set of transitions between locations, $\mathcal{W}^{det} : \mathcal{E} \to \mathbb{Z}_{\geq 1}$ captures the mapping from each transition to its travel duration as a positive integer based on a discretization of time $\Delta t > 0$. A stationary robot is represented by $(q, q) \in \mathcal{E}$ with $\mathcal{W}^{det}(q, q) = 1$. We consider states to be labeled with atomic propositions $\pi \in AP$ given by the map $\mathcal{L} : \mathcal{Q} \to AP$, where $\mathcal{L}^{-1}(\pi) = \{q \mid \mathcal{L}(q) = \pi\}$ denotes the set of all regions labeled with $\pi \in AP$.

We consider a heterogeneous team of robots $\mathcal{J}$ with capabilities from a finite set $\mathcal{C}$. The capability set of robot $j \in \mathcal{J}$ is $c_j \subseteq \mathcal{C}$, and defines the *robot's class* $g = c_j$. The set of all robot classes is $\mathcal{G} \subseteq 2^{\mathcal{C}}$. Each robot $j \in \mathcal{J}$ is characterized by an initial state $q_{0,j} \in \mathcal{Q}$, and capability set c_j, defining its robot class. An input signal for a robot j is a mapping $u_j : \mathbb{Z}_{\geq 0} \to \mathcal{E} \cup \{\varnothing\}$ where $u_j(t) = e$ indicates that robot $j \in \mathcal{J}$ starts traversing edge $e \in \mathcal{E}$ at time t. Traveling along e, implies that the value of $u_j(t) = \varnothing$ for the duration $\mathcal{W}(e)$. Note that $u_j(t)$ induces a trajectory of robot j in the environment Env captured via mapping function $s_j : \mathbb{Z}_{\geq 0} \to \mathcal{Q} \cup \mathcal{E}$ such that $s_j(t)$ represents robot j being at either state $q \in \mathcal{Q}$ or traversing edge $e \in \mathcal{E}$ at time $t \in \mathbb{Z}_{\geq 0}$, such that $s_j(0) = q_{0,j}$. The duration of a trajectory s_j of agent j is the sum of all its transitions' duration except self-loops. Formally, we have $\mathcal{W}^{det}(s_j) = \sum_{t=0}^{H-1} \mathcal{W}^{det}(u_j(t)) \cdot \mathbf{1}_{(u_j(t),(q,q))}$, where $H \in \mathbb{Z}_{\geq 1}$ is a given time horizon, and $\mathcal{W}(\varnothing) = 0$ by convention. The synchronous trajectory of the team of robots $\mathcal{J}$ is $s_{\mathcal{J}} : \mathbb{Z}_{\geq 0} \to (\mathcal{Q} \cup \mathcal{E})^{|\mathcal{J}|}$. The number of robots at state $q \in \mathcal{Q}$ with capability $c \in \mathcal{C}$ at time $t \in \mathbb{Z}_{\geq 0}$ is captured by the variable $n_{q,c}(t) = |\{j \in \mathcal{J} \mid q = s_j(t), c \in c_j\}|$. The total travel time of the team trajectory $s_{\mathcal{J}}$ for all agents is $\mathcal{W}^{det}(s_{\mathcal{J}}) = \sum_{j \in \mathcal{J}} \mathcal{W}^{det}(s_j)$.

Capability Temporal Logic [16]: Here, we define the syntax and semantics of CaTL, a language for a heterogeneous team of robots satisfying specified tasks.

Definition 1. *A* task *is a tuple $T = (d, \pi, cp)$, where $d \in \mathbb{Z}_{\geq 1}$ is a discrete duration of time (i.e., multiple time steps Δt), $\pi \in AP$ is an atomic proposition, $cp : \mathcal{C} \to \mathbb{Z}_{\geq 0}$ is a counting map specifying how many robots with each capability should be in each region labeled π, respectively.*

Definition 2. *The* syntax *of CaTL [16] is a fragment of Signal Temporal Logic (STL) [21]. The CaTL syntax is*

$$\phi ::= T \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \phi_1 \mathcal{U}_I \phi_2 \mid \Diamond_I \phi \mid \Box_I \phi$$

where ϕ is a CaTL formula, T is a task, $\wedge$ and $\vee$ are the Boolean conjunction and disjunction operators, $\mathcal{U}_I$, $\Diamond_I$, and $\Box_I$ are the time-bounded until, eventually, and always operators, respectively, with $I = [a \,..\, b]$ a time interval.

As CaTL is a fragment of STL, it also has qualitative and quantitative semantics recursively defined as follows.

Definition 3 (CaTL qualitative semantics)**.** *The Boolean (qualitative) semantics of CaTL are defined over trajectories $s_{\mathcal{J}}$ of robots. At time t, satisfaction of a task T is defined by $(s_{\mathcal{J}}, t) \vDash T \Leftrightarrow n_{q,c}(t') \geq cp(c)$, for all $t' \in [t \,..\, t + d], q \in \mathcal{L}^{-1}(\pi)$, and $c \in cp$. Boolean and temporal operators' semantics are the same as for STL [21]. A team trajectory satisfies a formula ϕ, denoted $(s_{\mathcal{J}}, t) \vDash \phi$, if $(s_{\mathcal{J}}, 0) \vDash \phi$.*

Note that the qualitative semantics captures the conditions on the timed trajectory $(s_{\mathcal{J}}, t)$ to satisfy a CaTL specification ϕ, over tasks T, Boolean and temporal operators. Similarly to STL [21], where the specification's margin of satisfaction or violation is defined recursively over the formula's structure (quantitative semantics). CaTL also has a quantitative semantics called robot availability robustness. It captures the surplus of robots needed to minimally satisfy or violate the task over its duration, the capabilities involved, and the locations required.

Definition 4 (Robot Availability Robustness)**.** *The robot availability robustness of a task is recursively defined as*

$$\begin{aligned}
\rho(s_{\mathcal{J}}, T, t) &= \min_{c \in cp_T} \min_{t' \in [t..t+d]} \min_{q \in \mathcal{L}^{-1}(\pi)} n_{q,c}(t') - cp(c), \\
\rho(s_{\mathcal{J}}, \phi_1 \wedge \phi_2, t) &= \min(\rho(s_{\mathcal{J}}, \phi_1, t), \rho(s_{\mathcal{J}}, \phi_2, t)), \\
\rho(s_{\mathcal{J}}, \phi_1 \vee \phi_2, t) &= \max(\rho(s_{\mathcal{J}}, \phi_1, t), \rho(s_{\mathcal{J}}, \phi_2, t)), \\
\rho(s_{\mathcal{J}}, \Box_I \phi, t) &= \min_{t' \in t+I} \rho(s_{\mathcal{J}}, \phi, t'), \\
\rho(s_{\mathcal{J}}, \Diamond_I \phi, t) &= \max_{t' \in t+I} \rho(s_{\mathcal{J}}, \phi, t'), \\
\rho(s_{\mathcal{J}}, \phi_1 \mathcal{U}_I \phi_2, t) &= \max_{t' \in t+I} \Big\{ \min\{\rho(s_{\mathcal{J}}, \phi_2, t'), \\
&\qquad \min_{t'' \in [t..t']} \rho(s_{\mathcal{J}}, \phi_1, t'')\}\Big\}.
\end{aligned} \tag{1}$$

The time horizon of a CaTL formula is defined as in [22]

$$\|\phi\| = \begin{cases} d, & \text{if } \phi = T, \\ \max\{\|\phi_1\|, \|\phi_2\|\}, & \text{if } \phi = \phi_1 \wedge \phi_2, \phi_1 \vee \phi_2 \\ \|\phi\| + \bar{I}, & \text{if } \phi = \Box_I \phi, \Diamond_I \phi \\ \bar{I} + \max\{\|\phi_1\|, \|\phi_2\|\}, & \text{if } \phi = \phi_1 \mathcal{U}_I \phi_2. \end{cases} \tag{2}$$

III. Problem Formulation

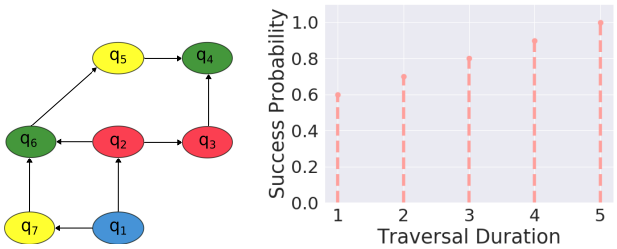
This section defines the heterogeneous multi-robot route planning problem with temporal logic goals and travel duration uncertainty. The notion of the travel duration uncertainty comes from a more realistic scenario where robots can adjust the low-level control inputs to change their moving velocity, thus changing the traversal time in the environment. Traveling fast or aggressively may impair the success probabilities and cause a rushing through a task, leading to errors. Conversely, traveling too conservatively might be overly cautious, thus resulting in slow task completion.

In [17], the authors consider the probability of success of agents to execute their routes. They introduced a chance

constraint in their problem with pre-defined (fixed) travel durations $\mathcal{W}^{det}$. Under reasonable independence assumptions, the chance constraint over the routes was reduced to each edge's given (fixed) success probabilities. In contrast, we tackle the route planning problem with uncertain edge travel durations where duration distributions are provided and their duration bounds need to be synthesized.

Specifically, instead of considering deterministic duration times $\mathcal{W}^{det}: \mathcal{E} \to \mathbb{Z}_{>0}$, we consider uncertainty in the travel time $\mathcal{W}: \mathcal{E} \to \mathcal{D}(\mathbb{Z}_{>0})$, where $\mathcal{D}$ is the set of probability distributions over bounded support positive integer sets. The weight $\mathcal{W}(e)$ captures the distribution over the duration of time to traverse edge e. Given $w \in \mathbb{Z}_{\geq 0}$, the probability of successfully traversing the edge e within duration w is given by $p_e(w) = \sum_{t=0}^{w} \mathcal{W}(e)(t) = \sum_{t=\underline{w}}^{w} \mathcal{W}(e)(t)$, where the support of $\mathcal{W}(e)$ is $\text{supp}(\mathcal{W}(e)) = [\underline{w}, \overline{w}]$. The success probability $p_e(w)$ is the duration distribution's Cumulative Density Function (CDF). Thus, as the duration w increases, the success probability also increases. Conversely, the shorter we set the time duration w, the lower the probability the robots will be able to traverse the edge e by the duration w. Thus, we must balance travel durations versus traversal success probabilities. The following example motivates the problem we want to tackle in this paper.

Example 1. *Consider the environment as shown in Fig. 1a. A* CaTL *formula is tasked to a group of robots with different capabilities c_1 and c_2 start at state q_1, where $\phi_{task} = \Diamond_{[0,10)}(5, \pi_{green}, (c_1, 2)) \wedge \Diamond_{[0,10)}(3, \pi_{green}, (c_2, 3))$. In English, this formula states, "Within 0 to 10 hours" after deployment, each region labeled in* green *requires 2 robots with capability c_1 working there for 5 hours. And within 0 to 10 hours after deployment, each region labeled in* green *requires 3 robots with capability c_2 working there for 3 hours". The transition between each node is stochastic, with the correlation between the traversal durations and success probabilities shown in Fig. 1b. We aim to maximize task satisfaction while minimizing traversal durations and maximizing the success probability.*



(a) (b)

Fig. 1: Ex. 1 (a) The environment is represented as a transition system: Nodes are labeled with colors indicating different atomic propositions. Each edge is associated with a stochastic transition duration. (b) The probability distribution with traversal duration.

Problem 1. *Given a team of robots $\mathcal{J}$ deployed in environment Env, with initial states $q_{0,j}$, capabilities $c_j \subseteq \mathcal{C}$, and a CaTL mission specification ϕ. find trajectories s_j for robots such that $s_{\mathcal{J}} \vDash \phi$, and (1) maximize the robustness $\rho(s_{\mathcal{J}}, \phi, k)$, (2) minimize the bound on the travel time ϵ_τ (3) maximize the success probability p_s of team trajectory $s_{\mathcal{J}}$. Formally, we have a non-linear optimization problem*

$$\begin{aligned} \max_{s_{\mathcal{J}}, \epsilon_\tau, p_s} \quad & \rho(s_{\mathcal{J}}, \phi) - \gamma \cdot \epsilon_\tau + p_s, \\ \text{s.t. } & s_{\mathcal{J}} \vDash \phi, \mathbb{P}[w(s_{\mathcal{J}}) \leq \epsilon_\tau] \geq p_s, \\ & \textit{(robots' dynamics given by } Env\textit{)}. \end{aligned} \tag{3}$$

where $w(s_{\mathcal{J}})$ the random total traveling time, $\gamma = \frac{\beta_u}{K \cdot |\mathcal{J}|}$, $\beta_u \in (0,1)$, and $K = \|\phi\|$.

Assumption 1. *As in [17], we consider three independence assumptions. 1) Probability of success of one agent $j \in \mathcal{J}$ traversing edge $e \in \mathcal{E}$ is independent of other agents traversing the environment. 2) The probability of success of one edge $e \in \mathcal{E}$ is independent of other edges in the environment. 3) The probability of success of an agent $j \in \mathcal{J}$ traversing an edge $e \in \mathcal{E}$ at any time t is independent of the same agent traversing the same edge at any other time, in the future and the past.*

Assumption 2. *A robot stationary at a state $q \in \mathcal{Q}$ is captured as transition $(q, q) \in \mathcal{E}$, and its duration of time is assumed to be one for both the deterministic $\mathcal{W}^{det}(q, q) = 1$ and stochastic $\mathcal{W}(q, q) = 1$ cases.*

IV. Solution

The optimization in Pb. 1 is nonlinear and poses significant challenges in using direct methods to encode and solve it. This section reformulates Pb. 1 as a bi-level optimization problem via an approximation of the chance constraint. The inner optimization is defined as a route planning problem with deterministic duration $\mathcal{W}^{det}$ for a heterogeneous team of robots $\mathcal{J}$ satisfying a CaTL specification. It is modeled as a Mixed Integer Linear Programming (MILP) encoding [16]. The outer optimization focuses on determining the selection of travel durations $\mathcal{W}$. We propose an iterative method to solve the bi-level optimization problem that cycles between (1) edge weight selection and, thus, success probabilities, and (2) solving multi-agent route planning.

Reformulation of Problem 1: We reformulate Pb. 1 by transforming the chance constraint using the independence Assumptions 1 and introducing decision variables for edge traversal duration bounds. Let us consider the travel time of robots navigating in the environment

$$w(s_{\mathcal{J}}) = \sum_{j \in \mathcal{J}} w(s_j) = \sum_{j \in \mathcal{J}} \sum_{e \in \mathcal{E}(s_j)} w(e), \tag{4}$$

$$= \sum_{j \in \mathcal{J}} \sum_{e \in \mathcal{E}(s_j)} w(e) - \mathcal{W}^{det}(e) + \mathcal{W}^{det}(e) \tag{5}$$

$$= \sum_{j \in \mathcal{J}} \sum_{e \in \mathcal{E}(s_j)} \Big(w(e) - \mathcal{W}^{det}(e) \Big) + \mathcal{W}^{det}(s_{\mathcal{J}}), \tag{6}$$

where $w(e) \sim \mathcal{W}(e)$ is the random travel duration of edge e, $\mathcal{E}(s_j) = \{u_j(t) \mid u_j(t) \neq \varnothing\}$ is the set of all edges traversed along trajectory s_j of agent j induced by control u_j. The edge duration bounds $\mathcal{W}^{det}(e)$ are deterministic and represent additional decision variables we introduce for the problem reformulation. Thus, the chance constraint on the

success probability in Pb. 1 is restated as

$$\mathbb{P}[w(s_{\mathcal{J}}) \le \epsilon_\tau] \tag{7}$$

$$= \mathbb{P}\left[\sum_{j\in\mathcal{J}}\sum_{e\in\mathcal{E}(s_j)}\Big(w(e) - \mathcal{W}^{det}(e)\Big) \le (\epsilon_\tau - \mathcal{W}^{det}(s_{\mathcal{J}}))\right] \tag{8}$$

$$= \mathbb{P}\left[\sum_{j\in\mathcal{J}}\sum_{e\in\mathcal{E}(s_j)}\Big(w(e) - \mathcal{W}^{det}(e)\Big) \le 0\right] \tag{9}$$

$$\ge \mathbb{P}\left[\bigcap_{j\in\mathcal{J}}\bigcap_{e\in\mathcal{E}(s_j)}\Big(w(e) - \mathcal{W}^{det}(e) \le 0\Big)\right] = \Theta, \tag{10}$$

where we set $\epsilon_\tau = \mathcal{W}^{det}(s_{\mathcal{J}})$. The chance constraint in (8) becomes (9) due to the specific choice of weights such that the total travel time matches the ϵ_τ bound. Thus, we eliminate the ϵ_τ decision variable. From (9), we obtain (10) by observing that the event in (10) implies the event in (9). Lastly, we obtain the formula for the lower bound Θ using Assumption 1, $\Theta = \prod_{j\in\mathcal{J}}\prod_{e\in\mathcal{E}(s_j)}\mathbb{P}[w(e) \le \mathcal{W}^{det}(e)]$. Thus, Pb. 1 is reformulated as follows.

Problem 2 (Reformulation of Problem 1)**.** *Under the same setting considered in Pb. 1 and reformulation above, formally, we have the following bi-level optimization problem*

$$\begin{aligned}\max_{e\in\mathcal{E}:\mathcal{W}^{det}(e)} \quad & \Lambda \\ s.t. \quad & \mathcal{W}^{det}(e) \in \mathrm{supp}(\mathcal{W}(e)) \\ \Lambda = \max_{s_{\mathcal{J}},\epsilon_\tau,p_s} \quad & \rho(s_{\mathcal{J}},\phi) - \gamma\cdot\epsilon_\tau + p_s, \\ & s.t.\ s_{\mathcal{J}} \vDash \phi,\ \epsilon_\tau = \mathcal{W}^{det}(s_{\mathcal{J}}) \\ & \textit{robots dynamics},\ \Theta \ge p_s.\end{aligned} \tag{11}$$

MILP encoding: We introduce robot dynamics and CaTL specification MILP encoding adapted from [16]. Let us consider a discrete-time variable k. Additionally, the time horizon of the encoding K is defined via $K = \|\phi\|$ as in (2).

Robot' dynamics encoding: Let $z_{q,g,k} \in \mathbb{Z}_{\ge 0}$ captures the number of robots of class $g \in \mathcal{G}$ at time $k \in [0..K]$, at state $q \in \mathcal{Q}$, and $u_{e,g,k} \in \mathbb{Z}_{\ge 0}$ the number of robots of class $g \in \mathcal{G}$ entering edge $e \in \mathcal{E}$ at time k, for all $k \in [0..K]$. Then, the robots' dynamics are captured as follows

$$z_{q,g,0} = |\{j \in \mathcal{J} \mid q_{0,j} = q, c_j = g\}|, \tag{12}$$

$$z_{q,g,k} = \sum_{(q',q)\in\mathcal{E}} u_{(q',q),g,k-\mathcal{W}((q',q))}, \tag{13}$$

$$\sum_{(q,q')\in\mathcal{E}} u_{(q,q'),g,k} = \sum_{(q',q)\in\mathcal{E}} u_{(q',q),g,k-\mathcal{W}((q',q))}, \tag{14}$$

where (12), capture the initial robot deployment over the environment. We ensure that robots cannot be simultaneously at two places and edges $\mathcal{E}$. Then, (13) and (14) capture the robot deployment at every time k, keeping conservation of robots in the environment during the time horizon K. Conservation is achieved if the number of robots entering a node $(q',q) \in \mathcal{E}$ equals the number of robots going out $(q,q') \in \mathcal{E}$.

CaTL specification encoding: We encode the CaTL specification ϕ into the MILP and couple it with the robot dynamics. We introduce the variable $z_{\pi,q,c,k} \in \mathbb{R}_{\ge 0}$ and add the following constraints to ensure that capabilities are not counted more than once in all regions with atomic proposition π labeled with $\mathcal{L}(q)$, $\sum_{\pi\in\mathcal{L}(q)} z_{\pi,q,c,k} = \sum_{g:c\in G} z_{q,g,k}$, where $z_{\pi,q,c,k}$ capture the amount of robots with capability $c \in \mathcal{C}$ and at time step $k \in [0..K]$ at every state $q \in \mathcal{Q}$ using label $\pi \in AP$. We introduce variable $z_{\mu(\pi,c),k} \in \mathbb{R}$, where $\mu(\pi,c) = \min_{q\in\mathcal{L}^{-1}(\pi)}\{n_{q,c}\} \ge cp(c)$, guarantees that the minimum number of robots with capability $c \in \mathcal{C}$ at every state $q \in \mathcal{L}^{-1}(\pi)$ is greater or equal to the requested in the counting proposition of the task. Then, the following constraint guarantees that the total number of robots at state $q \in \mathcal{Q}$ with capability $c \in \mathcal{C}$ satisfying the counting proposition does not surpass the number in Env, $z_{\mu(\pi,c),k} \le z_{\pi,q,c,k}$, for all $q \in \mathcal{L}^{-1}(\pi)$, $c \in \mathcal{C}$, $\pi \in AP$, and $k \in [0..K]$.

Lastly, we transform our CaTL *task* $T = (d,\pi,cp)$ definition and transformed into its equivalent STL formula $\phi_T = \square_{[0,d]}\bigwedge_{c\in\mathcal{C}}\mu(\pi,c)$. We translate the CaTL specification into an STL specification and use existing MILP encoding for STL as authors in [23]–[25]. These encodings are based on the recursive definition of the robustness definition of the STL specification. When having the specification in an STL form, the soundness property also holds

Theorem 1 (Soundness [26])**.** *Let $s_{\mathcal{J}}$ be the robot trajectories and ϕ an STL formula. It holds $\rho(s_{\mathcal{J}},\phi,k) > 0 \Rightarrow (s_{\mathcal{J}},k) \vDash \phi$ for satisfaction and $\rho(s_{\mathcal{J}},\phi,k) < 0 \Rightarrow (s_{\mathcal{J}},k) \nvDash \phi$ for violation.*

Therefore, we conclude that maximizing robustness enhances mission satisfaction.

Travel time regularization: Maximizing the robustness $\rho(s_{\mathcal{J}},\phi,k)$ aims to satisfy the specification. However, specification satisfaction does not guarantee robots would take efficient routes while traveling, and even robots that are not required for satisfaction may navigate the environment unnecessarily. These spurious movements of robots and optimal trajectories can be enforced by adding a travel time regularization term defined as follows

$$\epsilon_\tau = \sum_{k=0}^{K}\sum_{g\in\mathcal{G}}\sum_{e=(q,q')\in\mathcal{E},q\neq q'} \mathcal{W}^{det}(e)\cdot u_{e,g,k}, \quad \gamma = \frac{\beta_u}{K\cdot|\mathcal{J}|},$$

where γ is a term that normalizes the travel time objective to guarantee it never goes above one, and $\beta_u \in (0,1)$. This to guarantee maximizing robustness has higher priority.

Chance constraint: We reformulate the chance constraint $\Theta \ge p_s$ in (11) as a linear constraint by taking the logarithm, the detailed proof can be found in [17].

$$\ell_s = \sum_{k=0}^{K}\sum_{g\in\mathcal{G}}\sum_{e=(q,q')\in\mathcal{E},q\neq q'} \ell^{det}(e)\cdot u_{e,g,k} \le \epsilon_s, \tag{15}$$

$$\ell^{det}(e) = -\log(p_e(\mathcal{W}^{det}(e))) > 0 \tag{16}$$

$$\epsilon_s = -\log(p_s) \tag{17}$$

where the new variable ϵ_s is used in the objective of MILP encoding for the inner optimization problem in (11).

Iterative method: The MILP formulation tackles task satisfaction for a deterministic traversal duration and success probability for the inner optimization. This section presents an iterative method to handle multiple selections for traversal durations and solve the outer optimization in Pb. 2.

As the number of edges and edge duration selection increases, the computational time grows exponentially. Specifically, for n edges, each edge $e \in \mathcal{E}$ is associated with m distinct edge duration options, and then the total number of

combinations amounts to m^n. For simplicity, we assume that all edges' distributions have the same number of elements in their support sets, i.e., $m = |\text{supp}(\mathcal{W}(e))|$ for all $e \in \mathcal{E}$.

In our model, the probability function $p_e(w)$ implies a longer traversal duration corresponding to a higher success probability. Since our objective involves both maximizing probabilities and minimizing travel duration, as shown in (11), this opposing nature of the objectives inevitably leads to a trade-off, and the problem can have several optimal values and different local optima. Many algorithms can be used to tackle such optimization, such as simulated annealing and genetic algorithms [20], [27]. These iterative algorithms repeatedly evaluate the objective function and update the solution until a stopping criterion is met.

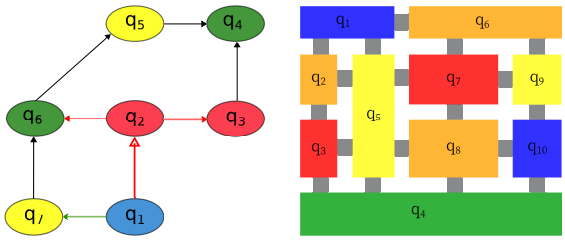
We adopt the simulated annealing algorithm as the framework shown in Alg. 1. The input is an initial solution guess corresponding to a vector of selected traversal durations for each edge from their associated distributions. Formally, $\boldsymbol{\eta} = [\eta_1, \eta_2, \cdots, \eta_n]$, where n represents the number of edge traversal durations to be determined, $\eta_i \in [0..m-1]$ denotes the selected traversal duration for edge e_i from its support set $\text{supp}(\mathcal{W}(e_i))$. Alg. 1 starts at a high temperature and decays at each iteration until the terminated condition is met. A new solution is generated at each iteration based on the previous solution in line 5. In the classic simulated annealing algorithm, the new solution is generated by randomly adjusting one of the elements from the original solution. Based on the current iteration temperature and the quality of the new objective value, the algorithm decides if the new solution is accepted in line 9. Then, the temperature decreases by a decaying coefficient $\alpha \in (0,1)$, and the best objective value is recorded at the end in lines 11-12.

Algorithm 1: Iterative approach

Input: $\boldsymbol{\eta}$ – initial solution, T_0 – initial temperature, T_{final} – final temperature
Output: C^* – best objective value

1 $T \leftarrow T_0$
2 $C \leftarrow MILP(\boldsymbol{\eta})$
3 $C^* \leftarrow C$
4 **while** $T > T_{final}$ **do**
5 $\quad \boldsymbol{\eta}' \leftarrow GeneratingNew(\boldsymbol{\eta})$
6 $\quad C' \leftarrow MILP(\boldsymbol{\eta}')$
7 $\quad \delta \leftarrow C' - C$
8 $\quad r \leftarrow random()$
9 $\quad$ **if** $r < \exp(\delta/T)$ *or* $\delta > 0$ **then** $\boldsymbol{\eta} \leftarrow \boldsymbol{\eta}'$
$\quad$ `// Reduce Temperature`
10 $\quad C \leftarrow C'$
11 $\quad T \leftarrow \alpha * T$
12 $\quad$ **if** $\delta > 0$ **then** $C^* \leftarrow C'$
13 **return** C^*

Iterative algorithms are generally applicable to a wide range of problems with good scalability. However, they can be improved by incorporating problem-specific information to guide the new iteration solution (line 5). For graph search scenarios, we use the graph's structure efficiently to update the solution at each iteration, as shown in Fig. 2a. Suppose we select the edge (q_1, q_2) to test one of the attributes, denoted in red, which can represent fast movement (small traversal duration). In contrast, slow movement is denoted in green. Since edges (q_2, q_3) and (q_2, q_6) are consecutive to the chosen edge (q_1, q_2), and considering that consecutive edges in graph search scenarios are often traversed together, agents are more likely to choose the same manner (fast motion) for these consecutive edges. Therefore, we also modify these consecutive edges in the same red manner, indicating fast motion. Furthermore, edges originating from or leading to the same source node can serve as alternatives to each other, such as the edge (q_1, q_7). By assigning contrasting manner (slow motion) to these alternative edges, agents can explore a more extensive search space by experiencing two opposing motion selections rather than having two alternative paths with identical or similar movement schemes.



(a) (b)

Fig. 2: (a) Example of adjustment scheme . (b) Environment Map: colors indicating different atomic propositions.

The procedure for generating a new solution is shown in Alg. 2. Without loss of generality, we assume the edge duration values $\text{supp}(\mathcal{W}(e))$ are in ascending order. In line 2, an edge is chosen for adjustment, and the consecutive and alternative edge groups are filtered in lines 3-4. The *Consecutive* edge set for edge (q, q') is defined as $\mathcal{E}_p = \{(q_+, q_-) \mid (q_+, q) \in \mathcal{E} \vee (q', q_-) \in \mathcal{E}\}$. The *Alternative* edge set is defined as $\mathcal{E}_n = \{(q_+, q_-) \mid (q_+, q') \in \mathcal{E} \vee (q, q_-) \in \mathcal{E}\}$. The heuristic adjustment will reduce the search space. Therefore, we use a probability ξ to determine whether this adjustment is applied in lines 8 and 11. Lastly, the new solution is updated in line 13 with the corresponding changes for the next iteration.

V. Computation Results

This section demonstrates our iterative algorithm's performance and computational effectiveness. All computations of the following case studies were performed on a PC with 20 cores at 3.7 GHz with 64 GB of RAM. We used Gurobi [28] as the MILP solver. For encoding of CaTL specifications we used PyTeLo [29] and ANTLRv4 [30], LOMAP [31] and networkx [32] for transition system models of environments.

Simulation specification: The simulated environment is shown in Fig. 2b, the set of labels is $\{\pi_{\text{blue}}, \pi_{\text{orange}}, \pi_{\text{yellow}}, \pi_{\text{red}}\}$. The set of capabilities is $\mathcal{C} = \{\text{IR}, \text{Vis}, \text{UV}, \text{Mo}\}$. The CaTL formula is $\phi = \Diamond_{[0,20]} T_1 \wedge \Box_{[20,40]} T_2 \wedge \Diamond_{[5,25]} T_3 \wedge \Diamond_{[3,15]} T_4 \wedge \Diamond_{[20,30]} T_5 \wedge \Box_{[3,12]} T_6 \wedge \Diamond_{[2,10]} T_7$, where $T_1 = \big(1, \pi_{\text{green}}, \{(\text{IR}, 2), (\text{Vis}, 2)\}\big)$,
$T_2 = \big(1, \pi_{\text{blue}}, \{(\text{UV}, 1), (\text{Mo}, 2)\}\big)$,
$T_3 = \big(2, \pi_{\text{yellow}}, \{(\text{UV}, 2), (\text{Vis}, 2)\}\big)$,
$T_4 = T_5 = \big(2, \pi_{\text{orange}}, \{(\text{Vis}, 2)\}\big)$,
$T_6 = \big(2, \pi_{\text{green}}, \{(\text{IR}, 2), (\text{Vis}, 3)\}\big)$,
$T_7 = \big(3, \pi_{\text{yellow}}, \{(\text{IR}, 3), (\text{Vis}, 2), (\text{UV}, 3), (\text{Mo}, 4)\}\big)$.

The following 6 edges are selected with stochastic duration: $\{(q_1, q_2), (q_1, q_5), (q_1, q_7), (q_1, q_6), (q_5, q_4), (q_6, q_9)\}$.

Algorithm 2: GeneratingNew

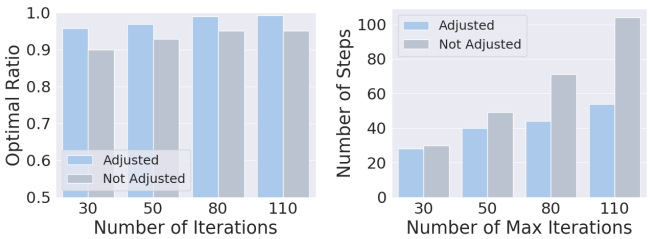
Input: $\boldsymbol{\eta}$ – previous solution, ξ – adjustment probability
Output: $\boldsymbol{\eta}'$ – new solution

```
// Randomly sample an edge from η
η_i ← random(η)
// Filter edges with consecutive or alternative correlation
E_p ← Consecutive(η, η_i)
E_n ← Alternative(η, η_i)
// Randomly select and assign a new edge traversal option
η_aux ← random(0, m − 1)
η_i ← η_aux
foreach η_p ∈ E_p do
    r ← random()
    if r < ξ then η_p ← η_aux
foreach η_n ∈ E_n do
    r ← random()
    if r < ξ then η_n ← m − η_aux
η' ← Update(η)
return η'
```

Each of these edges is associated with 4 different traversal options as $[(p_e(w), w)]_{w \in \text{supp}(\mathcal{W}(e))} = [(0.5, 1), (0.7, 2), (0.8, 3), (1, 4)]$, reflecting the positive correlation. All 21 Agents are initialized at state q_1.

Results and discussion: The optimal solution in this context can be computed through a brute force search, which requires 4096 (4^6) MILP optimizations. The run time increases exponentially with the number of edges and edge duration options. We use T_0 = 100 and T_{final} = 0.01 with a varying decaying coefficient α to adjust the maximum iterations numbers. Fig. 3 compares the modified edge adjustment method and the standard approach under varying numbers of iteration settings, with each data point being the average of 50 runs. Fig. 3a displays the iteration value comparison. This comparison is illustrated as an optimal ratio between the maximum value achieved during the iterations and the optimal value obtained through brute force search. The results demonstrate that employing the edge adjustment method increases the optimal percentage. Furthermore, the ratio for both cases is anticipated to increase as the number of iterations grows. Fig. 3b compares the average step at which the 99 percent near-optimal value is first reached during iterations. The maximum iteration number is recorded for cases where the near-optimal value is not attained. We observe that both methods perform similarly for smaller iteration numbers, with the average being close to the maximum iteration numbers. For the case of 30 iterations in Fig. 3b, this means that both approaches did not obtain the optimal values. As the number of iterations increases, the edge adjustment iteration steps converge. This means that the edge adjustment method, on average, reaches the optimal solution in less number of iterations than the standard approach. The standard approach is often close to the maximum iteration number, indicating that the optimal value was not reached. This occurs for the 50, 80, and 100 iterations cases in Fig. 3b.

Note that the number of steps continues to increase alongside the number of iterations for the case with edge adjustment. This is partly because, with higher iteration limits, the final step is larger for cases where the near-optimal value is not reached or reached in very late iteration stages, thus subsequently raising the average. The iteration steps reflect the run time performance because reaching the optimal value takes much fewer steps. The algorithm is



(a) (b)

Fig. 3: Performance comparison (a) Comparison of the optimal ratio from the max value achieved during the iteration over the optimal value. (b) Comparison between the number of steps to first find the near-optimal value.

evaluated under various settings that differ by the random assignment of color labeling in the environment, which in turn alters the motion scheme. The color (labeling) scheme shown in Fig. 2b corresponds to column C1 in Fig. 4. We illustrate the algorithm's performance in multiple settings regarding the optimal ratio with the maximum iteration number of 50, see Fig. 4. The figure shows the algorithm performs consistently over the six different labeling settings.

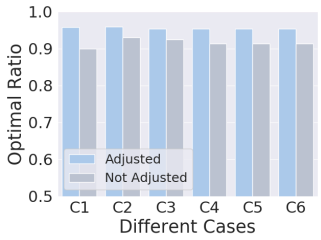


Fig. 4: Optimal ratio comparison in different cases in an average of 50 runs. The six cases differ in the color labeling.

VI. Conclusion

This work presented a framework that captures the planning problem for a team of heterogeneous robots satisfying CaTL specifications with uncertain traversal duration time in the environment. The problem with stochastic traversal durations and chance objectives on the total travel time is a non-linear problem, difficult to tackle using standard direct methods. Thus, we transformed the optimization problem into an equivalent bi-level optimization problem that allows us to decouple the problem and solve it by using an iterative approach. The outer level looks for the best probabilities of success and traversal time durations, allowing the inner level to maximize the robustness and probability of success and minimize the deterministic total travel time-bound. We introduced a simulated annealing algorithm and an edge adjustment method based on the correlations between edges in the environment to find optimal solutions faster than brute force search. We showed that our framework has increased performance and computational effectiveness compared to baseline approaches.

REFERENCES

[1] J. Gregory, J. Fink, E. Stump, J. Twigg, J. Rogers, D. Baran, N. Fung, and S. Young, "Application of multi-robot systems to disaster-relief scenarios with limited communication," in *Field and Service Robotics: Results of the 10th International Conference*, pp. 639–653, Springer, 2016.

[2] A. Ribeiro and J. Conesa-Muñoz, "Multi-robot systems for precision agriculture," *Innovation in Agricultural Robotics for Precision Agriculture: A Roadmap for Integrating Robots in Precision Agriculture*, pp. 151–175, 2021.

[3] K.-C. Chen, S.-C. Lin, J.-H. Hsiao, C.-H. Liu, A. F. Molisch, and G. P. Fettweis, "Wireless networked multirobot systems in smart factories," *Proceedings of the IEEE*, vol. 109, no. 4, pp. 468–494, 2020.

[4] G. A. Cardona and J. M. Calderon, "Robot swarm navigation and victim detection using rendezvous consensus in search and rescue operations," *Applied Sciences*, vol. 9, no. 8, p. 1702, 2019.

[5] J. Tumova and D. V. Dimarogonas, "Multi-agent planning under local ltl specifications and event-based synchronization," *Automatica*, vol. 70, pp. 239–248, 2016.

[6] Y. Kantaros and M. M. Zavlanos, "Stylus*: A temporal logic optimal control synthesis algorithm for large-scale multi-robot systems," *The International Journal of Robotics Research*, vol. 39, no. 7, pp. 812–836, 2020.

[7] G. A. Cardona, D. Saldaña, and C.-I. Vasile, "Planning for modular aerial robotic tools with temporal logic constraints," in *IEEE Conference on Decision and Control (CDC)*, pp. 2878–2883, IEEE, 2022.

[8] K. Liang and C.-I. Vasile, "Fair planning for mobility-on-demand with temporal logic requests," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1283–1289, IEEE, 2022.

[9] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, "Temporal-logic-based reactive mission and motion planning," *IEEE transactions on robotics*, vol. 25, no. 6, pp. 1370–1381, 2009.

[10] G. A. Cardona and C.-I. Vasile, "Partial Satisfaction of Signal Temporal Logic Specifications for Coordination of Multi-robot Systems," in *Workshop on the Algorithmic Foundations of Robotics*, pp. 223–238, Springer, 2022.

[11] A. Ulusoy, S. L. Smith, X. C. Ding, C. Belta, and D. Rus, "Optimality and robustness in multi-robot path planning with temporal logic constraints," *The International Journal of Robotics Research*, vol. 32, no. 8, pp. 889–911, 2013.

[12] G. A. Cardona, K. Leahy, and C.-I. Vasile, "Temporal logic swarm control with splitting and merging," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 12423–12429, IEEE, 2023.

[13] A. Bhatia, L. E. Kavraki, and M. Y. Vardi, "Sampling-based motion planning with temporal goals," in *2010 IEEE International Conference on Robotics and Automation*, pp. 2689–2696, IEEE, 2010.

[14] O. Maler and D. Nickovic, "Monitoring temporal properties of continuous signals," in *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems: Joint International Conferences on Formal Modeling and Analysis of Timed Systmes, FORMATS 2004, and Formal Techniques in Real-Time and Fault-Tolerant Systems, FTRTFT 2004, Grenoble, France, September 22-24, 2004. Proceedings*, pp. 152–166, Springer, 2004.

[15] V. Raman, A. Donzé, M. Maasoumy, R. M. Murray, A. Sangiovanni-Vincentelli, and S. A. Seshia, "Model predictive control with signal temporal logic specifications," in *53rd IEEE Conference on Decision and Control*, pp. 81–87, IEEE, 2014.

[16] K. Leahy, Z. Serlin, C.-I. Vasile, A. Schoer, A. M. Jones, R. Tron, and C. Belta, "Scalable and robust algorithms for task-based coordination from high-level specifications (scratches)," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2516–2535, 2021.

[17] M. Cai, K. Leahy, Z. Serlin, and C.-I. Vasile, "Probabilistic coordination of heterogeneous teams from capability temporal logic specifications," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1190–1197, 2021.

[18] D. P. Bertsekas, "Nonlinear programming," *Journal of the Operational Research Society*, vol. 48, no. 3, pp. 334–334, 1997.

[19] V. Daftardar-Gejji and H. Jafari, "An iterative method for solving nonlinear functional equations," *Journal of mathematical analysis and applications*, vol. 316, no. 2, pp. 753–763, 2006.

[20] P. J. Van Laarhoven, E. H. Aarts, P. J. van Laarhoven, and E. H. Aarts, *Simulated annealing*. Springer, 1987.

[21] O. Maler and D. Nickovic, "Monitoring temporal properties of continuous signals," in *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, pp. 152–166, Springer, 2004.

[22] A. Dokhanchi, B. Hoxha, and G. Fainekos, *5th International Conference on Runtime Verification, Toronto, ON, Canada.*, ch. On-Line Monitoring for Temporal Logic Robustness, pp. 231–246. Springer, 2014.

[23] S. Sadraddini and C. Belta, "Robust temporal logic model predictive control," in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 772–779, IEEE, 2015.

[24] D. Sun, J. Chen, S. Mitra, and C. Fan, "Multi-agent motion planning from signal temporal logic specifications," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3451–3458, 2022.

[25] V. Kurtz and H. Lin, "Mixed-integer programming for signal temporal logic with fewer binary variables," *IEEE Control Systems Letters*, vol. 6, pp. 2635–2640, 2022.

[26] A. Donzé and O. Maler, "Robust satisfaction of temporal logic over real-valued signals," in *International Conference on Formal Modeling and Analysis of Timed Systems*, pp. 92–106, Springer, 2010.

[27] J. H. Holland, "Genetic algorithms," *Scientific american*, vol. 267, no. 1, pp. 66–73, 1992.

[28] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2020.

[29] G. A. Cardona, K. Leahy, M. Mann, and C.-I. Vasile, "A flexible and efficient temporal logic tool for python: Pytelo," *arXiv preprint arXiv:2310.08714*, 2023.

[30] T. Parr, *The definitive ANTLR reference: building domain-specific languages*. Pragmatic Bookshelf, 2007.

[31] C.-I. Vasile and A. Ulusoy, "Ltl optimal multi-agent planner (lomap)." https://github.com/wasserfeder/lomap, 2024.

[32] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using networkx," in *Proceedings of the 7th Python in Science Conference* (G. Varoquaux, T. Vaught, and J. Millman, eds.), (Pasadena, CA USA), pp. 11 – 15, 2008.