2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2025)
October 19-25, 2025. Hangzhou, China

2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 979-8-3315-4393-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/IROS60139.2025.11247000

Automated Manipulation of Magnetic Microswarms for Temporal Logic Cargo Delivery Tasks in Complex Environments

Naifu Zhang, Tao Luo, Chongjie Jiang, Xiang Yin, Xiao Yu, and Rongrong Ji

***Abstract*— Micromanipulation using magnetic microswarms has garnered significant attention in recent years due to their potential in microscale cargo delivery tasks. While existing studies have demonstrated the capabilities of microswarms in basic manipulation tasks, they often lack the autonomy required to handle more complex specifications, particularly temporal logic tasks. In this paper, we propose a novel formal planning strategy for magnetic microswarms that enables cargo delivery in complex environments while satisfying finite linear temporal logic (LTL_f) specifications. Our approach consists of two key components. First, we develop a high-level path planner based on a bidirectional temporal logic rapid-explore random tree star (BTL-RRT*) algorithm, which facilitates efficient planning while ensuring compliance with the given task specifications. Second, we employ an automaton to manage the manipulation modes of the microswarm, enabling real-time control over the capture and release of cargoes. In addition, we implement the planning strategy on microswarms actuated by a visual feedback magnetic tweezers system. Extensive simulations and experimental results demonstrate the effectiveness of the proposed planning strategy. The results indicate that, using the proposed approach, microswarms can autonomously select and deliver multiple microbeads to designated regions in both static and dynamic environments, adhering to the LTL_f specifications.**

I. Introduction

Microrobots actuated by external magnetic fields have garnered significant attention due to their potential in micromanipulation and biomedical applications, including non-invasive surgery, targeted drug delivery, and microscale manufacturing [1]–[4]. Various types of magnetic microrobots have been extensively studied in recent years [5]–[7]. However, individual microrobots often face limitations in payload capacity and operational flexibility. To address these challenges, microswarms composed of millions of paramagnetic Fe_3O_4 nanoparticles have emerged as a promising solution [8]–[10]. By leveraging rotating magnetic fields, these nanoparticles can be magnetized and assembled into a vortex-like swarm through the combined effects of magnetic forces, hydrodynamic interactions, and inter-particle attractions [8]. Owing to their dynamic and reconfigurable morphologies, microswarms exhibit exceptional capabilities in the controllable transport and delivery of microscale cargoes [11]. These unique characteristics make microswarms highly suitable for micromanipulation tasks, particularly in cargo delivery.

Automated cargo delivery using microswarms has been widely investigated in recent years [12]–[15]. A notable study was presented in [15], where an automated control approach was proposed for reconfigurable microswarms to deliver multiple cargoes. In this method, the high-level planning problem was formulated as an open-loop traveling salesman problem (OTSP), and an enhanced genetic algorithm (eGA) was developed to optimize the delivery sequence of all cargoes. However, this approach still relies on manual cargo selection, limiting its autonomy for tasks with more complex specifications, such as selective delivery tasks.

Temporal logic task specifications, such as linear temporal logic (LTL) specifications [16], are widely used to describe manipulation tasks for robotic systems [17]. LTL provides a formal framework to describe temporal logic tasks using operators such as "always", "next", and "eventually". To complete delivery tasks with LTL specifications, microswarms must achieve autonomous high-level planning that ensures the correctness of the given specifications.

Among various high-level planning approaches, sampling-based methods stand out due to their exceptional adaptability to diverse environments [17]. For instance, high-level planners based on the rapid-explore random graph (RRG) method have been studied in [18] and [19]. Additionally, in [20], the temporal logic rapid-explore random tree star (TL-RRT*) algorithm was proposed to grow a sampling tree monitored by an automaton, ensuring the discovery of a feasible path that satisfies the task specifications. However, these existing methods often require substantial computational time to generate feasible planning, making them less practical for complex environments.

In this work, we focus on microswarms executing cargo delivery tasks in complex environments while adhering to given finite linear temporal logic (LTL_f) specifications. As illustrated in Fig. 1, the workspace for these tasks contains multiple microbeads. The delivery tasks require the microswarm to transport desired cargoes to a specified release region, following the given LTL_f specifications. For instance, a task might be described as: *"Eventually deliver two* $100\,\mu m$ *microbeads (color: green) to the release region, while always*

*This work was supported in part by the National Science and Technology Major Project under Grant 2021ZD0112600, in part by the National Natural Science Foundation of China under Grants 62173283, 62273285, and 62173226, and in part by the Fujian Provincial Natural Science Foundation of China under Grant 2024J010013. (*Corresponding author: Xiao Yu.*)

Naifu Zhang is with the Department of Automation and the Institute of Artificial Intelligence, Xiamen University, Xiamen, China (e-mail: naifuzhang@stu.xmu.edu.cn).

Tao Luo and Chongjie Jiang are with the Pen-Tung Sah Institute of Micro-Nano Science and Technology, Xiamen University, Xiamen, China (e-mail: luotao@xmu.edu.cn; jiangchongjie@stu.xmu.edu.cn).

Xiang Yin is with the Department of Automation, Shanghai Jiao Tong University, Shanghai, China. (e-mail: yinxiang@sjtu.edu.cn).

Xiao Yu and Rongrong Ji are with the Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, and the Institute of Artificial Intelligence, Xiamen University, Xiamen, China (e-mail: xiaoyu@xmu.edu.cn; rrji@xmu.edu.cn).

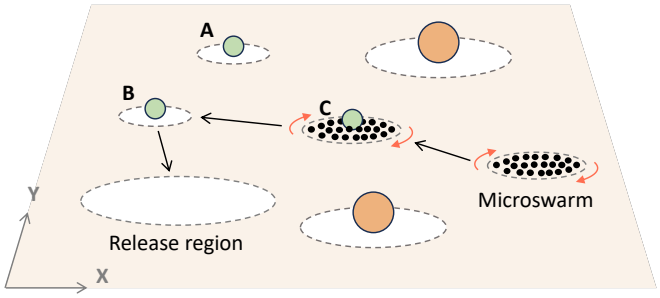


Fig. 1: Illustration of a workspace for cargo delivery tasks.

avoiding 200 μ*m microbeads (color: orange).*" Consequently, the microswarm must autonomously select the appropriate cargoes to transport, ensuring the delivery task is completed correctly with the minimal cost.

To address these challenges, we propose a formal planning strategy for manipulating magnetic microswarms to execute cargo delivery tasks described by LTL_f specifications in complex environments. Our approach comprises two key components. First, we develop a sampling-based high-level path planner that employs a Bidirectional TL-RRT* (BTL-RRT*) algorithm to enable rapid path planning while ensuring the correctness of the task specifications. Second, we propose an automaton-based mode planner to achieve real-time manipulation mode management. The planning strategy is implemented on vortex-like microswarms actuated by a visual-feedback magnetic-tweezers platform [21]. Both experiments and simulations are conducted to illustrate the effectiveness of the proposed strategy.

The main contributions of this work are as follows.

- We propose a novel formal planning strategy for magnetic microswarms to deliver cargoes while satisfying LTL_f specifications. Existing control schemes for microswarms, such as [15], are not capable of completing temporal logic tasks. In contrast, the proposed strategy enables the microswarm to autonomously select and deliver cargoes to a desired region while adhering to LTL_f specifications.
- We develop a BTL-RRT* method to achieve fast high-level path planning in complex workspaces while guaranteeing the correctness of the task specifications. Existing planning methods, such as those in [20], require substantial time for planning in crowded and dynamic workspaces. To address this problem, our method employs a bidirectional sampling tree to quickly identify a feasible path. Additionally, an optimization procedure is developed to refine the high-level path without violating the task specifications.
- Experiments are conducted in both static and dynamic workspaces on a magnetic tweezers platform to validate the effectiveness of the proposed formal planning strategy in diverse scenarios.

The remainder of this paper is organized as follows. In Section II, we review the preliminaries of LTL_f semantics and the definition of the deterministic finite automaton (DFA). In Section III, we introduce the high-level models of the motion and manipulation modes of the microswarm. In Section IV, we present the formal planning strategy for achieving cargo delivery tasks with LTL_f specifications. Simulations and experiments are conducted in Section V to highlight the advantages of the proposed strategy. Finally, the paper is concluded in Section VI.

II. Preliminaries

A. Finite Linear Temporal Logic

In this paper, we employ linear temporal logic over finite traces (LTL_f) to formally describe delivery tasks. Unlike standard LTL, which operates over infinite traces, LTL_f is specifically designed for finite sequences of states, making it more suitable for practical applications such as cargo delivery tasks. The syntax of LTL_f [22] over a set of atomic propositions Π is defined as

$$\phi ::= \text{true} \mid \pi \mid \neg\phi \mid \phi_1 \vee \phi_2 \mid X\phi \mid \phi_1 U \phi_2, \quad \pi \in \Pi \tag{1}$$

where $\neg$ (negation) and $\vee$ (disjunction) are logical operators, X (next) and U (until) are temporal operators, π represents a Boolean variable, and Π denotes the set of atomic propositions. The cardinality of Π is denoted by $|\Pi|$, and its power set is represented by 2^Π. Additional operators, such as $\wedge$ (conjunction), F (eventually), and G (always), can be derived from combinations of the aforementioned operators. For example, $F\pi$ signifies that π will be satisfied at some future point, while $G\pi$ indicates that π must be satisfied throughout the entire process.

Example 1: Consider a workspace as illustrated in Fig. 1 and a temporal logic delivery task as described in Section I. The corresponding LTL_f specification can be expressed as

$$\phi_0 = (\phi_1 \vee \phi_2 \vee \phi_3) \wedge G(\neg \pi^O) \tag{2}$$

$$\begin{cases} \phi_1 = F\Big(\pi^A \wedge F\big((\pi^B \vee \pi^C) \wedge F(\pi^R)\big)\Big) \\ \phi_2 = F\Big(\pi^B \wedge F\big((\pi^A \vee \pi^C) \wedge F(\pi^R)\big)\Big) \\ \phi_3 = F\Big(\pi^C \wedge F\big((\pi^A \vee \pi^B) \wedge F(\pi^R)\big)\Big) \end{cases} \tag{3}$$

where π^O denotes the microswarm reaching the 200 μm microbeads, π^R denotes its arrival at the release region, and $\{\pi^A, \pi^B, \pi^C\}$ denote the microswarm reaching the regions of the corresponding 100 μm microbeads, respectively.

B. Deterministic Finite Automaton

Every LTL_f expression ϕ can be translated into a deterministic finite automaton (DFA) A_ϕ that accepts ϕ.

Definition 1: (Deterministic Finite Automaton (DFA)). A deterministic finite automaton (DFA) is defined as a tuple $A_\phi = (\mathcal{Q}_A, 2^\Pi, \delta_A, \mathcal{Q}_A^0, \mathcal{Q}_A^F)$, where

- $\mathcal{Q}_A$ is a finite set of DFA states;
- 2^Π is the alphabet;
- $\delta_A : \mathcal{Q}_A \times 2^\Pi \to \mathcal{Q}_A$ represents the transition function;
- $\mathcal{Q}_A^0 \subseteq \mathcal{Q}_A$ and $\mathcal{Q}_A^F \subseteq \mathcal{Q}_A$ denote the sets of initial and accepting states, respectively.

A DFA run is defined as a finite sequence of automaton states $\sigma_A = q_A^0 q_A^1 \dots q_A^n$, where $q_A^i \in \mathcal{Q}_A$, $q_A^0 \in \mathcal{Q}_A^0$, $q_A^{i+1} \in \delta_A(q_A^i, \pi)$, $\pi \in 2^{\Pi}$, and $i \geq 0$. A run σ_A is accepted by A_ϕ, if $q_A^n \in \mathcal{Q}_A^F$. Intuitively, the high-level trajectory of the microswarm corresponds to a DFA run σ_A, and this trajectory satisfies the task specification ϕ if σ_A is accepted by A_ϕ.

III. High-Level Modeling

A. Motion Modeling

To ensure the correctness of the robotic system in adhering to the task specification, we model the high-level motion of the microswarm system as a transition system [20]:

$$T = (\mathcal{X}, x_0, \delta, \Pi, L) \tag{4}$$

where

- $\mathcal{X} \subset \mathbb{R}^2$ is a set of states representing all reachable points within the workspace;
- $x_0 \in \mathcal{X}$ denotes the initial position of the microswarm;
- $\delta \subseteq \mathcal{X} \times \mathcal{X}$ represents the transition relations, where a transition between two points exists if their connection 1) avoids obstacle regions and 2) crosses boundaries at most once;
- Π is the set of atomic propositions;
- $L : \mathcal{X} \to 2^{\Pi}$ is a labeling function that maps states to atomic propositions. States are assigned labels if they lie within the corresponding buffer regions.

The motion of the microswarm generates a path in (4). The path is represented as a sequence of states $P = x_1 x_2 \dots x_m$, where $(x_i, x_{i+1}) \in \delta_T$ and $i \geq 0$. A path is correct with respect to the task specification $P \models \phi$ if its trace $\tau = L(x_1)L(x_2)\dots L(x_m)$ is accepted by A_ϕ. Additionally, the cost J of a path P is defined as:

$$J(P) = \sum_{i=0}^{m-1} \|x_{i+1} - x_i\| \tag{5}$$

where $\|x_{i+1} - x_i\|$ is the Euclidean distance between x_{i+1} and x_i. The objective of high-level path planning is to find a correct path $P \models \phi$ while minimizing the cost $J(P)$.

B. Manipulation Mode Modeling

To achieve controllable capture and release of cargoes, a model capable of tracking the microswarm manipulation mode is essential. Therefore, we partition the microswarm into the following four modes:

- **Assembly:** Assemble the nanoparticles into a stable vortex-like microswarm.
- **Move:** The microswarm maintains a vortex-like shape and is actuated to track the high-level path.
- **Capture:** The microswarm maintains a vortex-like shape and is actuated to capture the cargoes.
- **Disassembly:** Disassemble the microswarm to release the cargoes at the desired region.

These four modes are dynamically switched based on real-time visual feedback information.

Algorithm 1: BTL-RRT*

Input: Position of the microswarm x_s^{current}, task state of the microswarm q_A^{current}, centroid of the release region x_R, DFA A_ϕ

Output: High-level path P

▷ Initialization

Set initial node $q^{\text{current}} \leftarrow (x_s^{\text{current}}, q_A^{\text{current}})$

Set objective node $q^F \leftarrow (x_R, q_A^F)$

$\mathcal{V}_f \leftarrow \{q^{\text{current}}\}, \mathcal{E}_f \leftarrow \emptyset, q^{\text{current}}.\text{cost} \leftarrow 0$

$\mathcal{V}_b \leftarrow \{q^F\}, \mathcal{E}_b \leftarrow \emptyset, q^F.\text{cost} \leftarrow 0$

Initialize $T_f \leftarrow (\mathcal{V}_f, \mathcal{E}_f), T_b \leftarrow (\mathcal{V}_b, \mathcal{E}_b)$

The connection between trees $\delta^{\text{con}} \leftarrow$ None

▷ Construction

for $n \leftarrow 1 : n_{max}$ **do**

  $x^{\text{rand}} \leftarrow \text{Sample}(\mathcal{X}_{\text{free}})$

  foreach $T \leftarrow \{T_f, T_b\}$ **do**

    $q^{\text{nearest}} \leftarrow \text{Nearest}(x^{\text{rand}}, \mathcal{V})$

    $(x^{\text{nearest}}, q_A^{\text{nearest}}) \leftarrow q^{\text{nearest}}$

    $x^{\text{new}} \leftarrow \text{Steer}(x^{\text{nearest}}, x^{\text{rand}}, \eta_s)$

    $Q^{\text{near}} \leftarrow \text{Near}(x^{\text{new}}, \mathcal{V}) \cup \{q^{\text{nearest}}\}$

    $\mathcal{Q}^{\text{con}} \leftarrow \text{Near}(x^{\text{new}}, \mathcal{V}')$

    foreach $q_A^{new} \in \mathcal{Q}_A$ **do**

      $q^{\text{new}} \leftarrow (x^{\text{new}}, q_A^{\text{new}})$

      $T \leftarrow \text{Extend}(Q^{\text{near}}, q^{\text{new}}, T)$

      if q^{new}*.parent is not None* **then**

        $T \leftarrow \text{Rewire}(q^{\text{new}}, Q^{\text{near}}, T)$

        ▷ Connectivity check

        $\delta^{\text{con}} \leftarrow \text{Connect}(q^{\text{new}}, \mathcal{Q}^{\text{con}})$

        if δ^{con} *is not None* **then**

          Break sampling

$P \leftarrow \text{FindPath}(T_f, T_b, \delta^{\text{con}})$

$P \leftarrow \text{OptimizePath}(P, A_\phi)$

IV. Formal Planning Strategy

A. System Diagram

The diagram of the magnetic tweezers system with the proposed formal planning strategy is illustrated in Fig. 2. Specifically, the system integrates an initialization procedure, a high-level planner (including a path planner and a manipulation mode planner), a low-level controller, and a magnetic tweezers system. During the initialization procedure, the cargoes are identified and labeled, and the given LTL$_\text{f}$ specification ϕ is translated into a DFA A_ϕ using the software *LTLf2DFA* [23]. The high-level planner, comprising a path planner and a mode planner, is subsequently employed to generate correct planning. The path planner identifies a feasible path $P^{(k)}$ that satisfies the task specification. Next, the mode planner determines the desired morphology and position of the microswarm in real-time. A low-level controller is then applied to calculate the input for the magnetic tweezers system to precisely actuate the microswarm. Real-time feedback $W^{(k)}$ is implemented through a microscope and a camera.

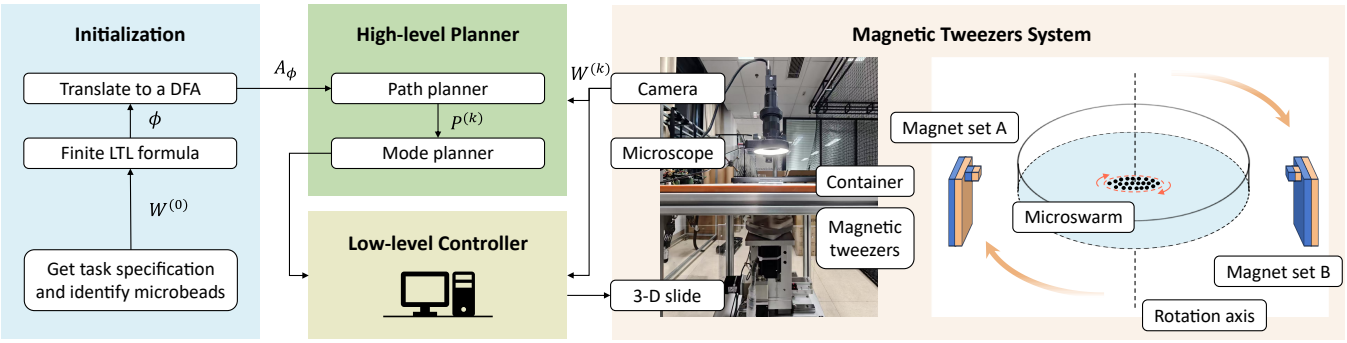


Fig. 2: Illustration of the magnetic tweezers system with the proposed formal planning strategy. The task specification and its DFA are denoted by ϕ and A_ϕ, respectively. The high-level path is $P^{(k)}$, and the visual feedback information is $W^{(k)}$.

B. High-Level Path Planner

The high-level path planner aims to identify a correct path on the transition system (4) with the minimal cost. To achieve fast high-level path planning while ensuring the correctness of the LTL_f specification, we propose a BTL-RRT* algorithm based on [20]. The key differences between the temporal logic RRT* and the conventional point-to-point RRT* are as follows. First, trees are constructed not on the transition system but on a product state space, where a node $q = (x, q_A)$ pairs a state from (4) with a state from A_ϕ. Second, the connectivity between nodes depends on A_ϕ.

Cargo delivery tasks often occur in crowded and dynamic workspaces, as microscale cargoes may experience displacement due to fluidic influences. Consequently, a fast planning method is essential to address potential changes in the workspace.

To better address these challenges, the proposed high-level path planner incorporates the following improvements:

- We develop a BTL-RRT* method that grows bidirectional trees simultaneously from both the microswarm and the release region. Existing methods, such as the standard TL-RRT* and the biased TL-RRT* in [20], are less efficient for generating planning in these workspaces. The standard TL-RRT* often produces excessive redundant branches, undermining planning efficiency. Additionally, while the biased TL-RRT* is advantageous for handling complex tasks, it consumes significant time when calculating biased samples in workspaces where task complexity is limited by the microswarm's loading capacity. The bidirectional tree approach is a less scalable but greedy method to rapidly identify a feasible path.
- We establish buffer regions for both cargoes and obstacles. The path planner's objective is to reach the buffer regions of the specified cargoes, preventing the microswarm from incorrectly capturing cargoes.
- We develop an algorithm to further optimize the produced path under the guidance of the DFA.

The main procedure of the BTL-RRT* algorithm is outlined in Algorithm 1. Initially, the bidirectional tree is initialized. The forward tree T_f grows from the microswarm

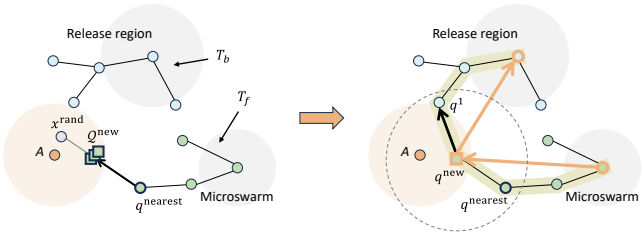


Fig. 3: An example of BTL-RRT* sampling.

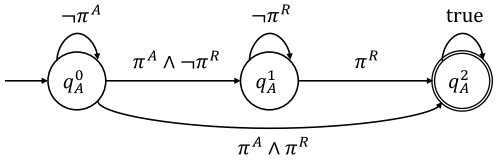


Fig. 4: An illustration of A_{ϕ_4}.

node q^{current} to the release region node q^F, where x_s^{current} represents the current position of the microswarm, q_A^{current} denotes the current task state of the microswarm, x_R is the centroid of the release region, and q_A^F is the only accepting state of A_ϕ. Conversely, the backward tree T_b grows from the release region toward the microswarm.

Starting with the initialized bidirectional tree, the procedure iteratively searches for a feasible path. The sampling process begins by generating a random sample $x^{\text{rand}} \in \mathcal{X}_{\text{free}}$, where $\mathcal{X}_{\text{free}} \subset \mathcal{X}$ denotes the obstacle-free space that maintains a safe distance from the workspace boundaries.

The function $Steer$ is then applied to guide x^{nearest} toward x^{rand} at a distance no greater than η_s, producing a new point x^{new}. Next, the algorithm selects a set of nodes near x^{new} using:

$$\text{Near}(x^{\text{new}}, \mathcal{V}) = \{(x, q_A) \in \mathcal{V} \mid \|x^{\text{new}} - x\| \leq \eta_{\text{near}}\} \quad (6)$$

where η_{near} is a predefined parameter. An additional set Q^{con}, comprising nearby nodes from the other tree, is constructed to facilitate inter-tree connectivity checks (Algorithm 1, line 14). Subsequently, x^{new} is paired with each DFA state to form a set of child node candidates:

$$\mathcal{Q}^{\text{new}} = \{q^{\text{new}} = (x^{\text{new}}, q_A^{\text{new}}) \mid q_A^{\text{new}} \in \mathcal{Q}_A\}. \quad (7)$$

Algorithm 2: *OptimizePath*

Input: Raw path P, DFA A_ϕ
Output: Optimized path P

```
for j ← 1 : j_max do
    if |P| ≤ 2 then
        break
    i ← Random()
    τ ← GetTrace(P)
    τ' ← Delete(τ, i + 1)
    if (x_i, x_{i+2}) ∈ δ_T and Accept(A_φ, q_A^current, τ') then
        P ← Delete(P, i + 1)
```

For each q^{new}, the function $Extend$ attempts to select a parent node from $q^{\text{near}} \in \mathcal{Q}^{\text{near}}$ with a valid connection and the minimal cost. The criteria for validating a connection between a parent node and a child node differ for the forward tree and the backward tree:

- For the forward tree, a connection is valid if $(x^{\text{parent}}, x^{\text{child}}) \in \delta_T$ and $q_A^{\text{child}} \in \delta_A\big(q_A^{\text{parent}}, L(x^{\text{child}})\big)$.
- For the backward tree, a connection is valid if $(x^{\text{child}}, x^{\text{parent}}) \in \delta_T$ and $q_A^{\text{parent}} \in \delta_A\big(q_A^{\text{child}}, L(x^{\text{parent}})\big)$.

Leveraging this criteria, a set comprising all valid parent candidates can be formed as $\mathcal{Q}^{\text{near}}_{\text{parent}} \subseteq \mathcal{Q}^{\text{near}}$. Thereby, the parent node of q^{new} can be selected as:

$$q^{\text{new}}.\text{parent} = \underset{q^{\text{near}} \in \mathcal{Q}^{\text{near}}_{\text{parent}}}{\operatorname{argmin}} \left(q^{\text{near}}.\text{cost} + C(q^{\text{near}}, q^{\text{new}})\right) \quad (8)$$

where the cost function $C(q^{\text{near}}, q^{\text{new}}) = \|x^{\text{near}} - x^{\text{new}}\|$ is the Euclidean distance between the corresponding states on (4), and the cost of a node is recursively defined as $q^{\text{near}}.\text{cost} = q^{\text{near}}.\text{parent}.\text{cost} + C(q^{\text{near}}, q^{\text{near}}.\text{parent})$.

If q^{new} successfully connects to a parent node, a $Rewire$ procedure is executed to rewire q^{new} as a parent node to nodes in $\mathcal{Q}^{\text{near}}$, identifying potential valid connections with lower cost. Specifically, a node is rewired as a child node of q^{new} if $q^{\text{near}}.\text{cost} > q^{\text{new}}.\text{cost} + C(q^{\text{near}}, q^{\text{new}})$. The function $Connection$ is then used to validate the connectivity of q^{new} to $\mathcal{Q}^{\text{con}}$. Once an inter-tree connection δ^{con} is established, the entire sampling procedure terminates. A path P satisfying the task specification ϕ is then constructed from the initial point of T_f, through δ^{con}, to the initial point of T_b.

To further optimize the path, we propose an optimization algorithm guided by A_ϕ. Each iteration begins by randomly selecting an index i. The trace τ of the path is then constructed. The algorithm attempts to delete the $(i+1)$-th atomic proposition of τ and verifies the correctness of the simplified trace τ'. If τ' is accepted by A_ϕ in the current DFA state q_A^{current} and the broken path satisfies $(x_i, x_{i+2}) \in \delta_T$, the corresponding node is removed from P. Consequently, the path is optimized without violating the task specification.

Finally, the produced high-level path can be partitioned into reference paths for the low-level system.

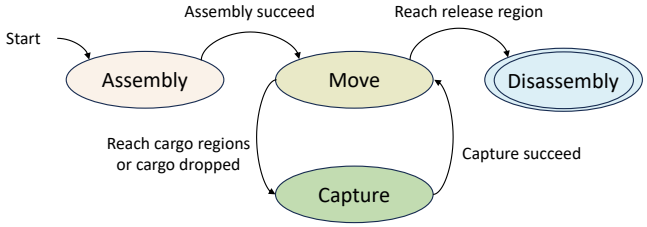


Fig. 5: The automaton-based manipulation mode planner.

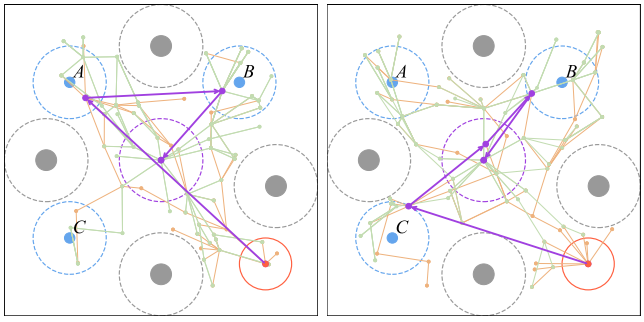
Example 2: An example of BTL-RRT* sampling is illustrated in Fig. 3, where the microswarm is assigned a simple delivery task: *"Eventually deliver microbead A to the release region."* The task can be written as a LTL_f specification $\phi_4 = F\big(\pi^A \wedge F(\pi^R)\big)$. The corresponding DFA A_{ϕ_4} is shown in Fig. 4. First, x^{rand} and q^{nearest} are selected through the sampling procedure. Then, x^{new} is generated using the $Steer$ function, and the DFA states are paired with x^{new} to form a set $\mathcal{Q}^{\text{new}}$ (marked by green squares). The nodes $q^{\text{nearest}} = (x^{\text{nearest}}, q_A^0)$ and $q^1 = (x^1, q_A^1)$ are collected into $\mathcal{Q}^{\text{near}}$ and $\mathcal{Q}^{\text{con}}$, respectively (marked by black dashed circles). One of the paired nodes, $q^{\text{new}} = (x^{\text{new}}, q_A^1)$, is connected to q^{nearest} as a child node. The $Rewire$ function does not alter the connection, as no rewiring results in a lower cost. Subsequently, the $Connect$ function links the bidirectional tree from q^{new} to q^1 (marked by a bold black arrow). Thus, a feasible path (bold green lines) is identified. The produced path can be further optimized, as indicated by the orange arrows.

C. High-Level Manipulation Mode Planner

The mode planner is responsible for determining the desired position and morphology of the microswarm. Inspired by [15], we employ an automaton-based mode planner to manage the four manipulation modes of the microswarm. As illustrated in Fig. 5, the modes are managed as follows:

- **Assembly**: Assembly is conducted before the execution of the entire task. The mode switches to *Move* upon completion of the assembly.
- **Move**: The microswarm moves along the high-level path. If a cargo region is reached or a cargo is dropped from the microswarm, the mode switches to *Capture*. If the boundary of the release region is reached, the mode switches to *Disassembly*.
- **Capture**: The microswarm is actuated to the centroid of the buffer region to capture the corresponding cargo. Once the cargo is successfully captured, the mode switches back to *Move*.
- **Disassembly**: The microswarm is actuated to the centroid of the release region, where disassembly is performed. The cargo delivery tasks are completed through disassembly.

By adhering to these rules, the mode planner establishes the desired morphology and position of the microswarm. Subsequently, the low-level system is guided to execute the high-level planning.



(a) (b)

Fig. 6: Simulation results of the high-level path planner. (a) BTL-RRT* planning on task ϕ_0. (b) BTL-RRT* planning on task ϕ_5 (ϕ_0 with a capacity constraint). The forward tree is marked by orange lines and dots. The backward tree is marked by green lines and dots. The dashed circles denote the buffer regions. The microswarm is marked by a red circle and dot. The release region is marked by a dashed purple circle. The produced path is marked by purple arrows.

TABLE I

PERFORMANCE COMPARISON OF HIGH-LEVEL PLANNING METHODS

Metric	TL-RRT* [20]	Biased TL-RRT* [20]	BTL-RRT*
$succ\%$	100	100	100
$iter$	149.37 ± 40.81	$\mathbf{41.31 \pm 14.30}$	52.45 ± 16.88
$time\,(s)$	0.83 ± 0.51	0.36 ± 0.18	$\mathbf{0.26 \pm 0.17}$
$cost$	880.37 ± 92.68	908.85 ± 78.21	$\mathbf{876.34 \pm 77.91}$

V. EXPERIMENTS

A. System Setup

As illustrated in Fig. 2, the planning strategy is implemented on a visual feedback magnetic tweezers system based on [21]. A pair of magnet sets (polar distance: 50 mm) is mounted on a servo motor to generate a rotating gradient magnetic field at 5 Hz. Each magnet set consists of a $50\,mm \times 50\,mm \times 5\,mm$ rectangular magnet and a $10\,mm \times 10\,mm \times 10\,mm$ cubic magnet. The servo motor is carried by a 3-DOF electric slide, with the magnets positioned approximately 10 mm below the base plane. A droplet of $0.8\,\mu$L spherical Fe_3O_4 particles (diameter: 0.5–$0.6\,\mu$m, concentration: $0.5\%\,w/v$) is added to a container filled with PBS solution to form the microswarm. The microswarm spontaneously follows the rotation center of the magnetic tweezers but may experience slight deviations due to disturbances [21]. Polystyrene (PS) microbeads (diameter: $100\,\mu$m and $200\,\mu$m, color: red) are placed in the workspace to serve as cargoes and obstacles. Calculations are performed by a computer, a motion controller card, and motor drivers.

B. Simulation of the Path Planner

We compare the BTL-RRT* method with the standard TL-RRT* method [20] and the biased TL-RRT method [20] to highlight its advantages. The simulations are conducted in a 600×600 workspace, where the microswarm is required to complete a delivery task ϕ_0, with $100\,\mu$m microbeads marked by blue disks and $200\,\mu$m microbeads marked by gray disks. Simulations are performed on a workstation equipped with an AMD core R7-8745H (3.80 GHz). The sampling parameters are set as $n_{\max} = 600$, $\eta_s = 100$, and $\eta_{\text{near}} = 200$.

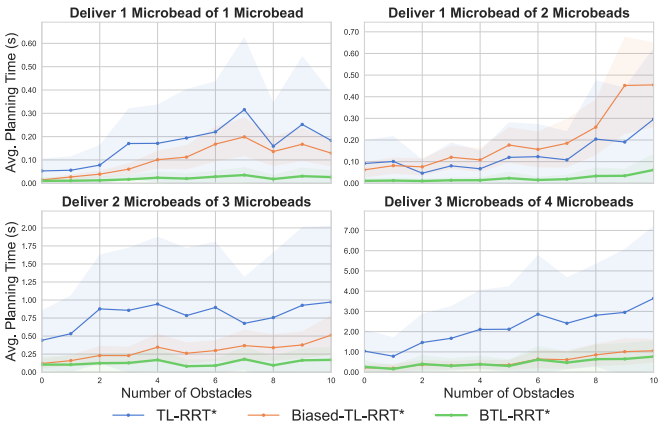


Fig. 7: Simulation results comparing three planning methods across four delivery tasks and workspaces with varying numbers of obstacles. The curves represent the average planning time, while the shaded areas indicate the standard deviation.

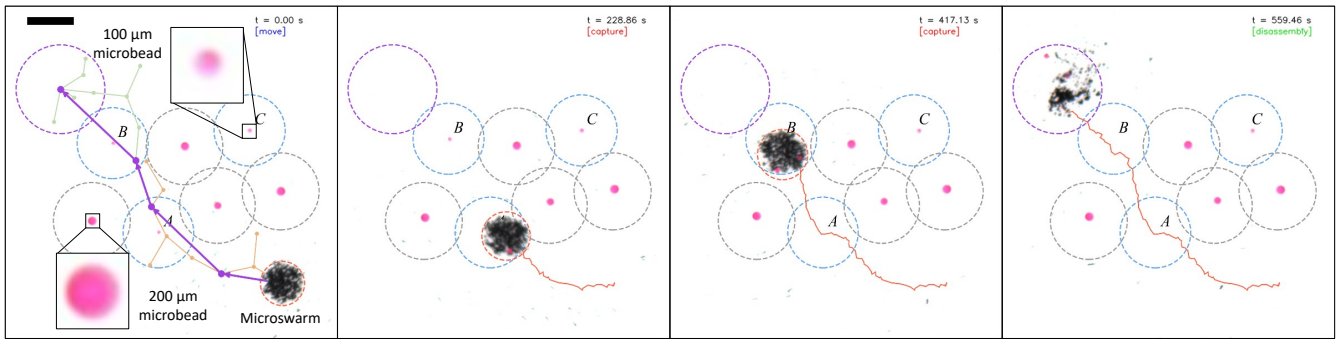
Table I compares the success rates, iterations, runtimes until a feasible path is found, and path costs calculated by (5) for the three algorithms. The results are presented as means and standard deviations over 100 simulations. A simulation result of the BTL-RRT* method planning on delivery task ϕ_0 is presented in Fig. 6a. The simulation results demonstrate that, while the biased TL-RRT* method finds a feasible path by fewer iterations, the biased sampling procedure consumes substantial time in each iteration, undermining overall efficiency. In contrast, the proposed BTL-RRT* method outperforms both the TL-RRT* method and the biased TL-RRT* method by finding a feasible path with less runtime.

We further validate the advantages of the BTL-RRT* method by evaluating its performance across four different tasks in workspaces with varying number of obstacles. For each setting, 100 simulations are conducted over 10 randomly generated workspaces. As illustrated in Fig. 7, the BTL-RRT* method outperforms both the standard TL-RRT* method and the biased TL-RRT* method by finding feasible paths with lower runtimes, highlighting its robustness and efficiency. The advantages over the biased TL-RRT* method are more significant in crowded workspaces, but gradually diminish with increasing task complexity.

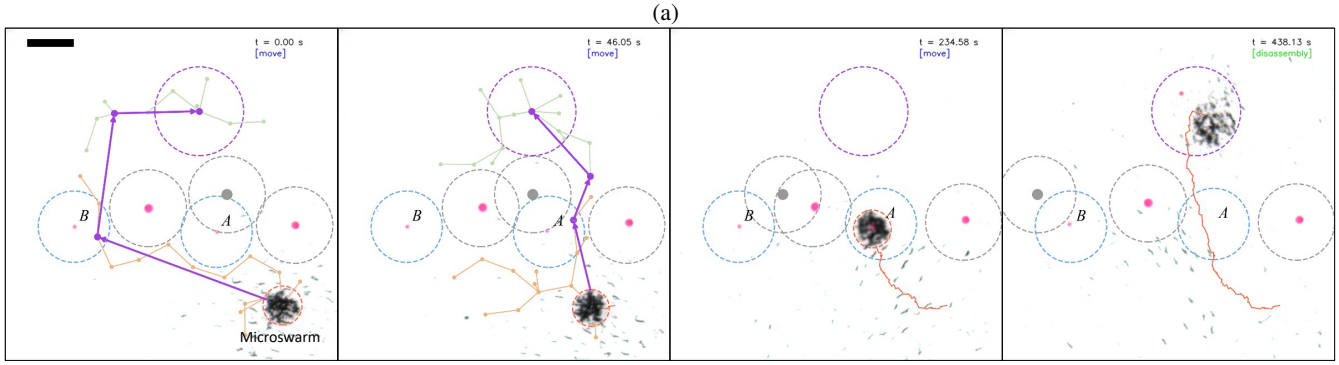
The proposed high-level planning method is also applicable to a wide range of task specifications. For instance, task ϕ_0 with a capacity constraint of one cargo at a time can be formulated as the following LTL_f expression:

$$\phi_5 = \phi_0 \wedge \phi_{\text{capacity}} \tag{9}$$

where $\phi_{\text{capacity}} = G\big(\pi^A \rightarrow X(\pi^R)\big) \wedge G\big(\pi^B \rightarrow X(\pi^R)\big) \wedge G\big(\pi^C \rightarrow X(\pi^R)\big)$ is the capacity constraint. The simulation result is shown in Fig. 6b, where the BTL-RRT* planner



(a)



(b)

Fig. 8: Results of real-world experiments. (a) Experiment I in a static workspace. (b) Experiment II in a dynamic workspace. The purple dashed circle denotes the release region. The forward sampling tree is marked by orange lines and dots. The backward sampling tree is marked by green lines and dots. The red dashed circle indicates the outer safety circle of the microswarm. The high-level path is marked by purple arrows. The microswarm trajectory is marked by a red curve. The gray dashed circles and the blue dashed circles represent the buffer region of the 200 μm PS microbeads and the 100 μm PS microbeads, respectively. The scale bars are 1 mm. The video clip of the results are available at the video attachment or https://xiaoyu.xmu.edu.cn/demos/microswarms-IROS2025.mp4.

finds a feasible path satisfying the given specification ϕ_5.

C. Real-World Experiments

To validate the effectiveness of the proposed planning strategy, we conduct two implementation experiments in a 7.20 mm $\times$ 7.20 mm workspace. To maintain a stable vortex-like morphology, the microswarm is programmed to move at a constant velocity. A 0.2 Hz reciprocating motion along the x-axis is applied to disassemble the microswarm in the release region [21]. Additionally, the PS microbeads may experience slight displacement due to the outer rotating particles. In this case, replanning is triggered when a PS microbead moves a certain distance away from the centroid of its buffer region. Conservative buffer regions are set based on the maximum radius of the microswarm.

1) Experiment in a Static Workspace: Experiment I is performed in a crowded workspace containing three 100 μm PS microbeads (labeled A, B, and C) and four 200 μm PS microbeads. The microswarm is tasked with ϕ_0.

The results are shown in Fig. 8a. At $t = 0.00$ s, the nanoparticles form a vortex-like microswarm, and the high-level path planner generates an optimal path to complete the delivery task by delivering A and B to the release region. Next, the microswarm bypasses the obstacles and captures A with the vortex-like trapping force caused by the rotating particles at $t = 228.86$ s. Subsequently, at $t = 417.13$ s, the microswarm successfully captures B. The task concludes by disassembling the microswarm to release the microbeads to the release region at $t = 559.46$ s.

2) Experiment in a Dynamic Workspace: Experiment II is performed in a dynamic workspace to validate the robustness of the proposed planning strategy. As illustrated in Fig. 8b, a workspace containing two 100 μm PS microbeads (labeled A and B), two 200 μm PS microbeads, and a simulated moving microbead (marked by a gray disk) is constructed. The microswarm is tasked with *"Eventually deliver one* 100 μ*m microbead to the release region, while always avoiding* 200 μ*m microbeads."*. The corresponding LTL_f specification can be written as:

$$\phi_6 = F\left((\pi^A \vee \pi^B) \wedge F(\pi^R)\right) \wedge G(\neg \pi^O) \tag{10}$$

In this case, the microswarm must autonomously decide the most appropriate cargo that minimizes travel distance in real time. At $t = 0.00$ s, since A is blocked by obstacles, the algorithm determines a path that captures B to complete the task. However, as the obstacle moves away, the algorithm identifies a shorter path at $t = 46.05$ s that carries A to the release region. At $t = 234.58$ s, the microswarm successfully

captures A. The task is completed by releasing A to the release region at $t = 438.13\,\mathrm{s}$. The average runtime of the high-level path planner is $0.02\,\mathrm{s}$.

As demonstrated by the experimental results, the proposed planning strategy enables magnetic microswarms to autonomously execute cargo delivery tasks with $\mathrm{LTL_f}$ specifications in both static and dynamic environments.

VI. Conclusion

In this work, we propose a formal planning strategy for magnetic microswarms to perform cargo delivery tasks in complex environments with $\mathrm{LTL_f}$ specifications. Unlike traditional point-to-point path planning problems, the proposed BTL-RRT* algorithm rapidly identifies a feasible high-level path while ensuring the correctness of the given $\mathrm{LTL_f}$ specification. Additionally, a mode planner based on a four-state automaton is developed to manage manipulation modes, enabling the microswarm to capture and release cargoes. The planning strategy is implemented on a vortex-like microswarm actuated by a visual-feedback magnetic tweezers system. Experiments demonstrate the effectiveness of the strategy, showing that the microswarms can autonomously select and deliver cargoes to the release region in both static and dynamic environments while adhering to $\mathrm{LTL_f}$ specifications.

This work advances the full autonomy of magnetic microrobotic systems for micromanipulation tasks with rich specifications. In future work, we will explore learning-based methods to enable magnetic microrobots to achieve efficient and safe reactive high-level planning in dynamic environments.

References

[1] J. Law, X. Wang, M. Luo, L. Xin, X. Du, W. Dou, T. Wang, G. Shan, Y. Wang, P. Song, X. Huang, J. Yu, and Y. Sun, "Microrobotic swarms for selective embolization," *Science Advances*, vol. 8, no. 29, p. eabm5752, 2022.

[2] C. L. Smart, T. G. Pearson, Z. Liang, M. X. Lim, M. I. Abdelrahman, F. Monticone, I. Cohen, and P. L. McEuen, "Magnetically programmed diffractive robotics," *Science*, vol. 386, no. 6725, pp. 1031–1037, 2024.

[6] J. Li, X. Li, T. Luo, R. Wang, C. Liu, S. Chen, D. Li, J. Yue, S. han Cheng, and D. Sun, "Development of a magnetic microrobot for carrying and delivering targeted cells," *Science Robotics*, vol. 3, no. 19, p. eaat8829, 2018.

[7] Z. Zeng, F. Wang, C. Li, M. Tan, S. Wang, and L. Feng, "Dung beetle optimizer-based high-precision localization for magnetic-controlled capsule robot," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 2347–2352.

[8] J. Yu, L. Yang, and L. Zhang, "Pattern generation and motion control of a vortex-like paramagnetic nanoparticle swarm," *The International Journal of Robotics Research*, vol. 37, no. 8, pp. 912–930, 2018.

[3] Q. Wang, K. F. Chan, K. Schweizer, X. Du, D. Jin, S. C. H. Yu, B. J. Nelson, and L. Zhang, "Ultrasound doppler-guided real-time navigation of a magnetic microswarm for active endovascular delivery," *Science Advances*, vol. 7, no. 9, p. eabe5914, 2021.

[4] Y. Kantaros, B. V. Johnson, S. Chowdhury, D. J. Cappelleri, and M. M. Zavlanos, "Control of magnetic microrobot teams for temporal micromanipulation tasks," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1472–1489, 2018.

[5] Y. Wang, Y. Xiong, K. Fang, and J. Yu, "Millipede-inspired multi-legged magnetic soft robots for targeted locomotion in tortuous environments," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 1406–1411.

[9] H. Xie, M. Sun, X. Fan, Z. Lin, W. Chen, L. Wang, L. Dong, and Q. He, "Reconfigurable magnetic microrobot swarm: Multimode transformation, locomotion, and manipulation," *Science Robotics*, vol. 4, no. 28, p. eaav8006, 2019.

[10] L. Yang, J. Yu, S. Yang, B. Wang, B. J. Nelson, and L. Zhang, "A survey on swarm microrobotics," *IEEE Transactions on Robotics*, vol. 38, no. 3, pp. 1531–1551, 2022.

[11] J. Jiang, L. Yang, and L. Zhang, "An overview of micro/nanorobot swarm control: From fundamental understanding to autonomy," *IEEE/ASME Transactions on Mechatronics*, pp. 1–17, 2024.

[12] L. Yang, J. Jiang, X. Gao, Q. Wang, Q. Dou, and L. Zhang, "Autonomous environment-adaptive microrobot swarm navigation enabled by deep learning-based real-time distribution planning," *Nature Machine Intelligence*, no. 5, p. 4, 2022.

[13] J. Jiang, L. Yang, and L. Zhang, "Automated microrobotic manipulation with micron-scale precision using multimodal magnetic microswarms," in *2023 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 2023, pp. 19–24.

[14] Q. Wang, L. Yang, and L. Zhang, "Micromanipulation using reconfigurable self-assembled magnetic droplets with needle guidance," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 2, pp. 759–771, 2022.

[15] J. Jiang, L. Yang, B. Hao, T. Xu, X. Wu, and L. Zhang, "Automated microrobotic manipulation using reconfigurable magnetic microswarms," *IEEE Transactions on Robotics*, vol. 40, pp. 3676–3694, 2024.

[16] A. Pnueli, "The temporal logic of programs," in *18th annual symposium on foundations of computer science (sfcs 1977)*. ieee, 1977, pp. 46–57.

[17] X. Yin, B. Gao, and X. Yu, "Formal synthesis of controllers for safety-critical autonomous systems: Developments and challenges," *Annual Reviews in Control*, vol. 57, p. 100940, 2024.

[18] S. Karaman and E. Frazzoli, "Sampling-based motion planning with deterministic mu-calculus specifications," in *2009 48h IEEE Conference on Decision and Control (CDC)*, 2009, pp. 2222–2229.

[19] C. I. Vasile, X. Li, and C. Belta, "Reactive sampling-based path planning with temporal logic specifications," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 1002–1028, Jul. 2020.

[20] X. Luo, Y. Kantaros, and M. M. Zavlanos, "An abstraction-free method for multirobot temporal logic optimal control synthesis," *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1487–1507, 2021.

[21] L. Zheng, H. Ji, and D. Sun, "Automated manipulation of microswarms without real-time image feedback using magnetic tweezers," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 6, pp. 5712–5723, 2022.

[22] G. De Giacomo and M. Y. Vardi, "Linear temporal logic and linear dynamic logic on finite traces," in *2013 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, 2013, pp. 854–860.

[23] F. Fuggitti, "LTLf2DFA," March 2019. [Online]. Available: http://ltlf2dfa.diag.uniroma1.it