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

Bayesian-Guided Evolutionary Strategy with RRT for Multi-Robot Exploration

Shuge Wu[1,*], Chunzheng Wang[2,*], Jiayi Pan[1], Dongming Han[2] and Zhongliang Zhao[1,3,†]

Abstract— **With the increasing demand for multi-robot exploration of unknown environments, how to accomplish this problem efficiently has become a focus of research. However, in this kind of task, the formulation of strategies for frontier point detection and task allocation largely determines the overall efficiency of the system. In the task of multi-robot exploration of unknown environments, the strategies of frontier point detection and task assignment determine the overall efficiency of the system. Most of the existing methods implement frontier point detection based on the Rapidly-Exploring Random Tree (RRT) and use greedy algorithms for task allocation. However, the classical RRT algorithm is a fixed growth step, which leads to the difficulty of growing branches in narrow environments, making the efficiency and correctness of detecting frontier points lower. Meanwhile, the allocation strategy of the greedy algorithm causes each robot to consider only the exploration area with the largest gain for itself, which easily leads to repeated exploration and reduces the overall efficiency of the system. To solve these problems, we propose an adaptive RRT tree growth strategy for frontier point detection, which can adjust the step size according to the known map information and thus improve the efficiency and accuracy of detection; and introduce a Bayesian-guided evolutionary strategy(BGE) for efficient task allocation, which can utilize the current and historical information to find the optimal allocation scheme in a global perspective. We conduct a comprehensive test of the proposed strategy in the ROS system as well as in the real world, which proves the efficiency of our strategy. Our code is open-sourced and can be provided under request.**

I. INTRODUCTION

As science and technology advance, multi-robot systems play an increasingly vital role in diverse scenarios, from deep-sea and Martian exploration to urban disaster response and forest fire monitoring. These systems can swiftly cover vast areas and collaborate for enhanced efficiency, especially in hazardous or inaccessible human environments. A primary challenge is enabling robots to explore unknown terrains, a hot research topic.

Addressing this exploration enhances multi-robot system efficiency and supports future missions. Several exploration strategies exist [1] [2] [3] [4], with frontier-based exploration being popular due to its probabilistic completeness and focus on uncharted areas [5] [6]. This strategy mainly involves two phases: frontier detection and task assignment. Detection relies on sensors (like cameras or radars) and algorithms (like RRT). Detected frontiers guide robots in map-building. Task allocation, the second phase, uses strategies like probability [8], machine learning [9], and auction theory [10] [11]. Efficient task allocation in robot teamwork is vital, but traditional strategies based on optimization or decision theory may falter in dynamic environments due to local optima. The challenges of multi-robot exploration can be broken down into the following lists:

This work was supported by the Defense Industrial Technology Development Program under Grant JCKY2021601B013.

[1] School of Electronic and Information Engineering, Beihang University, Beijing, 100191, PR China.
[2] School of Mechanical Engineering and Automation, Beihang University, Beijing, 100191, PR China.
[3] Department of Mathematics and Theories, Peng Cheng Laboratory, Shenzhen, 518055, PR China
∗ Authors with equal contribution.
† Corresponding Author, zhaozl@buaa.edu.cn

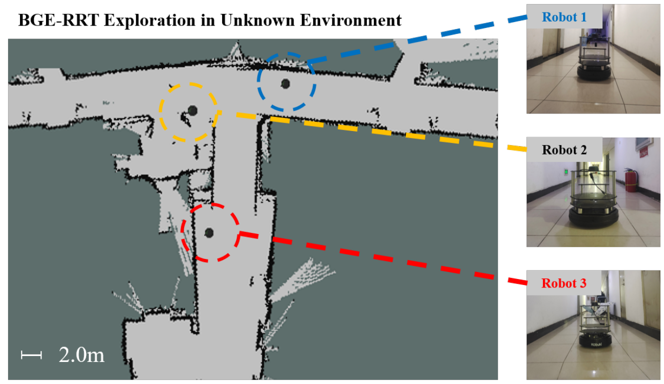


Fig. 1. Example of BGE-RRT exploration deployed in three Turtlebots

For multi-robot exploration, frontier detection often employs RRT trees. Yet, many studies [12] [13] fix the growth step length based on the environment and engineering experience, which isn't universally applicable. As exploration progresses, early step length settings may not suit later stages, leading to challenges in detecting frontier points or detecting incorrect ones. Thus, the RRT tree's growth step significantly impacts frontier detection efficiency.

Furthermore, during multi-robot exploration task assignments, considering inter-robot information is crucial for optimal coordination. The commonly used greedy strategy [1] is straightforward but tends to prioritize short-term gains over system-wide long-term optimization. While machine learning-based methods [9] and those rooted in auction and market theory [10] [11] show promise, their computational and communication demands can compromise system real-time performance.

We propose two strategies to meet exploration challenges. The adaptive strategy, responsive to the dynamic exploration context, modifies the RRT step size according to map data, improving frontier detection. For task assignment, we scale our approach: a speedy optimized greedy algorithm for smaller issues, and for greater complexity, we shift to the Bayes-guided evolutionary strategy (BGE), blending Bayesian optimization [14] [15]'s predictive power with genetic algorithms [16]' extensive search. These methods enable quick, robust robot tasking, demonstrated in our BGE-RRT physical trials (refer to Fig. 1).

In conclusion, the contribution of this paper is as follows:

1) We propose an adaptive RRT tree growth approach for frontier detection that adjusts step size using frontier points and map data, unlike traditional RRT trees. Initially, it employs a larger step size for swift exploration and gradually reduces it for precision as environmental knowledge grows, optimizing exploration efficiency and results throughout.

2) We propose a Bayes-guided evolutionary strategy where Bayesian optimization provides a high-quality start point for the genetic algorithm, and the genetic algorithm optimizes on the basis of this start point to find the best frontier point assignment. This combination method makes full use of the advantages of the two algorithms to achieve efficient and accurate frontier point allocation.

3) We introduce an adaptive task allocation method that chooses between Bayes-guided evolutionary strategies or greedy algorithms depending on the problem size. For smaller problems, we employ a greedy algorithm for quick responses, while larger scales utilize the Bayes-guided evolution strategy for global optimization. This offers an efficient and versatile allocation approach for multi-robot systems.

This paper is organized as follows. Section II describes the overall system architecture. Section III specifically describes the strategies used in the exploration process. Section IV gives the simulation and experimental setup along with specific results. The paper is summarized in Section V.

II. SYSTEM ARCHITECTURE

In this section, we describe the system architecture of BGE-RRT in detail. The architecture consists of a central computer and robot formation. The overall exploration strategy can be divided into four main parts, which are frontier point detection, frontier point filtering, frontier point assignment, and map merging.

From Fig. 2, it's evident that the robot formation operates in a distributed manner. On each individual robot, the local RRT frontier detection module, SLAM module, and navigation module are executed. Meanwhile, the central computer handles the global RRT frontier detection, frontier filter, frontier assignment, and map merging modules.

In our multi-robot system, a local RRT frontier detection module running on each robot and a global RRT frontier detection module running on a central computer are used to detect frontier points. The detected frontier points are sent to each robot through the filter module and assignment module, and then the robot will move to the target point to realize the process of exploration and mapping.

All robots and central computers are connected to the same LAN [17]. The ROS's topic and action mechanisms are used for communication, ensuring real-time and consistent data. Each robot has its unique namespace to avoid topic conflicts [18].

III. STRATEGY DESCRIPTION

In this section, we delve into the exploration process elaborated in Section II, encompassing four core components: frontier detection, frontier filtering, frontier assignment, and map merging. The frontier filter clusters detected frontier points, enhancing the precision of the detection. The map merging process integrates sub-maps from various robots to form a global map. Given that the functionalities of these two components align with the benchmark literature [1], our primary emphasis is on refining the frontier detection and frontier assignment modules.

A. Frontier Detection Module

The system's frontier detection module discerns unexplored boundaries between known and unknown areas. The global detector examines the entire map, while the local one focuses on the robot's immediate surroundings, both using the RRT algorithm.

In the global detector, the RRT randomly selects a point for each iteration. If it's in an unknown zone, the RRT expands towards the nearest known point. Successful expansions yield new RRT nodes. Nodes near uncharted areas become frontier points, directing the robot to new territories.

Traditional RRT exploration uses a consistent growth step. Initially, a larger step is favored due to sensor constraints. But as the robot collects data, this step might miss detailed areas or mislabel frontiers. We suggest an adaptive growth method adjusting the RRT's expansion rate based on the mapped region's size. Algorithm 1 details this approach. This method employs the following functions:

Nearest Node in RRT Tree: Given a point p_{rand} and an RRT tree T, the function $\text{NearestNode}(p_{\text{rand}}, T)$ returns the node in the RRT tree that is closest to p_{rand} based on the Euclidean distance.

$$\textit{NearestNode}(p_{\text{rand}}, T) = \arg\min_{p\in T} \|p - p_{\text{rand}}\|_2 \tag{1}$$

Adaptive Growth Rate Calculation: Given the RRT tree T and the known map M, the function $\textit{AdaptiveStepLength}(T, M)$ calculates the step length λ based on the following formula:

$$\lambda = \lambda_0 \cdot (1 - r) \tag{2}$$

Where λ_0 is a predefined initial step length and r is the ratio of the area covered by the RRT tree to the total known area of the map.

$$r = \epsilon \cdot \frac{S_T}{S_M} \tag{3}$$

$$S_T = n \cdot \lambda \tag{4}$$

Where S_M represents the currently known area of the map, S_T represents the area covered by the global tree, n represents the number of nodes in the global tree, and ϵ represents a constant weight that adjusts the order of magnitude between S_M and S_T.

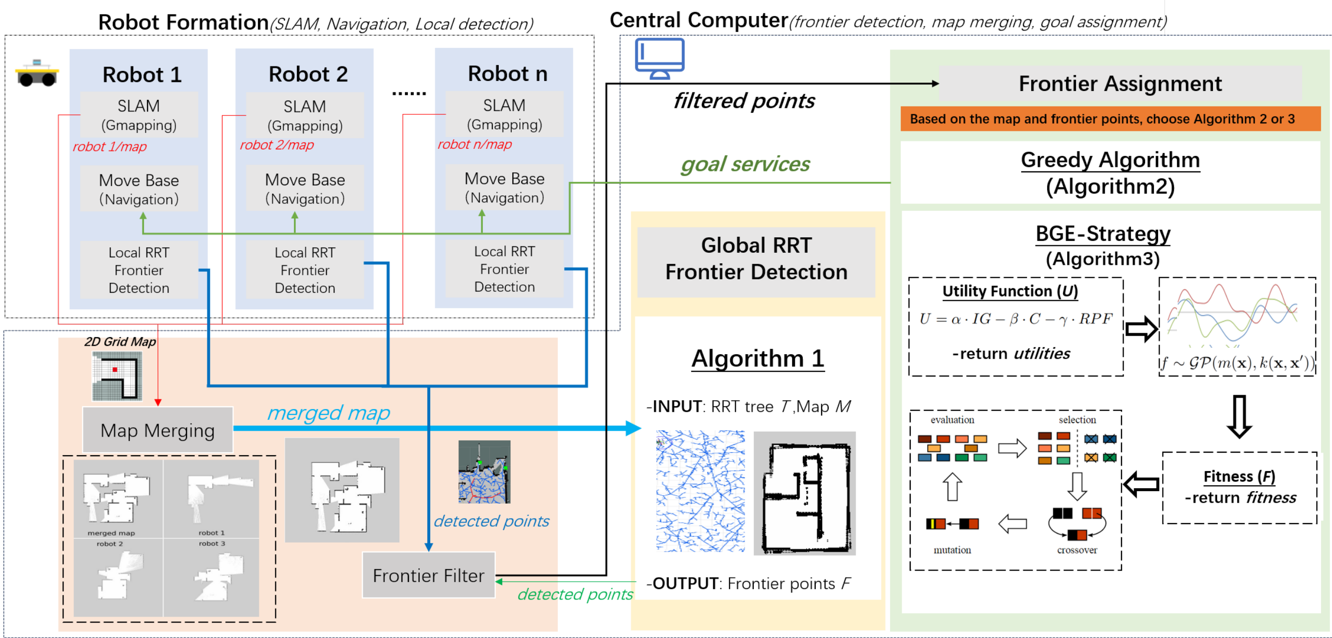


Fig. 2. System Architecture

The *Adaptive Global Frontier Detector* integrates the RRT algorithm with an adaptive growth strategy for optimized exploration. A point p_{rand} is sampled from map M. The closest node in T to p_{rand} is found using *NearestNode*. The *AdaptiveStepLength* function adjusts step length λ based on exploration status. This ensures rapid traversals in new zones and detailed study in known regions. If no collision, p_{new} is added to T. Edge points of known space are frontier points. This method enhances detection efficiency and accuracy in complex areas.

Algorithm 1 Adaptive Global Frontier Detector

Input: Map M, RRT tree T
Output: Frontier points P_f
while exploration not complete **do**
  $p_{\text{rand}} \leftarrow$ Random point from M
  $p_{\text{nearest}} \leftarrow \text{NearestNode}(p_{\text{rand}}, T)$
  $\lambda \leftarrow \text{AdaptiveStepLength}(T, M)$
  $p_{\text{new}} \leftarrow$ New point in the direction of p_{rand} from p_{nearest} with distance λ
  if no collision between p_{nearest} and p_{new} **then**
    Add p_{new} to T
    if p_{new} is on the frontier of known space **then**
      Add p_{new} to P_f
return P_f

B. Frontier Assignment Module

In our task assignment framework, we use a dual-algorithm approach, contrasting the standard Greedy method with the advanced BGE. BGE combines Bayesian Optimization for predictive analytics and Genetic Algorithms for extensive search, ensuring efficient and comprehensive decision-making.

1) Greedy Algorithm: The Greedy algorithm is a decision-making approach that prioritizes immediate benefits without considering the broader context. In the context of robotic frontier exploration, our Greedy algorithm is designed to assign each robot to the frontier that offers the highest immediate utility [19].

The utility of a frontier is determined by factors such as the distance from the robot and the potential information gained from exploring it. Specifically, the utility function U for a robot r and a frontier f is defined as:

$$U(r, f) = w_1 \cdot \text{InfoGain}(f) - w_2 \cdot \text{Distance}(r, f) \tag{5}$$

Where:

- $\text{InformationGain}(f)$ represents the expected amount of new information that can be obtained by exploring frontier f.
- $\text{Distance}(r, f)$ is the Euclidean distance between robot r and frontier f.
- w_1 and w_2 are weights that determine the relative importance of information gain and distance, respectively.

The Greedy algorithm then assigns each robot to the frontier with the highest utility value. The exact algorithm is described in Algorithm 2.

The key aspect of the proposed greedy algorithm is the utility function, which considers both the cost and potential information gain. This allows the robot to select the most cost-effective target and maximize short-term exploration benefits. The algorithm is primarily designed to solve small-scale problems in the early stages, with lower computational costs enabling the system to respond quickly and establish a strong starting point for the subsequent BGE strategy.

Algorithm 2 Greedy Frontier Assignment

procedure GREEDYASSIGNMENT(Robots, Frontiers)
  for each $robot$ in $Robots$ **do**
    $max\_utility \leftarrow -\infty$
    $best\_frontier \leftarrow$ None
    for each $frontier$ in $Frontiers$ **do**
      $utility \leftarrow U(robot, frontier)$
      if $utility > max\_utility$ **then**
        $max\_utility \leftarrow utility$
        $best\_frontier \leftarrow frontier$
    Assign $robot$ to $best\_frontier$

2) Bayesian-Guided Evolutionary Strategy: Our strategy integrates Bayesian optimization with genetic algorithms for efficient multi-robot exploration in unknown terrains. Bayesian optimization, a probabilistic global optimization method, predicts the potential value of frontier points, offering an initial search space for the genetic algorithm. The genetic algorithm then refines this by seeking the best robot-to-frontier point assignment, optimizing the allocation through simulated natural selection until a near-optimal solution is found.

Bayesian Optimization: Bayesian optimization, anchored by a Gaussian Process (GP) model, efficiently optimizes functions with high computational evaluation costs. Constructing a probabilistic model, not only forecasts function values but also judiciously selects the next sampling point, balancing exploration and exploitation. Within this project's scope, Bayesian optimization facilitates frontier utility assessments for robots, using historical data to estimate utility and reduce computational demands. The Utility Function, central to this method, quantifies exploration benefits, encompassing factors like expected information, travel cost, and robot avoidance, ensuring well-informed robot decisions.

The following are some definitions related to the utility function:

- **Information Gain (IG):** Given a frontier point f, the information gain is the potential amount of new information that a robot could obtain upon reaching that point. It can be computed based on the unexplored region around the frontier.
- **Cost (C):** Given a robot's current position x and a frontier point f, the cost is the effort required for the robot to reach that frontier.

$$C(x, f) = \|x - f\|_2 \tag{6}$$

- **Repulsive Potential Field (RPF):** Represents the influence of robots on each other to avoid them heading to the same frontier point.

$$RPF\left(x, y, r\right) = \sum_{i=1}^{N} \left(\frac{1}{\|x - y_i\|} - \frac{1}{r} \right) \cdot \frac{1}{\|x - y_i\|^2} \tag{7}$$

where x represents the robot's current position, y represents the position of the other robots, and r represents the effective range of the repulsive potential field.

- **Utility Function (U):** The utility of a frontier point f for a robot at position p is a combination of the information gain, cost, and repulsive potential field.

$$\begin{aligned} &U\left(x, y, \text{mapData}, f, r\right) = \alpha \cdot IG\left(f, \text{mapData}\right) \\ &-\beta \cdot C\left(x, f\right) - \gamma \cdot RPF\left(x, y, r\right) \end{aligned} \tag{8}$$

where α, β, and γ are weight coefficients.

$$U = \alpha \cdot IG - \beta \cdot C - \gamma \cdot RPF \tag{9}$$

Bayesian optimization employs Gaussian Processes (GPs) to model unknown objective functions. A GP is a non-parametric method that defines a distribution over functions, characterized by its mean function $m(x)$ and covariance function (or kernel) $k(x, x')$. Specifically, a GP is a collection of random variables, with any finite subset having a joint Gaussian distribution.

- Gaussian Process Prior:

$$f \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$$

Where:
  - $m(x)$ is the mean function, usually set to zero for simplicity.
  - $k(x, x')$ is the covariance function, which measures the similarity between x and x'.
- Kernel Function

$$k(x, x') = \sigma^2 \exp\left(-\frac{|x - x'|^2}{2l^2} \right) \tag{10}$$

Where:
  - σ^2 is the variance.
  - l is the length scale, determining the smoothness of the function.
- Posterior Distribution given data $D = \{X_{\text{history}}, Y_{\text{history}}\}$:

$$\begin{aligned} f(\mathbf{x})|D &\sim \mathcal{GP}(m_D(\mathbf{x}), k_D(\mathbf{x}, \mathbf{x}')) \\ m_D(\mathbf{x}) &= m(\mathbf{x}) + k(\mathbf{x}, X_{\text{history}}) \\ &\quad \times [k(X_{\text{history}}, X_{\text{history}}) + \sigma^2 I]^{-1} \\ &\quad \times (Y_{\text{history}} - m(X_{\text{history}})) \\ k_D(\mathbf{x}, \mathbf{x}') &= k(\mathbf{x}, \mathbf{x}') - k(\mathbf{x}, X_{\text{history}}) \\ &\quad \times [k(X_{\text{history}}, X_{\text{history}}) + \sigma^2 I]^{-1} \\ &\quad \times k(X_{\text{history}}, \mathbf{x}') \end{aligned}$$

where $m(\mathbf{x})$ is the mean function, $k(\mathbf{x}, \mathbf{x}')$ is the kernel function, and σ^2 is the noise variance.

Upon observing data $D = \{X_{\text{history}}, Y_{\text{history}}\}$, the GP's posterior distribution is updated. The GP then predicts the objective function's value and uncertainty at new points. An acquisition function, typically using Expected Improvement, determines the next sampling point by considering both the predicted value and uncertainty. The GP updates with this new sample, and the process iterates. This procedure is detailed in Algorithm 3.

In summary, Bayesian optimization, with its inherent advantages and the use of the RBF kernel, provides a principled and efficient approach to frontier evaluation in the project, aiding in the decision-making process for robot exploration.

Algorithm 3 Bayesian Optimization

function BO($x, frontiers, y, mapdata$)
  $X_{\text{current}} \leftarrow$ matrix of current frontiers
  for each f in $frontiers$ **do**
    $Y_{\text{current}}[f] \leftarrow$ U$(x, y, mapData, f, r)$
  Update X_{history} and Y_{history} with X_{current} and Y_{current}
  Train Gaussian Process with X_{history} and Y_{history}
  $Y_{\text{pred}}, \sigma_{\text{pred}} \leftarrow$ predict utilities and uncertainties for X_{current} using Gaussian Process
  return Y_{pred}

Genetic Algorithm: Based on the utility value predicted by Bayesian optimization, a genetic algorithm is used to find the best allocation strategy among all possible frontier point allocations.

The algorithm operates on a set of individuals, denoted as $population$, which represent possible solutions. The utility values for these solutions are computed using Bayesian Optimization and are stored in $utilities$. The algorithm takes as input a list of available robots, represented as $robots$, and a list of available frontiers, represented as $frontiers$. The output of the algorithm is an $assignmentdict$, which is a dictionary where the keys are robots and the values are the assigned frontiers.

The fitness of an individual in the genetic algorithm is a measure of the quality of the robot-to-frontier assignments. It is computed based on the following criteria:

1) **Total Utility**: For each robot's assignment to a frontier, the utility value from the 'utilities' matrix is summed up to compute the total utility of the assignment strategy.
2) **Penalty for Proximity(Penalty)**: If the distance between two robots' frontiers is below a set threshold, a major penalty is deducted from the total utility, ensuring robots don't redundantly explore the same area.

Mathematically, the fitness F of an individual can be represented as:

$$F(\text{individual}) = \sum_{i=1}^{n} utilities[\text{robot}_i][\text{individual}[\text{robot}_i]] - \text{Penalty} \tag{11}$$

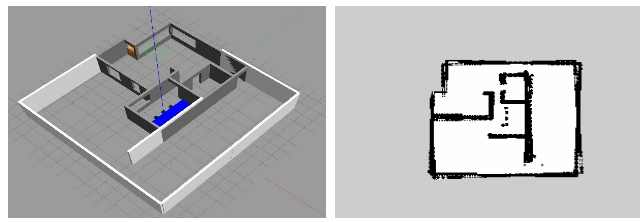
The most important steps in the genetic algorithm are crossover and mutation:

Crossover is a genetic algorithm operation that simulates the mating process in biological evolution. In this operation, two individuals are selected from the current population as parents, and their genes are combined to produce offspring. While mutation simulates the random gene mutations in biological evolution. In this operation, certain genes of an individual are altered based on a predefined mutation probability p_m.

Based on all the above descriptions, the proposed BGE strategy is shown in Algorithm 4.

Algorithm 4 BGE-Strategy

function BGE($robots, frontiers$)
  $population \leftarrow$ Initialize with random assignments of $frontiers$ to $robots$
  for each generation **do**
    for each $individual$ in $population$ **do**
      $fitness \leftarrow$ F($individual$)
    Select $individual$ for mating based on fitness
    Apply crossover and mutation to produce $Population'$
    $Population \leftarrow Population'$
  $best\_assignment \leftarrow$ highest fitness individual from $Population$
  Convert $best\_assignment$ to $assignment\_dict$
  return $assignment\_dict$



(a) Simulation environment (b) Results of occupancy grid map

Fig. 3. Multi-robot simulation environments

IV. SIMULATION AND EXPERIMENTAL RESULTS

Our proposed BGE-RRT strategy is compared with the benchmark model [1]. Both cases use the same frontier filtering module and map merging module, and the benchmark model requires the initial RRT tree growth step to be set as a parameter. Below we present a simulated map and a real map to verify our strategy in Fig. 3. For each map, The benchmark model runs 50 times each for a total of 350 times when the RRT tree growth steps are {0.5,1,2,4,6,10,15}, and the BGE-RRT strategy runs 50 times.

A. Simulation Setup And Results

1) Simulation Set Up: In the simulation process, we use the GAZEBO simulator, in which we can build robot simulation motion models, various scene models of the world, and sensor simulation models. The map area used in the simulation is about 182 square meters, the robot radius is 0.2 meters, and the 2D laser scanning distance is set to 40 meters.

2) Simulation Results: Fig. 4 reveals that, unlike the Benchmark strategy which often directs robots to redundant areas, BGE algorithm encourages robots to explore diverse regions, enhancing exploration efficiency. This advantage is evident in our reduced exploration time.

From Fig. 5, the BGE algorithm outperforms the benchmark by reducing time and distance by 28% and 45.9%, respectively. This efficiency ensures quicker task completion over shorter distances, a crucial advantage in real-world scenarios with time and energy constraints.

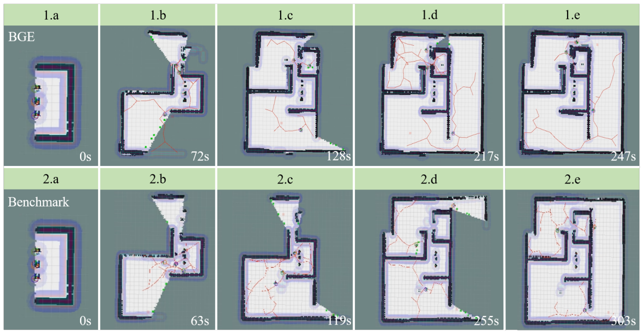
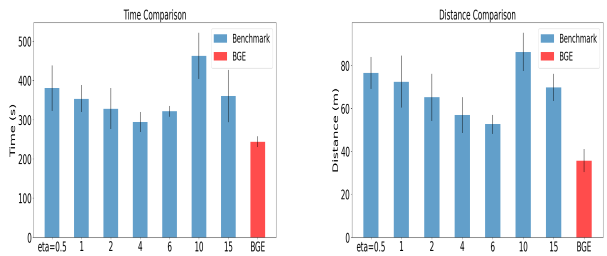


Fig. 4. Comparison of experimental results



(a) Time Comparison

(b) Distance Comparison

Fig. 5. Simulation of multi-robot results comparison

B. Experimental Setup And Results

1) Experimental Setup: We implemented the BGE-RRT strategy on three Turtlebot2 robots, each fitted with RPLidar sensors and a Kobuki mobile chassis. The system's computing infrastructure comprises four laptops: a host computer and three slave laptops mounted on the robots. Communication across these devices is facilitated by a WiFi 2.4GHz mesh network, anchored by a TP-Link router ensuring expansive network coverage in the test environment.

2) Experimental Results: Fig. 6 is the real scene as well as the floor plan during our physical experiment, and Fig. 7 is the result of the final build. The total area explored was about 350 square meters, the total time taken was 312 seconds, robot_1 traveled 53.03 meters, robot_2 traveled 33.06 meters, robot_3 traveled 24.73 meters, and the average distance traveled was 36.94 meters. It can be seen that there is some degradation and angular error in the merged map due to the inaccuracy of turtlebot2's odometer, but it can still be seen that our algorithm does assign the robots to different areas and avoids the problem of repeated exploration.

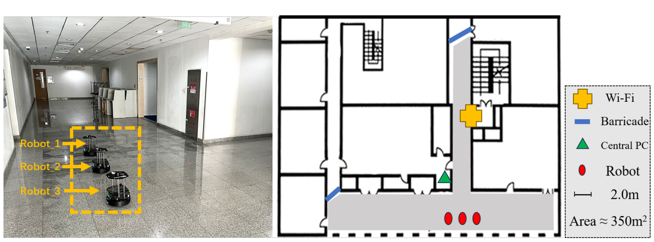


Fig. 6. Real maps and floor plans for physical experiments

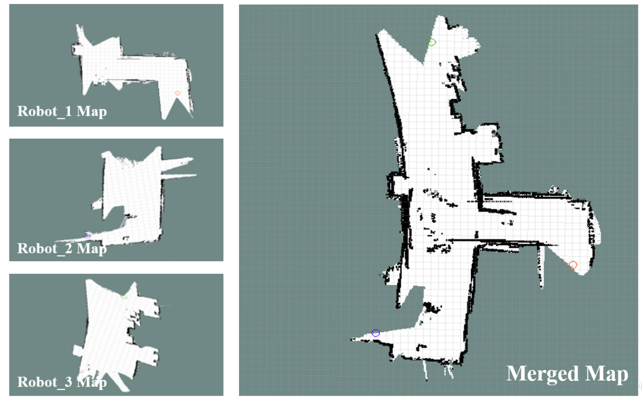


Fig. 7. Result of BGE-RRT exploration strategy

V. CONCLUSIONS

In this study, we present an innovative multi-robot exploration strategy based on the RRT algorithm that utilizes adaptive RRT trees to improve frontier point detection in various environments. Our BGE strategy combines Bayesian optimization and historical data to provide a solid foundation for a genetic algorithm with the goal of optimizing global task allocation. The combined use of the optimized greedy algorithm and the BGE strategy improves exploration efficiency. Through simulations and real-world testing, our method reduces exploration time by 28% and distance by 45.9%, consistently outperforming traditional RRT-based techniques to efficiently detect fronts and map regions.

For future endeavors, the BGE strategy can be further refined by:

1) Introducing map partitioning and applying the BGE strategy on sub-maps reduces the problem size and improves allocation efficiency through parallel solutions.

2) Extend this strategy to 3D exploration, incorporating terrain data and making the BGE strategy applicable to more complex terrain.

REFERENCES

[1] H. Umari and S. Mukhopadhyay, "Autonomous robotic exploration based on multiple rapidly-exploring randomized trees," 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 2017, pp. 1396-1402, doi: 10.1109/IROS.2017.8202319.

[2] Z. Tian, C. Guo, Y. Liu, and T. Cui, "Autonomous exploration of RRT robot based on seeded region growing," 2020 39th Chinese Control Conference (CCC), Shenyang, China, 2020, pp. 3936-3941, doi: 10.23919/CCC50068.2020.9188916.

[3] B. P. L. Lau et al., "Multi-AGV's Temporal Memory-Based RRT Exploration in Unknown Environment," in IEEE Robotics and Automation Letters, vol. 7, no. 4, pp. 9256-9263, Oct. 2022, doi: 10.1109/LRA.2022.3190628.

[4] D. Santosh, S. Achar and C. V. Jawahar, "Autonomous image-based exploration for mobile robot navigation," 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 2008, pp. 2717-2722, doi: 10.1109/ROBOT.2008.4543622.

[5] B. Yamauchi. (1998, May). Frontier-based exploration using multiple robots. In Proceedings of the second international conference on Autonomous agents (pp. 47-53).

[6] B. Yamauchi, "A frontier-based approach for autonomous exploration," Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation', Monterey, CA, USA, 1997, pp. 146-151, doi: 10.1109/CIRA.1997.613851.

[7] R. K. Williams, A. Gasparri and G. Ulivi, "Decentralized matroid optimization for topology constraints in multi-robot allocation problems," 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 293-300, doi: 10.1109/ICRA.2017.7989038.

[8] Zhang, Liwei, et al. "Rapidly-exploring Random Trees multi-robot map exploration under optimization framework." Robotics and Autonomous Systems 131 (2020): 103565.

[9] F. Niroui, K. Zhang, Z. Kashino and G. Nejat, "Deep Reinforcement Learning Robot for Search and Rescue Applications: Exploration in Unknown Cluttered Environments," in IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 610-617, April 2019, doi: 10.1109/LRA.2019.2891991.

[10] C. Nam and D. A. Shell, "Assignment Algorithms for Modeling Resource Contention in Multirobot Task Allocation," in IEEE Transactions on Automation Science and Engineering, vol. 12, no. 3, pp. 889-900, July 2015, doi: 10.1109/TASE.2015.2415514.

[11] Zlot R, Stentz A. Market-based Multirobot Coordination for Complex Tasks. The International Journal of Robotics Research. 2006;25(1):73-101. doi:10.1177/0278364906061160

[12] C. -Y. Wu and H. -Y. Lin, "Autonomous Mobile Robot Exploration in Unknown Indoor Environments Based on Rapidly-exploring Random Tree," 2019 IEEE International Conference on Industrial Technology (ICIT), Melbourne, VIC, Australia, 2019, pp. 1345-1350, doi: 10.1109/ICIT.2019.8754938.

[13] E. Sumer and H. Temeltas, "RRT Based Frontier Point Detection for 2D Autonomous Exploration," 2022 7th International Conference on Robotics and Automation Engineering (ICRAE), Singapore, 2022, pp. 305-311, doi: 10.1109/ICRAE56463.2022.10056191.

[14] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the Human Out of the Loop: A Review of Bayesian Optimization," in Proceedings of the IEEE, vol. 104, no. 1, pp. 148-175, Jan. 2016, doi: 10.1109/JPROC.2015.2494218.

[15] S. Greenhill, S. Rana, S. Gupta, P. Vellanki and S. Venkatesh, "Bayesian Optimization for Adaptive Experimental Design: A Review," in IEEE Access, vol. 8, pp. 13937-13948, 2020, doi: 10.1109/ACCESS.2020.2966228.

[16] Mitchell, Melanie. An introduction to genetic algorithms. MIT press, 1998.

[17] Martin, J., Chapman, K. K., & Leben, J. (1994). Local area networks architectures and implementations. Prentice-Hall, Inc.

[18] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., A. Y. (2009, May). ROS: an open-source Robot Operating System. In ICRA workshop on open source software (Vol. 3, No. 3.2, p. 5).

[19] Vince, A. (2002). A framework for the greedy algorithm. Discrete Applied Mathematics, 121(1-3), 247-260.