

Distributed maze exploration using multiple agents and optimal goal assignment

Manousos Linardakis, Iraklis Varlamis, Georgios Th. Papadopoulos

Abstract—Robotic exploration has long captivated researchers aiming to map complex environments efficiently. Techniques such as potential fields and frontier exploration have traditionally been employed in this pursuit, primarily focusing on solitary agents. Recent advancements have shifted towards optimizing exploration efficiency through multiagent systems. However, many existing approaches overlook critical real-world factors, such as broadcast range limitations, communication costs, and coverage overlap. This paper addresses these gaps by proposing a distributed maze exploration strategy (CU-LVP) that assumes constrained broadcast ranges and utilizes Voronoi diagrams for better area partitioning. By adapting traditional multiagent methods to distributed environments with limited broadcast ranges, this study evaluates their performance across diverse maze topologies, demonstrating the efficacy and practical applicability of the proposed method. The code and experimental results supporting this study are available in the following repository: <https://github.com/manousulinard/multiagent-exploration/>.

Index Terms—Cost-utility, distributed maze exploration, multiagent, potential fields, Voronoi partitioning.

I. INTRODUCTION

EXPLORATION of mazes has interested researchers in the recent years, offering a window into the cognitive capabilities of various organisms, such as mice [1], and more recently, the artificial intelligence capabilities of robotic systems [2], [3]. While algorithms specifically designed for single-agent maze exploration are widely utilized [4], [5], the adaptation of these algorithms to multiagent systems [6]–[8] presents persistent challenges, underscoring the necessity for novel solutions.

The task of mapping unknown mazes for exploration necessitates strategic methodologies. Robots traverse the maze, documenting their journey and observations to construct a map as they advance [9], [10]. By employing sensors, they identify walls and obstacles, enabling them to make informed decisions regarding their subsequent movements. Upon mapping a significant portion of the maze, robots strategize their routes towards their objectives, which could be reaching the exit or locating specific targets within the maze. This process involves

Research was possible due to the funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101073876 (Ceasefire).

Manousos Linardakis is with the Department of Informatics and Telematics, Harokopio University of Athens, Athens, Greece (e-mail: manousulinard@gmail.com, it22064@hua.gr). ORCID: 0009-0001-7060-848X.

Iraklis Varlamis is with the Department of Informatics and Telematics, Harokopio University of Athens, Athens, Greece (e-mail: varlamis@hua.gr). ORCID: 0000-0002-0876-8167.

Georgios Th. Papadopoulos is with the Department of Informatics and Telematics, Harokopio University of Athens, Athens, Greece (e-mail: g.th.papadopoulos@hua.gr). ORCID: 0000-0003-1686-421X.

analyzing the constructed map and selecting the most efficient path forward.

Using multiple agents to explore the maze expedites the mapping process and enhances the coverage of the resulting map. However, this approach introduces challenges, including the need for effective agent communication to share information and organize exploration to prevent collisions or obstructions in their paths [11]–[13]. Additionally, multiagent exploration may lead to revisiting previously explored areas, extending total mapping time. Distributing the maze exploration task to multiple agents offers a solution to these problems, as robots can first explore their assigned regions individually, reducing redundant coverage, the number of collisions, and the need for extensive communication compared to exploring the entire maze map collectively.

There are several methods to partition a maze into sub-areas, including clustering algorithms such as K-means, which typically suffer from local optima and potentially lead to higher robot dispersion [14]. Other approaches, like the Effective Regions of Movement (ERM) [15], [16], are available but require additional information (e.g., agent capacity, density, and speed), which is not easily acquired when exploring an unknown maze. Moreover, ERM is applied to known maps and is typically used for enhancing and understanding networks, such as those involving cars. In this paper, we employ Voronoi partitioning [17], [18] centered around each agent, due to its suitability for unknown areas. Voronoi partitioning has been proven effective for segmenting 2D spaces, as demonstrated by Wu *et al.* [19], who compared it with K-means for area coverage. Their results showed that Voronoi partitioning was consistently faster than K-means across all examined maps.

Our proposed approach, named CU-LVP, is designed to address the distributed maze problem. Experiments detailed in Section V showcase the method's efficiency, low computational complexity, minimal communication, and reduced repetitive coverage. We compare our method with other state-of-the-art techniques adapted to the distributed maze problem. To evaluate performance, we conduct extensive experimental evaluations across a variety of maze topologies, including 30x30 randomly generated mazes with varying obstacle densities. These experiments prove CU-LVP's efficacy and provide a comprehensive assessment of each method's performance. They offer valuable insights into their effectiveness in exploring distributed mazes and their practical applicability in real-world robotic systems. Additionally, by integrating constraints on broadcast ranges across all evaluated methods, we aim to emulate real-world communication limitations. By incorporating these constraints, we enhance the robustness of the

exploration strategies, making them more applicable to real-world scenarios.

The main contributions of this paper can be summarized as follows:

- A detailed review and classification of the recent exploration approaches using multiple agents.
- An introduction to the novel method CU-LVP, designed for covering distributed mazes efficiently.
- An extensive implementation of established multiagent maze exploration techniques, applied in distributed mazes through the integration of Voronoi diagrams.
- A comparative evaluation of the algorithms, using various metrics that examine different aspects of the exploration task (e.g. time, distance, computational and communication cost).

Section II offers an overview of related work in state-of-the-art multiagent maze and area exploration. Section III details the proposed approach and Section IV explains the experimental evaluation process that we applied. Section V examines the achieved results, whereas Section VI presents the main conclusions of this work and future work.

II. RELATED WORK

Exploring the field of multiagent maze exploration encompasses a variety of innovative strategies and methodologies. This section categorizes and reviews significant contributions, highlighting the implementation details of each approach. We begin with distributed area exploration techniques, which emphasize on area/maze partition for efficient task allocation and coordination among agents. Next, we examine potential field methods, where agents use potential/temperature fields for maze coverage. Finally, we discuss frontier-based exploration, where the exploration process is based on current frontiers—the unexplored boundaries of the known environment. Fig. 1 organizes the related work in a hierarchical structure for enhanced clarity and comparison.

A. Distributed Area Methods

Lozenguez *et al.* [20] introduce a method for efficiently allocating exploration tasks among multiple mobile robots. In the context of this paper, we will refer to this method as RMDP. The approach uses a combination of Road-Map techniques and Markovian Decision Processes (MDPs) to manage the exploration of areas marked by points of interest. By representing spatial knowledge as a dynamically adjustable graph of waypoints and partitioning this graph into regions, the method enables optimal task assignments based on real-time evaluations. As part of the R-Discover project, this research supports the use of UAVs for initial mapping, followed by ground robots for detailed exploration. Robots operate under a hierarchical control system, with leaders dynamically reallocating tasks to maximize efficiency during the mission.

Yan *et al.* [21] proposed MUI-TARE to address the intricate task of multiagent exploration in confined 3D spaces with unknown initial agent poses. Their work focuses on efficiently exploring these environments and effectively merging agents' sub-maps. Traditional strategies often struggle with merging

sub-maps due to false overlap detections, leading to inaccuracies. To combat this, the authors propose a novel lidar-based approach that intelligently balances sub-map merging robustness and exploration efficiency. The approach allows agents to adaptively repeat each other's trajectories based on merging quality, minimizing redundant exploration. Additionally, it extends a single-agent hierarchical strategy to multiple agents, enhancing exploration efficiency. Experimental results demonstrate efficiency and exploration accuracy over baselines, ensuring robust sub-map merging in complex 3D environments.

Gui *et al.* [22] tackled the challenge of efficient distributed exploration for UAVs, particularly in search and rescue scenarios. They present a novel cooperative strategy where UAVs dynamically explore distinct regions, enhancing efficiency while minimizing redundancy. Using a dynamic centroid-based approach, the 3D space is partitioned for each UAV, enabling independent target generation within their designated zones. They propose a next-best-view method employing a rapidly exploring random tree (RRT) to foster cooperative exploration and navigate unknown areas. Comparative evaluations against three classical methods underscore the superiority of their approach in both simulation and real-world experiments, marking a significant advancement in swarm UAV capabilities for dynamic environments like search and rescue missions.

Latif *et al.* [23] introduced CQLite, a distributed Q-learning technique tailored to address the challenges of coordinating multiple mobile robots in the autonomous exploration of complex environments. Unlike conventional approaches reliant on internal global maps for navigation, CQLite prioritizes minimizing communication and information exchange costs between robots while ensuring swift convergence and coverage. By leveraging ad hoc map merging and selectively sharing updated Q-values at newly discovered frontiers, the method significantly reduces communication overhead. Theoretical analysis and numerical validation conducted on simulated indoor maps with multiple robots underscore the efficiency of the proposed approach.

B. Potential Fields Methods

Jincheng Yu *et al.* [24] introduced the Multirobot Multi-target Potential Field (MMPF), a novel approach that uses potential fields for maze exploration. The robots operate concurrently, utilizing a simultaneous localization and mapping (SLAM) technique for navigation and mapping tasks. Notably, MMPF does not initially implement any partitioning of the environment. Instead, the research emphasizes localization, dispersing agents across disparate areas of the environment. As agents subsequently explore overlapping regions, localization, and information exchange occur to facilitate comprehensive mapping of the area.

Crnkovic *et al.* [25], [26] proposed HEDAC, a robust algorithm designed for both area and maze exploration. By leveraging temperature fields to compute attractive forces, HEDAC excels in prioritizing the exploration of unknown regions while mitigating collisions. Specifically, it creates a strong attractive force for unexplored areas and a weaker

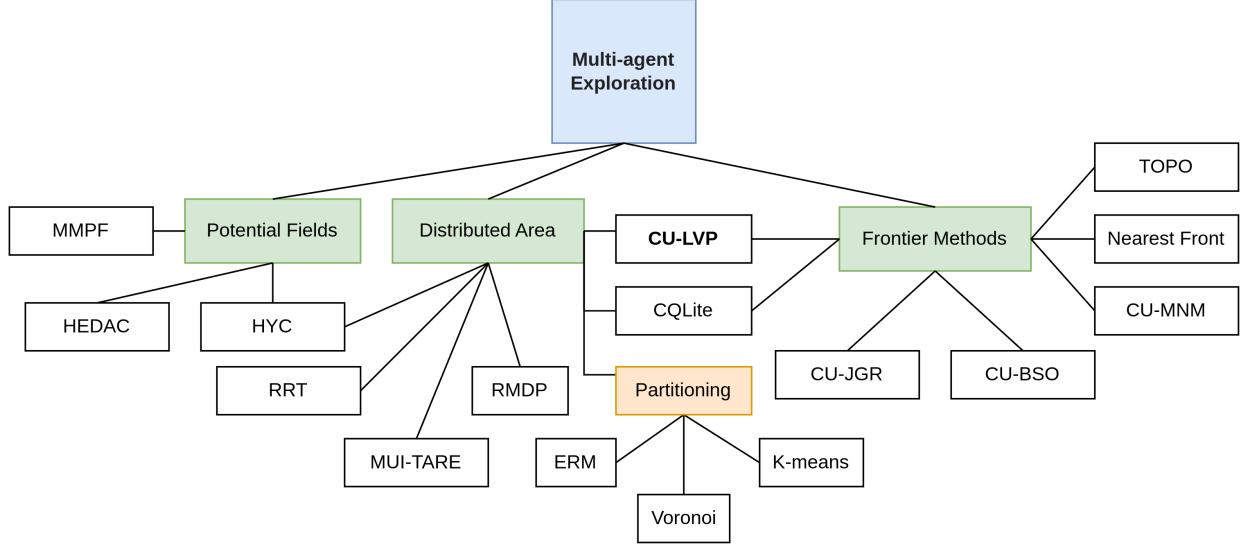


Fig. 1. Hierarchical Structure of the Related Work (Section II). Method in **bold** (CU-LVP) is our proposed method.

force for obstacles and other agents. HEDAC has demonstrated efficacy in uncertain conditions, such as variations in sensor readings and unpredictable environmental fluctuations, as documented in [27].

A similar approach to HEDAC that utilizes Voronoi diagrams to divide an area into subregions is presented by Zheng *et al.* [28], which we will refer to as HYC in this paper. In HYC, each subregion is explored by an agent using a temperature field-induced control strategy. For our study, we have implemented HEDAC instead of HYC because, although both methods use temperature fields, HEDAC is specifically designed for maze exploration with many or fewer obstacles, whereas HYC is not clearly implemented for such scenarios. The tested areas in the HYC paper are described as concave regions formed by removing small squares from larger ones, and the agent trajectories in the provided figures suggest that the tested areas are largely obstacle-free. Therefore, HEDAC has been deemed a more suitable choice for our task, given its effectiveness in both high and low-obstacle-density environments. Additionally, we have modified HEDAC to address the distributed maze problem, incorporating principles from the HYC paper, as detailed in Section IV-B.

C. Frontier Methods

Exploring multiagent systems has led to a variety of methods and algorithms, with the nearest-frontier method being notable for its simplicity and effectiveness. Initially proposed by Yamauchi [29], this method focuses on finding the shortest path to the closest frontier. Despite its efficiency, recent advancements in exploration strategies have introduced more sophisticated techniques, particularly those that use utility functions to enhance frontier selection. Cost-utility methods, for example, improve upon the deterministic nearest-frontier approach by incorporating additional utility functions, thus optimizing both exploration efficiency and decision-making processes.

Marjovi *et al.* [30] present a cost-utility approach, referred to here as CU-MNM, which effectively explores mazes while detecting fires. The utility function in CU-MNM is determined by the distance of the frontier from all robots. This decentralized approach promotes communication and collaborative decision-making among the robots.

Another cost-utility method, introduced by Julia *et al.* [31], uses the expected information gain in goal cells as the utility function to facilitate efficient exploration. For this paper, this method will be referred to as CU-JGR.

Bautin *et al.* [32] propose a computationally efficient frontier allocation method, referred to as CU-BSO in this paper, which encourages a balanced spatial distribution of robots within the environment. This method uses waveform propagation [33], originating from each frontier to quickly create a distance matrix (cost matrix) for all cells, and then selects frontiers with fewer nearby robots. The use of waveform propagation underscores reduced communication requirements among robots. An example of the waveform algorithm applied to a maze environment is shown in Fig. 2.

Beyond cost-utility functions, Zhang *et al.* [34] proposed the TOPO exploration method, designed to enhance multiagent exploration in communication-constrained environments. TOPO employs a frontier exploration strategy with a greedy approach, calculating the center of existing vertices whenever a new vertex is added. TOPO also constructs topological maps as robots navigate through the environment, reducing the data required for path planning and communication. Unlike traditional methods that depend on occupancy grid maps, TOPO significantly lowers data transfer requirements, as demonstrated in both simulated and real-world scenarios. When compared to the MMPF method [24], TOPO was shown to be faster in exploring the area through extensive experimentation.

Our approach (CU-LVP) contributes to the existing literature by introducing an efficient cost-utility method tailored to address the challenges of distributed maze coverage.

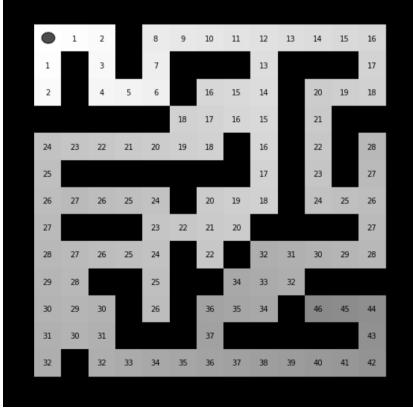


Fig. 2. Wavefront distance calculation for the entire maze. The gray dot indicates the agent's position, whereas each empty cell contains the distances from the agent. The black cells represent obstacles.

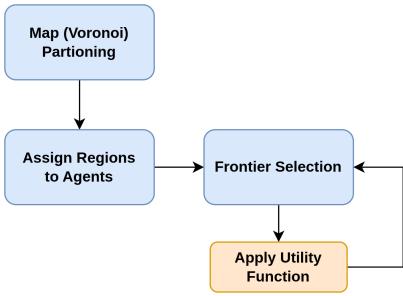


Fig. 3. Abstract overview of proposed methodology. Implementation details are provided in Section III.

Through rigorous experimentation, we have demonstrated that our method outperforms comparative approaches in terms of speed, achieving faster maze exploration while imposing minimal communication overhead, computational cost, and coverage overlap. Section V delves into this balance of metrics, showcasing results that validate CU-LVP's effectiveness.

III. PROPOSED METHOD

CU-LVP is a cost-utility approach that enhances frontier selection by combining modifications of previously implemented utility functions. It also effectively addresses the distributed maze problem by selecting frontiers assigned to each agent individually through Voronoi partitioning. Specifically, the process begins with map partitioning, where the maze is divided into subareas using Voronoi diagrams. Each agent is then assigned a specific subarea to explore. The next step involves frontier finding, where the best frontier is selected within each subarea, according to the utility function described in Section III-C. The shortest path to each frontier is then calculated, ensuring efficient navigation. Fig. 3 provides a high-level overview of our methodology, illustrating its application for distributed maze coverage. Before going into more details of the proposed approach it is necessary to define the main entity of our method, *the agent*.

A. Definition of the robotic agent

In our experiments, we define an agent class that represents the behavior of individual agents within a maze environment. The agents are capable of moving in four directions: up, down, left, and right. To avoid collisions, they are prohibited from moving into cells that are already occupied by other agents or obstacles. Each agent keeps a personal map of the areas it has explored in the maze. This map is continuously updated with information collected by other agents within broadcast range in each round. By sharing information about the explored sections of the maze, we simulate typical multiagent communication and behavior for maze exploration. Additionally, the agent's map is updated as it explores new areas within its sensor range, which, by default, extends two blocks in all orthogonal and diagonal directions. This simulates the sensing capabilities of real-world robots, which often have sensors on all sides. Moreover, the agent maintains a list of coordinates corresponding to its assigned Voronoi region earmarked for exploration. During each iteration, the agent employs the utility function described in Section III-C to choose a goal exclusively from its unexplored Voronoi coordinates. Once the agent discovers its designated Voronoi coordinate either autonomously or through information exchange with other agents, it redirects its focus to other (unexplored) Voronoi coordinates, thereby continuing its exploration trajectory.

Fig. 4 (in the bottom) shows an agent's explored map at the beginning, following initial information exchange between agents. The map is shown with a two-block view range, as configured in our experiments. If an obstacle is detected, the view beyond it is blocked to mimic real-sensor behavior. The agent class also stores essential parameters for exploration, including the agent's goal (marked with a red dot in Fig. 4), the next position along the shortest path to the target (indicated by a red line), the current position, and the view range. These parameters allow us to implement realistic agent behavior and exploration scenarios.

B. Voronoi Partitioning

CU-LVP relies on Voronoi partitioning to divide the maze, with each region centered around an agent. Consequently, the number of maze sub-regions in our experiments matches the number of agents. Agents are then tasked with exploring all cells within their assigned region. As they navigate the maze using the utility function described in Section III-C, agents within the broadcast range share information with each other, facilitating collaborative exploration. The broadcast range is defined as 25% of the maximum dimension of the maze. Furthermore, to expedite the exploration process, a mechanism is implemented to merge Voronoi regions when agents come into broadcast range. This merging facilitates a broader scope of exploration for agents, allowing them to explore the maze more effectively.

An example of the Voronoi partition process and broadcast range can be found in Fig. 4. On the top image, the starting Voronoi regions are depicted. Initially, all space in the maze is considered as free space, in order for the agents to explore it. The darker gray cells represent the obstacles that the agents

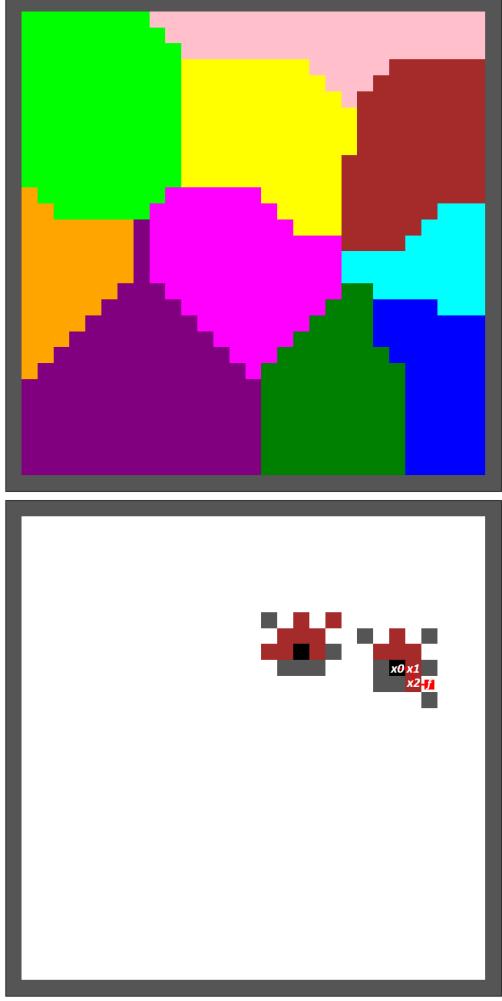


Fig. 4. Visualization of the initial Voronoi regions for a randomly 30x30 generated maze (top image). Agents within broadcast range (bottom image) have merged their Voronoi regions and exchanged information about their surroundings. Additionally, on the bottom, the red line depicts the waveform path (x_1, x_2) from an agent (x_0) to its goal (f) within its region. The black cells represent the agents. The colored cells depict the Voronoi regions, with each color representing a different partition, whereas the darker gray cells denote obstacles.

find along the way. Additionally, in the bottom image, the Voronoi regions of two agents (depicted as black cells) within the broadcast range have merged. This merging results in the region adopting a red hue, signifying consolidation into a single region. Furthermore, the two agents have exchanged information about their respective surroundings within a sensor range. It's worth noting that the agent's goal (depicted as a red dot in the bottom image) lies within its Voronoi region, signifying correct code execution.

Throughout the exploration process, agents periodically transmit their explored maps centrally, without pulling them. This allows for monitoring of exploration progress and facilitates experiment termination when the maze is explored. Upon completing the exploration of their designated region, agents move to the nearest unexplored Voronoi region, which is decided based on the central map.

Agents also possess the capability to traverse cells in neigh-

boring regions to access their next target within their Voronoi region, overcoming obstacles that might impede progress if they were to confine their movement solely within their designated area. Subsequently, they update their exploration map to reflect their progress. However, the primary focus remains on achieving complete coverage within the assigned region.

C. Goal selection

Initially, CU-LVP uses waveform propagation, starting from the agent's position, to efficiently calculate the shortest path lengths to the frontiers of its assigned region. Among the unexplored cells in the agent's Voronoi partition, the nearest frontier is selected based on distance. If multiple frontiers share the same minimum distance, for each frontier a_i in the nearest frontier list, the utility function described later in this section is computed. The frontier with the highest utility score is chosen as the agent's next target cell.

CU-LVP also ensures that all agents within the broadcast range have different goals by excluding frontiers already assigned to an agent, provided the number of unexplored frontiers is greater than or equal to the number of agents. This differentiation is applied when agents share the same Voronoi region, either due to merging their regions or exploring regions assigned to other agents once they have completed the exploration of their initial regions. By excluding assigned frontiers, agents avoid converging to the same goals, spreading out efficiently across different areas. It is also important to note that this differentiation can only be detected when agents are within broadcast range, enabling them to communicate and share their assigned frontiers, thus avoiding duplication. Additionally, when agents explore the same (or neighboring) regions, one agent may inadvertently explore a goal intended for another. To encounter this, CU-LVP checks at each round whether an agent's goal has been explored by others within its broadcast range. If the goal has been explored, the algorithm recalculates the next optimal goal, enhancing overall exploration efficiency.

The utility function of CU-LVP combines elements from the cost-utility function CU-MNM [30] (later referenced as u_{mnm}) and the CU-JGR approach [31]. Instead of merely considering the neighboring cells (in view range) of the goal cell in the cost-utility function, as CU-JGR does, our method considers the neighboring cells along the whole waveform path from the current position of the agent to its goal cell (later referenced as u_{jgr}). This comprehensive calculation provides a more accurate estimation of the expected explored cells and improves the overall efficiency of the maze exploration task, as demonstrated in the experimental evaluation. Specifically, the formula for this new cost-utility function is as follows:

$$\text{utility}(f) = N(u_{mnm}(f)) + \lambda \cdot N(u_{jgr}(f)) \quad (1)$$

where f represents the nearest frontier cell under examination within the agent's Voronoi region, and $N(x)$ denotes the min-max normalization function, ensuring values fall within the range $[0, 1]$. If cell x_0 is the current position of agent x , the path P is the sequence $x_1, \dots, x_i, \dots, x_n$ of cells that the agent

has to cross in order to reach the target f , which always lies within the agent's assigned region. These cells of path P may not all belong to the region assigned to this agent, allowing the agent to overcome obstacles that might impede progress if it were confined solely within its designated area. As the agent follows path P , several unexplored cells within its view range will be explored, enhancing the overall exploration efficiency if the agent selects the specific frontier f .

Moreover, the method incorporates a parameter λ , which was determined to be optimal at 0.2 after conducting 100 experiments across maze dimensions of 15x15, 30x30, and 50x50. A more comprehensive comparison of the outcomes across different λ values is shown in Fig. 5, using Copeland's method. Copeland's method [35]–[37] is a voting technique, which allows ranking multiple methods that are compared on a set of experiments with varying parameters (e.g. maze sizes, number of agents or partitions, etc.) to provide a unified ranking. The technique involves pairwise comparisons of the average scores obtained across the experiments for each λ value. Based on the results, we can determine which λ value consistently received more votes, thereby establishing its superiority across the experiments.

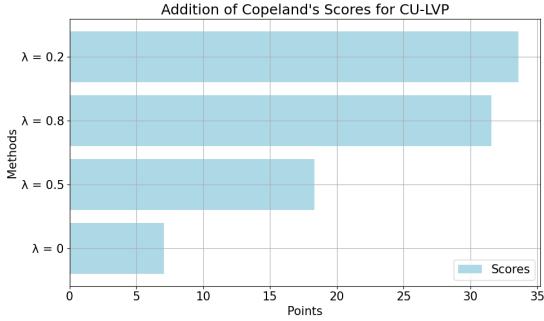


Fig. 5. Copeland Comparison for different λ values of CU-LVP across 100 experiments in various maze sizes (namely 15x15, 30x30, and 50x50).

The first utility score of the utility function (1), u_{mnm} , for the (nearest) frontier cell f is defined as:

$$u_{mnm}(f) = \sum_{i=1}^n dist[(X_f, Y_f), (X_{r_i}, Y_{r_i})] \quad (2)$$

where, $x_0 = (X_{r_i}, Y_{r_i})$ represents the position of agent i , $x_n = (X_f, Y_f)$ denotes the position of frontier f within its Voronoi region, and n signifies the number of agents. Also, u_{jgr} for the frontier cell f is defined as:

$$u_{jgr}(f) = \sum_{x_i \in P} Unex(x_i, r) \quad (3)$$

where $Unex(x, r)$ represents the number of unexplored cells within the agent's view range at cell x . x_i denotes the individual elements (cells) of the *path*, designated as the wavefront path, guiding the agent towards the target frontier f . Once again, the frontier(s) f considered are within the agent's currently assigned Voronoi region.

CU-LVP also incorporates collision avoidance strategies to navigate around other agents or obstacles. Specifically, if an

agent is unable to reach the nearest frontier due to obstacles or blocked paths caused by other agents within broadcast range, it remains stationary until the obstructing agents move. However, this is a rare occurrence due to the integration of Voronoi partitioning. As detailed in Section III-B, agents prioritize exploring their own Voronoi regions, which effectively prevents collisions throughout the experiments. Consequently, collisions may occur only when two or more agents navigate through the same Voronoi region.

IV. EVALUATION FRAMEWORK

To evaluate the performance of the newly proposed CU-LVP and compare it with other state-of-the-art techniques that we adapted to the distributed maze problem, we conducted experiments using randomly generated mazes and a varying density of obstacles. Several metrics were used to evaluate the performance of different methods under various conditions. The experiments ran 500 times and the average scores are reported. The code and results of our experiments are available in a code repository (<https://github.com/manouslinard/multiagent-exploration/>).

A. Maze Generation

For generating mazes, we adapted the methodology described in [38]. This method involves moving an agent randomly in orthogonal directions (up, down, left, and right) within a grid to construct the maze. Also, we extended this approach by introducing a probability factor for each obstacle cell, determining whether an obstacle should remain or be converted to empty space, thus allowing for the creation of mazes with varying levels of complexity. The function also permits the specification of maze dimensions.

The maze is represented as a 2D array with four cell states: unexplored cells (-1), free space (0), obstacles (1), and agent cells (2). When an agent views its surroundings, cells within its range are marked using 0 for free space and 1 for obstacles.

For our experiments, we generated 30x30 mazes with varying obstacle probabilities to create environments of differing complexities, set at 85% and 15%. This range enabled a thorough evaluation of the algorithms' performance under different conditions. An obstacle probability of 85% simulates a complex, maze-like stage, whereas 15% resembles a sparsely obstructed area. To ensure robustness in our evaluation, new mazes were generated for each experiment, providing fresh scenarios to test the efficiency of the algorithms. Fig. 6 shows two randomly generated 30x30 mazes, one with an obstacle probability of 15% and the other with 85%, demonstrating the diversity in maze complexity examined in our experiments.

B. Baseline methods

The baseline methods used to compare CU-LVP are described in Table I. Each method has been implemented from scratch based on the corresponding papers and adapted to the distributed maze problem using Voronoi partitioning. The specifics of the modifications for adapting these methods to partitioned mazes are detailed in Section IV-B1, whereas parameter tuning is discussed in Section IV-B2.

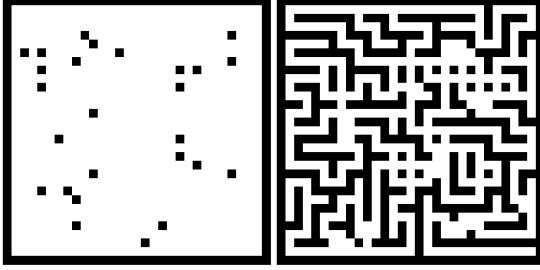


Fig. 6. Randomly generated 30x30 mazes from the experiments. The left maze has an obstacle probability of 15%, whereas the right maze has an obstacle probability of 85%.

TABLE I
METHODS USED FOR COMPARATIVE ANALYSIS IN DISTRIBUTED MAZE EXPLORATION.

Name	Description
HEDAC	Utilizes potential fields to explore the maze [25].
Nearest-Front	Selects the closest frontier to the agent [29].
CU-BSO	Uses wavefront propagation & selects the frontier with less robots near it [32].
CU-MNM	Cost-utility method considering the distances of frontiers from all agents [30].
CU-JGR	Cost-utility method estimating the expected explored cells in the target cell [31].

1) *Voronoi Partitioning for Baseline Methods:* The baseline methods were adapted to the distributed maze problem. Specifically, frontier exploration methods like CU-MNM, CU-JGR, CU-BSO, and Nearest Frontier follow the same principles for Voronoi integration as CU-LVP, as discussed in Section III-B. This is due to the fact that they are similar to CU-LVP, since CU-LVP is also a frontier exploration method.

HEDAC on the other hand differs in its adaptation for the distributed maze problem. Specifically, HEDAC constrains the agent's movements within its designated Voronoi region, similar to HYC [28]. This distinction arises from the characteristic of temperature/potential field methods, which select one neighboring cell at a time. To ensure navigation solely within the agent's Voronoi region, we consider only cells within that region, treating all other unexplored cells outside of it as obstacles for accurate potential field calculations.

However, this approach may encounter challenges in maze environments, where obstacles hinder progress, potentially causing the agent to become trapped and unable to explore its entire Voronoi region. To mitigate this, we refined the HEDAC algorithm. Once the agent has explored all reachable cells, we enhance the attraction force along the wavefront path, guiding the robot toward the nearest frontier. Consequently, the agent can navigate situations where its Voronoi region becomes inaccessible due to limited movement within it and the presence of obstacles, effectively addressing the distributed maze problem.

2) *Parameter Tuning for Baseline Methods:* Certain baseline methods require specific parameter settings to achieve optimal performance. For example, in HEDAC, the “iterations” parameter, as described in the original paper [25], determines how many times the attractive force should be recalculated to produce the final attractive force for each cell. Through

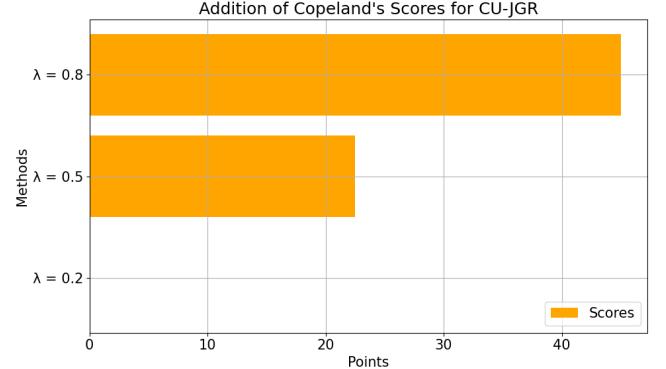


Fig. 7. Copeland Comparison for different λ_{jgr} values of CU-JGR across 100 experiments in various maze sizes (namely 15x15, 30x30, and 50x50).

experimentation in distributed mazes, we found that setting this parameter to 100 yielded the best results. Lower values caused agents to become trapped in local minima, impeding exploration progress, whereas higher values significantly increased computation time. Additionally, HEDAC includes a parameter denoted a , which we found to perform optimally when set to 10. Furthermore, the anti-collision (AC) condition outlined in the original HEDAC paper is set to ON in our experiments. This means that if an agent has information that one of the other agents is currently standing on one of its neighboring nodes, the agent does not consider this node for its next position. We also extended the original HEDAC approach by setting the agent's view range to two blocks.

The CU-JGR algorithm also requires tuning of the parameter λ_{jgr} . Through experimentation in partitioned mazes, we determined that setting λ_{jgr} to 0.8 resulted in more efficient maze exploration. Fig. 7 shows the results obtained for different λ_{jgr} values, using Copeland's method.

C. Evaluation metrics

To compare the methods, we utilized evaluation metrics proposed by Yan *et al.* [39]. Additionally, we tracked the number of exploration rounds to measure the repetition of the exploration process. A metric for estimating the communication cost between agents was also developed. The evaluation metrics used in this study include: i) Exploration Time, ii) Exploration Rounds, iii) Exploration Cost, iv) Exploration Efficiency, v) Map Quality, and vi) Communication Cost.

1) *Exploration Time:* The exploration time metric measures the total time required for a robot fleet to complete an exploration mission. The timer starts when at least one robot begins exploring and stops when the fleet collectively achieves 100% terrain exploration. This metric is measured in wall-clock time, reflecting the duration in days, hours, minutes, and seconds spent on the exploration task.

In our experiments, exploration time measures the computational time for each method, as the movement time is instantaneous. This is because the agents move within a simulated grid (a 2D array), where we simply change the cell value from 0 (free space) to 2 (occupied by an agent) to signify agent movement. Specifically, the (exploration) time

is calculated by finding the average time of each round and multiplying it by the number of rounds. Since the agents operate sequentially (one agent after the other performs a step in each round), we calculate the average round time by dividing the time needed for all agents to make a step to the number of agents:

$$\text{explorationTime}(n) = \frac{\sum_{i=1}^n t_{si}}{n} \cdot R \quad (4)$$

where t_{si} is the time needed for an agent to make a step.

2) *Exploration Rounds*: Exploration rounds (R) represent the iterative cycles of the exploration process, including both information exchange among agents and the algorithm's repetition until maze completion. These iterations measure the number of times the agents have communicated with each other, as information sharing occurs in each round for agents within broadcast range. Exploration iterations conclude when the entire maze has been fully explored.

3) *Exploration Cost*: The exploration cost is based on user specifications, including factors such as energy consumption by computational resources (e.g., CPU, RAM, and network bandwidth) and robot-related expenses like acquisition, handling, and maintenance costs. Their definition of the exploration cost metric requires summing the distances traveled by each robot during the collaborative maze exploration. This metric can also be utilized to measure repetitive coverage, as it is calculated using the following formula:

$$\text{explorationCost}(n) = \sum_{i=1}^n d_i \quad (5)$$

where n is the number of robots in the fleet, and d_i is the distance traveled by robot i .

4) *Exploration Efficiency*: Exploration efficiency relates to the amount of environmental information acquired, inversely proportional to the costs incurred by the robot fleet. It is described by the following formula:

$$\text{explorationEfficiency}(n) = \frac{M}{\text{explorationCost}(n)} \quad (6)$$

where n is the number of robots in the fleet and M is the total explored cells.

5) *Map Quality*: The map quality is defined as the overlap of the explored map and the ground truth map as a ratio to the total area of the ground truth map P :

$$\text{mapQuality} = \frac{M - A(\text{mapError})}{P} \quad (7)$$

where M are the total explored cells, P is the total area of the ground truth map, and $A(\text{mapError})$ is the area occupied by error cells. Error cells are those in the explored map that differ in value from the corresponding cells in the ground truth map. In our experiments, the *mapQuality* metric consistently reaches 100% due to the efficiency of the algorithms and the design of the mazes, which facilitate agents' access to all areas for exploration.

6) *Communication Cost*: The communication cost refers to the cells transferred from one agent to others within range during communication intervals. During each (exploration) round, an agent shares its entire explored stage with other agents within its communication range. Considering the four possible cell states (-1, 0, 1, and 2), we require 2 bits to represent each cell. Additionally, since each communication exchange occurs bidirectionally between two agents, the equation for calculating the communication cost is formulated as follows:

$$\text{commCost} = 4 \cdot \text{rows} \cdot \text{columns} \quad (8)$$

Here, the factor of 4 accounts for the 2 bits needed per cell multiplied by 2, representing the number of communicating agents. The variables *rows* and *columns* denote the dimensions of the maze, as the agent transmits its entire explored stage during each communication cycle. For instance, in a 30x30 maze, both the number of rows and columns is 30.

D. Hardware description

The final experiments were conducted on a PC running Linux equipped with an Intel(R) Core(TM) i5-1235U CPU. To optimize computational efficiency, the Python multiprocessing library was employed, utilizing all available processing cores. Detailed results of these experiments are presented in Section V.

V. EXPERIMENTAL RESULTS

The results consist of 500 experiment repetitions conducted with varying numbers of agents; specifically 1, 2, 4, 6, 8, 10, 15, and 20. Each agent configuration explored mazes of dimensions 30x30 with obstacle probabilities set at 85% and 15%. These experiments included both baseline methods and the newly proposed CU-LVP, as depicted in the resulting plots. Fig. 8 (refer to Section IV-C for details on the metrics) shows the average scores and standard deviations obtained from the experiments for each agent group and exploration method.

For the final evaluation, we employed the Composite Index Scoring method [40], [41]. This method is a powerful and popular tool for providing an overall measure of a subject (in our case, a method) by summarizing a group of measurements (component indices) of different aspects of the subject/method. It is widely used in economics, finance, policy evaluation, performance ranking, and many other fields. According to [40], the most widely used approach is to use a linear combination of the component indices with specified weights. In our case, the components are the metrics defined in Section IV-C, and the weights are all set to 1, as all metrics are equally important.

Specifically, we produced the Composite Index Scoring plot by normalizing all the metrics for each agent group using min-max scaling. This means, for example, that the maximum value of the min-max scaling for a setup with N agents is the largest (average) metric value (for the 500 repetitions) for the N -agents' setup only (and similarly for the minimum value). We ranked each method in each agent group for all metrics based on these normalized values. Then, we summed these rankings using equal weights to create the final Composite Index Scoring plot. The resulting Composite Index plot is

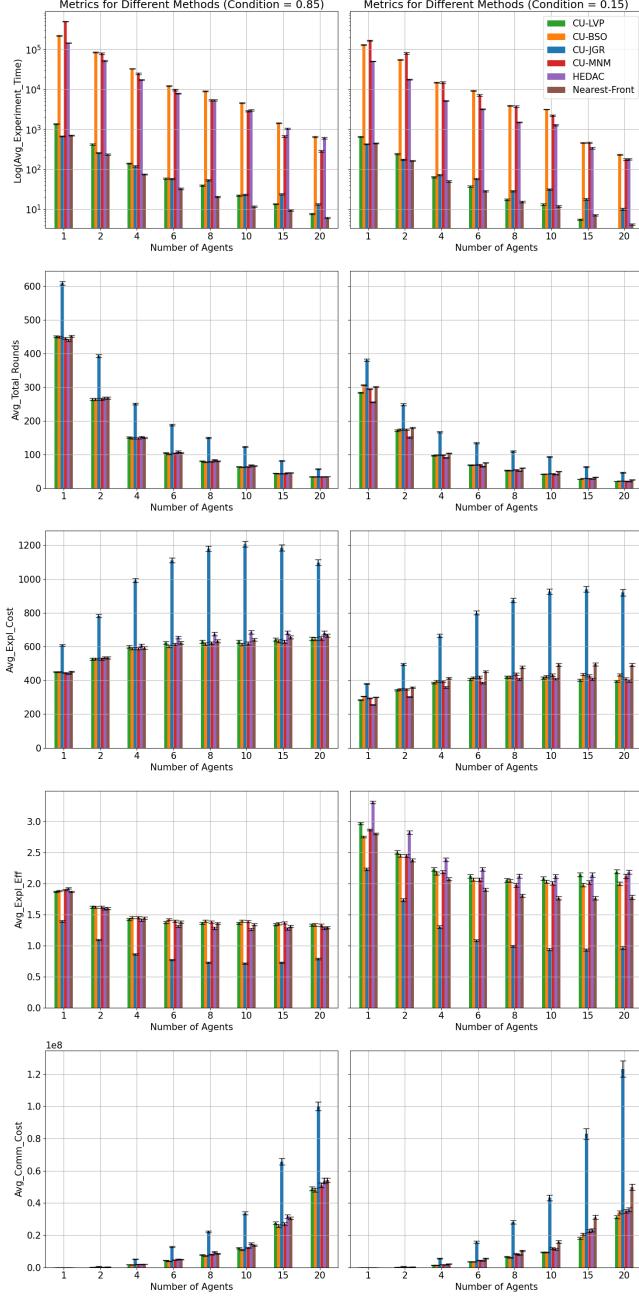


Fig. 8. Results of 500 experiments, utilizing the metrics described in Section IV-C for CU-LVP and baseline methods.

displayed in Fig. 9 and detailed in Section V-B. Greater values in the Composite Index Scoring denote better method performance.

A. Results of Metrics

The results of the metrics showcase the overall efficiency of CU-LVP. The first row of plots in Fig. 8 displays the Exploration Time results on a logarithmic scale for better visibility. CU-LVP is among the least computationally expensive methods across both maze topologies, comparable to Nearest-Frontier, i.e. the simplest baseline method. CU-JGR, another computationally efficient method, performs similarly

to CU-LVP in complex mazes (85% obstacles). However, in simpler mazes (15% obstacles) and with more than 4 agents, CU-LVP proves to be more efficient. Other methods like CU-MNM, CU-BSO, and HEDAC are more computationally expensive, leading to longer exploration times. This efficiency is attributed to CU-LVP's use of wavefront propagation to find the nearest frontiers, resulting in faster utility function calculations and overall algorithm performance.

The second row of Fig. 8 shows the Exploration Rounds Metric. In mazes with dense (i.e. 85%) obstacles, CU-LVP performs comparably to other methods except for CU-JGR, which has the highest number of exploration rounds. In mazes with few (i.e. 15%) obstacles, CU-LVP generally ranks second to HEDAC when there are fewer agents. As the number of agents increases, performance across methods becomes similar, except for CU-JGR, which consistently performs the worst, exhibiting the highest exploration rounds in both maze scenarios.

The third row presents the Exploration Cost results, which can also be interpreted as repetitive coverage results. CU-JGR again performs the worst in both scenarios. Similarly, HEDAC falls short compared to other methods in the dense obstacle setting, gradually ranking as the second-worst in exploration cost as the number of agents increases. CU-LVP remains competitive in complex mazes, improving its performance as the number of agents increases. In simpler mazes, CU-LVP typically ranks second to HEDAC with fewer agents, but this disparity diminishes as the number of agents rises, eventually achieving parity with HEDAC.

The fourth row displays Exploration Efficiency performance. CU-JGR has the lowest efficiency in both maze conditions, followed by HEDAC in 85% obstacle probability mazes. CU-LVP's efficiency is comparable to other cost-utility methods in dense mazes. In simpler mazes, HEDAC is the most efficient, followed by CU-LVP, with both methods performing equally well as the number of agents increases.

The fifth and final row shows Communication Cost results. CU-JGR consistently has the highest communication cost in all maze topologies. Communication costs increase for all methods as the number of agents rises. CU-LVP maintains one of the lowest communication costs overall, especially in simpler mazes where it consistently ranks first. This is likely due to the effective dispersion of robots by CU-LVP, reducing communication overhead.

B. Composite Index Scoring

Examining the Composite Index depicted in Fig. 9, it becomes evident that CU-LVP achieved the highest overall score in both maze topologies. Particularly, it outperforms all other methods in complex mazes (with 85% obstacle probability) and excels when more than 4 agents are used in the exploration of simpler mazes. However, for 1 and 2 agents in less complicated mazes, HEDAC exhibits superior performance. The advancement of CU-LVP over other baselines stems from its competitive results across all maze settings in metrics such as Exploration Rounds, Exploration Cost, Communication, and Efficiency, while maintaining significantly lower computa-

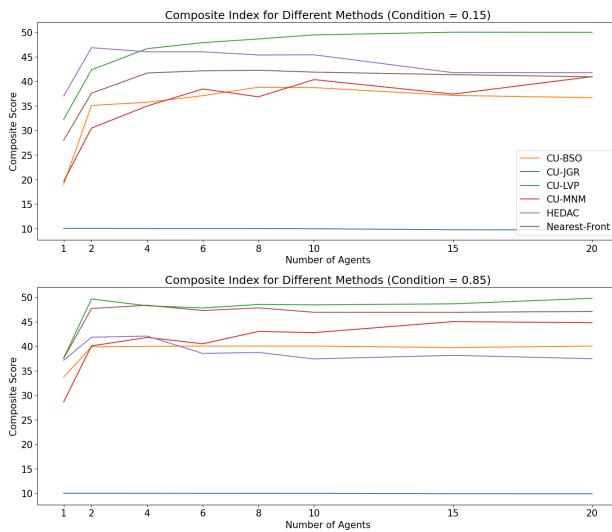


Fig. 9. Composite Index Scores across different methods.

tional complexity compared to other baselines, except Nearest Frontier, i.e. the simplest frontier exploration method.

VI. CONCLUSIONS

This work introduced CU-LVP, a novel method for multiagent maze exploration in distributed mazes, which demonstrates competitive results across various metrics. Notably, CU-LVP exhibits exceptional performance in Exploration Time (computational complexity), showcasing its efficiency by requiring minimal computations to explore mazes effectively. The effectiveness of CU-LVP is further evidenced in the Composite Index plot, where it emerges as the top performer across both complicated and simpler mazes, striking a balance between swift exploration, low computational overhead, minimized communication between agents, and reduced coverage overlap.

Moving forward, our research endeavors will focus on integrating successful elements from other methodologies, such as HEDAC, into the utility function to enhance performance in simpler mazes with fewer agents. Additionally, we plan to incorporate cutting-edge techniques in the baseline methods, such as the recently introduced reinforcement learning method CQLite [23], which shares similarities with CU-LVP in emphasizing frontier exploration in distributed mazes. Moreover, the application of these methods in real-world settings holds promise for providing invaluable insights into their performance and efficiency when confronted with practical complexities. Such insights will fuel further advancements and adaptations, enabling broader applicability.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101073876 (Ceasefire). This publication reflects only the authors views. The European Union is not liable for any use that may be made of the information contained therein.

REFERENCES

- [1] A.-K. Kraeuter, P. C. Guest, and Z. Sarnyai, "The y-maze for assessment of spatial working and reference memory in mice," *Pre-clinical models: Techniques and protocols*, pp. 105–111, 2019.
- [2] S. Alamri, H. Alamri, W. Alshehri, S. Alshehri, A. Alaklabi, and T. Alhmiedat, "An autonomous maze-solving robotic system based on an enhanced wall-follower approach," *Machines*, vol. 11, no. 2, p. 249, 2023.
- [3] S. Alamri, S. Alshehri, W. Alshehri, H. Alamri, A. Alaklabi, and T. Alhmiedat, "Autonomous maze solving robotics: Algorithms and systems," *International Journal of Mechanical Engineering and Robotics Research*, vol. 10, no. 12, pp. 668–675, 2021.
- [4] A. Kamalova, K. D. Kim, and S. G. Lee, "Waypoint mobile robot exploration based on biologically inspired algorithms," *IEEE Access*, vol. 8, pp. 190 342–190 355, 2020.
- [5] O. Doukhi and D. J. Lee, "Deep reinforcement learning for autonomous map-less navigation of a flying robot," *IEEE Access*, vol. 10, pp. 82 964–82 976, 2022.
- [6] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-agent systems: A survey," *IEEE Access*, vol. 6, pp. 28 573–28 593, 2018.
- [7] S. Li, S. He, Y. Zhang, X. Shi, G. He, and T. Jiang, "Edge intelligence enabled heterogeneous multi-robot networks: Hybrid framework, communication issues, and potential solutions," *IEEE Network*, vol. 36, no. 6, pp. 108–115, 2022.
- [8] G. T. Papadopoulos, M. Antonia, and C. Stephanidis, "Towards open and expandable cognitive ai architectures for large-scale multi-agent human-robot collaborative learning," *IEEE Access*, vol. 9, pp. 73 890–73 909, 2021.
- [9] F. Amigoni, J. Banfi, and N. Basilico, "Multirobot exploration of communication-restricted environments: A survey," *IEEE Intelligent Systems*, vol. 32, no. 6, pp. 48–57, 2017.
- [10] J. P. Queralta, J. Taipalmaa, B. Can Pullinen, V. K. Sarker, T. Nguyen Gia, H. Tenhunen, M. Gabouj, J. Raitoharju, and T. Westerlund, "Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision," *IEEE Access*, vol. 8, pp. 191 617–191 643, 2020.
- [11] P. Mazdin and B. Rinner, "Distributed and communication-aware coalition formation and task assignment in multi-robot systems," *IEEE Access*, vol. 9, pp. 35 088–35 100, 2021.
- [12] D. Cai, S. Wu, and J. Deng, "Distributed global connectivity maintenance and control of multi-robot networks," *IEEE Access*, vol. 5, pp. 9398–9414, 2017.
- [13] Y. Wu, X. Ren, H. Zhou, Y. Wang, and X. Yi, "A survey on multi-robot coordination in electromagnetic adversarial environment: Challenges and techniques," *IEEE Access*, vol. 8, pp. 53 484–53 497, 2020.
- [14] H. Li, H. He, and Y. Wen, "Dynamic particle swarm optimization and k-means clustering algorithm for image segmentation," *Optik*, vol. 126, no. 24, pp. 4817–4822, 2015.
- [15] E. R. Magsino, "Homogeneous map partitioning employing the effective regions of movement method," in *Pervasive Computing and Social Networking: Proceedings of ICPCSN 2022*. Springer, 2022, pp. 271–284.
- [16] E. R. Magsino and I. W.-H. Ho, "An enhanced information sharing roadside unit allocation scheme for vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15 462–15 475, 2022.
- [17] F. Aurenhammer and R. Klein, "Voronoi diagrams." *Handbook of computational geometry*, vol. 5, no. 10, pp. 201–290, 2000.
- [18] H. Moulinec, "A simple and fast algorithm for computing discrete voronoi, johnson-mehl or laguerre diagrams of points," *Advances in Engineering Software*, vol. 170, p. 103150, 2022.
- [19] M. García, D. Puig, L. Wu, and A. Solé, "Voronoi-based space partitioning for coordinated multi-robot exploration," *JoPha: Journal of Physical Agents, ISSN 1888-0258*, Vol. 1, N°. 1, 2007, pages. 37-44, vol. 1, 01 2007.
- [20] G. Lozenguez, L. Adouane, A. Beynier, P. Martinet, and A.-I. Mouaddib, "Map partitioning to approximate an exploration strategy in mobile robotics," in *Advances on Practical Applications of Agents and Multiagent Systems: 9th International Conference on Practical Applications of Agents and Multiagent Systems*. Springer, 2011, pp. 63–72.
- [21] J. Yan, X. Lin, Z. Ren, S. Zhao, J. Yu, C. Cao, P. Yin, J. Zhang, and S. Scherer, "Mui-tare: Cooperative multi-agent exploration with unknown initial position," *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 4299–4306, 2023.

- [22] J. Gui, T. Yu, B. Deng, X. Zhu, and W. Yao, "Decentralized multi-uav cooperative exploration using dynamic centroid-based area partition," *Drones*, vol. 7, no. 6, p. 337, 2023.
- [23] E. Latif and R. Parasuraman, "Communication-efficient multi-robot exploration using coverage-biased distributed q-learning," *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2622–2629, 2024.
- [24] J. Yu, J. Tong, Y. Xu, Z. Xu, H. Dong, T. Yang, and Y. Wang, "Smmr-explore: Submap-based multi-robot exploration system with multi-robot multi-target potential field exploration method," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8779–8785.
- [25] B. Crnković, S. Ivić, and M. Zovko, "Fast algorithm for centralized multi-agent maze exploration," *arXiv preprint arXiv:2310.02121*, 2023.
- [26] S. Ivić, B. Crnković, and I. Mezić, "Ergodicity-based cooperative multiagent area coverage via a potential field," *IEEE Transactions on Cybernetics*, vol. 47, no. 8, pp. 1983–1993, 2017.
- [27] S. Ivić, "Motion control for autonomous heterogeneous multiagent area search in uncertain conditions," *IEEE Transactions on Cybernetics*, vol. 52, no. 5, pp. 3123–3135, 2020.
- [28] Y. Zheng and C. Zhai, "Distributed coverage control of multi-agent systems in uncertain environments using heat transfer equations," *arXiv preprint arXiv:2204.09289*, 2022.
- [29] B. Yamauchi, "Frontier-based exploration using multiple robots," in *Proceedings of the second international conference on Autonomous agents*, 1998, pp. 47–53.
- [30] A. Marjovi, J. G. Nunes, L. Marques, and A. de Almeida, "Multi-robot exploration and fire searching," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 1929–1934.
- [31] M. Juliá, A. Gil, and O. Reinoso, "A comparison of path planning strategies for autonomous exploration and mapping of unknown environments," *Autonomous Robots*, vol. 33, pp. 427–444, 2012.
- [32] A. Bautin, O. Simonin, and F. Charpillet, "Towards a communication free coordination for multi-robot exploration," in *6th National conference on control architectures of robots*, 2011, pp. 8–p.
- [33] J. Barraquand, B. Langlois, and J.-C. Latombe, "Numerical potential field techniques for robot path planning," *IEEE transactions on systems, man, and cybernetics*, vol. 22, no. 2, pp. 224–241, 1992.
- [34] Z. Zhang, J. Yu, J. Tang, Y. Xu, and Y. Wang, "Mr-topomap: Multi-robot exploration based on topological map in communication restricted environment," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10794–10801, 2022.
- [35] G. Al-Sharrah, "Ranking using the copeland score: a comparison with the hasse diagram," *Journal of chemical information and modeling*, vol. 50, no. 5, pp. 785–791, 2010.
- [36] D. G. Saari and V. R. Merlin, "The copeland method: I.: Relationships and the dictionary," *Economic theory*, vol. 8, pp. 51–76, 1996.
- [37] S. Lestari, T. B. Adji, and A. E. Permanasari, "Performance comparison of rank aggregation using borda and copeland in recommender system," in *2018 international workshop on big data and information security (iwbis)*. IEEE, 2018, pp. 69–74.
- [38] M. Gold, "Python maze creation," 2023. [Online]. Available: <https://medium.com/@msgold/using-python-to-create-and-solve-mazes-672285723c96>
- [39] Z. Yan, L. Fabresse, J. Laval, and N. Bouraqadi, "Metrics for performance benchmarking of multi-robot exploration," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 3407–3414.
- [40] R. Chen, Y. Ji, G. Jiang, H. Xiao, R. Xie, and P. Zhu, "Composite index construction with expert opinion," *Journal of Business & Economic Statistics*, vol. 41, no. 1, pp. 67–79, 2022.
- [41] S. N. Chakrabarty, "Composite index: methods and properties," *Journal of Applied Quantitative Methods*, vol. 12, no. 2, pp. 25–33, 2017.



Manousos Linardakis is currently an undergraduate student at the Department of Informatics and Telematics, Harokopio University of Athens. He has conducted research in multirobot exploration and hand gesture recognition and has experience in managing laboratory systems, implementing DevOps practices, and developing decision support software. He also has participated in the Google Summer of Code as an open-source developer. His research interests include data science, machine learning, robotics, and multiagent systems.



Iraklis Varlamis (Member, IEEE) received the M.Sc. degree in information systems engineering from the University of Manchester Institute of Science and Technology, Manchester, U.K., and the Ph.D. degree from the Athens University of Economics and Business, Athens, Greece. He is currently a Professor of data management with the Department of Informatics and Telematics, Harokopio University of Athens (HUA), Kallithea, Greece. He has more than 250 articles published in international journals and conferences and more than 5000 citations on his work. He holds a patent from the Greek Patent Office for a system that thematically groups web documents using content and links. His research interests include data mining and social network analytics to recommender systems for social media and real-world applications. He is the Scientific Coordinator for HUA in several EU (H2020, ECSEL, REC) and Qatar (QNFR) projects as well as in national projects.



Georgios Th. Papadopoulos (M) is an Assistant Professor in the area of Computer Graphics and Computational Vision at the Department of Informatics and Telematics of the Harokopio University of Athens in Greece. He received the Diploma and Ph.D. degrees in electrical and computer engineering from the Aristotle University of Thessaloniki (AUTH), Thessaloniki, Greece. He has worked as a Post-doctoral Researcher at the Foundation For Research And Technology Hellas / Institute of Computer Science (FORTH/ICS) and the Centre for Research and Technology Hellas / Information Technologies Institute (CERTH/ITI). He has published over 70 peer-reviewed research articles in international journals and conference proceedings. His research interests include computer vision, artificial intelligence, machine/deep learning, human action recognition, human-computer interaction and explainable artificial intelligence. Dr. Papadopoulos is a member of the IEEE and the Technical Chamber of Greece.