ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

TRƯƠNG XUÂN HÙNG

DANH MỤC CÔNG TRÌNH KHOA HỌC CỦA TÁC GIẢ LIÊN QUAN ĐẾN LUẬN ÁN

NGHIÊN CỨU THUẬT TOÁN TỐI ƯU BẦY ONG TRONG HOẠCH ĐỊNH ĐƯỜNG ĐI VÀ PHÂN CỤM ĐỘNG CHO HỆ ROBOT

LUẬN ÁN TIẾN SĨ CƠ KỸ THUẬT

Hà Nội – 2026

DANH MỤC CÔNG TRÌNH KHOA HỌC CỦA TÁC GIẢ LIÊN QUAN ĐẾN LUẬN ÁN

1. **Hung Truong Xuan**, Thang Pham Manh, Nha Nguyen Quang, Hanh Nguyen Thi Hong Hanh, “Dynamic Clustering of Multi-Mobile Robot System using Gaussian Mixture Model”, *J. Robot. Control*, vol. 6, no. 5 SE-Articles, pp. 2129–2140, Aug. 2025, doi: 10.18196/jrc.v6i5.27184.
2. **Hung Truong Xuan**, Pham Manh Thang and Thuc Nguyen Van, "Digitizing Working Space of the Multi Mobile Robots System Using an Approximate Cell Decomposition Method", Proceedings of the 7th International Conference on Engineering Mechanics and Automation (ICEMA7), pp. 3-8, 2023, doi: 10.15625/vap.2023.0110.
3. **Hung X. Truong**, Thang M. Pham, Nha Q. Nguyen, Hanh T. H. Nguyen and Viet A. Dang, "Evaluation of Spatial Stability and Adaptability in a Dynamic Clustering Method for Multi-Robot Systems", Proceedings of the 8th International Conference on Engineering Mechanics and Automation (ICEMA8), pp. 12-18, 2025, doi: 10.15625/vap.2026.0003, ISBN: 978-604-357-500-2.
4. **Hung X. Truong**, Thang M. Pham, Nha Q. Nguyen, Hanh T. H. Nguyen and Viet A. Dang, "Artificial Bee Colony for Multi-Robot Path Planning: A Comparative Study of Population Structures", Proceedings of the 8th International Conference on Engineering Mechanics and Automation (ICEMA8), pp. 19-25, 2025, doi: 10.15625/vap.2026.0004, ISBN: 978-604-357-500-2.
5. Nguyễn Thị Hồng Hạnh (VN); **Trương Xuân Hùng** (VN); Nguyễn Đức Minh (VN); Bằng độc quyền sáng chế số 48666 "Cụm cơ cấu điều chỉnh hướng và robot vận chuyển hàng hoá trong nhà kho bao gồm cụm cơ cấu điều chỉnh hướng này", 2025.
6. **Trương Xuân Hùng** (VN); Nguyễn Thị Hồng Hạnh (VN); Trịnh Thăng Long (VN); Nguyễn Văn Thức (VN); Bằng độc quyền giải pháp hữu ích số 3615 "Phương pháp xác định trạng thái chuyển động của robot di động vận chuyển hàng hoá để phục vụ điều khiển và giám sát robot", 2024.

Danh mục này gồm 06 công trình.

Journal of Robotics and Control (JRC)
Volume 6, Issue 5, 2025
ISSN: 2715-5072, DOI: 10.18196/jrc.v6i5.27184

Dynamic Clustering of Multi-Mobile Robot System using Gaussian Mixture Model

Hung Truong Xuan [1*], Thang Pham Manh [2], Nha Nguyen Quang [3], Hanh Nguyen Thi Hong [4]
[1] Vietnam National Space Center, Vietnam Academy of Science and Technology, Cau Giay, Hanoi, Vietnam
[1, 2, 3] University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam
[4] Institute of Mechanics, Vietnam Academy of Science and Technology, Hanoi, Vietnam
Email: [1] txhung@vnsc.org.vn, [2] thangpm@vnu.edu.vn, [3] nhanq@vnu.edu.vn, [4] nguyenhonghanh.imech@gmail.com
*Corresponding Author

Abstract*—Managing large fleets of mobile robots poses significant challenges to system coordination and workload. An effective grouping strategy is crucial for enhancing operational performance and scalability. This paper introduces a two-stage dynamic clustering method (DCM), a novel framework for organizing robots into manageable groups. The methodology utilizes a Gaussian Mixture Model and the Expectation-Maximization algorithm to cluster robots based on their path intersection points. A unique "cost" parameter, formulated a least squares objective function, is proposed to guide the selection of near-optimal, workload-balanced configurations. The results from extensive simulations demonstrated the framework's effectiveness. On a single dataset, DCM exhibited exceptional reliability, maintaining a stable objective function value even as the number of robots per cluster fluctuated across runs. A sensitivity analysis over multiple unique datasets confirmed the model's adaptive strength, showing its ability to re-configure clusters. This adaptability was highlighted by the mean objective function value varying across different scenarios. Further analysis involving reduced robot populations and obstacle-filled environments validated DCM's generalizability and environment-independent nature. The robot distribution mechanism was consistently equitable and balanced. Statistical validation, including bootstrapping resamples, confirmed the stability and reliability of the performance estimates. The method also steadily maintained a high level of performance by adapting to internal variations. Moreover, every robot was successfully assigned to all clusters across all trials. The research concludes that DCM is a robust, adaptive, and environment-independent framework. It successfully balances performance stability with the flexibility to respond to new operational conditions, proving it is an effective solution for multi-robot coordination.

***Keywords*—Dynamic Cluster; Gaussian Mixture Model; Expectation-Maximization Algorithm; Path Planning; Workload Balancing; Multi-Mobile Robots.**

I. INTRODUCTION

The concept of multi-robot system or multi-mobile robot system emerged in the late 1980s, coming from foundational research in distributed robotic systems that focused on mobile robots [1], [2]. Subsequently, this field expanded to encompass groups of UAV, also called multi-UAV system. There are many important aspects that are considered to be investigated carefully in these multi-robot systems. Navigation is one of the most important features, heavily associated with localization and map modeling, global and local path planning [3] for static or dynamic environments [4]. Many algorithms and methods have been proposed and developed to deal with the path planning issues [5]-[8]. Besides, mapping is divided into two broad approaches, outdoor and indoor simultaneous localization and mapping [9]-[12]. Task allocation is also the next critical feature for efficient and scalable performance of multi-mobile robot systems [13]-[15]. It is greatly influenced by the overall structure of the system: centralized and decentralized approaches [6], [16]. Communication network is a main integral part, its capabilities are essential for accomplishing tasks as well as coordination among robots, including network architecture, routing protocols and data thread with IoT platform [17], [18]. Recently, artificial intelligence, such as reinforcement learning [19] and deep reinforcement learning [20]-[22], has also become very popular tools to support path planning, task allocation, and etc. The other interesting topics are formation and exploration; area coverage and surveillance; search and rescue; foraging and flocking; cooperative manipulation; team heterogeneity; adversarial environment [23]-[28].

Path planning involves generating a collision-free trajectory for a robot to move from an initial position to a goal position. In multi-robot systems, each robot has to deal with not only static environmental obstacles but also dynamic ones, particularly the other robots in the system [4], [29], [30]. Path planning is closely linked to self-localization and map awareness. Furthermore, path planning approaches are often designed to achieve several optimal goals, such as minimizing time or energy consumption. The path planning approaches are classified into several groups: classical types (cell decomposition, rapidly-exploring random tree, Dijkstra, A*, artificial potential field ...) [3]-[8], [31], bio-inspired method (genetic algorithm, ant colony optimization, particle swarm optimization, artificial bee colony algorithm ...) [3]-[6], [8], artificial intelligence [3], [5], [6], heuristic approaches including bio-inspired methods as well as artificial intelligence [7], [31], and others. When considering the specific task of coverage path planning, the approaches are typically categorized as geometric methods, grid-based searches, reward-based strategies, random incremental planners, and near-best-view planners [24].

Additionally, multi-mobile robot task allocation (MRTA) is a feature of multi-mobile robot systems, challenging research aims to achieve the goal of assigning a group of robots to a group of tasks in such a way that optimizes the overall system performance under a set of predefined constraints. There are two organizational paradigms for the overall structure of the system: centralized approaches and decentralized approaches [6], [16] together with several most commonly used approaches: market-based methods [13], [16], metaheuristic-based methods [16] and behavior-based approaches [14]. The research also states that some complex constraints remain unresolved and one of the groups can be classified as environment-related constraints. They include, but are not limited to, the dynamic and unpredictable nature of the environment as well as its partial observability and complexity.

Distributing tasks within a multi-mobile robot system is heavily relies on efficient task allocation and path planning. Subsequently, managing their activities to complete these assignments presents a significant challenge, primarily due to the substantial number of mobile robots involved. Following this, it is influenced by the system awareness for the workspace. A greater number of robots can enhance the system's awareness of the workspace by collecting more data, yet their movements also introduce uncertainties that need continuous identification over time in the working space. Moreover, while using more robots to complete tasks faster, it seems to increase the system workload significantly. A crucial challenge also lies in maintaining constant situational awareness of each robot. They have to warrant that their individual movements contribute to overall efficiency. The aforementioned facts highlight that clustering mobile robots into minor groups is an important and necessary solution to mitigate the workload of the systems with a large fleet of mobile robots. This grouping method is typically performed based on the specific correlation criteria among the individual units. Section II represents published works on techniques and algorithms in this area and it is a significant research activity for multi-mobile robots.

This paper proposes a two-stage dynamic clustering method that utilizes the gaussian mixture model (GMM) to distribute the entire robot fleet into several smaller, more manageable groups. Fig. 1 illustrates the relationship between DCM and the path planning as well as the task allocation approaches. DCM aims to reduce overall system workload by decreasing communication overhead and enabling more efficient local coordination during path execution. Each robot only needs to maintain communication and coordinate operations (if necessary) with others within its assigned group. In order to achieve that, this work offers several key contributions:

- First, while a variety of spatial or hard clustering methodologies, such as cell decomposition, Voronoi diagrams, and K-means, have been extensively reported in Section II, a notable gap exists regarding the utilization of GMM clustering for multi-mobile robot applications. The probabilistic nature of GMM, which facilitates soft clustering, offers a distinct advantage, promising reliable and complete clustering performance for adaptability and scalability. Thus, our contribution addresses this gap by proposing the integration of GMM with the Expectation-Maximization (EM) algorithm, for clustering multi-robot systems. Moreover, the log-likelihood function optimized by the EM algorithm is inherently non-convex. This function can produce multiple local maxima, rather than a singular global maximum. To overcome this, we introduce a novel "cost" parameter, when formulated a least squares objective function, enables the selection of a near-optimal result for workload balancing.
- Second, a specific correlation criterion is derived from intersection point datasets, which represent strictly direct relation between robots in the working space. This criterion is fundamental to the local coordination mechanism, essential not only for collision avoidance but also for situation awareness during operation. By strategically identifying and removing "unrelated" units, those without any intersection points, the local coordination is kept minimum, thereby simplifying system workload.

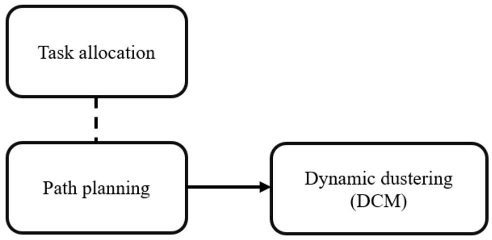

Fig. 1. Operation flow for DCM with path planning and task allocation

The remainder of this paper is organized as follows: Section II reviews the relevant literature on clustering approaches; section III describes the problem formulation and the proposed system methodology; section IV presents the simulation results along with a corresponding performance analysis; and finally, section V provides the concluding remarks and proposed future development.

II. LITERATURE REVIEW

Exploiting workspace is a very early way to develop algorithms for path planning, area exploration, area coverage, and task allocation later. In principle, the workspace consists of free space as well as obstacles. The former is the area where the mobile robots can move freely and the latter can be considered as the restricted space where the mobile robots are prohibited from entering. Cell decomposition is a specific method for exploiting space. In foundational research, path planning is directly derived from cell decomposition, a method where a continuous collision-free path is computed from a sequence of adjacent cells that constitute the free workspace. Furthermore, approximate cell decomposition techniques are also widely employed for applications requiring not only path planning but also comprehensive area coverage. Beyond decomposition, clustering offers another useful approach to identify meaningful groups within the workspace based on robot-specific features. Later, clustering is extended to other objects rather than the workspace based on the development of new clustering solutions or the application of clustering algorithms that have been applied in other research fields.

Cell decomposition methods are categorized as either exact or approximate, a distinction based on the geometry of the cells and their collective coverage of the workspace [32]. Exact decomposition methods partition the workspace into cells, whose union perfectly reconstructs the workspace, such as trapezoids [33], [34]. Meanwhile, approximate decomposition employs cells of a uniform, predefined shape, typically squares, whose union is strictly included in the workspace, thereby approximating its true geometry. Recently, there are a few proposed methods, such as polygonal grid cells which are squared shape [35], radial cell [36], adaptive cell [37] for approximate cell decomposition.

Approximate cell methods divide the working space into a two-dimensional grid of cells. While the most basic approach utilizes cells of a uniform size [35], [38], advanced techniques often employ variable-resolution grids to enhance efficiency. They are the quadtree method as well as recursive approaches to split each cell into the four smaller ones [39]-[42] or combine several adjacent cells into bigger one [43] [44], [45]. A prominent hierarchical strategy is the quadtree method, where "mixed" cells containing both free space and obstacles are recursively subdivided into four quadrants until a minimum resolution is achieved or all cells are homogeneous. Conversely, adjacent, homogeneous cells can be merged into a single larger cell. This hierarchical structure significantly reduces the number of cells required, thereby lowering data volume and accelerating computation. Alternative strategies for achieving variable resolution include adapting cell size based on proximity to the robot, using smaller cells nearby and larger ones farther away [43], [46], or based on the terrain's complexity, with smaller cells representing more intricate profiles [47]. A proposed quarter orbits algorithm is inspired by three types, including the regular grid, the adaptive cell decomposition, and the exact cell decomposition [48].

Clustering offers automated approaches to discover coherent groups (clusters) in workspace to model their unknown global organizational structure formed from the specific features of the robots [49]. Voronoi diagram, a computational geometry that divides multi-dimensional space into regions based on their proximity to a set of seeds, is also used for partitioning the working space in area coverage and area exploration. There are variations of this classic Voronoi diagram, such as Generalized Voronoi Diagram [15], [50] and Manhattan Voronoi [51]. Normally, they convert a multi-robot coverage problem into a number of single robot coverage problems. Some others are Voronoi partition-based coverage with optimal grid size relative to the sensor footprint [52], Voronoi cell repartitioning for load-balancing among the robots with limited communication ranges [53], and Voronoi diagram with a k-means clustering algorithm [54]. There are some other proposed approaches, for example virtual tokens in communication network [55], an extension on the spatial clustering algorithm designed for two classes [56]; particle clustering method [57]; density-based with relative distances to the centers of regions derived from the flying features of UAVs and the geographical locations of regions [58]. Clustering approaches are also introduced along with task allocation of multi-robots, such as Stochastic Clustering Auction [59]; Shapley value clustering for partitioning the initial task allocation into a set of the smaller and simpler task allocation [60]; decentralized coordination framework with cluster formation tracking for heterogeneous robots including aerial drones and ground robots [61]; correlation clustering [62]; adapted clustering based on static communication topology [63].

Initially designed for data clustering, the K-means algorithm inspires the visual navigation of the mobile robot and clustering approaches. The former is to improve the recognition ability of the mobile robot navigation to resist light source interference [64], use with Silhouette method for orchard navigation [65], improve image segmentation which strengthens the adaptability of image segmentation by using a pre-classification and a maximin method [66], improve achieve image segmentation for vegetation with particle swarm optimization [67]. The latter is to partition robot goals or tasks into distinct regions or clusters, aiming to balance the workload of multi-robots. The goals can be acquired by multi-robots [68]-[72] or even one robot [73]. The path planning is to determine near-optimal paths for each region and path connecting regions then. This strategy effectively transforms the problem of managing numerous individual tasks into managing a smaller set of clustered tasks, leading to more efficient task allocation and path planning in multi-robot systems. Their approaches are the combination of the improved K-means algorithm and the particle swarm optimization [68] or the K-means algorithm with another method: modified multiple travelling salesman problem [69], the grey wolf optimizer which improved by Kent chaotic algorithm [70], genetic algorithm [71], the auction based mechanism [72]. All tasks are grouped into clusters and robots are assigned to these clusters in a cost-effective manner, such as their travel costs or run-time.

When used for area exploration or area coverage, the K-means clustering approaches aim to divide the space into distinct regions. This initial fair partitioning ensures focused exploration [74], [75] and coverage [76], [77]. As robots discover new areas, re-clustering the remaining space is taken in order to rebalance the workload among the units. This is naturally promoting wider environmental dispersion for faster, more comprehensive parallel exploration with avoiding collision and repeated area. The Canopy Clustering is proposed as an initial step in K-Means Clustering [75]. An anti-flocking framework is introduced to execute the dynamical selection of the target position of the next step for each robot [76].

GMM is a data clustering method which uses a mixture of Gaussian distributions to model data clusters and the EM algorithm to learn the parameters of these distributions. GMM can be used to model motion using human demonstrations, representing either a continuous degree for every time step of a manipulation task [78], [79] or a set of segmentation points [80]. GMM and gaussian mixture regression (GMR) can implement a learning from demonstrations approach for cobot in which GMM can encode the demonstration trajectories [81], [82]. GMM can map the relationship between the observed variables of the object and the robot joint variables that can be used during the grasping process [83] or it can monitor daily movements of a robotic arm and enhance abnormality detection [84]. Some other critical contributions of GMM are path planning for a mobile robot and target search in dynamic environments. Path planning includes combining the fast marching square with Gaussian mixture models to represent people [85], and using the Risk-RRT planner with GMM-derived perception and prediction [86]. GMM allows for extracting feature nodes and generating collision-free paths considering the pedestrian density and the environmental structure. For target search, GMMs help determine the next best probability target for UAVs movement. They can be used with Infotaxis and a particle filter [87], or with Parzen windows [88].

The EM algorithm, which is an iterative method, is used for the estimation process. The algorithm performs an expectation step and a maximization step then to achieve maximum likelihood estimation [89]-[91]. These two steps are executed iteratively until a convergence condition is met, such as when the change between iterations becomes very small or a specified number of iterations is reached.

Cell decomposition and Voronoi diagrams are fundamentally geometric spatial partitioning, establishing distinct and rigid boundaries for robot groups or spatial segments. Similarly, K-means, as a hard clustering algorithm, assigns units to groups with equally definitive and non-overlapping boundaries. Meanwhile, GMM employs soft clustering, where cluster membership is determined by probability values rather than strict assignment. This inherent probabilistic nature provides greater flexibility and granularity in cluster structure, allowing GMM to present clusters with more "complex" boundaries that can overlap and reflect the significant probabilistic distribution of data points.

III. Problem Definition and System Approach

A. Problem Definition

When the number of robots increases, a key challenge in system operation is to coordinate all the units to execute the required missions effectively. It's also crucial for the system to be aware of the situation as each robot executes its individual path to handle the assigned task in the workspace. Robots with intersecting paths inherently have the potential for spatial and temporal interaction, establishing their natural correlation. Thus, clustering these robots allows us to concentrate the coordination efforts more effectively. Robots can share their location, navigation information, and state movement with other cluster members, minimizing conflicts and optimizing task execution. This selective information sharing significantly enhances situation awareness; each robot not only understands its own trajectory but can also anticipate and react to the movements of other related robots. Moreover, optimized communication is another key advantage of the clustering. Robots only need to keep contact with the cluster members, those with whom they have the highest likelihood of interaction, substantially reducing communication bandwidth. Furthermore, by operating independently across different cluster data streams, the system can achieve better decentralization and improved scalability.

Considering a working space for N robots that is populated with static obstacles. It has a well-defined size and is represented by the following expression:

$$\mathcal{W} = \mathbb{R}^n \tag{1}$$

where n = 2 or 3.

We define an obstacle space $\mathcal{O}$, the region of $\mathcal{W}$ containing static obstacles. A static obstacle has a fixed position that does not change over time from the moment it appears in the workspace. Assuming that $\mathcal{W}$ has m obstacles, $\mathcal{O}$ can be determined by the following formula:

$$\mathcal{O} = \mathcal{O}_1 \cup \mathcal{O}_2 \ldots \cup \mathcal{O}_m \subset \mathcal{W} \tag{2}$$

where $\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_n$ are the space for each obstacle, respectively.

Free space $\mathcal{F}$ is a specific part of the working space $\mathcal{W}$ where robots can move freely. We have a relation between $\mathcal{F}$ and $\mathcal{W}$ as follow:

$$\mathcal{F} \subset \mathcal{W} \tag{3}$$

and

$$(\mathcal{F} \cup \mathcal{O}) \subset \mathcal{W} \tag{4}$$

The (4) indicates that there are some regions that belongs to $\mathcal{W}$ and do not belong to $\mathcal{O}$, but robots still cannot reach them due to their size factor.

When N robots, called $A_1, A_2, \ldots, A_N$, move along independent paths from their own starting points to the goals in the working space $\mathcal{W}$. Assuming the starting points of N robots are $(q_{10}, q_{20}, \ldots, q_{N0})$ and the goals are $(q_{1G}, q_{2G}, \ldots, q_{NG})$, the paths of N robots are:

$$\begin{bmatrix} Path_1(t) \\ \ldots \\ Path_N(t) \end{bmatrix}$$

where path of A_i is defined as follows:

$$Path_i(t) = f_{pp}(q_{i0} \quad q_{iG}) \tag{5}$$

and i is index of A_i; q_{i0} is starting point and q_{iG} is goal of A_i; $(q_{10}, q_{20}, \ldots, q_{N0})$ and $(q_{1G}, q_{2G}, \ldots, q_{NG})$ are all in free space $\mathcal{F}$.

If the path of A_i intersects the path of A_j, we define the intersection points as follows:

$$\left(Co_{i,j}\right) = Path_i(t) \cap Path_j(t) \neq \emptyset \tag{6}$$

So, there is a data set for the intersection points between the paths of N robots:

$$\mathcal{C} = \cup_{i,j=0}^{N} \left(Co_{i,j}\right) \tag{7}$$

and the relationships shown are:

$$A_i \Leftrightarrow Path_i(t) \Leftrightarrow Co_{i,j} \tag{8}$$

As indicated in (8), the correlation between robots that have intersections with their paths is much stronger than those without intersections. Consequently, the intersection point dataset, as illustrated in (7), is chosen as the basis for the clustering method of robots.

B. Gaussian Mixture Model

Given the aforementioned advantages, GMM is the chosen method for grouping multi-mobile robot system by using the intersection point dataset in this paper. GMM is represented by the overall probability density function which is a weighted sum of every components [89]-[91], namely:

$$p(x) = \sum_{i=1}^{K} w_i N(x|\mu_i, \Sigma_i) \tag{9}$$

where w_i is mixing weight of i^{th} component with $w_i > 0$,

$$\sum_{i=1}^{K} w_i = 1 \tag{10}$$

and $N(x|\mu_i, \Sigma_i)$ is a Gaussian density function which also called a component is defined as follows:

$$N(x|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{\frac{n}{2}}(|\Sigma_i|)^{\frac{1}{2}}} exp\left(-\frac{1}{2}(x-\mu_i)^T {\Sigma_i}^{-1}(x-\mu_i)\right) \tag{11}$$

where $x \in \mathbb{R}^n$ represents a data vector, $\mu_i \in \mathbb{R}^n$ is the mean and Σ_i is the $n \times n$ covariance matrix of the i^{th} Gaussian density. We have three critical unknown parameters of GMM in the (9) and (11). There is the mean μ_i, the covariance matrix Σ_i, and the weight w_i. If these parameters are estimated well, we can model the distribution of the data set by using GMM. The EM algorithm can be obtained by two parts:

- Expectation step: calculate the most likely values that component C_i generate sample x_j each iteration based on the current understanding of the model. The computation equation is as follows:

$$f_E(C_i|x_j) = \frac{w_i N(x_j|\mu_i, \Sigma_i)}{\sum_{k=1}^{K} w_i N(x_j|\mu_k, \Sigma_k)} \tag{12}$$

- Maximization step: update the parameters of each component, the mean μ_i, the covariance matrix Σ_i and the weight w_i based on the most likely values C_i calculated in the expectation step. The equations for maximization step are as follows:

$$w_i = \frac{\sum_{j=1}^{s} f_E(C_i|x_j)}{s} \tag{13}$$

$$\mu_i = \frac{\sum_{j=1}^{s} f_E(C_i|x_j) x_j}{s w_i} \tag{14}$$

$$\Sigma_i = \frac{\sum_{j=1}^{s} f_E(C_i|x_j)(x_j-\mu_i)(x_j-\mu_i)^T}{s} \tag{15}$$

where s is number of samples.

C. System Approach

By leveraging the intersection data set and the relation between the intersection points and the robots illustrated in (8), GMM effectively separates units into several smaller, more manageable groups for the N-robot system. This strategy significantly reduces system workload by decreasing communication overhead and facilitating local coordination. Specifically, each robot only needs to maintain communication and coordinate operations (if necessary) with others within its assigned group. Besides, a critical characteristic of the log-likelihood function optimized for GMM by the EM algorithm is its non-convexity. This inherent property means that the function can produce multiple local maxima rather than a single global maximum, leading to numerous clustering results that satisfy the aforementioned requirements.

Given the need for effective local coordination among robots, it's crucial to select clustering results that ensure a comparable number of robots within each cluster. This approach enables a balanced workload distribution across clusters. Consequently, a system with such workload-balanced groups becomes inherently more scalable, allowing it to maintain performance and adapt flexibly to changing workloads. To achieve this, a novel 'cost' parameter is proposed to evaluate the local coordination of each cluster in the mentioned system. This parameter is determined by the number of robots participating in sharing information with each other in a well-defined given time. The value can be calculated as following expression:

$$\delta_k = \frac{1}{N_K} \tag{16}$$

with $k = 1..K$, K is the number of clusters and N_K is the number of robots allocated in the cluster k^{th}.

Subsequently, a least squares objective function is formulated to derive the near-optimal clustering result, specifically:

$$f_{obj} = \sqrt{\sum_{k=1}^{K}(\delta_k)^2} \to min \tag{17}$$

The proposed objective function aims to optimize the distribution of robots across clusters, not merely by minimizing the sum of squared parameters, but by leveraging the least squares criterion to achieve a near best-fit clustering result. Specifically, this function endeavors to make the values of $(\delta_1, \delta_2, ..\delta_K)$ as similar and as small as possible. This direct approach effectively ensures an equivalent number of robots within each cluster, thereby fostering a balanced workload distribution across the entire system.

The proposed solution is dynamic clustering method (DCM) and it has two main stages with the intersection data set as the input:

- Stage 1: determine the parameter sets for the GMM model based on the EM algorithm. The GMM model is illustrated as (9) and (11). The input is the intersection data set. The parameter sets are the mean μ_i, the covariance matrix Σ_i and the weight w_i for each i^{th} cluster. The (10), (12) to (15) are used for iterative estimation of the EM algorithm.
- Stage 2: determine the number of robots in each cluster based on (8). Based on that basis, we calculate the cost value of each cluster and objective function according to (16) and (17).

Repeating the stage 1 and stage 2 to find the smallest value of the objective function after a certain iteration. The corresponding parameter sets for the GMM is also the near-optimal result.

IV. RESULTS AND DISCUSSION

The performance of the dynamic clustering technique was assessed through simulation scenarios within a discrete 25×25 map. A fleet of 50 robots was simulated, with each robot tasked to navigate from a randomly generated starting coordinate $(q_{10}, q_{20}, \ldots, q_{N0})$ to a randomly assigned goal coordinate $(q_{1G}, q_{2G}, \ldots, q_{NG})$ and Fig. 2 illustrates such a dataset. Because of the random selection, the simulation run could yield each unique dataset of path intersections. Due to the computational limitations of the simulation computer, the number of robots and the map size are restricted accordingly as mentioned above in the simulation model.

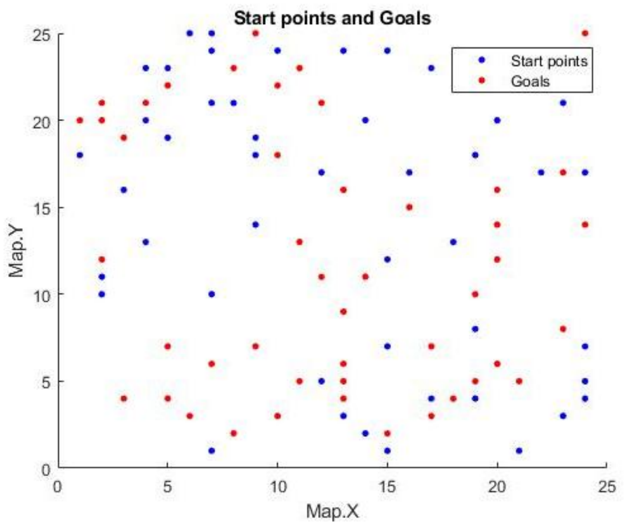

Fig. 2. Starting points and goals

The A* pathfinding algorithm was implemented to compute the optimal path for each robot. The algorithm critically depends on the precise configuration of static obstacles in the working space, as these obstacles directly impact the search space and profoundly modify the geometry, length, and location of the computed optimal paths. Consequently, the given set of intersection points of these paths will inherently vary with the presence and arrangement of obstacles. In contrast, GMM and the EM algorithm operate independently of the environmental context. Their function is to model the distribution of a given dataset, relying solely on the intrinsic properties of the data points themselves, such as coordinates, data density, and point-to-point distances, rather than the environment from which these points are generated. While the distribution of intersection points may differ significantly between obstacle-filled and obstacle-free environments, the fundamental mathematical principles of GMM and EM remain constant.

Based on the above statement, the evaluation is carried out based on quantitative data determined from three proposed simulation scenarios, mentioned as below:

- In the first simulation scenario, DCM was executed 100 times on a single dataset of intersection points to perform an uncertainty quantification. We applied descriptive statistical methods to gain deeper insight into the variability and reliability of the cluster configurations. Specifically, we calculated the mean, standard deviation, and interquartile range across these runs for two key aspects: the number of robots assigned to each cluster and the minimum objective function value. This analysis allowed us to characterize the variability in the cluster composition and the stability of DCM's performance.
- In the second simulation scenario, a sensitivity analysis was conducted to assess the model's performance against varied input data. To do this, we generated four more distinct datasets of intersection points by the A* algorithm with four different sets of 50 random start-and-goal pairs. DCM was then executed 100 times on each of these datasets. This analysis allowed us to evaluate cluster sensitivity by systematically comparing cluster characteristics across the different input conditions and multiple simulation runs.
- The third simulation scenario was designed to assess the generalizability of the proposed DCM framework and empirically demonstrate its environment-independent nature. To achieve this, the model's performance was evaluated under two distinct conditions designed to introduce complexity and constraints. The first condition featured a reduced agent population, with 25 robots operating within a 25×25 obstacle-free map. In contrast, the second condition retained the full contingent of 50 robots but introduced environmental complexity via randomly placed obstacles within the same map. For both cases, the A* algorithm was employed to generate two distinct datasets of intersection points. DCM was executed 100 times on each dataset then. By analyzing the performance under these varied conditions, this evaluation provided robust evidence of the DCM's adaptability and broad applicability.

The number of clusters, K, for the GMM was empirically selected as 3. The selection is motivated by an intuitive experienced partitioning of the workspace, suggesting potential natural groupings of the robots. The EM algorithm was executed with a maximum of 100 iterations. Convergence of the EM algorithm was determined by monitoring the cumulative maximum likelihood achieved over each iteration. The termination happens when the change in this cumulative likelihood between successive iterations fell below a threshold of 1×10^{-6}. It means that the parameter estimation has fully stabilized.

Table I presented the final data from the first simulation scenario. The data included the minimum objective function value (f_{objmin}), the number of robots in each group for every clustering configuration, and r_{cl}, which showed the percentage of the total robot population that was clustered. Table II then provided a statistical summary for the set of f_{objmin} values and the number of robots per cluster.

The statistical indicators of f_{objmin} values demonstrated remarkable stability and consistency. Specifically, the strong central tendency was confirmed by the convergence of the mean of 0.0567 and median of 0.0565, which this difference of 0.0003 is very small compared to the standard deviation of 0.0026. It is suggested a relatively symmetrical data distribution with negligible skew. The top chart of Fig. 3 displayed the distribution of the f_{objmin} values. Furthermore, the low variability was substantiated by a small standard deviation at 0.0026 and a compact interquartile range of 0.0026, indicating that the data points are tightly grouped around the mean. This consistency was reinforced by the close alignment between the calculated 95% confidence interval (2σ) of (0.0515, 0.0619) and the observed empirical range of the data, from 0.0518 to 0.0644.

TABLE I. FINAL DATA OF THE FIRST SIMULATION SCENARIO

f_{objmin}	Number of robots in each cluster			r_{cl} (%)	f_{objmin}	Number of robots in each cluster			r_{cl} (%)
	#1	#2	#3			#1	#2	#3	
0.0593	27	25	43	100	0.0564	31	43	25	100
0.0641	25	24	36	100	0.0563	26	36	33	100
0.0565	36	25	35	100	0.0575	37	25	32	100
0.0553	36	33	27	100	0.0565	37	25	34	100
0.0635	34	28	23	100	0.0565	34	25	37	100
0.0558	25	33	41	100	0.0519	37	31	33	100
0.0577	29	25	43	100	0.0612	39	35	21	100
0.0565	34	37	25	100	0.0553	33	27	36	100
0.0557	33	35	27	100	0.0614	25	32	29	100
0.0606	23	29	41	100	0.0593	27	43	25	100
0.0575	32	37	25	100	0.0593	27	43	25	100
0.0565	37	25	34	100	0.0544	34	28	35	100
0.0565	36	35	25	100	0.0570	25	30	43	100
0.0565	36	25	35	100	0.0525	37	30	33	100
0.0558	41	25	33	100	0.0565	37	34	25	100
0.0553	33	27	36	100	0.0541	33	35	29	100
0.0593	25	27	43	100	0.0584	34	39	23	100
0.0565	25	34	37	100	0.0531	29	35	35	100
0.0558	33	25	41	100	0.0538	31	35	31	100
0.0563	33	36	26	100	0.0595	30	24	38	100
0.0531	35	29	35	100	0.0584	43	28	25	100
0.0531	29	35	35	100	0.0561	32	42	25	100
0.0586	23	33	40	100	0.0565	25	37	34	100
0.0565	34	25	37	100	0.0577	29	43	25	100
0.0563	33	26	36	100	0.0558	25	32	43	100
0.0561	25	42	32	100	0.0586	40	33	23	100
0.0558	43	32	25	100	0.0563	33	26	36	100
0.0525	32	33	34	100	0.0558	32	43	25	100
0.0521	35	32	33	100	0.0593	27	25	43	100
0.0565	35	36	25	100	0.0553	34	35	27	100
0.0565	37	34	25	100	0.0577	29	43	25	100
0.0538	31	35	31	100	0.0570	30	43	25	100
0.0570	25	43	30	100	0.0565	25	36	35	100
0.0563	36	33	26	100	0.0558	25	32	43	100
0.0536	34	35	29	100	0.0628	25	34	26	100
0.0564	31	43	25	100	0.0565	36	25	35	100
0.0541	29	35	33	100	0.0549	28	33	35	100
0.0538	31	35	31	100	0.0565	25	34	37	100
0.0521	35	33	32	100	0.0538	35	31	31	100
0.0614	29	25	32	100	0.0541	33	35	29	100
0.0545	36	28	33	100	0.0552	29	35	31	100
0.0539	30	32	35	100	0.0589	38	31	24	100
0.0614	29	25	32	100	0.0553	33	27	36	100
0.0565	37	34	25	100	0.0538	31	35	31	100
0.0531	35	35	29	100	0.0579	36	38	23	100
0.0586	23	40	33	100	0.0579	36	23	38	100
0.0579	36	23	38	100	0.0565	25	37	34	100
0.0593	25	43	27	100	0.0644	23	36	26	100
0.0558	25	41	33	100	0.0614	29	32	25	100
0.0544	28	34	35	100	0.0577	25	29	43	100

TABLE II. STATISTICAL INDICATORS OF THE FIRST SIMULATION SCENARIO

	f_{objmin}	Number of robots in each cluster		
		#1	#2	#3
Dataset #1				
Mean	0.0567	31.38	32.67	31.82
Standard deviation	0.0026	4.89	5.73	5.9
Median	0.0565	32	33	32.5
First quartile (Q1)	0.0553	27	27.75	25.75
Third quartile (Q3)	0.0579	35	35.25	35.25
Interquartile range (IQR)	0.0026	8	7.5	9.5

A bootstrapping technique was employed with 10,000 bootstrapped resamples to robustly assess the reliability of the mean f_{objmin} values. The bottom chart of Fig. 3 showed the distribution of the bootstrapped f_{objmin} means calculated from the bootstrapped samples. The plot appeared as a normal distribution and was tightly centered around a mean of 0.0567, a value identical to the original sample mean, indicating an accurate and unbiased estimate. The standard deviation of this bootstrap distribution, known as the standard error, was found to be exceptionally small at 0.00026. This was substantially lower than the standard deviation of the original data at 0.0026, which reflects the dispersion of individual data points. It confirmed that the estimate of the mean is highly reliable, despite very few variabilities among the individual f_{objmin} values themselves. Furthermore, the narrow 95% confidence interval derived from the bootstrap analysis of (0.0561, 0.0572) further supported the above conclusion.

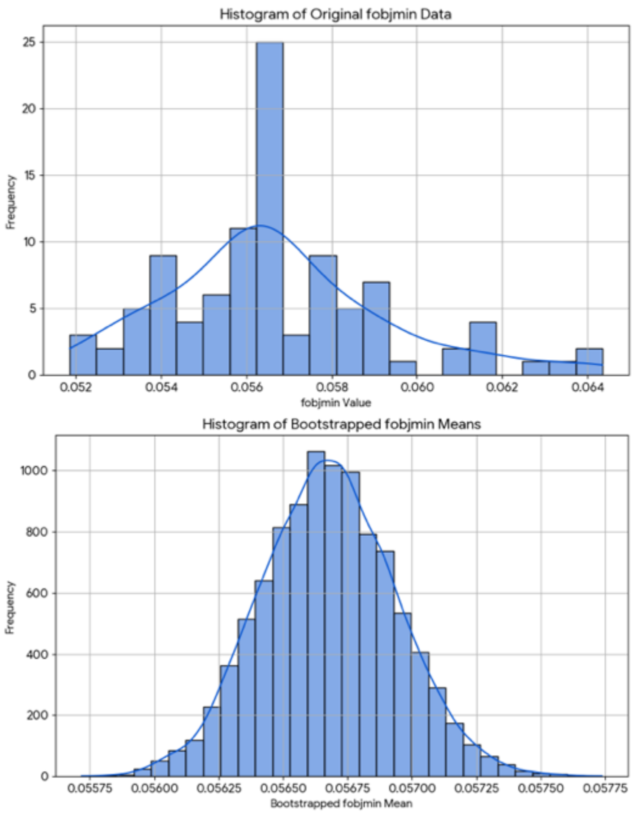

Fig. 3. Histogram of f_{objmin} data (top) and bootstrapped f_{objmin} means (bottom)

The statistical indicators revealed strong similarities in the central tendencies of robot distribution across the three clusters. Specifically, the mean and median values for all clusters were concentrated within a narrow range of 31 to 33 robots, and the proximity of these two metrics within each cluster suggested a relatively symmetrical data distribution. Meanwhile, the primary distinction among the clusters, however, lied in their respective levels of dispersion but not in their central value. Cluster #3 exhibited the highest variability, evidenced by the largest standard deviation of 5.90 and interquartile range of 9.50. This indicated that the number of robots was more dispersed in Cluster #3 than in the others. In contrast, Cluster #1 and Cluster #2 showed lower variability signifying more stability in their robot counts. In summary, while the overall robot distribution is fairly equitable, the clusters differ notably in their consistency. Cluster #3 is characterized by significant population variability, whereas Clusters #1 and #2 maintain more stable configurations.

The first simulation scenario disclosed the performance reliability in the face of cluster configuration variation. The overall performance, quantified by the f_{objmin} values, exhibited exceptional stability. The conclusion was substantiated by the narrow 95% confidence interval of (0.0561, 0.0572) derived via bootstrapping. This consistency was particularly noteworthy when contrasted with the significant variations in the underlying cluster configurations, where the number of robots per cluster fluctuates between different simulation runs. The ability to maintain a stable and consistent level, even when the distribution of robots changed, demonstrated the performance reliability of DCM. The method could effectively adapt to internal variations without compromising its overall objective.

Table III presented the summary of the statistical indicators for four datasets of the second simulation scenario, from Dataset #2 to Dataset #5. There were key metrics for the f_{objmin} values and the number of robots per cluster (#1, #2, and #3). Additionally, it included the statistical results derived from bootstrapping the original f_{objmin} samples. The bootstrapping method was applied with 10,000 samples to estimate the stability of the mean.

While analysis of individual datasets, from Dataset #2 to Dataset #5, suggested that DCM maintained a stable and consistent performance level, mirroring the reliability observed in the first simulation scenario, corresponding to Dataset #1. A comprehensive sensitivity analysis across all five datasets confirmed the different core strength of DCM, its remarkable responsiveness to varying conditions. This pronounced sensitivity was quantitatively evident in both the objective function and the distribution of robots. The mean value of f_{objmin} fluctuated significantly, ranging from a lowest data of 0.0563 in Dataset #4 to a highest data of 0.0645 in Dataset #5, underscoring a strong dependency on the specific dataset. This was further corroborated by robot distribution metrics, where the standard deviation per cluster varied considerably across Datasets #2 through #5, reflecting DCM's success in dynamically reallocating robots to match each scenario's specific input set. Therefore, the sensitivity analysis confirmed the adaptive strength of DCM without the existence of a single, universally optimal clustering structure. DCM had the ability to tolerate internal configuration changes to maintain stability for one problem while completely re-forming that configuration to adapt to new ones.

TABLE III. STATISTICAL INDICATORS OF THE SECOND SIMULATION SCENARIO

	boot-strapped samples	f_{objmin}	**Number of robots in each cluster**		
			#1	**#2**	**#3**
Dataset #2					
Mean	0.0642	0.0642	29.81	30.28	29.75
Standard deviation	0.00021	0.0021	7.65	7.88	8.21
Median	0.0642	0.0653	29	29	29
First quartile (Q1)	0.0640	0.0646	23	23	22
Third quartile (Q3)	0.0643	0.0655	36.25	38	38
Interquartile range (IQR)	0.00028	0.0009	13.25	15	16
Confidence Interval (2σ)	(0.0638, 0.0646)	(0.0599, 0.0685)			
Dataset #3					
Mean	0.0611	0.0611	29.96	29.46	31.29
Standard deviation	0.00014	0.0014	6.89	6.13	7.18
Median	0.0611	0.0615	28	27	28
First quartile (Q1)	0.0610	0.0594	24	24	25.5
Third quartile (Q3)	0.0612	0.0623	37.25	37	38
Interquartile range (IQR)	0.00018	0.0029	13.25	13	12.5
Confidence Interval (2σ)	(0.0608, 0.0613)	(0.0583, 0.0638)			
Dataset #4					
Mean	0.0563	0.0563	32.15	32.13	31.27
Standard deviation	0.00017	0.0017	5.12	5.33	4.94
Median	0.0563	0.0566	29	29	29
First quartile (Q1)	0.0562	0.0564	27	27	27
Third quartile (Q3)	0.0564	0.0570	37	37	36
Interquartile range (IQR)	0.00022	0.0006	10	10	9
Confidence Interval (2σ)	(0.0560, 0.0566)	(0.0529, 0.0596)			
Dataset #5					
Mean	0.0645	0.0645	30.57	29.3	27.69
Standard deviation	0.0003	0.0031	5.99	6.08	6.32
Median	0.0645	0.0636	34	32	28
First quartile (Q1)	0.0643	0.0625	25.5	24	22
Third quartile (Q3)	0.0647	0.0667	35	34	34
Interquartile range (IQR)	0.0004	0.0042	9.5	10	12
Confidence Interval (2σ)	(0.0639, 0.0651)	(0.0584, 0.0706)			

Table IV presented the summary of the statistical indicators for two datasets, Dataset #6 and Dataset #7, tương ứng với two cases of the third simulation scenario. They are reduced agent population, with 25 robots operating within a 25x25 obstacle-free map, and the full contingent of 50 robots within the 25x25 obstacle-filled map. There were key metrics for the f_{objmin} values and the number of robots per cluster

(#1, #2, and #3). Meanwhile, it also included the statistical results derived from bootstrapping the original f_{objmin} samples. The bootstrapping method was applied with 10,000 samples to estimate the stability of the mean.

A comparative statistical analysis of Dataset #6 and Dataset #7 revealed a consistently stable and reliable system performance across different operational cases. The mean of f_{objmin} proved to be a stable estimator of the true population mean, a conclusion strongly supported by bootstrapping methods. For instance, in the Dataset #7, the bootstrapped mean of 0.0626 was identical to the original sample mean, and its corresponding standard error of 0.00023 was significantly smaller than the original data's standard deviation of 0.0023. The narrow 95% confidence interval, from 0.06210 to 0.06301, and the convergence of the bootstrapped sampling distribution to an approximately normal shape, even when the original data was skewed, reinforced the validity and stability of the mean estimate.

TABLE IV. STATISTICAL INDICATORS OF THE THIRD SIMULATION SCENARIO

	boot-strapped samples	f_{objmin}	**Number of robots in each cluster**		
			#1	**#2**	**#3**
Dataset #6					
Mean	0.1413	0.1413	13.24	13.34	13.82
Standard deviation	0.0011	0.0107	3.3	3.2	3.14
Median	0.1413	0.1355	14	14	15
First quartile (Q1)	0.1406	0.1355	10	10	11
Third quartile (Q3)	0.142	0.15	15	16	16
Interquartile range (IQR)	0.0014	0.0145	5	6	5
Confidence Interval (2σ)	(0.1391, 0.1434)	(0.1200, 0.1626)			
Dataset #7					
Mean	0.0626	0.0626	30.76	30.14	31.45
Standard deviation	0.00023	0.0023	8.52	8.25	8.41
Median	0.0625	0.0623	27.5	26	30
First quartile (Q1)	0.0624	0.0619	23	23	23
Third quartile (Q3)	0.0627	0.0625	38	37.25	37.25
Interquartile range (IQR)	0.00031	0.00069	15	14.25	14.25
Confidence Interval (2σ)	(0.06210, 0.06301)	(0.05799, 0.06712)			

Furthermore, the robot distribution mechanism demonstrated a high degree of equity and balance in both analyses. The clusters consistently exhibited similar measures of central tendency and dispersion, indicating that the partitioning algorithm was unbiased and that assignment variability was uniform across experimental runs. Crucially, these analyses presented a comparable stability in the DCM's performance under two distinct cases: reduced agent population, with 25 robots in the obstacle-free map, and the full contingent of 50 robots in the obstacle-filled map. When considered alongside findings from the first simulation scenario, this empirically demonstrated the generalizability and environment-independent nature of the proposed DCM.

V. CONCLUSION AND FUTURE WORK

The dynamic clustering method is proven a robust and adaptive approach for grouping robots based on their path intersections. The framework uses a Gaussian Mixture Model and the Expectation-Maximization algorithm to cluster the intersection points into a predetermined number of groups, which was empirically set to three for these simulations. Across the trials, 100% of the robots were successfully assigned to a cluster. A key "cost" parameter, formulated a least squares objective function, is introduced in this method. The function is used for evaluating the performance of a given clustering configuration. Analysis reveals that the DCM's performance is exceptionally stable and reliable for a single dataset, maintaining a consistent objective function value even as the number of robots per cluster fluctuates between runs. Rather than simply creating smaller groups, the system identifies near-optimal configurations that ensure overall stable performance. Meanwhile, the adaptive sensitivity, when presented with entirely new datasets, is also achieved. DCM re-configures the clusters to adapt to the new configurations, demonstrating its ability to generalize across different scenarios. This dual capacity for maintaining stability while remaining adaptive to new conditions confirms the DCM is a highly effective and environment-independent framework for robot coordination.

Although DCM has shown encouraging initial results, the proposed future developments aim to deepen the analytical understanding of this method. Specifically, the analysis will extend beyond performance metrics to evaluate the spatial stability of the clusters. Furthermore, the initially subjective choice of K=3 will be validated through a data-driven, objective selection process using established indices to determine the optimal model complexity. Finally, the research will investigate the quantitative relationship between the topological properties of the input data, such as the density or dispersion of intersection points or their dynamic updates, and DCM's performance to realize the system's sensitivity. A separate avenue of future research will involve assessing the performance and viability of DCM when adapted for IoT-enabled robot systems.

ACKNOWLEDGMENT

This research was supported by the Program "Supporting Research, Development and Technology Application of Industry 4.0" (KC-4.0/19-30) of the Ministry of Science and Technology in Vietnam through the funding of the Project "Research, design, manufacture, and integration for a system of multiple autonomous flying robots, applying for environmental exploration, search and rescue", No. KC-4.0-49/19-25.

REFERENCES

[1] F. Rubio, F. Valero, and C. Llopis-Albert, “A review of mobile robots: Concepts, methods, theoretical framework, and applications,” *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, p. 1729881419839596, 2019, doi: 10.1177/1729881419839596.

[2] R. Raj and A. Kos, “A Comprehensive Study of Mobile Robot: History, Developments, Applications, and Future Research Perspectives,” *Applied Sciences*, vol. 12, no. 14, 2022, doi: 10.3390/app12146951.

[3] L. Liu, X. Wang, X. Yang, H. Liu, J. Li, and P. Wang, “Path planning techniques for mobile robots: Review and prospect,” *Expert Systems*

with Applications, vol. 227, p. 120254, 2023, doi: 10.1016/j.eswa.2023.120254.

[4] J. A. Abdulsaheb and D. J. Kadhim, "Classical and Heuristic Approaches for Mobile Robot Path Planning: A Survey," *Robotics*, vol. 12, no. 4, 2023, doi: 10.3390/robotics12040093.

[5] H. Qin, S. Shao, T. Wang, X. Yu, Y. Jiang, and Z. Cao, "Review of Autonomous Path Planning Algorithms for Mobile Robots," *Drones*, vol. 7, no. 3, 2023, doi: 10.3390/drones7030211.

[6] S. Lin, A. Liu, J. Wang, and X. Kong, "A Review of Path-Planning Approaches for Multiple Mobile Robots," *Machines*, vol. 10, no. 9, 2022, doi: 10.3390/machines10090773.

[7] A. Loganathan and N. S. M. Ahmad, "A systematic review on recent advances in autonomous mobile robot navigation," *Engineering Science and Technology, an International Journal*, vol. 40, p. 101343, 2023, doi: 10.1016/j.jestch.2023.101343.

[8] M. N. Wahab, S. Nefti-Meziani, and A. Atyabi, "A comparative review on mobile robot path planning: Classical or meta-heuristic methods?," *Annual Reviews in Control*, vol. 50, pp. 233–252, 2020, doi: 10.1016/j.arcontrol.2020.10.001.

[9] A. S. Aguiar, F. N. dos Santos, J. B. Cunha, H. Sobreira, and A. J. Sousa, "Localization and Mapping for Robots in Agriculture and Forestry: A Survey," *Robotics*, vol. 9, no. 4, 2020, doi: 10.3390/robotics9040097.

[10] Y. Liu, S. Wang, Y. Xie, T. Xiong, and M. Wu, "A Review of Sensing Technologies for Indoor Autonomous Mobile Robots," *Sensors*, vol. 24, no. 4, 2024, doi: 10.3390/s24041222.

[11] M. A. K. Niloy *et al.*, "Critical Design and Control Issues of Indoor Autonomous Mobile Robots: A Review," *IEEE Access*, vol. 9, pp. 35338–35370, 2021, doi: 10.1109/ACCESS.2021.3062557.

[12] R. B. Sousa, H. Sobreira, and A. P. Moreira, "A systematic literature review on long-term localization and mapping for mobile robots," *Journal of Field Robotics*, vol. 40, no. 5, pp. 1245–1322, 2023, doi: 10.1002/rob.22170.

[13] F. Quinton, C. Grand, and C. Lesire, "Market approaches to the multi-robot task allocation problem: a survey," *Journal of Intelligent \& Robotic Systems*, vol. 107, no. 2, p. 29, 2023, doi: 10.1007/s10846-022-01803-0.

[14] H. Chakraa, F. Guérin, E. Leclercq, and D. Lefebvre, "Optimization techniques for Multi-Robot Task Allocation problems: Review on the state-of-the-art," *Robotics and Autonomous Systems*, vol. 168, p. 104492, 2023, doi: 10.1016/j.robot.2023.104492.

[15] R. J. Alitappeh and K. Jeddisaravi, "Multi-robot exploration in task allocation problem," *Applied Intelligence*, vol. 52, no. 2, pp. 2189–2211, 2022, doi: 10.1007/s10489-021-02483-3.

[16] A. Khamis, A. Hussein, and A. Elmogy, "Multi-robot task allocation: a review of the state-of-the-art," in *Cooperative Robots and Sensor Networks 2015*, pp. 31–51, 2015, doi: 10.1007/978-3-319-18299-5_2.

[17] H. Kabir, M.-L. Tham, and Y. C. Chang, "Internet of robotic things for mobile robots: Concepts, technologies, challenges, applications, and future directions," *Digital Communications and Networks*, vol. 9, no. 6, pp. 1265–1290, 2023, doi: 10.1016/j.dcan.2023.05.006.

[18] J. Gielis, A. Shankar, and A. Prorok, "A Critical Review of Communications in Multi-robot Systems," *Current Robotics Reports*, vol. 3, no. 4, pp. 213–225, 2022, doi: 10.1007/s43154-022-00090-9.

[19] L. C. Garaffa, M. Basso, A. A. Konzen, and E. P. de Freitas, "Reinforcement Learning for Mobile Robotics Exploration: A Survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 8, pp. 3796–3810, 2023, doi: 10.1109/TNNLS.2021.3124466.

[20] J. Orr and A. Dutta, "Multi-Agent Deep Reinforcement Learning for Multi-Robot Applications: A Survey," *Sensors*, vol. 23, no. 7, 2023, doi: 10.3390/s23073625.

[21] H. Sun, W. Zhang, R. Yu, and Y. Zhang, "Motion Planning for Mobile Robots—Focusing on Deep Reinforcement Learning: A Systematic Review," *IEEE Access*, vol. 9, pp. 69061–69081, 2021, doi: 10.1109/ACCESS.2021.3076530.

[22] K. Zhu and T. Zhang, "Deep reinforcement learning based mobile robot navigation: A review," *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 674–691, 2021, doi: 10.26599/TST.2021.9010012.

[23] S. Cohen and N. Agmon, "Recent Advances in Formations of Multiple Robots," *Current Robotics Reports*, vol. 2, no. 2, pp. 159–175, 2021, doi: 10.1007/s43154-021-00049-2.

[24] R. Almadhoun, T. Taha, L. Seneviratne, and Y. Zweiri, "A survey on multi-robot coverage path planning for model reconstruction and mapping," *SN Applied Sciences*, vol. 1, no. 8, p. 847, 2019, doi: 10.1007/s42452-019-0872-y.

[25] N. Sharma, J. K. Pandey, and S. Mondal, "A Review of Mobile Robots: Applications and Future Prospect," *International Journal of Precision Engineering and Manufacturing*, vol. 24, no. 9, pp. 1695–1706, 2023, doi: 10.1007/s12541-023-00876-7.

[26] D. S. Drew, "Multi-Agent Systems for Search and Rescue Applications," *Current Robotics Reports*, vol. 2, no. 2, pp. 189–200, 2021, doi: 10.1007/s43154-021-00048-3.

[27] R. N. Darmanin and M. K. Bugeja, "A review on multi-robot systems categorised by application domain," in *2017 25th Mediterranean Conference on Control and Automation (MED)*, pp. 701–706, 2017, doi: 10.1109/MED.2017.7984200.

[28] Y. Rizk, M. Awad, and E. W. Tunstel, "Cooperative Heterogeneous Multi-Robot Systems: A Survey," *ACM Computing Surveys*, vol. 52, no. 2, 2019, doi: 10.1145/3303848.

[29] S. G. Tzafestas, "Mobile Robot Control and Navigation: A Global Overview," *Journal of Intelligent \& Robotic Systems*, vol. 91, no. 1, pp. 35–58, 2018, doi: 10.1007/s10846-018-0805-9.

[30] S. Huang, R. S. H. Teo, and K. K. Tan, "Collision avoidance of multi unmanned aerial vehicles: A review," *Annual Reviews in Control*, vol. 48, pp. 147–164, 2019, doi: 10.1016/j.arcontrol.2019.10.001.

[31] A. N. Rafai, N. Adzhar, and N. I. Jaini, "A Review on Path Planning and Obstacle Avoidance Algorithms for Autonomous Mobile Robots," *Journal of Robotics*, vol. 2022, no. 1, p. 2538220, 2022, doi: 10.1155/2022/2538220.

[32] J.-C. Latombe, "Exact Cell Decomposition and Approximate Cell Decomposition," in *Robot Motion Planning*, pp. 200–294, 2012, doi: 10.1007/978-1-4615-4022-9_6.

[33] A. Abbadi and R. Matousek, "Hybrid rule-based motion planner for mobile robot in cluttered workspace," *Soft Computing*, vol. 22, no. 6, pp. 1815–1831, 2018, doi: 10.1007/s00500-016-2103-4.

[34] R. Bähnemann, N. Lawrance, J. J. Chung, M. Pantic, R. Siegwart, and J. Nieto, "Revisiting Boustrophedon Coverage Path Planning as a Generalized Traveling Salesman Problem," in *Field and Service Robotics*, pp. 277–290, 2021, doi: 10.1007/978-981-15-9460-1_20.

[35] S. W. Cho, H. J. Park, H. Lee, D. H. Shim, and S.-Y. Kim, "Coverage path planning for multiple unmanned aerial vehicles in maritime search and rescue operations," *Computers \& Industrial Engineering*, vol. 161, p. 107612, 2021, doi: 10.1016/j.cie.2021.107612.

[36] O. A. A. Salama, M. E. H. Eltaib, H. A. Mohamed, and O. Salah, "RCD: Radial Cell Decomposition Algorithm for Mobile Robot Path Planning," *IEEE Access*, vol. 9, pp. 149982–149992, 2021, doi: 10.1109/ACCESS.2021.3125105.

[37] S. K. Debnath and others, "Different Cell Decomposition Path Planning Methods for Unmanned Air Vehicles-A Review," in *Lecture Notes in Electrical Engineering*, pp. 99–111, 2020, doi: 10.1007/978-981-15-5281-6_8.

[38] I. Iswanto, O. Wahyunggoro, and A. I. Cahyadi, "Quadrotor Path Planning Based On Modified Fuzzy Cell Decomposition Algorithm," *TELKOMNIKA*, vol. 14, no. 2, pp. 655–664, 2016, doi: 10.12928/TELKOMNIKA.v14i1.2989.

[39] F. Lingelbach, "Path planning using probabilistic cell decomposition," in *IEEE International Conference on Robotics and Automation*, vol. 1, pp. 467–472, 2004, doi: 10.1109/ROBOT.2004.1307193.

[40] X. Huang, M. Sun, H. Zhou, and S. Liu, "A Multi-Robot Coverage Path Planning Algorithm for the Environment With Multiple Land Cover Types," *IEEE Access*, vol. 8, pp. 198101–198117, 2020, doi: 10.1109/ACCESS.2020.3027422.

[41] A. Abbadi and V. Prenosil, "Safe path planning using cell decomposition," in *International Conference on Distance Learning, Simulation and Communication*, pp. 8–14, 2015.

[42] K. R. Guruprasad and T. D. Ranjitha, "CPC Algorithm: Exact Area Coverage by a Mobile Robot Using Approximate Cellular

Decomposition," *Robotica*, vol. 39, no. 7, pp. 1141–1162, 2021, doi: 10.1017/S026357472000096X.

[43] T. Arney, "An efficient solution to autonomous path planning by Approximate Cell Decomposition," in *2007 Third International Conference on Information and Automation for Sustainability*, pp. 88–93, 2007, doi: 10.1109/ICIAFS.2007.4544785.

[44] P. T. Kyaw, A. Paing, T. T. Thu, R. E. Mohan, A. V. Le, and P. Veerajagadheswar, "Coverage Path Planning for Decomposition Reconfigurable Grid-Maps Using Deep Reinforcement Learning Based Travelling Salesman Problem," *IEEE Access*, vol. 8, pp. 225945–225956, 2020, doi: 10.1109/ACCESS.2020.3045027.

[45] V. G. Nair and K. R. Guruprasad, "MR-SimExCoverage: Multi-robot Simultaneous Exploration and Coverage," *Computers \& Electrical Engineering*, vol. 85, p. 106680, 2020, doi: 10.1016/j.compeleceng.2020.106680.

[46] R. V Cowlagi and P. Tsiotras, "Beyond quadtrees: Cell decompositions for path planning using wavelet transforms," in *2007 46th IEEE Conference on Decision and Control*, pp. 1392–1397, 2007, doi: 10.1109/CDC.2007.4434146.

[47] C. Cai and S. Ferrari, "Information-Driven Sensor Path Planning by Approximate Cell Decomposition," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 3, pp. 672–689, 2009, doi: 10.1109/TSMCB.2008.2008561.

[48] Z. E. Kanoon, A. S. Al-Araji, and M. N. Abdullah, "Enhancement of Cell Decomposition Path-Planning Algorithm for Autonomous Mobile Robot Based on an Intelligent Hybrid Optimization Method," *International Journal of Intelligent Engineering and Systems*, vol. 15, no. 3, pp. 161–175, 2022, doi: 10.22266/ijies2022.0630.14.

[49] O. Arslan, D. P. Guralnik, and D. E. Koditschek, "Coordinated Robot Navigation via Hierarchical Clustering," *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 352–371, 2016, doi: 10.1109/TRO.2016.2524018.

[50] P. Bhattacharya and M. L. Gavrilova, "Density-Based Clustering Based on Topological Properties of the Data Set," in *Generalized Voronoi Diagram: A Geometry-Based Approach to Computational Intelligence*, pp. 197–214, 2008.

[51] V. G. Nair, R. S. Adarsh, K. P. Jayalakshmi, M. V Dileep, and K. R. Guruprasad, "Cooperative Online Workspace Allocation in the Presence of Obstacles for Multi-robot Simultaneous Exploration and Coverage Path Planning Problem," *International Journal of Control, Automation and Systems*, vol. 21, no. 7, pp. 2338–2349, 2023, doi: 10.1007/s12555-022-0182-9.

[52] V. G. Nair and K. R. Guruprasad, "2D-VPC: An Efficient Coverage Algorithm for Multiple Autonomous Vehicles," *International Journal of Control, Automation and Systems*, vol. 19, no. 8, pp. 2891–2901, 2021, doi: 10.1007/s12555-020-0389-6.

[53] A. Dutta, A. Bhattacharya, O. P. Kreidl, A. Ghosh, and P. Dasgupta, "Multi-robot informative path planning in unknown environments through continuous region partitioning," *International Journal of Advanced Robotic Systems*, vol. 17, no. 6, 2020, doi: 10.1177/1729881420970461.

[54] J. Kim and H.-I. Son, "a voronoi diagram-based workspace partition for weak cooperation of multi-robot system in orchard," *IEEE Access*, vol. 8, pp. 20676–20686, 2020, doi: 10.1109/ACCESS.2020.2969449.

[55] N. B. Cruz, N. Nedjah, and L. de M. Mourelle, "Robust distributed spatial clustering for swarm robotic based systems," *Applied Soft Computing*, vol. 57, pp. 727–737, 2017, doi: 10.1016/j.asoc.2016.06.002.

[56] G. Di Caro, F. Ducatelle, and L. M. Gambardella, "A fully distributed communication-based approach for spatial clustering in robotic swarms," in *Proceedings of the 2nd Autonomous Robots and Multirobot Systems Workshop (ARMS)*, pp. 153–171, 2012.

[57] A. Prorok, A. Bahr, and A. Martinoli, "Low-cost collaborative localization for large-scale multi-robot systems," in *2012 IEEE International Conference on Robotics and Automation*, pp. 4236–4241, 2012, doi: 10.1109/ICRA.2012.6225016.

[58] J. Chen, C. Du, Y. Zhang, P. Han, and W. Wei, "A Clustering-Based Coverage Path Planning Method for Autonomous Heterogeneous UAVs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25546–25556, 2022, doi: 10.1109/TITS.2021.3066240.

[59] K. Zhang, E. G. Collins, and D. Shi, "Centralized and distributed task allocation in multi-robot teams via a stochastic clustering auction," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 7, no. 2, 2012, doi: 10.1145/2240166.2240171.

[60] J. G. Martin, F. J. Muros, J. M. Maestre, and E. F. Camacho, "Multi-robot task allocation clustering based on game theory," *Robotics and Autonomous Systems*, vol. 161, p. 104314, 2023, doi: 10.1016/j.robot.2022.104314.

[61] J. Hu, P. Bhowmick, I. Jang, F. Arvin, and A. Lanzon, "A Decentralized Cluster Formation Containment Framework for Multirobot Systems," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1936–1955, 2021, doi: 10.1109/TRO.2021.3071615.

[62] A. Dutta, U. Vladimir, A. Asai, and E. Czarnecki, "Coalition Formation for Multi-Robot Task Allocation via Correlation Clustering," *Cybernetics and Systems*, vol. 50, no. 8, pp. 711–728, 2019, doi: 10.1080/01969722.2019.1677334.

[63] N. Nedjah, L. M. Ribeiro, and L. de Macedo Mourelle, "Communication optimization for efficient dynamic task allocation in swarm robotics," *Applied Soft Computing*, vol. 105, p. 107297, 2021, doi: 10.1016/j.asoc.2021.107297.

[64] Q. Du, D. Wang, and L. Sha, "Recognition of Mobile Robot Navigation Path Based on K-Means Algorithm," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 34, no. 08, p. 2059028, 2019, doi: 10.1142/S0218001420590284.

[65] S. Opiyo, C. Okinda, J. Zhou, E. Mwangi, and N. Makange, "Medial axis-based machine-vision system for orchard robot navigation," *Computers and Electronics in Agriculture*, vol. 185, p. 106153, 2021, doi: 10.1016/j.compag.2021.106153.

[66] H. Rong, A. Ramirez-Serrano, L. Guan, and Y. Gao, "Image Object Extraction Based on Semantic Detection and Improved K-Means Algorithm," *IEEE Access*, vol. 8, pp. 171129–171139, 2020, doi: 10.1109/ACCESS.2020.3025193.

[67] H. Zhang and Q. Peng, "PSO and K-means-based semantic segmentation toward agricultural products," *Future Generation Computer Systems*, vol. 126, pp. 82–87, 2022, doi: 10.1016/j.future.2021.06.059.

[68] Y. Yuan, P. Yang, H. Jiang, and T. Shi, "A Multi-Robot Task Allocation Method Based on the Synergy of the K-Means++ Algorithm and the Particle Swarm Algorithm," *Biomimetics*, vol. 9, no. 11, 2024, doi: 10.3390/biomimetics9110694.

[69] E. Murugappan, S. Nachiappan, R. Shams, G. Mark, and H. K. Chan, "Performance analysis of clustering methods for balanced multi-robot task allocations," *International Journal of Production Research*, vol. 60, no. 14, pp. 4576–4591, 2022, doi: 10.1080/00207543.2021.1955994.

[70] J. Li and F. Yang, "Task assignment strategy for multi-robot based on improved Grey Wolf Optimizer," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 12, pp. 6319–6335, 2020, doi: 10.1007/s12652-020-02224-3.

[71] F. Janati, F. Abdollahi, S. Shiry Ghidary, M. Jannatifar, J. Baltes, and S. Sadeghnejad, "Multi-robot Task Allocation Using Clustering Method," in *Advances in Intelligent Systems and Computing*, pp. 233–247, 2016, doi: 10.1007/978-3-319-31293-4_19.

[72] M. Elango, S. Nachiappan, and M. K. Tiwari, "Balancing task allocation in multi-robot systems using K-means clustering and auction based mechanisms," *Expert Systems with Applications*, vol. 38, no. 6, pp. 6486–6491, 2011, doi: 10.1016/j.eswa.2010.11.097.

[73] W.-C. Wang and R. Chen, "An Effective Way to Large-Scale Robot-Path-Planning Using a Hybrid Approach of Pre-Clustering and Greedy Heuristic," *Applied Artificial Intelligence*, vol. 34, no. 14, pp. 1159–1175, 2020, doi: 10.1080/08839514.2020.1824094.

[74] A. Solanas and M. A. Garcia, "Coordinated multi-robot exploration through unsupervised clustering of unknown space," in *International Conference on Intelligent Robots and Systems (IROS)*, vol. 1, pp. 717–721, 2004, doi: 10.1109/IROS.2004.1389437.

[75] S. Sharma, R. Tiwari, A. Shukla, and J. Yadav, "Canopy Clustering Based Multi Robot Area Exploration," *IFAC Proceedings Volumes*, vol. 47, no. 1, pp. 505–510, 2014, doi: 10.3182/20140313-3-IN-3024.00253.

[76] H. Zhang, L. Bai, J. Qi, and Y. Xiao, "Area Coverage of Swarm Robotics Based on Anti-Flocking Framework with Dynamical Clustering," in *International Conference on Unmanned Systems (ICUS)*, pp. 209–213, 2022, doi: 10.1109/ICUS55513.2022.9986539.

[77] N. A. A. Rahman, K. S. M. Sahari, N. A. Hamid, and Y. C. Hou, "A coverage path planning approach for autonomous radiation mapping

with a mobile robot," *International Journal of Advanced Robotic Systems*, vol. 19, no. 4, 2022, doi: 10.1177/17298806221116483.

[78] N.-J. Cho, S.-H. Lee, and I.-H. Suh, "Modeling and evaluating Gaussian mixture model based on motion granularity," *Intelligent Service Robotics*, vol. 9, no. 2, pp. 123–139, 2016, doi: 10.1007/s11370-015-0190-1.

[79] J. Yan, Y. Wu, K. Ji, C. Cheng, and Y. Zheng, "A novel trajectory learning method for robotic arms based on Gaussian Mixture Model and k-value selection algorithm," *PLoS One*, vol. 20, pp. 1–19, 2025, doi: 10.1371/journal.pone.0318403.

[80] S.-H. Lee, I.-H. Suh, S. Calinon, and R. Johansson, "Autonomous framework for segmenting robot trajectories of manipulation task," *Autonomous Robots*, vol. 38, no. 2, pp. 107–141, 2015, doi: 10.1007/s10514-014-9397-9.

[81] Y. Wang, Y. Hu, S. El Zaatari, W. Li, and Y. Zhou, "Optimised Learning from Demonstrations for Collaborative Robots," *Robotics and Computer-Integrated Manufacturing*, vol. 71, p. 102169, 2021, doi: 10.1016/j.rcim.2021.102169.

[82] C. Ye, J. Yang, and H. Ding, "Bagging for Gaussian mixture regression in robot learning from demonstration," *Journal of Intelligent Manufacturing*, vol. 33, no. 3, pp. 867–879, 2022, doi: 10.1007/s10845-020-01686-8.

[83] J. Lou, "Crawling robot manipulator tracking based on gaussian mixture model of machine vision," *Neural Computing and Applications*, vol. 34, no. 9, pp. 6683–6693, 2022, doi: 10.1007/s00521-021-06063-x.

[84] T. M. N. U. Akhund and others, "IoST-Enabled Robotic Arm Control and Abnormality Prediction Using Minimal Flex Sensors and Gaussian Mixture Models," *IEEE Access*, vol. 12, pp. 45265–45278, 2024, doi: 10.1109/ACCESS.2024.3380360.

[85] A. Mora, A. Prados, A. Mendez, R. Barber, and S. Garrido, "Sensor Fusion for Social Navigation on a Mobile Robot Based on Fast Marching Square and Gaussian Mixture Model," *Sensors*, vol. 22, no. 22, 2022, doi: 10.3390/s22228728.

[86] Y. Yuan, J. Liu, W. Chi, G. Chen, and L. Sun, "A Gaussian Mixture Model Based Fast Motion Planning Method Through Online Environmental Feature Learning," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 4, pp. 3955–3965, 2023, doi: 10.1109/TIE.2022.3177758.

[87] M. Park, S. An, J. Seo, and H. Oh, "Autonomous Source Search for UAVs Using Gaussian Mixture Model-Based Infotaxis: Algorithm and Flight Experiments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 6, pp. 4238–4254, 2021, doi: 10.1109/TAES.2021.3098132.

[88] P. Yao, Q. Zhu, and R. Zhao, "Gaussian Mixture Model and Self-Organizing Map Neural-Network-Based Coverage for Target Search in Curve-Shape Area," *IEEE Transactions on Cybernetics*, vol. 52, no. 5, pp. 3971–3983, 2022, doi: 10.1109/TCYB.2020.3019255.

[89] D. Reynolds, "Gaussian Mixture Models," in *Encyclopedia of Biometrics*, pp. 827–832, 2015, doi: 10.1007/978-1-4899-7488-4_196.

[90] F. Najar, S. Bourouis, N. Bouguila, and S. Belghith, "A Comparison Between Different Gaussian-Based Mixture Models," in *International Conference on Computer Systems and Applications (AICCSA)*, pp. 704–708, 2017, doi: 10.1109/AICCSA.2017.108.

[91] H. Wan, H. Wang, B. Scotney, and J. Liu, "A Novel Gaussian Mixture Model for Classification," in *International Conference on Systems, Man and Cybernetics (SMC)*, pp. 3298–3303, 2019, doi: 10.1109/SMC.2019.8914215.

DOI: 10.15625/vap.2023.0110

Digitizing Working Space of the Multi Mobile Robots System Using an Approximate Cell Decomposition Method

Hung Truong Xuan
Vietnam National Space Center, Vietnam Academy of Science and Technology
Hanoi, Vietnam
txhung@vnsc.org.vn

Thang Pham Manh
University of Engineering and Technology, Vietnam National University
Hanoi, Vietnam
thangpm@vnu.edu.vn

Thuc Nguyen Van
Vietnam National Space Center, Vietnam Academy of Science and Technology
Hanoi, Vietnam
nvthuc@vnsc.org.vn

Abstract*— Having a large number of units, the operation of the multi mobile robots system relies on several crucial features, such as path planning, searching optimal paths, navigation, localization and tasks allocation. The system performance is influenced by not only external environmental factors but also its own ones. The interaction between robots is considered as one of the most important internal factors during the system operation. A working space of the system is continuous, bounded and well-determined. It might be having some uncertainty but they will be determined over time. In order to be aware of the workspace, a map is proposed to describe the working space of the system. This tool can be used for sharing the data between robots, helping robots perform navigation and positioning. In addition, the map also provides the necessary data to execute path planning, finding optimal paths and task allocations. The paper introduces an approach for mapping using an Approximate Cell Decomposition method. The proposed approach divides the continuous and bounded working area into a finite number of cells then the workspace awareness can be converted to accessible data about the cells in the map. The robot is located by finding out the coordinates of the occupied cell in the map and the robot route consists of a series of consecutive cells forming one path. The expanded data can contain the coordinates of the static obstacles as well as the robots being active. In the future, the mapping approach is used together with other algorithms for path planning and finding optimal paths of the robots.

***Keywords*— Mobile robot, multi mobile robots system, approximate cell decomposition.**

I. INTRODUCTION

The initial concept of multi-robots system or multi mobile robots system was formed in the late 1980s along with several works in the distributed robotic systems in which the research objects are mobile robots. Since then, there are a lot of issues and topics are considered to look into, for example, biological inspirations; communication; architectures; task allocation and control; localization and mapping; exploration; object transport and manipulation; motion coordination and control; formation control; reconfigurable robots and multi-agent robots; multi-sensor fusion and so on [1], [2], [3]. The robots can share tasks to achieve a common goal by exchanging information, coordination and cooperation during the operation. They may operate under the same environmental conditions and are capable of performing complex and difficult missions that are not accomplished by a single robotic unit. Several major application areas of the systems are surveillance, search and rescue; foraging and flocking; formation and exploration; cooperative manipulation; team heterogeneity; adversarial environment [4], [5].

The working space of the MMRS is normally continuous, bounded and well-determined. There might be some uncertainties but they can be determined over time. The dimension and the shape are given, the positions of static obstacles are also known. The only dynamic part is the mobile robots, which move in the working space according to the path planning feature. Path planning aims for collision-free movement of the robots from initial positions to reach goal positions within expected finited time. Each robot must deal with not only static obstacles, if any, but also dynamic ones, which are other robots. A good path planning approach is also expected to achieve several optimal goals, such as time, energy etc. According to [6], the path planning approaches are classified into several groups: heuristic, artificial potential field (APF), artificial intelligence types, behavior decomposition, cased-based learning, and rolling windows. Considering the coverage path planning (CPP) of multi-robots [7], the path planning approaches are classified into grid-based search, geometric, reward based, near best view, random incremental planners. Fig. 1 shows a warehouse layout in which the path planning is based on an APF algorithm with a grid-based map.

The MMRS has a large fleet of mobile robots so its performance relies heavily on the path planning and other features, especially having hundred or more units. Following this, the path planning is influenced by other factors, such as the system awareness for the workspace, or how often the system processes the tasks etc. In the former, the system receives information on the working space based on the data collected from the robot units. Therefore, the larger the number, the more information is received, the more awareness is achieved. On the other hand, having a large number of mobile robots, those robot movements may produce some uncertainties in the working space to be identified over time. In the latter, by using more robot units, there are a greater number of tasks to be finished quickly so it likely reduces the system workload.

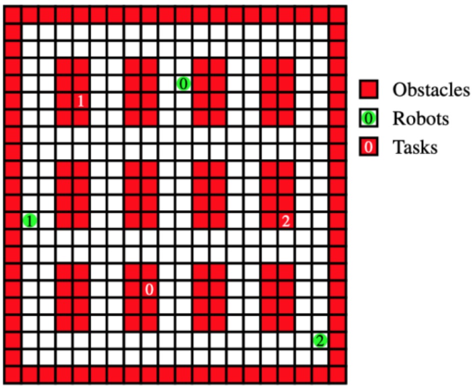

Fig. 1. Warehouse layout in [8]

However, the main drawback of a large fleet system is how the system coordinates all the units during the path planning and finding optimal paths, then aware of the situation in the workspace when each robot executes its own path. In order to solve these matters, the paper introduces a digital mapping process based on an approximate cell decomposition method, that can work along with the path planning approaches later. The map consists of the information on the static characteristics of the working space and can be implemented information on movements of all units. The rest of this article is organized into four sections as follows. In Section II, we present an overview of related works on approximate cell decomposition algorithms and clustering approaches. Subsequently, Section III describes the digital mapping process in detail and Section IV introduces the simulation results and the analysis to evaluate the performance of the mapping. Finally, Section VI contains some conclusions.

II. Related Works

Considering motions in a continuous and bounded workspace as robotics problems, there are several classical solutions to search for the available paths and cell decomposition method is one of them. Sometimes, unknown objects exist in the workspace and they might be unidentified static obstacles or other dynamic objects. In these cases, the workspace can be well-determined after some time when the robots start to move around. The principle of cell decomposition is to divide the free workspace into a finite number of simple and connected regions called "cells" so converting the space into a discrete search problem. The simple cells are a kind of basic geometric shapes such as square, rectangle, trapezoid or even rhombus. Meanwhile, connected cells are adjacent with no overlap or having gaps between them. A connectivity graph is created to represent the adjacency relation between the cells.

In the earlier research, path planning can be derived directly from cell decomposition. When splitting cells from the free workspace and defining nodes from them, two adjacent nodes are connected by a link. The outcome of the search is a sequence of cells called a channel. A continuous free path can be computed from this sequence.

Cell decomposition methods can be broken down into exact and approximate methods based on how to choose the shape of the cells [9]. Fig. 2 demonstrates the difference between exact cell decomposition and approximate cell decomposition. Firstly, exact cell decomposition methods decompose the free workspace into cells whose union is exactly the free workspace. Secondly, approximate cell decomposition methods produce cells of predefined shape (e.g. rectangles) whose union is strictly included in the free workspace. Recently, there are a few classifications for cell decomposition such as radial cell decomposition [10], adaptive cell decomposition [11], but ultimately they still are a subset of approximate cell decomposition.

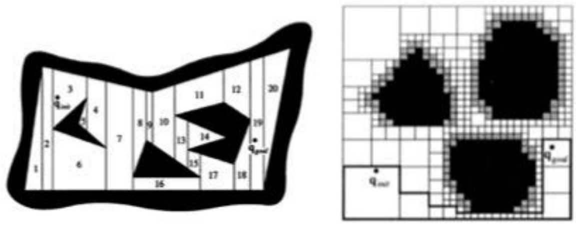

Fig. 2. Exact (left) and approximate (right) cell decomposition [9]

Meanwhile, exact cell decomposition is used, such as trapezoidal cells [22] or other types [23], approximate cell decomposition methods are widely used when focusing on path planning and area coverage. The methods divide the workspace into two-dimensional grids, a matrix of cells. Two critical factors of the division process are selecting cell size and shape as well as distinguishing free area and obstacles in the workspace. Aparting from having equal size [12], some other works propose quadtree methods as well as recursive approaches to split each cell into the four smaller ones [16, 17, 19] or combine four adjacent cells into bigger one [15]. The former applies to a cell mixed between free space and obstacle and the process repeats until there are no mixed cells left or reach minimum condition. The latter assigns to the cells having the same feature, free space, or obstacle. The advantage of those proposals is to reduce the amount of data, corresponding to the number of cells, and to speed up the computational process then. Another way to get small data volume and high computational speed is to create grids, which mix of different sizes, for example the closer to the robot, the smaller, the farther away, the larger [20, 21]. The size can change according to the complexity of the terrain, such as the large cell for simple terrain or the small cell for terrain with many complex profiles [18]. Based on the defined grids, several other path planning methods are used for finding the suitable path, such as a fuzzy algorithm modified by a potential field algorithm [12], a RRT and fuzzy rule-based method [22], an A* algorithm [15, 20], the Dijkstra algorithm [19], using squared L2 norm, L1 norm and L∞ norm [23].

Besides the cell decomposition method, the clustering approaches are solutions to separate the working space into smaller regions (clusters), such as based on terrain density

[25, 26], using k-means algorithm [27, 28, 29] and exact clustering [24], [30]. As the density-based clustering, each region is divided based on density factor, the cost per unit distance traveled by the unit is homogeneous and isotropic, varied geographic features of the terrain [25] or relative distances to the centers of regions derived from the flying features of UAVs and the geographical locations of regions [26]. When using k-means algorithm, the separation of regions is processed to gather points with the nearest mean to the cluster center, such as given positions or the random positions. The former are the goals that the robot needs to be acquired through the optimal paths in each cluster [28]. The latter are arbitrary positions in the non-explored space and the number of clusters are equalled to the number of units, such as mobile robots or UAVs [27, 29]. The purpose of these approaches are to divide the working space to several regions for area coverage or area exploration. The regions can be partitioned by Canopy K Mean using a fast approximate distance metric and two distance thresholds T1 > T2 for processing [27]. While density-based clustering and k-means clustering are approached from the features of the workspace, exact clustering is formed from the features of the robots [24], [30]. Accordingly, the grouping process is executed on the coherent characteristics of the robots to model their unknown global organizational structure and determine collision-free local neighborhoods of robot configurations.

Based on clustering approaches, path planning is proposed to execute for single unit [25], [28] or multi units [24], [30] as well as there is area coverage for multi units [26], [27], [29]. Units of a multi mobile robot system share the same working space which has some obstacles. Avoiding collisions is not only for those obstacles but also between units and becomes a mandatory requirement of the system. When the number of units increases and the task assignments are large enough, the probability of collision of the units tends to grow. Therefore, mapping and clustering techniques can help to support the path planning with collision avoidance.

III. Digital Mapping Process

Considering a working space $\boldsymbol{\mathcal{W}}$, it is continuous and bounded

$$\boldsymbol{\mathcal{W}} \subset \mathbb{R}^2 \#[1]$$

Because of the continuous feature, the mobile robot can move freely in this two-dimension workspace. The boundary of $\boldsymbol{\mathcal{W}}$ can be completely arbitrary shape, including rectangle or square. The workspace contains non-movable obstacles, called static obstacles. If the positions of all static obstacles are known, this working space $\boldsymbol{\mathcal{W}}$ can be considered well-determined.

Given n, a number of obstacles, the obstacle space can be described as below

$$\boldsymbol{\mathcal{O}} = \boldsymbol{\mathcal{O}}_1 \cup \boldsymbol{\mathcal{O}}_2 \ldots \cup \boldsymbol{\mathcal{O}}_n$$

whereas $\boldsymbol{\mathcal{O}}_1, \boldsymbol{\mathcal{O}}_2 \ldots, \boldsymbol{\mathcal{O}}_n$ are the occupied spaces of each static obstacle. $\boldsymbol{\mathcal{O}}$ can be dis-continuous due to the different positions of the obstacles and the relation between $\boldsymbol{\mathcal{O}}$ and $\boldsymbol{\mathcal{W}}$ is adopted as

$$\boldsymbol{\mathcal{O}} \subset \boldsymbol{\mathcal{W}} \#[2]$$

A space that the robots are able to move freely is called the freespace $\boldsymbol{\mathcal{F}}$

$$\boldsymbol{\mathcal{F}} \subset \boldsymbol{\mathcal{W}} \#[3]$$

In fact, the robots are not small enough so that they can not reach out to some areas, which do not belong to $\boldsymbol{\mathcal{O}}$, in $\boldsymbol{\mathcal{W}}$. Then, we have the following relation

$$(\boldsymbol{\mathcal{F}} \cup \boldsymbol{\mathcal{O}}) \subset \boldsymbol{\mathcal{W}} \#[4]$$

The digital mapping is to convert the relationship mentioned in [4] to a map. There are also other basic requirements for the process, following as below:

- Being able to share information between the robots, generating a united awareness for the working space;

- The number of robots is large, they perform the tasks simultaneously. This is dynamic characteristic of the working space, forming from the movements of the robots;

- Being able to avoid collision, one robot only occupies one specified space at a time;

- The robots have similar structure and size.

A. Bounded conditions

Using the approximate cell decomposition methods partitions the workspace $\boldsymbol{\mathcal{W}}$ to cells of the same size. One cell should have enough space so that one robot can stay inside it completely. Given the dimensions of one robot, D and L, then the size of one cell is determined by the following expression

$$d = max(D, L) + \delta \#[5]$$

whereas δ is allowable tolerance to warrant no collision when two robots are in two adjacent cells. δ is determined from experience with the following condition

$$\delta > 0{,}05 \times max(D, L) \#[6]$$

The bounding condition of the cell decomposition process is the notation below

$$\begin{matrix} N_W \times d \leq L_W \\ M_W \times d \leq D_W \end{matrix} \#[7]$$

where: N_W and M_W are the positive integer values, L_W and D_W are the length and the width of the workspace. $\boldsymbol{\mathcal{W}}$ is equivalently converted into a matrix $\mathbb{W}$ which consists of N_W rows and M_W columns corresponding to the number of cells divided by rows and column respectively.

TABLE I. SET OF CELL IDS

Type	*Size of* $\mathbb{W}$	*Set of cell IDs*
8-bit	lower than 256 elements	$\{0x00..0xFF\}$
10-bit	lower than 1024 elements	$\{0x0000..0x03FF\}$
16-bit	lower than 65536 elements	$\{0x0000..0xFFFF\}$

B. Mapping with IDs

It is important to achieve the united awareness for $\boldsymbol{\mathcal{W}}$, each element of $\mathbb{W}$ has an unique value corresponding to one specific cell in the workspace, a cell ID. If so, the robots can aware its own position and know about the locations of the other robots. Futhermore, the information of the workspace environment can be spread out all robots based on the set of cell IDs, or $\mathbb{W}$. The notation of $\mathbb{W}$ is to illustrate below

$$\mathbb{W} = \begin{bmatrix} w_{1,1} & \cdots & w_{M_w,1} \\ \vdots & \ddots & \vdots \\ w_{1,N_w} & \cdots & w_{M_w,N_w} \end{bmatrix} \#[8]$$

where $\{w_{i,j}\}$ is the set of unique values. The number of the unique values is decided based on the size of $\mathbb{W}$. We can describe this connection as Table I.

Starting from the expression [4], an another matrix ($\mathbb{F}$) is proposed to distinguish and represent the state of the cells. $\mathbb{F}$ is the corresponding matrix of $\mathbb{W}$ so it has the same size.

$$\mathbb{F} = \begin{bmatrix} f_{1,1} & \cdots & f_{M_w,1} \\ \vdots & \ddots & \vdots \\ f_{1,N_w} & \cdots & f_{M_w,N_w} \end{bmatrix} \#[9]$$

The element value is determined from cell states, called state IDs, such as:

$f_{i,j} = 0xFF$ correlate with the cell $[i,j]$ belonging to obstacle space or consisting of part of obstacle space and part of free space;

$f_{i,j} = 0x00$ correlate with the cell $[i,j]$ belonging to free space only.

In order to apply the avoidance collision approach, the state of robots has to be identified when they moved in the free space. We add two more state IDs in the $\mathbb{F}$: $f_{i,j} = 0x55$ corresponds to the cell $[i,j]$ where a robot has been occupied, then the surrounding cells get the $0xAA$ value. This association can be illustrated as Fig. 3 below.

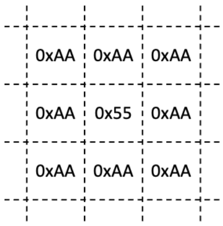

Fig. 3. The association between two state IDs 0x55 and 0xAA

C. Algorithm of digital mapping process

$\mathbb{W}$, with the cell IDs, and $\mathbb{F}$, with the state IDs, are the expected results of the digital mapping process of the workspace $\boldsymbol{\mathcal{W}}$, including the information on both of $\boldsymbol{\mathcal{O}}$ and $\boldsymbol{\mathcal{F}}$. The proposed algorithm of the digital mapping is described as below.

%==============================

% Given parameters

maximum Depth of workspace *Dw*

maximum Length of workspace *Lw*

Depth of mobile robot *D*

Length of mobile robot *L*

Tolerance *delta*

% Initialization of bounded conditions

d = $max(D, L)$ + *delta*

Nw = $uint(Lw/d)$

Mw = $uint(Dw/d)$

% Mapping with IDs

Map (W) with Size (Nw, Mw)

For i=1:Nw

For j=1:Mw

Cell resolution W[i,j] = Select (Nw, Mw)

Map (F) with Size (Nw, Mw)

For i=1:Nw

For j=1:Mw

If Cell[i,j] belong Free Then F[i,j] = 0x00

Else F[i,j] = 0xFF

%==============================

IV. Simulation And Analysis

We propose an experiment with a given workspace and the size of the robot is about 600 x 500 mm in length and width. Two scenario of obstacles placement are following:

- Simple configuration, several convex polygons with different sizes, illustrated as in Fig. 4.

- Complex configuration, mix between one convex polygon and two concave polygons, illustrated as in Fig. 5. Concave polygons can create more junctions between free space and obstacle space.

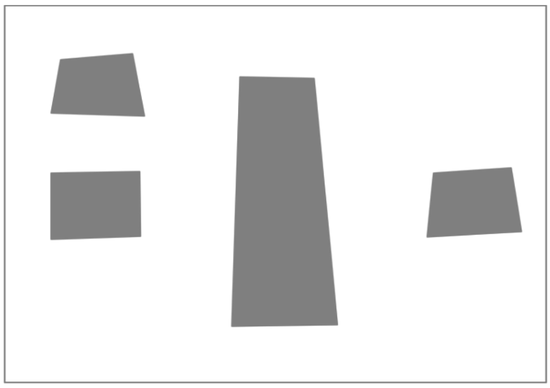

Fig. 4. Simple configuration with convex polygons

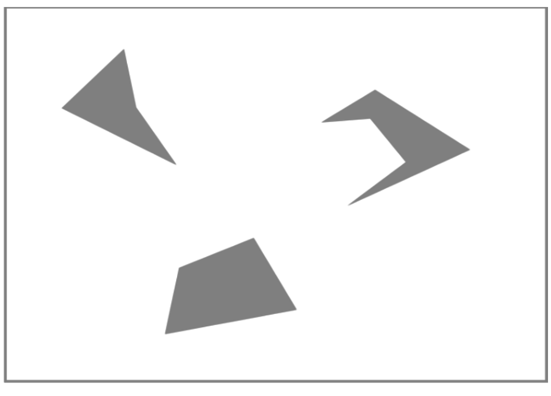

Fig. 5. Complex configuration with both convex polygon and concave polygon

The results of digital mapping are illustrated as Fig. 7, Fig. 8, and Fig. 9 in which,

- The matrix 𝕎 with the 8-bit cell IDs data set in Fig. 7;

- The matrix 𝔽 with the state IDs data set correlated with the simple configuration in Fig. 8;

- The matrix 𝔽 with the state IDs data set correlated with the complex configuration in Fig. 9.

The data set of cell IDs is no change since we keep the same workspace in both configurations. Meanwhile, the data sets of state IDs are changed for each configuration. These data sets are matched perfectly with the proposed configurations. The profile of concave polygons seems to narrow down the available free space that the robot can reach out. It can be a limitation of the digital mapping process using approximate cell decomposition methods.

0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
0x10	0x11	0x12	0x13	0x14	0x15	0x16	0x17	0x18	0x19	0x1A	0x1B	0x1C	0x1D	0x1E	0x1F
0x20	0x21	0x22	0x23	0x24	0x25	0x26	0x27	0x28	0x29	0x2A	0x2B	0x2C	0x2D	0x2E	0x2F
0x30	0x31	0x32	0x33	0x34	0x35	0x36	0x37	0x38	0x39	0x3A	0x3B	0x3C	0x3D	0x3E	0x3F
0x40	0x41	0x42	0x43	0x44	0x45	0x46	0x47	0x48	0x49	0x4A	0x4B	0x4C	0x4D	0x4E	0x4F
0x50	0x51	0x52	0x53	0x54	0x55	0x56	0x57	0x58	0x59	0x5A	0x5B	0x5C	0x5D	0x5E	0x5F
0x60	0x61	0x62	0x63	0x64	0x65	0x66	0x67	0x68	0x69	0x6A	0x6B	0x6C	0x6D	0x6E	0x6F
0x70	0x71	0x72	0x73	0x74	0x75	0x76	0x77	0x78	0x79	0x7A	0x7B	0x7C	0x7D	0x7E	0x7F
0x80	0x81	0x82	0x83	0x84	0x85	0x86	0x87	0x88	0x89	0x8A	0x8B	0x8C	0x8D	0x8E	0x8F
0x90	0x91	0x92	0x93	0x94	0x95	0x96	0x97	0x98	0x99	0x9A	0x9B	0x9C	0x9D	0x9E	0x9F
0xA0	0xA1	0xA2	0xA3	0xA4	0xA5	0xA6	0xA7	0xA8	0xA9	0xAA	0xAB	0xAC	0xAD	0xAE	0xAF
0xB0	0xB1	0xB2	0xB3	0xB4	0xB5	0xB6	0xB7	0xB8	0xB9	0xBA	0xBB	0xBC	0xBD	0xBE	0xBF

Fig. 6. 8-bit cell IDs data set

0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0xFF	0xFF	0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0xFF	0xFF	0xFF	0x00	0x00	0xFF	0xFF	0xFF	0xFF	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0xFF	0xFF	0xFF	0x00	0x00	0xFF	0xFF	0xFF	0xFF	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00	0x00	0x00	0xFF	0xFF	0xFF	0xFF	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0xFF	0xFF	0xFF	0x00	0x00	0xFF	0xFF	0xFF	0xFF	0x00	0x00	0xFF	0xFF	0xFF	0xFF
0x00	0xFF	0xFF	0xFF	0x00	0x00	0xFF	0xFF	0xFF	0xFF	0x00	0x00	0xFF	0xFF	0xFF	0xFF
0x00	0xFF	0xFF	0xFF	0x00	0x00	0xFF	0xFF	0xFF	0xFF	0x00	0x00	0xFF	0xFF	0xFF	0xFF
0x00	0x00	0x00	0x00	0x00	0x00	0xFF	0xFF	0xFF	0xFF	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00	0x00	0x00	0xFF	0xFF	0xFF	0xFF	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00	0x00	0x00	0xFF	0xFF	0xFF	0xFF	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00

Fig. 7. State IDs data set of simple configuration

0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0xFF	0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0xFF	0xFF	0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0xFF	0xFF	0xFF	0x00	0x00	0x00
0x00	0xFF	0xFF	0xFF	0xFF	0x00	0x00	0x00	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0x00	0x00
0x00	0x00	0x00	0xFF	0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0xFF	0xFF	0xFF	0x00	0x00
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0xFF	0xFF	0xFF	0x00	0x00	0x00
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0xFF	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00	0x00	0x00	0xFF	0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00	0xFF	0xFF	0xFF	0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00

Fig. 8. State IDs data set of complex configuration

V. CONCLUSIONS

Mapping the working space of the multi mobile robot system relies on clear separation between the static characteristics of this workspace and the dynamic one that changes over time. Movement of robots can be considered as one of the main dynamic factors. The map can provide basic information for all units in the system to share awareness of the surrounding environment with each other, then forming a united awareness. It is so important to the system with a very large number of units and has a considered impact on the application of finding optimal paths, path planning, and task allocation at the system level. The limitation of the digital mapping is the requirement of complete knowledge about the workspace including the location of all static obstacles. Besides, the volume of generated data and calculation amount might be large. However, they are not the big problems for high performance MCUs nowaday. In the future, further research will be conducted to implement the digital map with the path planning approach to evaluate the effectiveness of the combination.

ACKNOWLEDGMENT

This research was supported by the Program "Supporting Research, Development and Technology Application of Industry 4.0" (KC-4.0/19-25) of the Ministry of Science and Technology in Vietnam through the funding of the Project "Research and Development for a Multi Mobile Robots System that Integrates some Key Technologies of the 4th Industrial Revolution (Industry 4.0), Application in logistics", No. KC-4.0-34/19-25.

REFERENCES

[1] T. Arai, E. Pagello, and L. E. Parker, "Guest editorial advances in multirobot systems," IEEE Trans. Robot. Autom., vol. 18, no. 5, pp. 655-661, 2002, doi: 10.1109/TRA.2002.806024.

[2] A. Gautam and S. Mohan, "A review of research in multi-robot systems," in 2012 IEEE 7th International Conference on Industrial and Information Systems (ICIIS), 2012, pp. 1-5, doi: 10.1109/ICIInfS.2012.6304778.

[3] Y. Cai and S. X. Yang, "A Survey on multi-robot systems," in World Automation Congress 2012, 2012, pp. 1-6.

[4] R. N. Darmanin and M. K. Bugeja, "A review on multi-robot systems categorised by application domain," in 2017 25th Mediterranean Conference on Control and Automation (MED), 2017, pp. 701-706, doi: 10.1109/MED.2017.7984200.

[5] Y. Rizk, M. Awad, and E. W. Tunstel, "Cooperative Heterogeneous Multi-Robot Systems: A Survey," ACM Comput. Surv., vol. 52, no. 2, Apr. 2019, doi: 10.1145/3303848.

[6] H. Zhang, W. Lin, and A. Chen, "Path Planning for the Mobile Robot: A Review," Symmetry (Basel)., vol. 10, no. 10, p. 450, Oct. 2018, doi: 10.3390/sym10100450.

[7] R. Almadhoun, T. Taha, L. Seneviratne, and Y. Zweiri, "A survey on multi-robot coverage path planning for model reconstruction and mapping," SN Appl. Sci., vol. 1, no. 8, p. 847, 2019, doi: 10.1007/s42452-019-0872-y.

[8] G. Sun, R. Zhou, B. Di, Z. Dong, and Y. Wang, "A Novel Cooperative Path Planning for Multi-robot Persistent Coverage with Obstacles and Coverage Period Constraints," Sensors, vol. 19, no. 9, p. 1994, Apr. 2019, doi: 10.3390/s19091994.

[9] J. C. Latombe, "Robot motion planning," Kluwer Academic Publishers, Springer, 1991.

[10] O. A. A. Salama, M. E. H. Eltaib, H. A. Mohamed, and O. Salah, "RCD: Radial Cell Decomposition Algorithm for Mobile Robot Path Planning," IEEE Access, vol. 9, pp. 149982–149992, 2021, doi: 10.1109/ACCESS.2021.3125105.

[11] S. K. Debnath et al., "Different Cell Decomposition Path Planning Methods for Unmanned Air Vehicles-A Review," in Proceedings of the 11th National Technical Seminar on Unmanned System Technology 2019, 2021, pp. 99-111.

[12] I. Iswanto, O. Wahyunggoro and A. I. Cahyadi, "Quadrotor Path Planning Based On Modified Fuzzy Cell Decomposition Algorithm," TELKOMNIKA, vol. 14, no. 2, pp. 655–664, 2016, doi: 10.12928/TELKOMNIKA.v14i1.2989.

[13] B. Dugarjav, S.-G. Lee, D. Kim, J. H. Kim, and N. Y. Chong, "Scan matching online cell decomposition for coverage path planning in an unknown environment," Int. J. Precis. Eng. Manuf., vol. 14, no. 9, pp. 1551–1558, 2013, doi: 10.1007/s12541-013-0209-5.

[14] R. A. Brooks and T. Lozano-Pérez, "A subdivision algorithm in configuration space for findpath with rotation," IEEE Trans. Syst. Man. Cybern., vol. SMC-15, no. 2, pp. 224-233, 1985, doi: 10.1109/TSMC.1985.6313352.

[15] T. Arney, "An efficient solution to autonomous path planning by Approximate Cell Decomposition," in 2007 Third International Conference on Information and Automation for Sustainability, 2007, pp. 88–93, doi: 10.1109/ICIAFS.2007.4544785.

[16] F. Lingelbach, "Path planning using probabilistic cell decomposition," in IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004, 2004, vol. 1, pp. 467-472 Vol.1, doi: 10.1109/ROBOT.2004.1307193.

[17] X. Huang, M. Sun, H. Zhou, and S. Liu, "A Multi-Robot Coverage Path Planning Algorithm for the Environment With Multiple Land Cover Types," IEEE Access, vol. 8, pp. 198101-198117, 2020, doi: 10.1109/ACCESS.2020.3027422.

[18] C. Cai and S. Ferrari, "Information-Driven Sensor Path Planning by Approximate Cell Decomposition," IEEE Trans. Syst. Man, Cybern. Part B, vol. 39, no. 3, pp. 672-689, 2009, doi: 10.1109/TSMCB.2008.2008561.

[19] A. Abbadi and V. Přenosil "Safe Path Planning Using Cell Decomposition Approximation," In Mirosav Hrubý, International Conference Distance Learning, Simulation And Communication, první. Brno: University of Defence, Brno, p. 8-14, 2015, ISBN 978-80-7231-992-3.

[20] S. Behnke, "Local Multiresolution Path Planning," in RoboCup 2003: Robot Soccer World Cup VII, 2004, pp. 332-343.

[21] R. V Cowlagi and P. Tsiotras, "Beyond quadtrees: Cell decompositions for path planning using wavelet transforms," in 2007 46th IEEE Conference on Decision and Control, 2007, pp. 1392-1397, doi: 10.1109/CDC.2007.4434146.

[22] A. Abbadi and R. Matousek, "Hybrid rule-based motion planner for mobile robot in cluttered workspace," Soft Comput., vol. 22, no. 6, pp. 1815-1831, 2018, doi: 10.1007/s00500-016-2103-4.

[23] M. Kloetzer, C. Mahulea, and R. Gonzalez, "Optimizing cell decomposition path planning for mobile robots using different metrics," in 2015 19th International Conference on System Theory, Control and Computing (ICSTCC), 2015, pp. 565-570, doi: 10.1109/ICSTCC.2015.7321353.

[24] Omur Arslan, Dan P. Guralnik, and Daniel E. Koditschek, "Clustering-Based Robot Navigation and Control," 2016 IEEE International Conference on Robotics and Automation, Workshop on Emerging Topological Techniques in Robotics, 2016.

[25] P. Bhattacharya and M. L. Gavrilova, "Density-Based Clustering Based on Topological Properties of the Data Set BT - Generalized Voronoi Diagram: A Geometry-Based Approach to Computational Intelligence," M. L. Gavrilova, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 197-214.

[26] J. Chen, C. Du, Y. Zhang, P. Han, and W. Wei, "A Clustering-Based Coverage Path Planning Method for Autonomous Heterogeneous UAVs," IEEE Trans. Intell. Transp. Syst., vol. 23, no. 12, pp. 25546-25556, 2022, doi: 10.1109/TITS.2021.3066240.

[27] S. Sharma, R. Tiwari, A. shukla, and J. yadav, "Canopy Clustering Based Multi Robot Area Exploration," IFAC Proc. Vol., vol. 47, no. 1, pp. 505–510, 2014, doi: https://doi.org/10.3182/20140313-3-IN-3024.00253.

[28] W. C. Wang and R. Chen, "An Effective Way to Large-Scale Robot-Path-Planning Using a Hybrid Approach of Pre-Clustering and Greedy Heuristic," Appl. Artif. Intell., vol. 34, no. 14, pp. 1159–1175, Dec. 2020, doi: 10.1080/08839514.2020.1824094.

[29] H. Zhang, L. Bai, J. Qi, and Y. Xiao, "Area Coverage of Swarm Robotics Based on Anti-Flocking Framework with Dynamical Clustering," in 2022 IEEE International Conference on Unmanned Systems (ICUS), 2022, pp. 209-213, doi: 10.1109/ICUS55513.2022.9986539.

[30] O. Arslan, D. P. Guralnik, and D. E. Koditschek, "Coordinated Robot Navigation via Hierarchical Clustering," IEEE Trans. Robot., vol. 32, no. 2, pp. 352-371, 2016, doi: 10.1109/TRO.2016.2524018.

The 8th International Conference on
Engineering Mechanics and Automation (ICEMA 2025)
Hanoi, November 14-15, 2025

DOI: 10.15625/vap.2026.0003

Evaluation of Spatial Stability and Adaptability in a Dynamic Clustering Method for Multi-Robot Systems

Hung X. Truong
Vietnam National Space Center, Vietnam Academy of Science and Technology,
Hanoi, Vietnam
txhung@vnsc.org.vn

Thang M. Pham
University of Engineering and Technology, Vietnam National University,
Hanoi, Vietnam
thangpm@vnu.edu.vn

Nha Q. Nguyen
University of Engineering and Technology, Vietnam National University,
Hanoi, Vietnam
nhanq@vnu.edu.vn

Hanh T. H. Nguyen
Institute of Mechanics, Vietnam Academy of Science and Technology,
Hanoi, Vietnam
nguyenhonghanh.imech@gmail.com

Viet A. Dang
University of Engineering and Technology, Vietnam National University,
Hanoi, Vietnam
vietda@vnu.edu.vn

Abstract **- A two-stage dynamic clustering method is proposed for partitioning multi-robot system into smaller, manageable groups to enhance coordination and workload balance. Based on encouraging initial results, this novel framework had clearly proved a robust and adaptive method. The approach leverages a Gaussian Mixture Model, optimized via the Expectation-Maximization algorithm and newly proposed objective function, to cluster robots based on their path intersection points, with randomized start and destination points. This paper presents a deeper analytical evaluation of the method, focusing the spatial stability and adaptability of the resultant clusters, specifically their means. The core of the analysis is a rigorous three-tiered simulation performed in a 25x25 grid environment with up to 50 robots, using path intersection points as the sole input for the clustering model. First, an uncertainty analysis confirmed the model's high consistency and the stability of its outputs when presented with a constant input. Second, a sensitivity analysis revealed that cluster characteristics, particularly dispersion measured by IQR and skewness, were highly sensitive to input variations. This finding indicates that the identified clusters are not fixed geographical zones but dynamic "hotspots" fundamentally dependent on the specific traffic patterns. The final generalizability test provided compelling evidence of the framework's adaptability. The model dynamically adjusted its clustering behavior to reflect environmental constraints, with cluster distributions transforming from a moderate right-skew in the open space to a more pronounced right-skew in an obstructed environment. Ultimately, results demonstrate that the DCM is a robust, reliable, and highly adaptive solution for managing multi-robot systems.**

Keywords: Dynamic Cluster, Gaussian Mixture Model, Expectation-Maximization Algorithm, Path Planning, Workload Balancing, Spatial Stability.

I. INTRODUCTION

The field of multi-robot systems, originating in the late 1980s, addresses the coordination of multiple mobile robots, including UAVs [1] [2]. Foundational research focuses on several key areas: navigation, which encompasses localization, mapping, and path planning for static and dynamic environments; task allocation, which can be managed through centralized or decentralized architectures; and communication networks. A wide array of algorithms has been developed for these challenges, ranging from classical methods to bio-inspired and artificial intelligence techniques. These approaches aim to optimize system performance by solving complex problems like generating collision-free trajectories and efficiently assigning tasks to individual robots.

As the number of robots in a system increases, the interplay between task allocation and path planning becomes a significant challenge. A larger fleet can improve workspace awareness but also introduces greater dynamic uncertainty and computational workload, as each robot becomes a moving obstacle for others [3] [4] [5]. The core problem lies in ensuring that the actions of numerous individual agents contribute to a cohesive and efficient overall objective. This complexity highlights the necessity of clustering, grouping robots based on specific criteria, as a critical strategy to manage the system's workload and streamline operations, making it a vital research area for scalable multi-robot applications.

The two-stage Dynamic Clustering Method (DCM) was proposed to manage large-scale robotic systems. It utilizes the Gaussian Mixture Model (GMM) to partition the entire robot fleet into several smaller, more efficient subgroups [6]. The DCM framework integrates with path planning and task allocation processes, as illustrated in Fig. 1, to reduce the overall system workload. This reduction is achieved by minimizing communication overhead and enabling streamlined local coordination, as each robot primarily needs to communicate and coordinate actions only with other members within its own designated cluster. Performance analysis has established DCM as a robust, stable, and adaptive framework. It maintained exceptional reliability for a given intersection point dataset while also demonstrating the ability to reconfigure clusters effectively when presented with new scenarios, proving its effectiveness as an environment-independent solution. This paper extends the previous analysis, moving beyond these performance metrics to evaluate the spatial stability of the clusters.

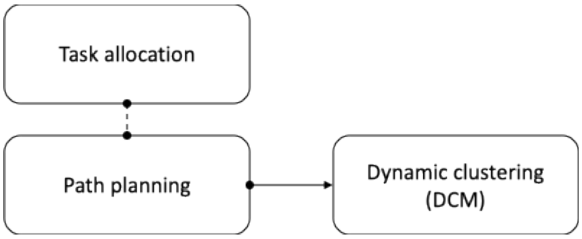

Fig. 1. *Operation flow for DCM with path planning and task allocation*

The rest of the paper is organized as follows: Section II reviews the relevant literature on clustering approaches and GMM; Section III describes the system approach of DCM; Section IV presents the simulation results along with a corresponding performance analysis related to spatial stability; and finally, Section V provides the concluding remarks and proposed future development.

II. RELATED WORKS

In multi-robot systems, hard clustering encompasses methods that partition a workspace, tasks, or objectives into distinct, non-overlapping groups. The foundational principle is to organize a large set of entities, such as robots, physical locations, or tasks, into several smaller groups by assigning each entity to exactly one cluster. This definitive assignment creates clear boundaries, simplifying complex coordination problems by breaking them down into more manageable, independent sub-problems.

One major category of hard clustering involves the physical partitioning of the environment. For instance, cell decomposition is a geometric technique used for path planning that divides the workspace into smaller cells, the Voronoi Diagram partitions space based on proximity to a set of robots or points. Approximate cell decomposition methods represent a robot's workspace by dividing it into a two-dimensional grid. While basic approaches use a grid of uniform cells [7] [8], advanced techniques employ variable-resolution grids to improve computational efficiency, such as the recursive approach [9] [10], the Quadtree method [11] [12]; using smaller cells near the robot and larger ones farther away [13], or adapting cell size to the complexity of the terrain [14]. The Voronoi diagram, a computational geometry technique, partitions a space into distinct regions based on proximity to a set of points. In multi-robot systems, this is highly effective for converting a complex problem, such as area coverage, into a set of simpler, single-robot tasks. There are variations of this classic Voronoi diagram, such as Generalized Voronoi Diagram[15], Manhattan Voronoi [16], and some other Voronoi partition-based [17] [18] [19].

Meanwhile, the K-Means algorithm also stands out as a particularly versatile hard clustering technique in robotics, applied primarily in two ways. First, it can simplify task allocation and path planning by grouping a large number of individual goals into a smaller set of clusters, to which robots are then assigned [20] [21] [22]. Path planning is to find the optimal route within each cluster and then determines an efficient path connecting the clusters. This approach simplifies a complex multi-task problem into a more manageable one, improving the overall efficiency of multi-robot systems. Second, it is used to systematically divide an area into distinct regions for exploration and coverage. The dynamic capability is a feature in this application; as robots discover new information, the algorithm can re-cluster the remaining space to rebalance the workload and ensure efficient, comprehensive coverage [23] [24] [25].

In contrast to the definitive assignments of hard clustering, soft clustering allows an entity to belong to multiple clusters simultaneously, but with varying degrees of membership or probability. Instead of creating clear-cut boundaries, this method models uncertainty and overlap. The most prominent method is the Gaussian Mixture Model (GMM), which is highly useful in robotics for representing uncertain target locations, learning complex motion trajectories, or modeling continuous environmental data like pedestrian density. GMM are a probabilistic clustering method that models data as a finite mixture of Gaussian distributions, with its parameters typically learned via the Expectation-Maximization (EM) algorithm [26] [27] [28]. This iterative process alternates between an expectation (E) step and a maximization (M) step to find maximum likelihood estimates for the model parameters until a convergence criterion is met. This approach enables a system to handle ambiguous and overlapping scenarios, better reflecting real-world complexity.

GMMs have been extensively applied across numerous domains in robotics. They feature prominently in model continuous movements/discrete segmentation points [29] [30] [31] or Learning from Demonstration for robot [32] [33] [34]. GMMs can also map the complex relationships between object observations and robot joint variables for tasks like grasping [35] and serve as abnormality detection in routine robotic arm movements [36]. Beyond manipulation, GMMs are instrumental in autonomous navigation, including path planning, such as the fast marching square with Gaussian mixture models to represent people [37], using the Risk-RRT planner with GMM [38], strategies for information extraction and goal selection that significant reduce data transmission [39]. Furthermore, GMMs help determine the next best target by modeling probability distributions, often integrated with frameworks such as Infotaxis and particle filters [40] [41].

III. DYNAMIC CLUSTERING METHOD

In large-scale multi-robot systems, a significant operational hurdle is the effective coordination of all units.

The inherent spatial and temporal correlations among units with intersecting trajectories provide a logical basis for partitioning the fleet. The coordination can be localized and intensified by grouping these robots into clusters. The exchange of critical data, such as location, state, and trajectory information, within each cluster enhances situational awareness; thereby minimizing conflicts and optimizing task execution. A critical advantage of this architecture is communication efficiency; by restricting communication to intra-cluster members, overall network bandwidth is substantially conserved. Consequently, this method fosters greater decentralization and system scalability by enabling parallel operations across different cluster data streams.

The operational environment for a system of *N* robots is a well-defined workspace, denoted by $\mathcal{W}$. This working space exists in either two or three dimensions and is represented as the following expression:

$$\mathcal{W} \subseteq \mathbb{R}^n \tag{1}$$

where: n = 2 or 3. We define an obstacle space $\mathcal{O}$, the region of $\mathcal{W}$ containing static obstacles, which are objects whose positions remain fixed over time. Assuming that $\mathcal{W}$ has m obstacles, $\mathcal{O}$ can be determined by the following formula:

$$\mathcal{O} = \bigcup_{i=1}^{m} \mathcal{O}_i \subset \mathcal{W} \tag{2}$$

where: $\mathcal{O}_i$ is the space occupied by each individual obstacle.

Considering a system of *N* robots, denoted $A_1, A_2, \ldots, A_N$, navigating from their starting points $(q_{10}, q_{20}, \ldots, q_{N0})$ to their goal points $(q_{1G}, q_{2G}, \ldots, q_{NG})$, all start and goal configurations are located within the obstacle-free workspace. The paths for each robot A_i, denoted $Path_i(t)$ generated by a planning function f_{pp} based on its assigned start and goal:

$$Path_i(t) = f_{pp}(q_{i0} \quad q_{iG}) \tag{3}$$

If the path of any two robots, A_i and A_j, cross each other, we define the intersection points as follows:

$$\left(Co_{i,j}\right) = Path_i(t) \cap Path_j(t) \neq \emptyset \tag{4}$$

The complete dataset of all such intersection points for the entire fleet is the union of all unique pairwise intersections:

$$\mathcal{C} = \bigcup_{i,j=0}^{N} \left(Co_{i,j}\right) \tag{5}$$

and the relationships shown are

$$A_i \Leftrightarrow Path_i(t) \Leftrightarrow Co_{i,j} \tag{6}$$

As indicated in (6), the existence of an intersection point establishes a strong spatial correlation between the corresponding robots as they are more likely to require coordination. For this reason, the intersection point dataset, as illustrated in (5), is chosen as the foundational input for our robot clustering method.

GMM is represented by the overall probability density function which is a weighted sum of every components, namely:

$$p(x) = \sum_{i=1}^{K} w_i N(x|\mu_i, \Sigma_i) \tag{7}$$

where: w_i is mixing weight of i^{th} component with $w_i > 0$,

$$\sum_{i=1}^{K} w_i = 1 \tag{8}$$

and $N(x|\mu_i, \Sigma_i)$ is a Gaussian density function which also called a component is defined as follows:

$$N\left(x|\mu_i, \Sigma_i\right) = \frac{1}{\left(2\pi\right)^{\frac{n}{2}} \left(\left|\Sigma_i\right|\right)^{\frac{1}{2}}} exp\left(-\frac{1}{2}\left(x-\mu_i\right)^T \Sigma_i^{-1}\left(x-\mu_i\right)\right) \tag{9}$$

where $x \in \mathbb{R}^n$ represents a data vector, $\mu_i \in \mathbb{R}^n$ is the mean and Σ_i is the $n \times n$ covariance matrix of the i^{th} Gaussian density. We have three critical unknown parameters of GMM in the (7) and (9).The EM algorithm can be utilized to estimate the mean μ_i, the covariance matrix Σ_i, and the weight w_i. The computation equation of the expectation step is as follows:

$$f_E\left(C_i|x_j\right) = \frac{w_i N\left(x_j|\mu_i, \Sigma_i\right)}{\sum_{k=1}^{K} w_i N\left(x_j|\mu_k, \Sigma_k\right)} \tag{10}$$

and he equations for maximization step are as follows

$$w_i = \frac{\sum_{j=1}^{s} f_E\left(C_i|x_j\right)}{s} \tag{11}$$

$$\mu_i = \frac{\sum_{j=1}^{s} f_E\left(C_i|x_j\right) x_j}{s w_i} \tag{12}$$

$$\Sigma_i = \frac{\sum_{j=1}^{s} f_E\left(C_i|x_j\right)\left(x_j-\mu_i\right)\left(x_j-\mu_i\right)^T}{s} \tag{13}$$

where: *s* is number of samples.

It is crucial to balance the number of robots within each cluster to promote effective local coordination and system scalability. This approach enables a balanced workload distribution across clusters. Consequently, a system with such workload-balanced groups becomes inherently more scalable, allowing it to maintain performance and adapt flexibly to changing workloads. First, a cost parameter, δ_k, is defined for each cluster, which is the reciprocal of the number of robots, N_K, in that cluster, as following expression:

$$\delta_k = \frac{1}{N_K} \tag{14}$$

with: $k = 1\ldots K$, K is the number of clusters and N_K is the number of robots allocated in the cluster k^{th}. Second, we minimize a least squares objective function formulated to find the most balanced configuration, specifically:

$$f_{obj} = \sqrt{\sum_{k=1}^{K}\left(\delta_k\right)^2} \rightarrow min \tag{15}$$

The proposed objective function inherently drives all cost parameters to be as small and as uniform as possible. Since $\delta_k$is inversely proportional to the cluster size, this process directly results in a near-optimal distribution where each cluster contains an approximately equal number of robots, ensuring a balanced workload across the system.

IV. RESULTS AND DISCUSSION

To evaluate the performance of DCM, particularly the spatial stability of the clusters, an experimental framework was established within a simulated 25×25 discrete grid environment. A fleet of 50 robots was tasked with navigating between random start and goal positions, with the classical A* algorithm employed to compute optimal paths. The number of robots and map size were limited due to computational constraints. The dataset of path intersection points generated from these trajectories served as the sole input for the clustering model.

The clustering method itself, which uses a Gaussian Mixture Model and the Expectation-Maximization algorithm, is environment-agnostic. It only operates on the intrinsic properties of the intersection point dataset (e.g., coordinates, density, distances) and is not concerned with how the points were generated. Key hyperparameters were empirically determined: the number of clusters was set to K = 3 based on an intuitive workspace partitioning, and the EM algorithm was configured for a maximum of 100 iterations, terminating early if the change in log-likelihood fell below a convergence threshold of 1×10^{-6}.

The performance evaluation of DCM was rigorously tested through three distinct simulation scenarios, mentioned as below:

i. In the initial stage, an uncertainty analysis was performed to quantify the model's consistency. DCM was executed 100 times on a single dataset of intersection points. Descriptive statistical metrics, including the mean, standard deviation, and interquartile range, were then applied to the key spatial outputs: the cluster mean vectors. This process allowed for a precise measurement of the model's reliability and the stability of its clustering results when presented with a constant input.

ii. Secondly, a sensitivity analysis was conducted to assess how the model performs with different inputs. Two other distinct datasets of intersection points were generated using different sets of random start/goal pairs. DCM was run 100 times on each of these two datasets. This measured how sensitive the resulting cluster characteristics were to input data variations.

iii. Finally, a generalizability test was performed under two varied conditions to demonstrate the framework's adaptability and environment-independent nature. They are: condition A, featuring a reduced robot population, a smaller fleet of 25 robots, in the 25x25 map without obstacles; condition B, which introduced environmental complexity with randomly placed obstacles. Analyzing the model's performance in these contrasting scenarios provides robust evidence of its adaptability.

Table 1 presented a statistical summary of the coordinate data for three distinct cluster centroids from the first simulation scenario. These key statistics, the mean, standard deviation, median, and interquartile range (IQR), revealed the underlying distribution of the X and Y coordinates for each cluster. By quantifying the data's central tendency and dispersion, these values provided a quantitative assessment of each cluster's spatial arrangement and consistency.

A comparison of the standard deviation and the Interquartile Range (IQR) offered valuable insights into the data's dispersion. The standard deviation, which measured the average deviation from the mean, was sensitive to outliers, while the IQR, representing the spread of the central 50% of the data, was more robust to extreme values. The results also indicated that the standard deviation values across all clusters were quite similar, ranging narrowly from 3.8358 to 4.4685.

Table 1. *Statistical indicators of the cluster centroids, the first simulation scenario*

		Mean	Standard Deviation	Median	IQR
Dataset #1					
Cluster #1	X	11.2104	4.1715	12.1503	7.8841
	Y	13.7347	4.3226	13.4684	8.0918
Cluster #2	X	10.8981	4.0555	11.8624	7.8761
	Y	13.7915	4.4685	13.2703	9.4265
Cluster #3	X	11.5118	3.8358	12.2046	7.4063
	Y	14.0077	4.1988	13.4684	7.9833

Specificially, the coordinates of Cluster #2 showed the greatest variability, evidenced by its Y-coordinates having the largest standard deviation of 4.4685 and the widest IQR of 9.4265 for Y-value. Similarly, the X-coordinates of Cluster #2 had the standard deviation of 4.0555 and the IQR of 7.8761. In contrast, the coordinates of Cluster #3 were the most concentrated, with its X-coordinates displaying the lowest standard deviation of 3.8358 and the smallest IQR of 7.4063. The Y-coordinates of Cluster #3 showed a slightly larger spread, with the standard deviation of 4.1988 and the IQR of 7.9833. A consistent pattern across all coordinate sets indicated that the standard deviation was smaller than the IQR, which pointed to a distribution that was not perfectly symmetrical and contained outliers influencing the spread.

Further investigation into the data's distribution involved comparing the mean and the median, which can reveal skewness and the influence of outliers. For the X-coordinates across all three clusters, the mean was consistently lower than the median, an inconsistency that suggested a negative skew from a few low-value data points. Conversely, the mean for all Y-coordinates was slightly higher than the median, implying a slight positive skew caused by high-value outliers. However, the differences between these two measures of central tendency were relatively small, suggesting that while outliers were actually present, their impact on the overall distribution was moderate. In conclusion, the analysis revealed that the cluster coordinate data was relatively centralized, with minor skewness introduced by a few outlier points in each dimension.

Table 2 showed the summary of the statistical indicators for two distinct cluster centroids derived from the second simulation scenario. These key statistics represent two additional datasets, providing further insight into the model's performance under varied conditions.

The dispersion of the clusters demonstrated significant sensitivity to the input data, particularly when observing the Interquartile Range (IQR). For example, the IQR for the X- coordinate of Cluster #2 was moderate in Dataset #1 (7.88), expanded to a very wide distribution in Dataset #2 (11.46), and then contracted to a highly concentrated cluster in Dataset #3 (3.05). This abrupt shift could be explained by the random start/goal pairs in this scenario inadvertently creating a highly concentrated traffic flow in a specific area, leading to the formation of outliers in an unanticipated direction. The volatility of IQR indicated clearly that the spread of the central 50% of the data is highly dependent on the input scenario. In contrast, the standard deviation remained more stable, with its values fluctuating within a relatively narrow range across all datasets. This suggested that while the core data spread was sensitive, the average deviation from the mean was a more robust characteristic.

Table 2. *Statistical indicators of the cluster centroids, the second simulation scenario*

		Mean	Standard Deviation	Median	IQR
Dataset #2					
Cluster #1	X	12.9062	4.345	13.7079	4.3731
	Y	11.7953	5.0474	12.3247	10.4779
Cluster #2	X	11.4155	5.0107	12.5457	11.4618
	Y	12.182	4.6467	12.3745	8.2482
Cluster #3	X	11.5598	5.0451	12.4675	11.539
	Y	11.9362	4.5711	12.2486	8.2493
Dataset #3					
Cluster #1	X	13.2477	3.6791	13.4516	4.5116
	Y	10.8499	4.6917	10.4646	8.5004
Cluster #2	X	13.4294	3.4385	13.4713	3.0461
	Y	9.9421	4.5823	8.0729	8.8684
Cluster #3	X	12.0134	3.4473	13.4352	5.9579
	Y	11.7233	4.9798	10.6778	11.2516

Furthermore, the impact of outliers and the resulting skewness of the data distribution also showed clear sensitivity. In Dataset #1 and Dataset #3, a consistent pattern emerged where X-coordinates tended to be negatively skewed, the mean value is lower than the median value, while Y-coordinates were positively skewed, the mean value is higher than the median value. However, this pattern was completely disrupted in Dataset #2, where nearly all coordinates exhibited a strong negative skew. This abrupt shift demonstrated that the distributional shape and the directional influence of outliers are not fixed properties but are instead highly sensitive to the specific characteristics of the input data.

In conclusion, this sensitivity analysis revealed that the resulting cluster characteristics were quite sensitive to changes in the input data. Both the dispersion, as measured by the IQR, and the distributional shape, inferred from the relationship between the mean and median, varied substantially. This implied that the model's output is heavily dependent on the specific dataset of intersection points, which in turn is generated from different sets of random start/goal pairs.

Table 3 presented the summary of the statistical indicators for two datasets, Dataset #4 and Dataset #5, corresponding to two cases of the third simulation scenario. They are reduced agent population, with 25 robots operating within a 25x25 obstacle-free map, and the full contingent of 50 robots within the 25x25 obstacle-filled map. There were key metrics, the mean, standard deviation, median, and interquartile range (IQR), for three distinct cluster centroids.

Dataset #4 demonstrated varied characteristics between the coordinates in the open environment. The X-coordinates generally exhibited higher dispersion and right-skewness, evidenced by the larger gaps between the mean and median and the relatively high standard deviations compared to the IQRs. This effect was most pronounced in Cluster #3, where the mean for X of 9.065 was substantially larger than its median of 6.9854, strongly suggesting the influence of high-value outliers. In contrast, the Y-coordinates across all clusters in this dataset showed less spread and more symmetry, with means and medians closely aligned. This pattern shifted significantly in the more complex obstacle-filled environment of Dataset #5. Here, all clusters exhibited greater data dispersion for both X and Y coordinates, reflected in universally larger standard deviations and IQRs. A consistent trend emerged where the mean was noticeably higher than the median across all clusters and coordinates, indicating a pervasive right-skew. This systematic skew suggested that the introduction of randomly placed obstacles had a substantial impact, generating outliers as robots navigated the more complex terrain and pulling the mean away from the central tendency of the data.

Table 3. *Statistical indicators of the cluster centroids, the third simulation scenario*

		Mean	Standard Deviation	Median	IQR
Dataset #4					
Cluster #1	X	11.3759	3.1236	10.4987	6.5131
	Y	10.5559	2.0367	10.7478	4.2595
Cluster #2	X	11.0005	3.3615	10.4987	7.5278
	Y	10.6413	1.8948	10.7478	4.2595
Cluster #3	X	9.065	2.81	6.9854	3.515
	Y	10.4636	1.7834	10.7478	3.0242
Dataset #5					
Cluster #1	X	13.908	4.0592	11.5057	7.0225
	Y	11.7269	5.1565	11.4828	7.6165
Cluster #2	X	14.0918	3.9118	11.4931	6.873
	Y	11.5559	5.0124	11.4932	7.6959
Cluster #3	X	13.3515	3.7813	11.4369	6.6379
	Y	11.8418	5.1516	11.4347	7.6009

The results of this generalizability analysis provided compelling evidence for the framework's high adaptability to varying environmental conditions. The pronounced changes in both the dispersion and skewness of the cluster data between the two conditions demonstrated that the model did not produce rigid, static results. Instead, it dynamically adjusted its clustering behavior to accurately reflect the physical constraints of the environment. This adaptability was clearly shown in the distribution's transformation, from a moderate right-skew primarily in the X-coordinates in the open space to a pervasive and more pronounced right-skew across both X and Y-coordinates in the obstructed environment. This shift strongly supported the framework's flexibility.

V. CONCLUSION AND FUTURE WORK

This study successfully demonstrated the robustness and adaptability of the Dynamic Clustering Method through a rigorous three-tiered simulation analysis. The uncertainty analysis confirmed that the model is highly consistent and reliable, producing stable cluster characteristics when applied to a constant input dataset. The sensitivity analysis further revealed that the framework's outputs are not static; rather, they are highly sensitive to variations in the input trajectory data. This indicates that the identified clusters are not merely fixed geographical zones but dynamic "hotspots" whose shape and dispersion are fundamentally dependent on the specific traffic patterns generated. Most importantly, the generalizability test provided compelling evidence of the framework's adaptability, showing its capacity to dynamically adjust its clustering behavior to accurately reflect significant changes in environmental complexity, such as the introduction of obstacles. The model's ability to transition from identifying moderately skewed distributions in an open environment to capturing pervasively skewed patterns in an obstructed one underscores its flexibility and environment-agnostic nature.

While this research establishes a strong foundation, several avenues for future work could extend its applicability and scope. Firstly, addressing the computational constraints noted in this study is a critical next step. Future research should investigate the framework's scalability by applying it to larger maps, significantly larger robot fleets, and more computationally intensive pathfinding algorithms. Furthermore, a deeper analysis of the cluster covariance matrices could provide significant insights. While this study focused on the centroids, examining the fluctuations in the covariance matrices might reveal how the shape, orientation, and volume of the congestion hotspots adapt to different traffic scenarios and environmental layouts. Finally, the ultimate goal is to bridge the gap between simulation and reality. Validating the DCM framework with trajectory data from real-world robotic systems would be an invaluable step toward its practical implementation in logistics, autonomous navigation, and fleet management.

ACKNOWLEDGMENT

This research was supported by the Program "Supporting Research, Development and Technology Application of Industry 4.0" (KC-4.0/19-30) of the Ministry of Science and Technology in Vietnam through the funding of the Project "Research, design, manufacture, and integration for a system of multiple autonomous flying robots, applying for environmental exploration, search and rescue", No. KC-4.0-49/19-25.

REFERENCES

[1] F. Rubio, F. Valero, and C. Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications," Int. J. Adv. Robot. Syst., vol. 16, no. 2, p. 1729881419839596, 2019, doi: 10.1177/1729881419839596.

[2] R. Raj and A. Kos, "A Comprehensive Study of Mobile Robot: History, Developments, Applications, and Future Research Perspectives," Appl. Sci., vol. 12, no. 14, 2022, doi: 10.3390/app12146951.

[3] J. A. Abdulsaheb and D. J. Kadhim, "Classical and Heuristic Approaches for Mobile Robot Path Planning: A Survey," Robotics, vol. 12, no. 4, 2023, doi: 10.3390/robotics12040093.

[4] S. G. Tzafestas, "Mobile Robot Control and Navigation: A Global Overview," J. Intell. Robot. Syst., vol. 91, no. 1, pp. 35–58, 2018, doi: 10.1007/s10846-018-0805-9.

[5] S. Huang, R. S. H. Teo, and K. K. Tan, "Collision avoidance of multi unmanned aerial vehicles: A review," Annu. Rev. Control, vol. 48, pp. 147–164, 2019, doi: 10.1016/j.arcontrol.2019.10.001.

[6] H. T. Xuan, T. P. Manh, N. N. Quang, and H. N. T. Hong, "Dynamic Clustering of Multi-Mobile Robot System using Gaussian Mixture Model," J. Robot. Control, vol. 6, no. 5 SE-Articles, pp. 2129–2140, Aug. 2025, doi: 10.18196/jrc.v6i5.27184.

[7] S. W. Cho, H. J. Park, H. Lee, D. H. Shim, and S.-Y. Kim, "Coverage path planning for multiple unmanned aerial vehicles in maritime search and rescue operations," Comput. Ind. Eng., vol. 161, p. 107612, 2021, doi: 10.1016/j.cie.2021.107612.

[8] I. Iswanto, O. Wahyunggoro, and A. I. Cahyadi, "Quadrotor Path Planning Based On Modified Fuzzy Cell Decomposition Algorithm," TELKOMNIKA, vol. 14, no. 2, pp. 655–664, 2016, doi: 10.12928/TELKOMNIKA.v14i1.2989.

[9] P. T. Kyaw, A. Paing, T. T. Thu, R. E. Mohan, A. Vu Le, and P. Veerajagadheswar, "Coverage Path Planning for Decomposition Reconfigurable Grid-Maps Using Deep Reinforcement Learning Based Travelling Salesman Problem," IEEE Access, vol. 8, pp. 225945–225956, 2020, doi: 10.1109/ACCESS.2020.3045027.

[10] V. G. Nair and K. R. Guruprasad, "MR-SimExCoverage: Multi-robot Simultaneous Exploration and Coverage," Comput. Electr. Eng., vol. 85, p. 106680, 2020, doi: 10.1016/j.compeleceng.2020.106680.

[11] X. Huang, M. Sun, H. Zhou, and S. Liu, "A Multi-Robot Coverage Path Planning Algorithm for the Environment With Multiple Land Cover Types," IEEE Access, vol. 8, pp. 198101–198117, 2020, doi: 10.1109/ACCESS.2020.3027422.

[12] K. R. Guruprasad and T. D. Ranjitha, "CPC Algorithm: Exact Area Coverage by a Mobile Robot Using Approximate Cellular Decomposition," Robotica, vol. 39, no. 7, pp. 1141–1162, 2021, doi: 10.1017/S026357472000096X.

[13] R. V Cowlagi and P. Tsiotras, "Beyond quadtrees: Cell decompositions for path planning using wavelet transforms," in 2007 46th IEEE Conference on Decision and Control, 2007, pp. 1392–1397, doi: 10.1109/CDC.2007.4434146.

[14] C. Cai and S. Ferrari, "Information-Driven Sensor Path Planning by Approximate Cell Decomposition," IEEE Trans. Syst. Man, Cybern. Part B, vol. 39, no. 3, pp. 672–689, 2009, doi: 10.1109/TSMCB.2008.2008561.

[15] R. J. Alitappeh and K. Jeddisaravi, "Multi-robot exploration in task allocation problem," Appl. Intell., vol. 52, no. 2, pp. 2189–2211, 2022, doi: 10.1007/s10489-021-02483-3.

[16] V. G. Nair, R. S. Adarsh, K. P. Jayalakshmi, M. V Dileep, and K. R. Guruprasad, "Cooperative Online Workspace Allocation in the Presence of Obstacles for Multi-robot Simultaneous Exploration and Coverage Path Planning Problem," Int. J. Control. Autom. Syst., vol. 21, no. 7, pp. 2338–2349, 2023, doi: 10.1007/s12555-022-0182-9.

[17] V. G. Nair and K. R. Guruprasad, "2D-VPC: An Efficient Coverage Algorithm for Multiple Autonomous Vehicles," Int. J. Control. Autom. Syst., vol. 19, no. 8, pp. 2891–2901, 2021, doi: 10.1007/s12555-020-0389-6.

[18] Ayan Dutta, Amitabh Bhattacharya, O Patrick Kreidl, Anirban Ghosh, and Prithviraj Dasgupta, "Multi-robot informative path planning in unknown environments through continuous region partitioning," Int. J. Adv. Robot. Syst., vol. 17, no. 6, p. 1729881420970461, Nov. 2020, doi: 10.1177/1729881420970461.

[19] J. Kim and H. Il Son, "A Voronoi Diagram-Based Workspace Partition for Weak Cooperation of Multi-Robot System in Orchard," IEEE Access, vol. 8, pp. 20676–20686, 2020, doi: 10.1109/ACCESS.2020.2969449.

[20] Y. Yuan, P. Yang, H. Jiang, and T. Shi, "A Multi-Robot Task Allocation Method Based on the Synergy of the K-Means++ Algorithm and the Particle Swarm Algorithm," Biomimetics, vol. 9, no. 11, 2024, doi: 10.3390/biomimetics9110694.

[21] E. Murugappan, S. Nachiappan, R. Shams, G. Mark, and H. K. and Chan, "Performance analysis of clustering methods for balanced multi-robot task allocations," Int. J. Prod. Res., vol. 60, no. 14, pp. 4576–4591, Jul. 2022, doi: 10.1080/00207543.2021.1955994.

[22] J. Li and F. Yang, "Task assignment strategy for multi-robot based on improved Grey Wolf Optimizer," J. Ambient Intell. Humaniz. Comput., vol. 11, no. 12, pp. 6319–6335, 2020, doi: 10.1007/s12652-020-02224-3.

[23] S. Sharma, R. Tiwari, A. Shukla, and J. Yadav, "Canopy Clustering Based Multi Robot Area Exploration," IFAC Proc. Vol., vol. 47, no. 1, pp. 505–510, 2014, doi: 10.3182/20140313-3-IN-3024.00253.

[24] H. Zhang, L. Bai, J. Qi, and Y. Xiao, "Area Coverage of Swarm Robotics Based on Anti-Flocking Framework with Dynamical Clustering," in 2022 IEEE International Conference on Unmanned Systems (ICUS), 2022, pp. 209–213, doi: 10.1109/ICUS55513.2022.9986539.

[25] Nur Aira Abd Rahman, Khairul Salleh Mohamed Sahari, Nasri A Hamid, and Yew Cheong Hou, "A coverage path planning approach for autonomous radiation mapping with a mobile robot," Int. J. Adv. Robot. Syst., vol. 19, no. 4, p. 17298806221116484, Jul. 2022, doi: 10.1177/17298806221116483.

[26] D. Reynolds, "Gaussian Mixture Models BT - Encyclopedia of Biometrics," in Encyclopedia of Biometrics, S. Z. Li and A. Jain, Eds. Boston, MA: Springer US, 2009, pp. 659–663.

[27] F. Najar, S. Bourouis, N. Bouguila, and S. Belghith, "A Comparison Between Different Gaussian-Based Mixture Models," in 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA), 2017, pp. 704–708, doi: 10.1109/AICCSA.2017.108.

[28] H. Wan, H. Wang, B. Scotney, and J. Liu, "A Novel Gaussian Mixture Model for Classification," in 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), 2019, pp. 3298–3303, doi: 10.1109/SMC.2019.8914215.

[29] N. J. Cho, S. H. Lee, and I. H. Suh, "Modeling and evaluating Gaussian mixture model based on motion granularity," Intell. Serv. Robot., vol. 9, no. 2, pp. 123–139, 2016, doi: 10.1007/s11370-015-0190-1.

[30] J. Yan, Y. Wu, K. Ji, C. Cheng, and Y. Zheng, "A novel trajectory learning method for robotic arms based on Gaussian Mixture Model and k-value selection algorithm," PLoS One, vol. 20, no. 2 February, pp. 1–19, 2025, doi: 10.1371/journal.pone.0318403.

[31] S. H. Lee, I. H. Suh, S. Calinon, and R. Johansson, "Autonomous framework for segmenting robot trajectories of manipulation task," Auton. Robots, vol. 38, no. 2, pp. 107–141, 2015, doi: 10.1007/s10514-014-9397-9.

[32] Y. Q. Wang, Y. D. Hu, S. El Zaatari, W. D. Li, and Y. Zhou, "Optimised Learning from Demonstrations for Collaborative Robots," Robot. Comput. Integr. Manuf., vol. 71, p. 102169, 2021, doi: 10.1016/j.rcim.2021.102169.

[33] C. Ye, J. Yang, and H. Ding, "Bagging for Gaussian mixture regression in robot learning from demonstration," J. Intell. Manuf., vol. 33, no. 3, pp. 867–879, 2022, doi: 10.1007/s10845-020-01686-8.

[34] C. Feng, Z. Liu, W. Li, X. Lu, Y. Jing, and Y. Ma, "Improved Gaussian mixture model and Gaussian mixture regression for learning from demonstration based on Gaussian noise scattering," Adv. Eng. Informatics, vol. 65, p. 103192, 2025, doi: 10.1016/j.aei.2025.103192.

[35] J. Lou, "Crawling robot manipulator tracking based on gaussian mixture model of machine vision," Neural Comput. Appl., vol. 34, no. 9, pp. 6683–6693, 2022, doi: 10.1007/s00521-021-06063-x.

[36] T. M. N. U. Akhund et al., "IoST-Enabled Robotic Arm Control and Abnormality Prediction Using Minimal Flex Sensors and Gaussian Mixture Models," IEEE Access, vol. 12, pp. 45265–45278, 2024, doi: 10.1109/ACCESS.2024.3380360.

[37] A. Mora, A. Prados, A. Mendez, R. Barber, and S. Garrido, "Sensor Fusion for Social Navigation on a Mobile Robot Based on Fast Marching Square and Gaussian Mixture Model," Sensors, vol. 22, no. 22, 2022, doi: 10.3390/s22228728.

[38] Y. Yuan, J. Liu, W. Chi, G. Chen, and L. Sun, "A Gaussian Mixture Model Based Fast Motion Planning Method Through Online Environmental Feature Learning," IEEE Trans. Ind. Electron., vol. 70, no. 4, pp. 3955–3965, 2023, doi: 10.1109/TIE.2022.3177758.

[39] Y. Wu et al., "MR-GMMExplore: Multi-Robot Exploration System in Unknown Environments based on Gaussian Mixture Model," in 2022 IEEE International Conference on Robotics and Biomimetics (ROBIO), 2022, pp. 1198–1203, doi: 10.1109/ROBIO55434.2022.10011789.

[40] M. Park, S. An, J. Seo, and H. Oh, "Autonomous Source Search for UAVs Using Gaussian Mixture Model-Based Infotaxis: Algorithm and Flight Experiments," IEEE Trans. Aerosp. Electron. Syst., vol. 57, no. 6, pp. 4238–4254, 2021, doi: 10.1109/TAES.2021.3098132.

The 8th International Conference on
Engineering Mechanics and Automation (ICEMA 2025)
Hanoi, November 14-15, 2025

DOI: 10.15625/vap.2026.0004

Artificial Bee Colony for Multi-Robot Path Planning: A Comparative Study of Population Structures

Hung X. Truong
Vietnam National Space Center, Vietnam Academy of Science and Technology,
Hanoi, Vietnam
txhung@vnsc.org.vn

Thang M. Pham
University of Engineering and Technology, Vietnam National University,
Hanoi, Vietnam
thangpm@vnu.edu.vn

Nha Q. Nguyen
University of Engineering and Technology, Vietnam National University,
Hanoi, Vietnam
nhanq@vnu.edu.vn

Hanh T. H. Nguyen
Institute of Mechanics, Vietnam Academy of Science and Technology,
Hanoi, Vietnam
nguyenhonghanh.imech@gmail.com

Viet A. Dang
University of Engineering and Technology, Vietnam National University,
Hanoi, Vietnam
vietda@vnu.edu.vn

***Abstract*—Multi-robot path planning (MRPP) remains a fundamental challenge in robotics, crucial for system coordination and task allocation, demanding solutions that are both optimal and collision-free. While numerous classical and intelligent algorithms have been developed to address this NP-hard problem, bio-inspired metaheuristics, such as the Artificial Bee Colony (ABC) algorithm, offer robust optimization capabilities. This research conducts a comparative analysis of two control paradigms within the ABC framework: a decentralized approach and a centralized, coordinated approach enhanced with a seeding strategy that uses the output of the decentralized approach as its initial solution. The strategies were evaluated for a three-robot system on a 50x50 map with varying obstacle layouts. The simulation results demonstrate that the decentralized approach provides a highly stable and reliable performance baseline but lacks the necessary coordination for inter-robot collision avoidance. In contrast, the seeded centralized approach proves to be a promising strategy, successfully finding safe, coordinated paths with costs that are often comparable, and sometimes superior, to the uncoordinated baseline. However, the analysis also reveals that the efficiency of the centralized method is sensitive to the environment's configuration, as different obstacle layouts significantly altered its performance and success rate. Nonetheless, its performance variability and the small scale of the current experiments highlight the need for further research, particularly in evaluating its scalability as the number of robots and environmental complexity increase.**

Keywords*— *Multi-robot path planning, NP-hard problem, Artificial Bee Colony algorithm, Population Structures

I. INTRODUCTION

A multi-robot system (MRS) is a collection of autonomous robots designed to collaboratively perform tasks that would be difficult or impossible for a single robot. The core principle of an MRS lies in achieving a common objective through coordinated action in a shared environment. The principal features of such systems include communication, cooperative strategies, and control architectures that can be either centralized or decentralized, often utilizing a mix of homogeneous or heterogeneous agents. The utilization of these features yields a multitude of advantages over single-robot solutions, making MRS increasingly prevalent in fields like logistics, manufacturing, and exploration. A primary benefit is robustness through redundancy; the failure of a single agent does not lead to a total system failure, as other robots can adapt and continue the mission, leading to graceful degradation of performance. Secondly, MRS provides superior scalability and flexibility, as the number of robots can be easily increased or decreased to match the complexity and scale of a given task. This modularity also allows for the efficient execution of spatially distributed tasks. Finally, these systems can achieve significant gains in operational efficiency. By executing tasks in parallel, an MRS can drastically reduce the time required for completion compared to a single, sequential approach, thereby optimizing throughput and resource utilization.

Among the challenges of multi-robot systems, one of the most fundamental and critical problems is multi-robot path planning (MRPP). Path planning is a prerequisite for any mobile robot system, as it provides the foundational capability for navigation from one point to another. In a multi-robot context, the problem's complexity is significantly amplified compared to the single-robot case, as it involves determining an optimal, collision-free path for each robot by not only avoiding static obstacles but also simultaneously considering the dynamic movements of other robots. To address this challenge, many MRPP approaches are often categorized into three main groups: classical, heuristic, and artificial intelligence (AI)-based methods, with a clear trend towards using hybrid heuristic-AI solutions and AI-based approaches [1]. A particularly prominent method is swarm intelligence (SI), which applies principles from natural swarms to enhance the efficiency and scalability of MRPP [2] [3]. The computational complexity of this problem grows exponentially with the number of robots, making it an NP-hard problem. Therefore, the development of efficient and scalable MRPP algorithms is paramount to unlocking the full operational potential of multi-robot systems and remains a primary focus of robotics research.

SI leverages the collective behavior of decentralized, self-organized systems to solve complex optimization problems. The global, intelligent behavior of the swarm is not explicitly programmed but rather emerges from simple, local interactions among individual agents and between the agents and their environment. These approaches draw their inspiration from natural systems, such as the intricate trail-laying of ant colonies, the synchronized movements of a flock of birds, and the collaborative foraging of honeybees. Inspired by this last example, the Artificial Bee Colony (ABC) method stands out as one of the most remarkable algorithms [4] [5]. The ABC algorithm models a population of candidate solutions as a set of food sources, It then utilizes a colony of artificial bees with three distinct roles: employed, onlooker, and scout bees, to cooperatively explore promising food sources and exploit the most profitable ones, thereby guiding the search process toward a global optimum.

A core challenge in applying the ABC algorithm to MRPP lies in the fundamental choice of the control architecture. One paradigm is the decentralized approach, wherein an independent ABC instance is assigned to each robot. While this model promotes scalability and computational speed, it risks converging to globally suboptimal path solutions or generating inter-robot conflicts. In contrast, the centralized approach utilizes a single, unified ABC colony to optimize all robot paths simultaneously as one high-dimensional problem. This method can theoretically achieve global optimality but at the cost of a significant increase in computational complexity, often referred to as the curse of dimensionality. Therefore, this paper undertakes an investigation to evaluate and directly compare these two paradigms. The primary goal is to quantify the foundational trade-off between solution quality and computational cost, thereby providing clear insights into the practical application of the ABC algorithm for MRPP.

The remainder of this paper is organized as follows: Section II reviews the relevant works on existing ABC approaches; Section III describes the proposed methodology; Section IV presents the simulation results along with a corresponding performance analysis; and finally, Section V provides the concluding remarks and proposed future development.

II. RELATED WORKS

The ABC algorithm has been extensively modified and hybridized to address the complexities of robot path planning. A notable trend is the hybridization of ABC with other methodologies to enhance performance. For instance, several studies have combined ABC with Evolutionary Programming (EP), using ABC for initial path discovery with EP to smooth the final trajectory [6], selecting the next best point of the path based on its distance from both the goal and nearby obstacles [7]. This hybrid concept has been extended into a novel framework incorporating a Probabilistic Roadmap (PRM) with ABC and EP for robust planning in complex industrial environments [8]. Other hybridizations have also proven effective, such as ABC with Ant Colony Optimization (IACO-IABC) to improve path smoothing and reduce turns [9], and ABC with a Genetic Algorithm (HABC-GA) to solve multi-objective path planning by balancing path length, safety, and smoothness [10].

Beyond hybridization, the core ABC algorithm itself has been improved for path planning. One approach, ABCL, incorporates the global best individual and a learning method from Teaching-Learning-Based Optimization (TLBO) to boost exploitation in multi-robot scenarios, resulting in better collision-free paths and faster run times [11]. For grid-based environments, an improved ABC (IABC) integrates path linearization and local search strategies to achieve significant path length improvements [12]. Other modifications include the ADL-ABC, which uses an adaptive control limit to reduce computational time for finding smooth paths around obstacles [13]. In direct comparisons, ABC demonstrates superior performance over classical methods like A* as well as other metaheuristics such as IPSO and IGWO, calculating shorter paths in a fraction of the time [14].

The ABC algorithm is also a powerful tool for solving complex task allocation and scheduling problems across various domains, from smart agriculture to industrial manufacturing. In the agricultural sector, various discrete versions of ABC address specific optimization tasks. These include the MFDABC for weeding robots across multiple farms [15], the multi-objective MODABC for picking robots [16], the CDABC for scheduling of multiple robots in smart farms using dynamic neighborhoods [17], and a discrete ABC (DABC) for the integrated scheduling of assigning tasks to agricultural picking robots and distribution vehicles [18]. In parallel, ABC applies to sophisticated scheduling and optimization challenges in industrial and manufacturing contexts. For robotic welding systems, a closed-loop rescheduling method combining an improved ABC with Deep Reinforcement Learning handles robot failures effectively [19]. Applications for disassembly lines include the MPCCABC to manage preventive maintenance scenarios [20], an ABC variant integrated with Q-learning is to solve bi-objective scheduling problems [21], and a multi-objective ABC with stochastic simulation for complex human-robot collaborative disassembly [22]. Furthermore, ABC supported by Deb’s rules optimizes the multi-objective problem of a single robotic arm palletizing items from multiple production lines [23].

The algorithm's applicability extends to various engineering and scientific fields. In control systems, a hybrid ABC-fuzzy algorithm offers a solution for the real-time machine learning control of robotic manipulators [24], a highly effective two-stage methodology uses a

Firefly algorithm for global allocation and a hybrid of Quantum Genetic and ABC algorithms for local allocation [25]. In logistics, hybridized ABC variants address the fuzzy capacitated distribution center problem under uncertain demand [26], while another hybrid ABC integrates concepts from Tabu Search and the Artificial Immune Algorithm to solve task planning in heterogeneous multi-robot warehouses [27]. For energy applications, a specialized binary ABC with a multi-dimensional update mechanism proves effective for the optimal placement of wind turbines [28].

A significant body of research focuses not on a specific application, but on improving the fundamental performance of the ABC algorithm itself, targeting common issues like slow convergence or an imbalance between exploration and exploitation. FOABC, which incorporates fractional-order calculus to add a memory property, has been proposed to enhance local search and exploitation [29]. Similarly, CCABC introduces horizontal and vertical search mechanisms to accelerate convergence, proving highly effective in benchmark tests and for image segmentation tasks [30]. The ABC-MNT variant addresses the balance of exploration and exploitation by using multiple neighborhood topologies with distinct search equations [31], while another approach uses a coevolution framework and a global-best leading strategy to achieve similar goals [32]. To make the algorithm more adaptive, FLABC has been designed with an online fitness landscape analysis technique, allowing it to select the most appropriate search equation based on the problem's structural features [33]. Finally, another enhanced ABC introduces innovations across all algorithmic phases, including hybrid initialization, objective-oriented operators, and self-learning mechanisms, to achieve substantial performance gains [34].

The literature on the ABC algorithm demonstrates its extensive application and continuous evolution across a wide range of optimization problems. A substantial body of research focuses on its use in robotics, particularly for complex robot path planning and for task allocation and scheduling in various engineering domains. To enhance its performance for these tasks, researchers frequently develop hybrid versions by integrating ABC with other metaheuristics or improve the fundamental mechanics of the core algorithm itself. These modifications consistently target issues such as slow convergence, path quality, and the balance between exploration and exploitation, showcasing the algorithm's versatility and robustness as an optimization tool. Despite this breadth of research, few provide a direct comparative analysis of how different control architectures, centralized versus decentralized, impact the performance of the ABC algorithm within MRPP contexts. This oversight highlights the need for investigation into the structural application of the algorithm.

III. PROPOSED METHODOLOGY

3.1. Artificial Bee Colony algorithm

The ABC algorithm's structure is modeled on the principle of dividing labor in honeybee colonies, comprising three distinct groups with specialized roles. The first group, employed bees, handles the exploitation of known solutions, represented as "food sources". Each employed bee focuses on a single food source, explores its immediate neighborhood for better alternatives, and then shares information about the source's quality, or fitness, through a metaphorical waggle dance. Following this, onlooker bees observe these dances and probabilistically select promising food sources to exploit, establishing a positive feedback mechanism that concentrates the search in promising regions of the solution space. Conversely, scout bees are responsible for exploration; when a food source is exhausted, i.e. not improved after a limited number of trials, its associated employed bee becomes a scout and conducts a random search for a new food source. This action creates a negative feedback loop that is crucial for maintaining population diversity and enabling the algorithm to escape local optima, thereby ensuring a robust balance between exploitation and exploration.

The core of the ABC algorithm is an iterative process that cycles through the following four main phases. In the initialization phase, the algorithm generates a randomly distributed initial population of SN solutions, where *SN* denotes the size of population. Each solution x_i (i = 1,2,..., *SN*) is initialized using the following expression:

$$x_{ij} = x_j^{min} + rand(0,1)\left(x_j^{max} - x_j^{min}\right) \quad (1)$$

where x_{ij} is the value of the *j*-th dimension of the *i*-th solution; x_j^{min} and x_j^{max} are the lower and upper bounds for the *j*-th dimension; $rand(0,1)$ is a random number in the range [0, 1].

In the employed bee phase, each agent searches for a new solution v_i , potentially better one, in the neighborhood of its currently assigned solution x_i. A new candidate is generated using the equation:

$$v_{ij} = x_{ij} + \emptyset_{ij}\left(x_{ij} - x_{kj}\right) \quad (2)$$

where: v_{ij} is the new candidate for the current solution x_{ij}; x_{kj} is a randomly selected neighboring solution, where $k \neq i$; and $\emptyset_{ij}$ is a random number between [−1, 1]. After generating the candidate, a greedy selection is performed based on the fitness value. If the fitness of the new solution v_{ij} is better than the old one x_{ij}, the bee memorizes the new position and discards the old one.

In the onlooker bee phase, the probability p_i of an onlooker bee selecting the *i*-th particular solution is also proportional to its fitness:

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \quad (3)$$

where: fit_i is the fitness of the *i*-th solution. Once a solution is selected, the onlooker generates a new candidate using (2), and a greedy selection is applied.

In the scout bee phase, if a solution is not be improved after a predetermined number of cycles, it is abandoned. The associated employed bee becomes a scout bee and discovers a new random solution using (1). This mechanism helps the algorithm avoid local optima and explore new regions of the search space.

3.2. Objective functions and fitness values

The operational environment for a multi-robot system is a well-defined workspace, denoted by $\mathcal{W}$. This working space exists in either two or three dimensions and is represented as the following expression:

$$\mathcal{W} \subseteq \mathbb{R}^n \tag{4}$$

where: n = 2 or 3. We define an obstacle space $\mathcal{O}$, the region of $\mathcal{W}$ containing static obstacles, which are objects whose positions remain fixed over time. Assuming that $\mathcal{W}$ has m obstacles, $\mathcal{O}$ can be determined by the following formula:

$$\mathcal{O} = \bigcup_{i=1}^{m} \mathcal{O}_i \subset \mathcal{W} \tag{5}$$

where: $\mathcal{O}_i$ is the space occupied by each individual obstacle.

Considering a system of N robots, denoted $A_1, A_2, \ldots, A_N$, navigating from their starting points $(q_{10}, q_{20}, \ldots, q_{N0})$ to their goal points $(q_{1G}, q_{2G}, \ldots, q_{NG})$, all start and goal configurations are located within the obstacle-free region. The paths for each robot A_i, denoted $Path_i(t)$, generated by the ABC algorithm based on its assigned start and goal.

$$Path_i(t) = f_{ABC}(q_{i0} \quad q_{iG}) \tag{6}$$

Decentralized approach

Each of the N robots has its own objective function and to be unaware of the system's global state in this approach. The goal of path planning is only to minimize each robot's own path length. The objective function for i-th robot is also its path length:

$$f_{obj,i} = L_i + \omega_1 \sum_{j=1}^{m} S(i,j) \tag{7}$$

where: L_i is the Euclidean distance between two points $p_{i,j+1}$ and $p_{i,j}$, waypoint step j+1 and step j of i-th robot

$$L_i = \sum_{j=1}^{m-1} \left\| p_{i,j+1} - p_{i,j} \right\|_2 \tag{8}$$

ω_1 is a high penalty weight for avoiding collision at very high priority, and $S(i,j)$ is the static obstacle collision function as follow:

$$S(i,j,k) = \begin{cases} 1, & if\ p_{i,j} \in \mathcal{O} \\ 0, & if\ p_{i,j} \notin \mathcal{O} \end{cases} \tag{9}$$

where: $p_{i,j}$ is the waypoint step j of i-th robot and $\mathcal{O}$ is the obstacle space.

This objective function does not inherently account for collisions with other robots. Collision avoidance in a decentralized approach can be handled by separate, real-time mechanisms rather than being part of this high-level path optimization function. This cost function must be converted into a fitness value to be maximized.

$$fit_i = \frac{1}{1 + f_{obj,i}} \tag{10}$$

Centralized approach

The goal of path planning aims to find a set of optimal, collision-free routes for the entire system. Therefore, the objective function consists of three primary components: a total path length, an inter-robot collision penalty, and a static obstacle collision penalty:

$$f_{obj} = f_{obj,L} + f_{obj,O} + f_{obj,C} \tag{11}$$

where: $f_{obj,L}$ is the total path length component, the sum of all individual path lengths

$$f_{obj,L} = \sum_{i=1}^{N} L_i = \sum_{i=1}^{N} \left(\sum_{j=1}^{m-1} \left\| p_{i,j+1} - p_{i,j} \right\|_2 \right) \tag{12}$$

$f_{obj,O}$ and $f_{obj,C}$ are the collision penalties for static obstacle and for inter-robot

$$f_{obj,O} = \omega_1 \sum_{j=1}^{m} S(i,j) \tag{13}$$

$$f_{obj,C} = \omega_2 \sum_{i=1}^{N-1} \sum_{k=i+1}^{N} \sum_{j=1}^{m} C(i,j,k) \tag{14}$$

where: ω_1 and ω_2 are high penalty weights to make collision avoidance a high priority; and $C(i,j,k)$ is collision function

$$C(i,j,k) = \begin{cases} 1, & if\ \left\| p_{i,j+1} - p_{i,j} \right\|_2 < d_{safe} \\ 0, & otherwise \end{cases} \tag{15}$$

where: d_{safe} is the safe distance between two robots.

Given that the objective function f_{obj} is a cost function requiring minimization, it must be transformed into a fitness value for maximization-based optimizers

$$fit = \frac{1}{1 + f_{obj}} \tag{16}$$

IV. RESULTS AND DISCUSSION

An experimental framework was conducted on a 50x50 discrete map where a small group of three robots navigated the working space with randomly placed obstacles occupying 20% of the total area. To comparatively analyze path-planning strategies, the ABC algorithm, with parameters defined in Table I, was implemented under two distinct strategies using given start and destination points illustrated in Fig. 1. The first, Scenario A, employed a decentralized approach, computing the optimal path for each robot independently. The second, Scenario B, utilized a centralized approach to performed a simultaneous optimization for the entire system. Critically, to accelerate convergence and enhance solution quality, the initial population for Scenario B was seeded with the final results from Scenario A.

Table 1. *Parameters of ABC algorithm*

Parameter	Value	Description
Population size	25	the number of employed bees or onlooker bees in the colony.
Number of waypoints	5	the number of intermediate editable points that form the path between the start and end points.
Abandonment threshold	50	the number of consecutive trials that the employed and onlooker bees perform before abandoning a solution.
Maximum iterations	200	the maximum cycles that the algorithm perform.

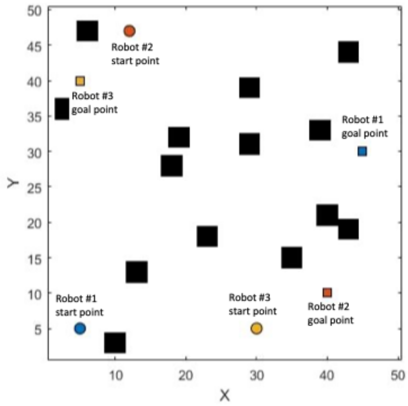

Fig. 1. The starting points and the goal points

An initial set of 50 simulation runs, illustrated in Table II, reveals a significant performance trade-off between the two approaches. The results indicate that while Scenario A performs as a reliable and highly stable benchmark, consistently converging to solutions with an average cost of approximately 137.44 and a negligible standard deviation of 0.58. This confirms its predictability for summing individual near-optimal paths.

In contrast, Scenario B presents a more dynamic performance profile but highly competitive performance profile. Although it has a higher variability with a standard deviation of 7.40, its average cost is only about 3.8% greater than Scenario A's. Moreover, the centralized approach produced a lower cost solution in 18 of the 50 runs, achieving a remarkable 36% success rate. This data demonstrates that for this specific problem instance, the seeded centralized strategy is both effective and robust. While its cost distribution is wider than Scenario A's, the ranges of the two scenarios overlap significantly, highlighting its competitiveness. Furthermore, its worst-case performance, a cost of 164.03, remains within a reasonable bound, suggesting the risk of a catastrophic, high-cost trap.

Table 2. *Best cost of two scenarios*

Scenario A	Scenario B	Scenario A	Scenario B
137.8825	137.5957	136.9654	136.9654
137.3696	147.443	136.7787	144.2803
136.9831	157.8091	137.6869	138.9455
137.0834	137.0834	137.6966	137.1626
138.2145	138.1011	136.967	155.2074
137.9076	137.7116	136.8084	136.8084
136.9987	137.5754	136.8095	136.7961
136.9792	136.887	137.3172	153.8574
137.2899	137.2671	137.6315	147.0125
137.8204	137.3796	138.5851	138.5826
136.9417	136.9417	137.1278	154.0538
138.5848	143.8318	137.4449	157.4428
137.448	146.0667	138.4788	141.8372
138.3083	155.8515	136.8684	164.0271
137.3003	149.5187	136.9394	136.9168
137.6448	137.3174	137.8625	156.091
138.4807	138.0832	137.0834	137.0472
137.1826	137.075	136.994	143.7745
137.6953	137.6953	136.9978	136.984
136.9218	136.9218	137.0048	146.7371
137.3088	140.6135	137.1948	140.667
137.2984	138.8697	136.797	136.797
139.0569	152.1447	137.0686	137.0193
136.7454	144.9842	137.7255	144.8694
138.3354	137.9712	137.4235	137.3522

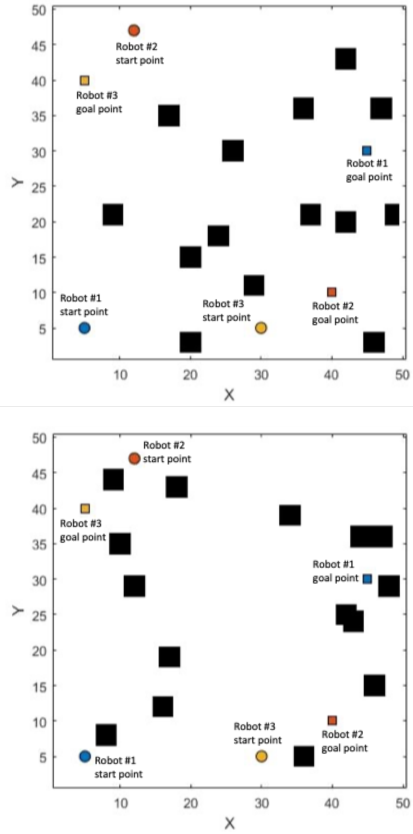

Fig. 2. *The starting points and the goal points of Layout #1 and Layout #2*

To evaluate the robustness and adaptability of the path planning strategies, the simulation was expanded by subjecting them to different static obstacle layouts, as illustrated in Fig. 2, while unchanging the robot start and goal positions. Across both new layouts, Scenario A reaffirmed its role as a stable baseline, largely insensitive to the environmental changes. Its average cost remained nearly identical in Layout #1, with mean of 137.82 and standard deviation of 0.70, and Layout #2, with mean of 138.24 and standard deviation of 0.62.

Conversely, the performance of the seeded Scenario B showed a clear dependency on the map's configuration. In Layout #1, the algorithm performed exceptionally well in 25 of the 50 runs, achieving a high 50% success rate, and indicating that the obstacle arrangement was highly conductive to efficient coordination. However, in Layout #2, the challenge of coordination increased, causing the success rate to drop significantly to 26%, as 13 of the 50 runs. This performance difference suggests that the configuration of Layout #2 creates more complex conflicts, making it more difficult for the algorithm to find an efficient escape from the initial seeded state, thereby forcing slightly less optimal detours to ensure safety.

V. CONCLUSION AND FUTURE WORK

This study demonstrates that a centralized, seeded ABC algorithm is a promising strategy for MRPP. It successfully finds safe, coordinated paths with costs that are often competitive with, and sometimes superior to, a stable decentralized baseline. However, its efficiency is largely sensitive to the specific configuration of the environment, as different obstacle layouts alter the complexity of the coordination required, a factor that warrants more in-depth research and analysis.

The current experiments are defined by two key limitations: the small scale of the three-robot system, where the number of potential path intersections is limited, and and the use of fixed start and destinational points. Therefore, future work must evaluate the strategy's performance as the system is expanded. Increasing the number of robots will expand the search space for the centralized model exponentially, providing a more rigorous test of the algorithm's capabilities and exposing any limitations regarding the balance between exploitation and exploration. Additionally, the algorithm's robustness should be investigated under varying start and goal configurations. Analyzing different task geometries would provide deeper insights into how the "cost of coordination" is influenced by the density and complexity of potential path conflicts.

ACKNOWLEDGMENT

This research was supported by the Program "Supporting Research, Development and Technology Application of Industry 4.0" (KC-4.0/19-30) of the Ministry of Science and Technology in Vietnam through the funding of the Project "Research, design, manufacture, and integration for a system of multiple autonomous flying robots, applying for environmental exploration, search and rescue", No. KC-4.0-49/19-25.

REFERENCES

[1] S. Banik, S. C. Banik, and S. S. Mahmud, “Path Planning Approaches in Multi-robot System: A Review,” Eng. Reports, vol. 7, no. 1, p. e13035, Jan. 2025, doi: 10.1002/eng2.13035.

[2] L. V. Nguyen, “Swarm Intelligence-Based Multi-Robotics: A Comprehensive Review,” AppliedMath, vol. 4, no. 4, pp. 1192–1210, 2024, doi: 10.3390/appliedmath4040064.

[3] S. Liu, Y. Niu, and B. Wu, “Introduction to Multi-Robot Coordination Algorithms,” in 2022 IEEE 4th International Conference on Civil Aviation Safety and Information Technology (ICCASIT), 2022, pp. 832–837, doi: 10.1109/ICCASIT55263.2022.9986713.

[4] D. Karaboga and B. Basturk, “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm,” J. Glob. Optim., vol. 39, no. 3, pp. 459–471, 2007, doi: 10.1007/s10898-007-9149-x.

[5] D. Karaboga and B. Akay, “A comparative study of Artificial Bee Colony algorithm,” Appl. Math. Comput., vol. 214, no. 1, pp. 108–132, 2009, doi: 10.1016/j.amc.2009.03.090.

[6] A. Q. Faridi, S. Sharma, A. Shukla, R. Tiwari, and J. Dhar, “Multi-robot multi-target dynamic path planning using artificial bee colony and evolutionary programming in unknown environment,” Intell. Serv. Robot., vol. 11, no. 2, pp. 171–186, 2018, doi: 10.1007/s11370-017-0244-7.

[7] S. Kumar and A. Sikander, “Optimum Mobile Robot Path Planning Using Improved Artificial Bee Colony Algorithm and Evolutionary Programming,” Arab. J. Sci. Eng., vol. 47, no. 3, pp. 3519–3539, 2022, doi: 10.1007/s13369-021-06326-8.

[8] S. Kumar and A. Sikander, “A novel hybrid framework for single and multi-robot path planning in a complex industrial environment,” J. Intell. Manuf., vol. 35, no. 2, pp. 587–612, 2024, doi: 10.1007/s10845-022-02056-2.

[9] G. Li, C. Liu, L. Wu, and W. Xiao, “A mixing algorithm of ACO and ABC for solving path planning of mobile robot,” Appl. Soft Comput., vol. 148, p. 110868, 2023, doi: 10.1016/j.asoc.2023.110868.

[10] F. Ye, P. Duan, L. Meng, and L. Xue, “A hybrid artificial bee colony algorithm with genetic augmented exploration mechanism toward safe and smooth path planning for mobile robot,” Biomim. Intell. Robot., vol. 5, no. 2, p. 100206, 2025, doi: 10.1016/j.birob.2024.100206.

[11] Y. Cui, W. Hu, and A. Rahmani, “Multi-robot path planning using learning-based Artificial Bee Colony algorithm,” Eng. Appl. Artif. Intell., vol. 129, p. 107579, 2024, doi: 10.1016/j.engappai.2023.107579.

[12] M. Y. Yildirim and R. Akay, “An efficient grid-based path planning approach using improved artificial bee colony algorithm,” Knowledge-Based Syst., vol. 318, p. 113528, 2025, doi: 10.1016/j.knosys.2025.113528.

[13] R. Kamil, M. Mohamed, and B. Oleiwi, “Path Planning of Mobile Robot Using Improved Artificial Bee Colony Algorithm,” Eng. Technol. J., vol. 38, no. 9, pp. 1384–1395, 2020, doi: 10.30684/etj.v38i9a.1100.

[14] N. Promkaew, S. Thammawiset, P. Srisan, P. Sanitchon, T. Tummawai, and S. Sukpancharoen, “Development of metaheuristic algorithms for efficient path planning of autonomous mobile robots in indoor environments,” Results Eng., vol. 22, p. 102280, 2024, doi: 10.1016/j.rineng.2024.102280.

[15] J.-Y. Chen, Q.-K. Pan, J. S. Neufeld, and Z.-H. Miao, “Optimization of task assignment for multi-farm multi-weeding robots based on discrete artificial bee colony algorithm,” Expert Syst. Appl., vol. 267, p. 126182, 2025, doi: 10.1016/j.eswa.2024.126182.

[16] L.-L. Dai, Q.-K. Pan, Z.-H. Miao, P. N. Suganthan, and K.-Z. Gao, “Multi-Objective Multi-Picking-Robot Task Allocation: Mathematical Model and Discrete Artificial Bee Colony Algorithm,” IEEE Trans. Intell. Transp. Syst., vol. 25, no. 6, pp. 6061–6073, 2024, doi: 10.1109/TITS.2023.3336659.

[17] H. Guo, Z. Miao, J. C. Ji, and Q. Pan, “An effective collaboration evolutionary algorithm for multi-robot task allocation and scheduling in a smart farm,” Knowledge-Based Syst., vol. 289, p. 111474, 2024, doi: 10.1016/j.knosys.2024.111474.

[18] Y. Pan and Z. Miao, “Smart Farm Picking and Distribution Integration : Discrete Artificial Bee Colony Algorithm,” in 2024 5th International Conference on Information Science, Parallel and Distributed Systems (ISPDS), 2024, pp. 535–539, doi: 10.1109/ISPDS62779.2024.10667521.

[19] P. Zhang, M. Wang, G. Zhang, P. Zheng, M. Jin, and J. Zhang, “Multi-robot multi-station welding flow shop closed-loop rescheduling with deep reinforcement learning and improved artificial bee colony algorithm,” Comput. Ind. Eng., vol. 193, p. 110295, 2024, doi: 10.1016/j.cie.2024.110295.

[20] J. Guo, Y. Li, B. Du, X. Sun, and K. Wang, “A multi-population cooperative coevolution artificial bee colony algorithm for partial multi-robotic disassembly line balancing problem considering preventive maintenance

scenarios," Adv. Eng. Informatics, vol. 62, p. 102750, 2024, doi: 10.1016/j.aei.2024.102750.

[21] Y. Ren, K. Gao, Y. Fu, D. Li, and P. N. Suganthan, "Ensemble artificial bee colony algorithm with Q-learning for scheduling Bi-objective disassembly line," Appl. Soft Comput., vol. 155, p. 111415, 2024, doi: 10.1016/j.asoc.2024.111415.

[22] C. Xu, H. Wei, X. Guo, S. Liu, L. Qi, and Z. Zhao, "Human-Robot Collaboration Multi-objective Disassembly Line Balancing Subject to Task Failure via Multi-objective Artificial Bee Colony Algorithm," IFAC-PapersOnLine, vol. 53, no. 5, pp. 1–6, 2020, doi: 10.1016/j.ifacol.2021.04.076.

[23] R. Szczepanski, K. Erwinski, M. Tejer, A. Bereit, and T. Tarczewski, "Optimal scheduling for palletizing task using robotic arm and artificial bee colony algorithm," Eng. Appl. Artif. Intell., vol. 113, p. 104976, 2022, doi: 10.1016/j.engappai.2022.104976.

[24] H.-C. Huang and C.-C. Chuang, "Artificial Bee Colony Optimization Algorithm Incorporated With Fuzzy Theory for Real-Time Machine Learning Control of Articulated Robotic Manipulators," IEEE Access, vol. 8, pp. 192481–192492, 2020, doi: 10.1109/ACCESS.2020.3032715.

[25] F. Zitouni, R. Maamri, and S. Harous, "FA–QABC–MRTA: a solution for solving the multi-robot task allocation problem," Intell. Serv. Robot., vol. 12, no. 4, pp. 407–418, 2019, doi: 10.1007/s11370-019-00291-w.

[26] Y. M. Ayid, M. Zakaraia, and M. M. Eltoukhy, "An artificial bee colony optimization algorithms for solving fuzzy capacitated logistic distribution center problem," MethodsX, vol. 13, p. 102964, 2024, doi: 10.1016/j.mex.2024.102964.

[27] Z. Zhuang, Z. Huang, Y. Sun, and W. Qin, "Optimization for cooperative task planning of heterogeneous multi-robot systems in an order picking warehouse," Eng. Optim., vol. 53, no. 10, pp. 1715–1732, Oct. 2021, doi: 10.1080/0305215X.2020.1821198.

[28] H. Hakli, "The optimization of wind turbine placement using a binary artificial bee colony algorithm with multi-dimensional updates," Electr. Power Syst. Res., vol. 216, p. 109094, 2023, doi: 10.1016/j.epsr.2022.109094.

[29] Y. Cui, W. Hu, and A. Rahmani, "Fractional-order artificial bee colony algorithm with application in robot path planning," Eur. J. Oper. Res., vol. 306, no. 1, pp. 47–64, 2023, doi: 10.1016/j.ejor.2022.11.007.

[30] H. Su et al., "Horizontal and vertical search artificial bee colony for image segmentation of COVID-19 X-ray images," Comput. Biol. Med., vol. 142, p. 105181, 2022, doi: 10.1016/j.compbiomed.2021.105181.

[31] X. Zhou, Y. Wu, M. Zhong, and M. Wang, "Artificial bee colony algorithm based on multiple neighborhood topologies," Appl. Soft Comput., vol. 111, p. 107697, 2021, doi: 10.1016/j.asoc.2021.107697.

[32] F. Xu et al., "A new global best guided artificial bee colony algorithm with application in robot path planning," Appl. Soft Comput., vol. 88, p. 106037, 2020, doi: 10.1016/j.asoc.2019.106037.

[33] X. Zhou, J. Song, S. Wu, and M. Wang, "Artificial bee colony algorithm based on online fitness landscape analysis," Inf. Sci. (Ny)., vol. 619, pp. 603–629, 2023, doi: 10.1016/j.ins.2022.11.056.

[34] F. Ye, P. Duan, L. Meng, H. Sang, and K. Gao, "An enhanced artificial bee colony algorithm with self-learning optimization mechanism for multi-objective path planning problem," Eng. Appl. Artif. Intell., vol. 149, p. 110444, 2025, doi: 10.1016/j.engappai.2025.110444.

BỘ KHOA HỌC VÀ CÔNG NGHỆ
CỤC SỞ HỮU TRÍ TUỆ

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

BẰNG ĐỘC QUYỀN
SÁNG CHẾ
Số: 48666

Tên sáng chế: CỤM CƠ CẤU ĐIỀU CHỈNH HƯỚNG VÀ ROBOT VẬN CHUYỂN HÀNG HOÁ TRONG NHÀ KHO BAO GỒM CỤM CƠ CẤU ĐIỀU CHỈNH HƯỚNG NÀY

Chủ Bằng độc quyền: Viện Cơ học, Viện Hàn Lâm Khoa Học và Công Nghệ Việt Nam (VN)
264 Đội Cấn, quận Ba Đình, thành phố Hà Nội, Việt Nam

Tác giả: 1. Nguyễn Thị Hồng Hạnh (VN)
Viện Cơ học, Viện Hàn lâm Khoa học và Công nghệ Việt Nam, 264 Đội cấn, Ba Đình, Hà Nội, Việt Nam
2. (Danh sách kèm theo)

Số đơn: 1-2023-04428

Ngày nộp đơn: 06/07/2023

Số điểm yêu cầu bảo hộ: 34 **Số trang mô tả:** 37

Cấp theo Quyết định số: 104022/QĐ-SHTT.IP, ngày: 12/06/2025

hiệu lực từ ngày cấp đến hết 20 năm tính từ ngày nộp đơn (Hiệu lực bảo hộ cần duy trì hàng năm).

C.H.X.H.C.N VIỆT NAM
CỤC SỞ HỮU TRÍ TUỆ
INTELLECTUAL PROPERTY OFFICE OF VIET NAM
BỘ KHOA HỌC VÀ CÔNG NGHỆ

VN 1-004

KT. CỤC TRƯỞNG
PHÓ CỤC TRƯỞNG

CỤC SỞ HỮU TRÍ TUỆ

Lê Huy Anh

BẰNG ĐỘC QUYỀN SÁNG CHẾ SỐ: 48666

Tác giả khác:

2. TRƯƠNG XUÂN HÙNG (VN)

Trung tâm Vũ trụ Việt Nam, Viện Hàn lâm Khoa học và Công nghệ Việt Nam, tòa nhà A6, 18 Hoàng Quốc Việt, Cầu Giấy, Hà Nội, Việt Nam

3. NGUYỄN ĐỨC MINH (VN)

Trung tâm Vũ trụ Việt Nam, Viện Hàn lâm Khoa học và Công nghệ Việt Nam, tòa nhà A6, 18 Hoàng Quốc Việt, Cầu Giấy, Hà Nội, Việt Nam

(12) BẢN MÔ TẢ SÁNG CHẾ THUỘC BẰNG ĐỘC QUYỀN SÁNG CHẾ

(19) **Cộng hòa xã hội chủ nghĩa Việt Nam (VN)**
CỤC SỞ HỮU TRÍ TUỆ

(11) **1-0048666**

(51)2022.01 **G05D 1/00**; G05D 1/02; B65G 1/00; B65G 43/00

(13) **B**

(21) 1-2023-04428
(22) 06/07/2023
(45) 25/07/2025 448
(43) 27/11/2023 428A

(73) Viện Cơ học, Viện Hàn Lâm Khoa Học và Công Nghệ Việt Nam (VN)
264 Đội Cấn, quận Ba Đình, thành phố Hà Nội, Việt Nam

(72) Nguyễn Thị Hồng Hạnh (VN); Trương Xuân Hùng (VN); Nguyễn Đức Minh (VN).

(54) CỤM CƠ CẤU ĐIỀU CHỈNH HƯỚNG VÀ ROBOT VẬN CHUYỂN HÀNG HOÁ TRONG NHÀ KHO BAO GỒM CỤM CƠ CẤU ĐIỀU CHỈNH HƯỚNG NÀY

BỘ KHOA HỌC VÀ CÔNG NGHỆ
CỤC SỞ HỮU TRÍ TUỆ

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

BẰNG ĐỘC QUYỀN
GIẢI PHÁP HỮU ÍCH
Số: 3615

Tên giải pháp hữu ích: PHƯƠNG PHÁP XÁC ĐỊNH TRẠNG THÁI CHUYỂN ĐỘNG CỦA ROBOT DI ĐỘNG VẬN CHUYỂN HÀNG HOÁ ĐỂ PHỤC VỤ ĐIỀU KHIỂN VÀ GIÁM SÁT ROBOT

Chủ Bằng độc quyền: Trung tâm Vũ trụ Việt Nam, Viện Hàn lâm Khoa học và Công nghệ Việt Nam (VN)
Tòa Nhà A6, 18 Hoàng Quốc Việt, Cầu Giấy, Hà Nội, Việt Nam

Tác giả: 1. Trương Xuân Hùng (VN)
2. (Danh sách kèm theo)

Số đơn: 2-2023-00779

Ngày nộp đơn: 06/01/2022

Số điểm yêu cầu bảo hộ: 01 **Số trang mô tả:** 20

Cấp theo **Quyết định** số: 52452/QĐ-SHTT.IP, **ngày:** 07/05/2024

Có hiệu lực từ ngày cấp đến hết 10 năm tính từ ngày nộp đơn (Hiệu lực bảo hộ cần duy trì hàng năm).

VN 2-[illegible]15

KT. CỤC TRƯỞNG
PHÓ CỤC TRƯỞNG

CỤC SỞ HỮU TRÍ TUỆ

Lê Huy Anh

BẢNG ĐỘC QUYỀN GIẢI PHÁP HỮU ÍCH SỐ: 3615

Danh sách các Tác giả tiếp theo:

2. Nguyễn Thị Hồng Hạnh (VN)
3. Trịnh Thăng Long (VN)
4. Nguyễn Văn Thức (VN)

(12) **BẢN MÔ TẢ GIẢI PHÁP HỮU ÍCH THUỘC BẰNG ĐỘC QUYỀN GIẢI PHÁP HỮU ÍCH**

(19) **Cộng hòa xã hội chủ nghĩa Việt Nam (VN)**
CỤC SỞ HỮU TRÍ TUỆ

(11) **2-0003615**

(51) 2022.01 **G05D 1/00**; B60P 1/00; B62D 61/00

(13) **Y**

(21) 2-2023-00779
(22) 06/01/2022
(67) 1-2022-00060
(45) 25/06/2024 435
(43) 25/03/2022 408

(73) Trung tâm Vũ trụ Việt Nam, Viện Hàn lâm Khoa học và Công nghệ Việt Nam (VN)
Tòa Nhà A6, 18 Hoàng Quốc Việt, Cầu Giấy, Hà Nội, Việt Nam

(72) Trương Xuân Hùng (VN); Nguyễn Thị Hồng Hạnh (VN); Trịnh Thăng Long (VN); Nguyễn Văn Thức (VN).

(54) PHƯƠNG PHÁP XÁC ĐỊNH TRẠNG THÁI CHUYỂN ĐỘNG CỦA ROBOT DI ĐỘNG VẬN CHUYỂN HÀNG HOÁ ĐỂ PHỤC VỤ ĐIỀU KHIỂN VÀ GIÁM SÁT ROBOT

(57) Giải pháp hữu ích liên quan đến việc điều khiển và giám sát hoạt động của robot di động ứng dụng trong lĩnh vực vận chuyển hàng hóa trong nhà kho, trong đó hàng hóa được bố trí trong giá hàng có chiều cao lớn và nằm ở phía trên robot. Đây là hệ hai vật liên kết robot-giá hàng có tọa độ điểm trọng tâm nằm cao hơn robot. Cụ thể hơn, giải pháp hữu ích đề cập đến phương pháp xác định trạng thái chuyển động của robot di động, bao gồm, nhưng không chỉ giới hạn ở, hướng, quay, gia tốc (gồm tăng tốc và giảm tốc), sốc, rung động, độ nghiêng dựa trên sự kết hợp giữa bố trí cảm biến, kết nối phần cứng và giải pháp tính toán, trong đó, giải pháp tính toán dựa trên dữ liệu đọc về định kỳ từ cảm biến đo lường quán tính (ĐLQT). Các trạng thái và dữ liệu tính toán được sử dụng để ra lệnh điều khiển hoạt động robot di chuyển theo chế độ phù hợp hoặc giám sát cảnh báo các tình huống bất thường.