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.

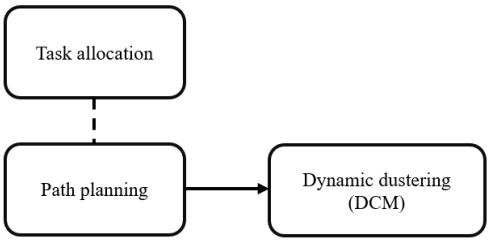

Fig. 1. Operation flow for DCM with path planning and task allocation

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:

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

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

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

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:

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

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

$$\sum_{i=1}^{K} w_i = 1 \quad (10)$$

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

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

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:

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

- 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:

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

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

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

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:

$$\delta_k = \frac{1}{N_K} \quad (16)$$

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:

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

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.

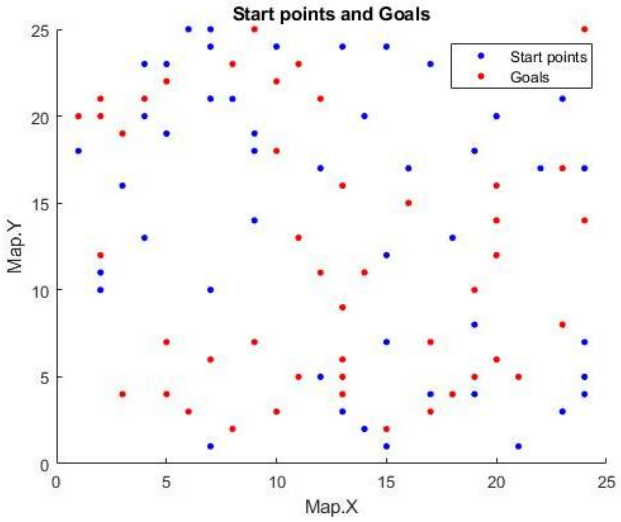

Fig. 2. Starting points and goals

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.

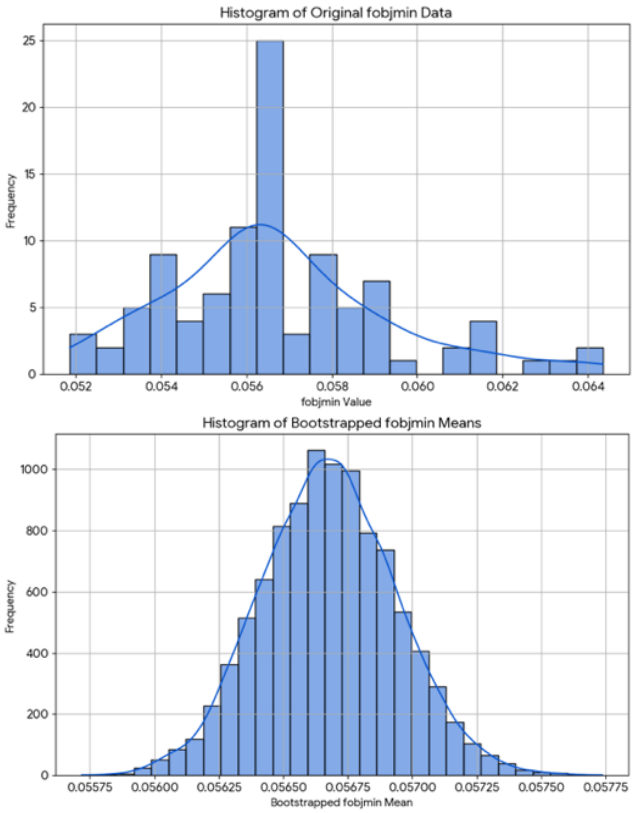

Fig. 3. Histogram of f_{objmin} data (top) and bootstrapped f_{objmin} means (bottom)

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.