IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received August 7, 2021, accepted August 23, 2021, date of publication August 26, 2021, date of current version September 3, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3108177

A Comprehensive Review of Coverage Path Planning in Robotics Using Classical and Heuristic Algorithms

CHEE SHENG TAN[1], ROSMIWATI MOHD-MOKHTAR[1], (Senior Member, IEEE), AND MOHD RIZAL ARSHAD[1,2], (Senior Member, IEEE)

[1]School of Electrical and Electronic Engineering, Universiti Sains Malaysia, Engineering Campus, Nibong Tebal, Pulau Pinang 14300, Malaysia
[2]Universiti Malaysia Perlis (UniMAP), Kampus Pauh Putra, Arau, Perlis 02600, Malaysia

Corresponding author: Rosmiwati Mohd-Mokhtar (eerosmiwati@usm.my)

This work was supported by the Fundamental Research Grant Scheme FRGS/1/2019/TK04/USM/02/12, and the Universiti Sains Malaysia (USM) Fellowship.

ABSTRACT The small battery capacities of the mobile robot and the un-optimized planning efficiency of the industrial robot bottlenecked the time efficiency and productivity rate of coverage tasks in terms of speed and accuracy, putting a great constraint on the usability of the robot applications in various planning strategies in specific environmental conditions. Thus, it became highly desirable to address the optimization problems related to exploration and coverage path planning (CPP). In general, the goal of the CPP is to find an optimal coverage path with generates a collision-free trajectory by reducing the travel time, processing speed, cost energy, and the number of turns along the path length, as well as low overlapped rate, which reflect the robustness of CPP. This paper reviews the principle of CPP and discusses the development trend, including design variations and the characteristic of optimization algorithms, such as classical, heuristic, and most recent deep learning methods. Then, we compare the advantages and disadvantages of the existing CPP-based modeling in the area and target coverage. Finally, we conclude numerous open research problems of the CPP and make suggestions for future research directions to gain insights.

INDEX TERMS Coverage path planning, exploration, heuristic algorithm, deep reinforcement learning.

I. INTRODUCTION

Mobile robots such as unmanned aerial vehicles (UAVs), unmanned ground vehicles (UUVs), autonomous underwater vehicles (AUVs), autonomous surface vehicles (ASVs), and industrial robots have been used to perform autonomous area coverage tasks for field exploration. Although the industrial robot arm generally manipulates the end-effector to reach the goal position along a predetermined path to cover a specified target area, such a method is not optimized to avoid static or dynamic obstacles in the path space domain. Hence, autonomous robots must overcome the obstacles by resolving the coverage path planning (CPP) problem for interacting in a complex environment.

CPP has become a hot research topic in robotic applications such as autonomous cleaning [1], [2], lawn mowing [3], structural inspection [4], [5], agriculture [6], [7], and surveillance [8], including exploration, mapping, search, and rescue [9], [10]. Robotic end-effector could also be beneficial from CPP such as surface treatment applications (milling [11], laser cleaning [12], spray painting [13], [14], fused deposition modeling printing, and manufacturing inspection [15], [16]). CPP is the determination of the path that cover all points from an initial state to a final state while detecting and avoiding obstacles in a target environment [17]. The goal of the CPP algorithm is to compute the optimal path and project a collision-free trajectory to ensure the robot fully covers an area of interest (AOI) within a certain time. Firstly, a decomposition technique decomposes the AOI into a set of sub-areas. Then, it sets an initial position of the robot and determines the covering direction of each sub-area. Effective optimization solver computes the sequence connection of the sub-areas to cover each cell. Finally, the robot covers all the sub-areas by using simple movements such as back-and-forth

The associate editor coordinating the review of this manuscript and approving it for publication was Zheng H. Zhu.

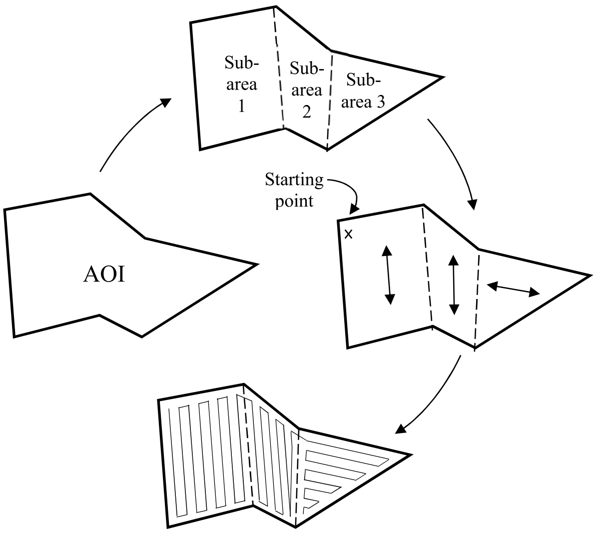

FIGURE 1. The concept of coverage path planning (CPP).

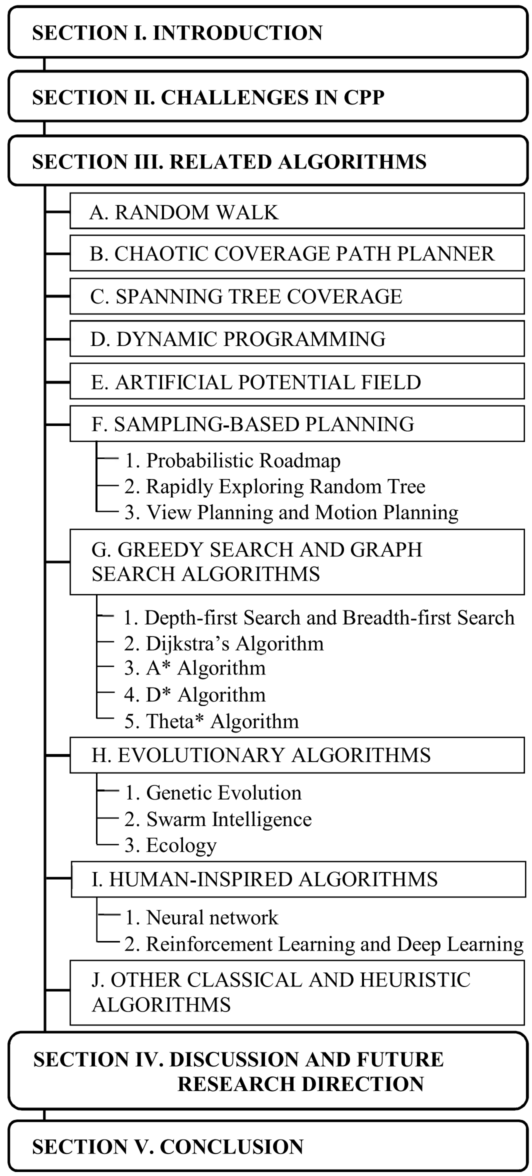

FIGURE 2. The organization of the paper.

motion. The concept of the CPP is illustrated in Fig. 1. The robustness and performance of CPP efficiency are based on several parameters, such as the percentage of covered area, travel time, path overlap rate, and energy consumption of robots.

CPP is an integral part of mobile robot exploration to deal with area coverage optimization. Area coverage is generalized as a completely or partially enclosed area with a non- overlapping path by robots. Depending on the prior knowledge of the surrounding environment with onboard sensors, the CPP algorithm can be categorized into off-line and online algorithms [18]. The off-line algorithm allows the mobile robot to perform the coverage with a static and well-known environment. The CPP is generally based on global sequential point-to-point coverage, and the robot follows the route besides obstacle avoidance on the given map. However, in practice, the robot needs to deal with an unknown or partially known environment. Therefore, the online algorithm is preferred, whereby the exploration strategy changes whilst the robot moves, executes, acts, and observes the location of the obstacle to explore an unknown area within the region of interest. The robot will resolve for a suitable path by acquiring real-time data from the local sensor and extract distinctive features in the dynamic environment. In the end, the robot must create a finite mapping of the environment under exploration with the CPP technique [21].

In the past decade, Galceran and Carreras [18] have reviewed CPP for robotics literature. The works reported are surveys on an environment modeling based on various surface partitioning methods used in solving the CPP problem, i.e., cellular, grid-based, and graph-based methods of the respective 2D and 3D structures. The literature reported in recent years is a review on multi-robot CPP for model reconstruction and mapping [19] and specifically a review on drones [20]. The difference between past review papers and the present review is a comprehensive and state-of-the-art study particularly in terms of optimization criteria. In the current review article, an extensive review of CPP focusing mainly on the classical and heuristic algorithms used to solve the optimization problem. The collision-free path, coverage cost function (shortest and smoothness paths), and coverage sequence (set covering problem, SCP and traveling salesman problem, TSP) directly correlate with CPP problems, in which how well the optimization problem can be solved. Furthermore, no literature review exists on addressing the CPP problems using deep reinforcement learning methods. We believe that this review will provide a comprehensive understanding of CPP in robotics in terms of design variations, the characteristic of optimization algorithms, and various technical features, i.e., searching time, path optimality, dynamic performance, convergence speed, and computational complexities.

This paper provides a review of CPP techniques. The remaining of the paper goes as follows. Section II presents the objective of CPP and the specific challenges regarding the platform, environment, and path optimality. Next in Section III gives the recent development of CPP based on various classical and heuristic algorithms. The existing reviews are related to coverage efficiency issues and performance metrics. Section IV analyzes and summarizes the applications of various CPP algorithms with the advantages and disadvantages and discusses the open problems in the CPP for future research to provide the directions. Finally, Section V concludes the paper. The organization of the paper is shown in Fig. 2.

II. CHALLENGES IN CPP

CPP is still an open problem in robotics in improving the efficiency of planning an optimal path to cover the target area, as well as generating a collision-free pathway with less computation. The generated coverage path should be optimal to ensure minimal logistical costs, such as overlapping, number of turns, travel time, and energy consumption. The CPP problems include potential uncertainty failures, unknown obstacles in a complex environment, and path optimality, which are considered the major challenges in robotics. An overview of CPP problems with the objective, challenges, and design features is shown in Fig. 3.

Area coverage using a single robot has been presented in many works, whereby only one autonomous vehicle executes a simple task in small areas such as room cleaning. In the case of broader area coverage, the robot may suffer mission incompletion due to uncertainty in malfunction and potential failures, such as mechanical or electronic breakdown, sensor and actuator faults, and battery drainage. Thus, many researchers focus on improving the efficiency of the area coverage by deploying multi-robot systems. Multi-robot coverage provides more significant advantages over a single robot in minimizing operational time and enhancing the robustness of CPP [22]. However, developing the CPP technology of multi-robot is still challenging to fulfill complex and large-scale environments because it must address many CPP constraints.

Meanwhile, limited sensing capability and communication bottlenecking are the significant factors to deal with in the face of positioning failures of the multi-robot system. Thus, the distributed control network system is either broadcasted by centralized or decentralized methods to avoid the scalability problem in such a limitation [23]. Besides, the strategic resilience of a team robot is equally important, where the neighbor robots could overtake re-planning tasks to fill the functional gap in the case of robot failure [24]. Improper task schedules could also lead to an idling problem. Specifically, coordination and task allocation are the core problems in the multi-robot distribution for area coverage, highly depending on each robot's position. Therefore, the efficiency of the CPP is highly reliant on coordination and task allocation strategies in the effort of minimizing the total coverage time and balancing the workload of each robot. In the end, the multi-robot system could provide system redundancy and high fault-tolerant compared to a single robot.

Environmental factors like wind, wave and underwater current are still considered a great challenge for the CPP in robotics. Vehicles such as UAV, AUV, and human-centered intelligent robots [25] must stabilize themselves in a position when collecting the data under the physical influence of the environmental conditions as well as the impact of human motion. Other than counteracting with external forces, obstacle avoidance is also a common practice to prevent physical damage to the vehicle by physical collision. The CPP for large-scale environments (especially multi-robot systems) is often an off-line planned algorithm due to the limited onboard sensor and battery limitation. Generally, many CPP techniques of robots are only considering in a two-dimensional (2D) workspace due to the complexity of kinematic and dynamic constraints. That, in turn, limiting the robots capable of three-dimensional (3D) space coverage, especially in an underwater environment [26]. Despite the simplicity in the 2D model that only requires a small amount of computation. Hence, many studies create a 2D model on a cross-section of a surface, neglecting the height information in 3D modeling since most robots can perform 2D specific area coverage tasks. However, the significant aspect of the CPP problem in the artificial 2D workspace is the overlapped coverage of the sensor footprint along the sweeping path [10]. In reality, the height at a constant depth varies when UAV or AUV covers the region of interest (ROI) on a non-planar surface (large degree of the surface slope). When the environment is prior known, cellular decomposition is the simplest method to segment the region into smaller sub-areas, either regular grid cells or polygon shapes [27]–[29]. CPP in the 3D space mainly focuses on the target coverage in such a way as to cover the critical ROI for evaluating the quality of the structure (3D model). The effective coverage of the target area can be achieved by generating viewpoints and optimizing the sequence of visiting the viewpoints. However, most research works only focused on 3D targets with a smooth surface (less interest in rough surface or hidden surface) [30].

Path optimality is related to the shortest coverage path or TSP, where typically in terms of planning a path with minimal travel cost to visit all points through the multi-ROIs. Thus, it introduces a significant challenge to address the CPP problem since TSP and CPP problems are NP-hard [31]. Many integrated TSP and CPP studies on finding the visiting order for the set of regions by TSP solver, as well as planning the optimal path to fully cover all the sub-regions in the back-and-forth manner [32]–[34]. Hence, the connection of local and global coverage paths should be concerned to address the integrated TSP and CPP problems, including coverage path in each ROI, the sequence of visiting order within sub-regions, and the entry-exit path. Additionally, in a 3D surface, single or multi-robot typically generate a set of viewpoints to cover the target surface areas through view planning and

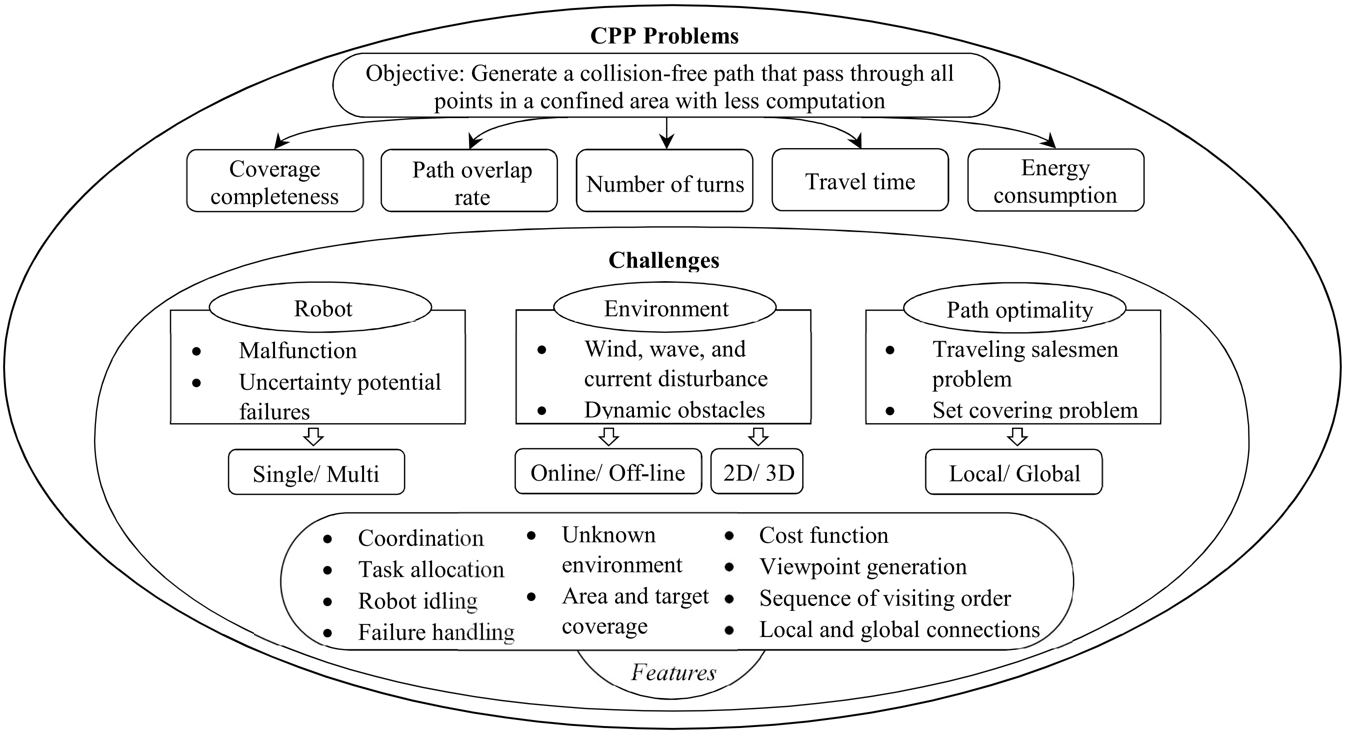

FIGURE 3. The objective and challenges in coverage path planning (CPP) problems.

find the shortest path with collision-free to visit the selected viewpoint [35], [36]. The view planning problem in model-based is typically regarded as an SCP that the goal is to reduce the number of viewpoints, then the TSP or multi-TSP solves the set of selected viewpoints for tackling the path planning problem [23]. Therefore, the challenge of the CPP path optimality is to minimize the total travel time along the coverage path and reduce the turning cost.

III. RELATED ALGORITHMS

CPP algorithms can be categorized into two approaches, classical algorithms, and heuristic-based algorithms. The summarized details of CPP algorithms according to the characteristics of the algorithms are classified as shown in Fig. 4. Notably, sampling-based planning and bio-inspired algorithms are hot research topics for solving CPP problems. There are ten highlights in the existing literature, i.e., random walk, chaotic coverage path planner, spanning tree coverage, artificial potential field, sampling-based planning algorithms, dynamic programming, greedy search and graph search algorithms, evolutionary algorithms, human-inspired algorithms, and other classical-heuristic algorithms.

A. RANDOM WALK

Random walk (RW) is a stochastic process that describes the animal search pattern or movement in the attempt to scan and explore the unexplored area [37]. Different variants of the RW have been studies for environmental exploration and coverage [38], [39]. There are two methods for area coverage based on the RW, i.e., fixed linear method and variable step method. Robot of fixed linear approach randomly turns at an angle and frequently moves at the straight line until it collides with the wall or obstacle boundaries. Hasan *et al.* [40] introduced CPP algorithms that involve the RW, spiral motion, boustrophedon motion, and wall follower in the cleaning system. Liu *et al.* [41] proposed an online random coverage method that improves the coverage rate. However, to ensure that the robot covers the whole area, the variable step method computes a set of RW directions based on the probability distribution of step lengths taken by the robot.

The variable step method is popular in a collaborative mobile robot swarm system, including Brownian motion (BM) [42] and Lévy flight (LF) [43]. The robot based on BM repeatedly moves in a step length with a given distribution (i.e., Gaussian or von Mises [44]) and randomly turns in a direction. Conversely, the robot of LF travels a distance in which the step length depends on Lévy's probability distribution [45]. The BM step length is of finite variance, whereas the LF step length is of infinite variance. Therefore, BM has a high target density (local walk) and short-range movement compared to LF (global walk). Martinez *et al.* [46] proposed a swarm robot using BM-based RW to enhance area coverage. Each robot is considered a particle whose motion is controlled by signals in the environment. In [47], pheromone-based communication [48] is utilized to control multi-robots and the LF search strategy is implemented to improve the efficiency of searching and coverage in an unknown environment. Whereas [49] proposed gradient following combined

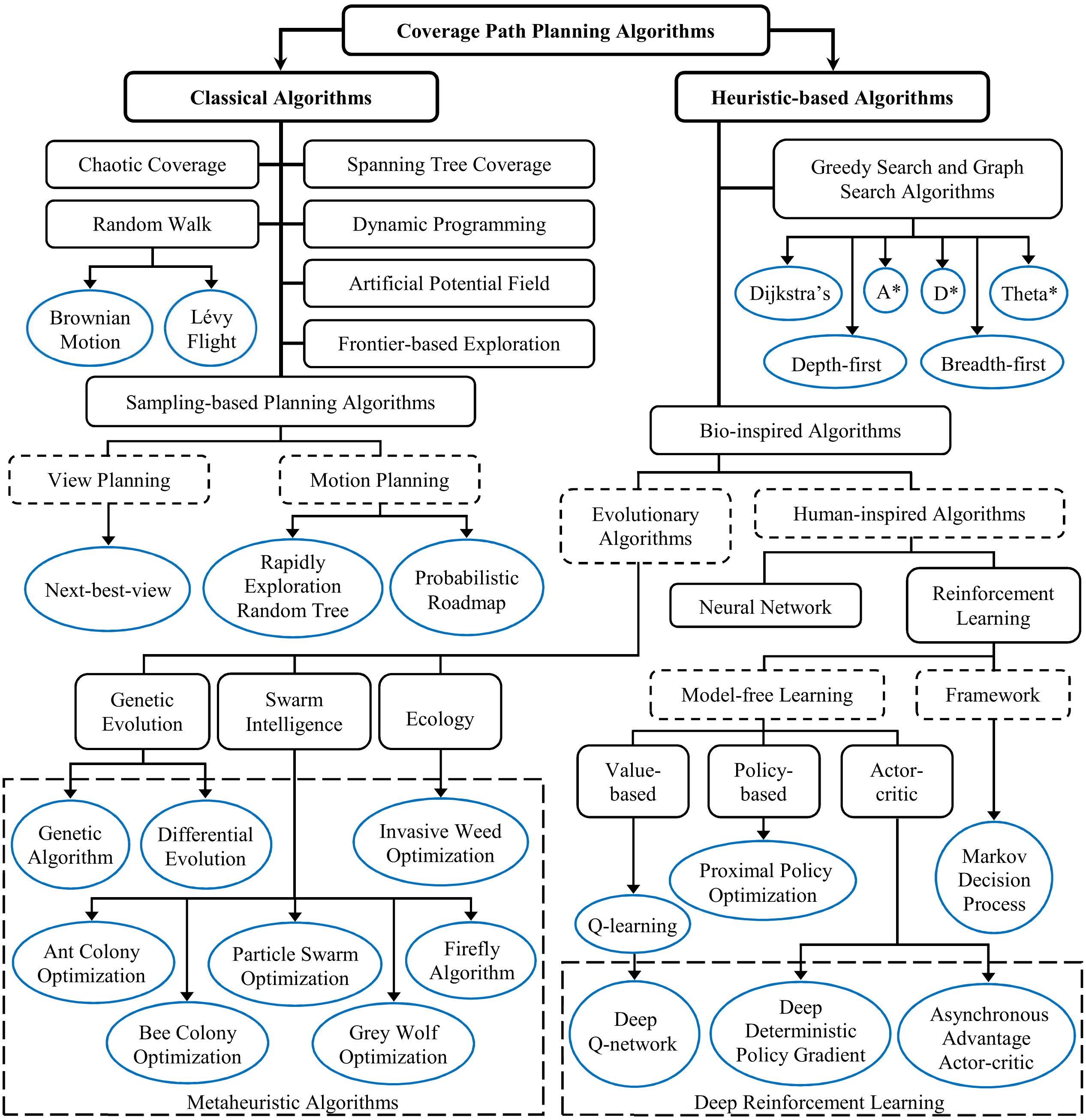

FIGURE 4. **The classification of coverage path planning (CPP) algorithms.**

with the LF approach using a virtual pheromone-based model in the control to provide better performance in area coverage.

The main advantage of the random walk approach is that the platform does not require sensors for localization. The robot only requires simple onboard sensors to sense and detect the boundaries of an area for obstacle avoidance. Thus, it is very flexible and easy to deploy due to a simple algorithm with less memory requirement. However, the RW path valid only for a small environment and hard to cover all areas in the presence of obstacles. The robot may also cross the same path several times, leading to an inefficient overall path.

B. CHAOTIC COVERAGE PATH PLANNER

Chaotic CPP is a deterministic technique that consists of a chaotic system to generate a coverage trajectory based on chaotic motion. Chaotic CPP ensures high coverage efficiency in the entire workspace in terms of the robot's trajectory, guaranteeing faster coverage in the working space because the motion is pre-determined. Arnold's dynamical system is a well-known chaotic system, first introduced by Sekiguchi and Nakamura [50]. A controller is designed and built with a combination of chaotic dynamic variables and kinematic equations of the mobile robot to construct chaotic motion. This system could also perform surveillance tasks by achieving the highest coverage rate without the requirement of obstacle avoidance along the boundaries [51].

In the case of a 3D non-linear chaotic system, the Lorenz dynamical system and Chua circuit are similar to the Arnold dynamical system. In [52], the Lorenz system speeded up the workspace coverage by utilizing the hyperchaotic technique with a non-linear open-loop controller, showing a good chaotic characteristic compared to the Arnold system and RW [53], [54]. Chua patterns used in mobile robot also provides a better coverage performance [55], [56]. A random number generators based on chaotic attractors using the Chua circuit, Lorenz system, and multiple scroll attractors have been proposed in the CPP [57]. Nasr *et al.* [58] utilized a multi-scroll Chua chaotic mirror mapping method to determine the low-cost coverage path.

Standard (Taylor–Chirikov) and logistic map are both the discrete-time dynamical system model of a 2D iterated map and 1D iterated map, respectively. Volos *et al.* [59], [60] design a chaotic logistic map random bit generator to generate the coverage trajectory for the mobile robot. Angular transformations could further improve the evenness of the coverage path planner [61]. Whereas [62] implemented a pseudo-random bit generator combined with an inverse pheromone method, achieving less memory requirement while providing higher coverage in a given terrain. In the case of the standard map, [63] presented the terrain space covering using a discontinuous control law. Whilst [64], [65] suggested a fusion strategy with the iteration cycles between large and small divided regions as well as mapping (affine transformations) correspond to the standard map. Meanwhile, Li *et al.* [66] used a 2D Chebyshev map with a similar affine transformation technique for chaotic CPP.

Most of the chaotic CPP motion does the exploration and surveillance mission in an unpredictable random, small number of steps and provides fast scanning in an unknown environment compared to RW because RW is not continuous [67]. Thus, the continuous motion of chaotic CPP enables the robot to move in searching and finding the target effectively with a more uniform coverage density. However, the existing literature only highlighted the coverage rate, ignoring the cost of coverage time. The unpredictable trajectory is also hugely dependent on the kinematic motion of the robot subjected to kinematic constraint, and it needs to be studied.

C. SPANNING TREE COVERAGE

The spanning-tree coverage (STC) based CPP algorithm subdivides the workspace into a finite sequence of disjoint cells, either by cell decomposition-based method or grid-based method [68], [69]. Then, it constructs a spanning tree of the graph in the corresponding mega cells that spit into four subcells, whereby the size of the corresponding cells equals the size of the robot. This algorithm enables the robot to cover each unoccupied cell by finding the optimal path using a tree traversal algorithm, such as depth-first search. However, the robot fails to cover the mega-cell if an obstacle within an entire mega-cell occupies a sub-cell. In [70], the authors proposed a full-STC algorithm, where a robot can cover the free sub-cells to maximize the area coverage. The STC has been extended in focusing on the online strategy for a multi-robot system to increase the coverage efficiency [71], [72]. However, the traveled path is dependent on the initial positions of each robot and might lead to backtracking issues among other robots. The robots suffer from a high overlap rate, significantly deteriorating energy efficiency. Kapoutsis *et al.* [73] proposed an area division algorithm concerned with the initial positions of robots to optimal cell assignment in matrix conditions. A minimum spanning tree is constructed in each divided space for balanced task assignment. Still, it cannot deal with the pathway through the free sub-cells situation in which the cells are occupied by obstacles, especially in robot placement along the same axis. In [74], the workspace is divided into different cell sizes based on the hierarchical quadtree structure, following the construction of the spanning tree by considering different edge lengths. This method could minimize the repeated coverage and balance the task assignment, but introduces over-segmentation in the cell, leading to extra task costs.

Gao and Xin [75] proposed the STC algorithm based on auction and bidding processes for solving multi-robot CPP. In [76], a pseudo-STC is constructed to create the virtual edges, providing that the obstacles occupy the mega-cells. The wall following algorithm enables the robot to move along the obstacle boundary through the sub-nodes. Meanwhile, Pham *et al.* [77] improved the algorithm to find the optimal path, focusing on minimizing the backtracking and increasing the coverage rate by considering the mega-cells that are partially occupied by obstacles in building the C-space boundary contour. The path is planned through the spanning-tree edge in an anti-clockwise direction to find the next unvisited mega-cell. In the case of the next mega-cell that is partially occupied by obstacles, the robot moved along the C-space edge and returned to the parent node. The experimental results show that the proposed algorithm achieves a high coverage rate as compared to the full-STC method. Similarly, [78] proposed the adjacency graphs structure based on connectivity between the minor nodes to allow the robot to cover the mega-cells that are partially occupied by obstacles. Typically, the robot-based online CPP needs to provide sensing feedback, resulting in considerable energy usage. Hence, [21] proposed a hybrid CPP without the aid of scanners by combining the frontier-based exploration and STC algorithm to improve energy efficiency.

In the latest studies, most of the multi-robots-based STC algorithms rely on centralized control techniques, involving communication and task allocation. The sensor information significantly burdens computation and memory complexity. That might result in system failures when a breakdown occurs in the central control agent. Dong *et al.* [79] proposed an artificially weighted STC based on a decentralization strategy to perform the coverage task in a distributed manner. The tasks burdened by each robot are equally distributed and the algorithm could re-generate the STC path if the robot failure occurred. Thus, the system could ensure the completion of the coverage task in the case of robot failure. However, the path re-planner could ignore the task burdened by the operating robots, leading to an unbalanced workload problem. Fault tolerance is still a big challenge in a real-world situation.

D. DYNAMIC PROGRAMMING

Dynamic programming (DP) is an approach for solving optimization problems by recursively dividing the complex problem into a set of simple sub-problems and recombining the results of all the sub-problems to obtain the solution [80]. The DP problem exhibits the overlapping sub-problems and optimal substructure in CPP to optimize the sequence of global coverage sub-spaces based on the distance matrix [81]. In [82], the DP and TSP reduction could optimize a greedy construction of a set of segments and the connection of all the segments respectively to find the shortest coverage path and minimize the number of turns. The DP framework is developed in [83] to optimize the coverage overlaps within an area of interest. Coombes *et al.* [84] used the bottom-up strategy to save memory space and accelerate through the recombination process of the decomposed cells. A DP has been used to solve the TSP in CPP with global planning for finding the shortest path that sequentially covers all the regions [33]. However, the generated tour might not be optimal due to the enormous scale of the problem. Thus, [34] proposed the nearest neighbor-based or genetic algorithm (GA) based 2-Opt algorithm to solve many regions, further optimizing the tours by the DP-based exact approach. Cheng *et al.* [85] introduced the graph model of the environment according to the sets of morphology layer and stripe layer, requiring cost calculation of each strip layer to be memorized by DP, developing re-calculations precaution to speed up the computation. However, the robot cannot adapt to a complex dynamic environment. Ghaddar and Merei [86] suggested an online CPP algorithm by utilizing DP to improve performance in terms of adaptability and energy efficiency.

E. ARTIFICIAL POTENTIAL FIELD

The artificial potential field (APF) algorithm is commonly used in detecting obstacles when the robot is towards the goal position. A fictional repulsive force and attractive force are created in the surrounding obstacles and around the goal respectively to ensure the robot in achieving the target while keeping the distance between the robot and obstacles [87]. Sutantyo *et al.* [88] employed the LF algorithm to explore the unknown environment. The dispersion is enhanced by adding the APF technique for producing the repulsion among the robots. In [89], the coverage path is re-planned by calculating the cost according to the artificial potentials when the sensor detects the defect for surface treatment. However, the robot may fail to escape from the dead zone due to the APF method has a local optimum problem. Hence, Wei *et al.* [90] implemented the inspection strategy by combining the APF and particle swarm optimization (PSO) algorithms to overcome the problem of local optimal by optimizing the speed and position of particles. Wang *et al.* [91] introduced a potential field based on the information gain and path cost, in which the robot can find the optimized trajectory to avoid being trapped in local minima. In [92], the authors improved the APF algorithm by introducing the concept of seeds for CPP in a grid environment. Different kinds of path seeds can be generated according to the environment map to cover the area. Huang *et al.* [93] utilized the APF method to cover the area by the formation control of the multi-robot system. The simulation results proved that the approach achieves better area coverage and real-time planning. In a specific case such as a robot pass through a narrow space, the robot might not be able to reach the target. Hence, Jiang and Deng [94] improved the APF algorithm by modifying the repulsive potential function to avoid the obstacles in the inspection mission effectively. Despite all the research effort, there is still a lack of planning for collision avoidance between multiple robots when simultaneously access to the goal under the potential field.

F. SAMPLING-BASED PLANNING ALGORITHMS

The traditional algorithm applies a random sampling method to a coverage issue for solving a planning problem [95], [96]. Recently, probability sampling-based planning (SBP) algorithms have been used to solve complex planning problems heuristically and optimally. Generally, the algorithm is the process of mapping the environment from configuration space by using a node sampling strategy (random generation of a set of nodes in the search environment). The probabilistic completeness of SBP is effective for optimizing sensor-based (visual-based) inspection in terms of exploration. SBP based planner includes probabilistic roadmap (PRM) [97] and rapidly exploring random tree (RRT) [98].

1) PROBABILISTIC ROADMAP

The PRM planner is a process of planning and query by establishing a roadmap for creating a path in the configuration space [99]. The planning phase randomly generates the number of nodes in the robot's configuration space and connects the pairs of nodes in a straight line without crossing the obstacles to form a roadmap. Then, the query phase plans a path between initial and goal configuration by using the result from the planning phase [100]. Dias *et al.* [101] deployed grid-based PRM for search and rescue in an earthquake situation. PRM is widely used to optimize path and obstacle avoidance by combining a search algorithm such as the A* algorithm [97]. In [102], the collision-free path and optimal sequence path between the measurement position of an industrial robot are generated based on PRM and A* algorithm, respectively. The simulation result shows that the proposed algorithm could reduce the cycle time by adding a TSP solver. However, the PRM method limits the robot coverage area near the boundaries and obstacles due to the random placement of nodes. The PRM also removes the corresponding nodes and edges when an obstacle collision occurs. Besides, the PRM may lead to high complexity and computation time despite the advantage of probabilistic completeness with massive nodes.

2) RAPIDLY EXPLORING RANDOM TREE

The RRT algorithm is an efficient search planner by using an incremental technique in tree structure form to construct a graph to search and explore in the free or obstacle configuration space [103]. The algorithm is designed to search in high dimensional spaces effectively and handle kinodynamic planning. The RRT is faster than PRM for a single-query problem because the algorithm does not require a sampled configuration to build a roadmap during the learning phase [1034]. Zaheer *et al.* [105] analyzed that the RRT has better performance in terms of computation time and has a better smooth path compared to PRM. Meanwhile, [106] proposed a bidirectional search approach between the initial and goal trees to rapidly grow towards each other, making a connection on both trees to generate the shortest path for uniform searching. However, the generated path based on RRT is not optimal in solving the planning problem. The modified variant of RRT called RRT* can improve path quality by providing an asymptotically optimal solution [107]. Englot and Hover [35] presented a CPP based on the sampling-based approach to solve both coverage sampling and multi-goal planning problems independently. The first coverage sampling problem determines the minimal set of views that provide guaranteed coverage. Then, the multi-goal planning problem addresses a shorter tour that visits all the views. The approach asymptotically finds the globally optimal solution to improve the feasible coverage path by using the RRT* algorithm. Similarly, [108] proposed a rapidly exploring random tree of trees algorithm to find the optimal coverage path for real-time 3D reconstruction. A meta-tree structure contains multiple sub-trees, and each sub-tree grows according to its own RRT* planner for every number of iterations to provide full visibility. However, the algorithm requires a large memory to store notes in the tree, leading to high planning costs. Hence, an optimal CPP algorithm is utilized based on two-scale algorithms to produce the shortest coverage path by reducing memory requirement [109]. The multi-directional fixed nodes RRT* algorithm is developed to generates a minimum cost trajectory planning for each point of interest (POI) from a given initial point to a goal point by exploring the neighborhood. Then, the GA is used to find the shortest path to visit a sequence of POIs by dealing with the problem of TSP, following a return to the initial point. Similarly, [110] utilized an incremental random inspection roadmap search to optimize the number of POIs in the constructed graph. The tree is iteratively generated based on RRT, constructing the roadmap that induces the subset of the POIs. Then, it computes the shortest path to cover the POIs with a suitable graph search algorithm. The results [109], [110] show that the approach can minimize coverage planning time by limiting the size of memory (number of nodes in the tree). Faghihi *et al.* [111] introduced a random kinodynamic inspection tree (RKIT) algorithm, integrating the CPP problem and kinodynamic planning problem. In the 3D model structure, the starting point and goal point are located at the center of the front and back faces, respectively. Then, the structure is remodeled in which several hypothetical cubes are developed where the size of the cubes in respect to the dimension of the front (or back) face and sensor coverage. The path-creating module computes the intermediate points that refer to the critical points (outward spiral path, helix spiral path, and inward spiral path). Finally, the coordinate of the intermediate point on a given area is taken to perform sampling by RKIT in every iteration. The algorithm also utilized a steering function to deal with differential constraints effectively. Hence, the authors proved that the algorithm successfully identifying a feasible coverage plan in 3D structure. Nevertheless, the research study does not involve the simulation result in the presence of static and dynamic obstacles.

The recent development of the RRT* algorithm has realized a breakthrough in terms of searching time and path cost (shorter and smooth path). However, fewer related studies tackle the narrow passage problem when the robot is performing the coverage task. Therefore, the robot moves through a narrow unstructured environment cluttered with obstacles using RRT* variant (to optimize the area coverage in near difficult regions) would be an interesting research endeavor in the future.

3) VIEW PLANNING AND MOTION PLANNING

Apart from the sensor-based planning method [112], [113], the sampling-based view planning approach [114], [115] is another solution for solving the optimization problem, requiring both view planning and motion planning tasks [116], [117]. View planning mainly applies to modeling applications and exploration tasks [118]. The sensors are crucial to enable the robot vision system to handle the viewpoint planning problem and CPP problem for target covering. The SCP and TSP solve the minimal set of viewpoints to cover the whole target structure and the viewpoints, respectively [119], [120]. Then, the variant of planning algorithms solves the coverage planning problems, i.e., greedy strategy, optimal strategy, or decompose planner [36]. In addressing the online CPP problem, most studies utilized the next-best-view (NBV) approach [121] for solving suitable view selection in which the viewpoint is planned based on the current robot location and the information acquired from the sensor. The robot onboard sensor explores and senses the target region before the planner generates the viewpoint to reconstruct the structure model [122], [123].

Meanwhile, [115] proposed a structural inspection planner (SIP) by implementing the Lin-Kernighan-Helsgaun (LKH) algorithm [124] to optimize the tour of viewing poses. Palomeras *et al.* [125] introduced the NBV planner by using probabilistic analysis for utility calculation. Osswald *et al.* [126] used the inverse reachability map combined with the NBV algorithm to improve robot poses and viewpoint planning problems by filtering possible view candidates. Ardiyanto and Miura [127] presented a visibility coverage based on polygon search using skeletonization technique to generate coverage viewpoints and improve the viewpoint planner further to minimize the energy consumptions of the robot's movement, thus, maintaining visibility of a moving target [128]. However, the robot may fail to track if an occlusion occurred.

The sequential viewpoint is part of the viewpoint planning problem as well, requiring modeling of information gain in a 3D environment, such as voxel [129] or surface mesh [130] in the NBV planning. Wu *et al.* [131] proposed the learning-based NBV to compute an optimal viewpoint by estimating a set of voxels for planning the next scan following the ray casting along the voxels. The inverse kinematic solver computes collision avoidance as well as finding a good sequence of the viewpoint by using the calibration of relative position between the onboard sensor and viewpoints [131], [132]. Mansouri *et al.* [133], [134] utilized the structure from motion method to reconstruct the target region, generating high-quality cover map 3D data. This method highlighted the cost-effectiveness compared to laser or range scanning. Meanwhile, [135] presented multi-view cameras based on structure from motion in CPP. Meng *et al.* [136] constructed 3D models using the probabilistic volumetric map based on Octomap structure [137] and the information gain could select the frontier viewpoints for solving the variant of TSP. Paratama *et al.* [138] proposed a search space CPP algorithm to maximize the information gain of the waypoints and calculate the entropy of each waypoint based on Octomap in the heuristic cost function. The experiment results showed that the proposed algorithm could provide a higher coverage percentage as compared to SIP, LKH with RRT, and LKH with Euclidean heuristic methods.

Most research focuses on large unknown search space without looking at less informative areas, leading to inaccurate and incomplete structure models, disregarding global path, and results in long path overlapping. Hence, most researchers studied the receding horizon planning approaches, including NBV planner and exploration planner, utilize the RRT or RRT* algorithm to explore an unknown environment [122], [139]–[141]. The optimization process repeats in the next iteration in such a way that, only the first viewpoint is executed, and the path is selected based on the best viewpoint. However, the robot often falls into dead-end sub-optimal traps due to the limited look-ahead sensing for a fixed horizon. Thus, Jung *et al.* [142] introduced a multi-layer CPP technique, dividing the 3D model structure into several layers and resample viewpoints in each layer to obtain the local path, following all the layers connected for global coverage. Oleynikova *et al.* [143] introduced an online local re-planning to maximize exploration gain by deploying an intermediate goal selection strategy. Providing a collision-free path in exploration in an unknown indoor environment with narrow and large-scale space is challenging. Thus, [144], [145] presented the combination of local and global exploration techniques by utilizing a sampling-based algorithm and frontier exploration. Similarly, Almadhoun *et al.* [146] proposed a switching approach between the frontier and adaptive grid viewpoint generators to enhance the qualities in terms of local minima avoidance and utility function. However, high coverage density in a particular area increases the traveled cost. Thus, Schmid *et al.* [147] studied the potential influence of information gain and cost formulation on tackling the balance between the gain and cost in the utility function. To improve the completeness of the target coverage, [147], [148] introduced an informative sampling algorithm to maximize the utility value in terms of global coverage and trajectories by using an online approach, reducing the sampling range by employing a streaming set cover algorithm.

Furthermore, Jing *et al.* [149] proposed a novel CPP framework, including viewpoint generation, path primitive generation, visibility estimation, primitive coverage graph encoder formulation, and coverage graph search. The computation of an iterative adaptation of uniform could provide full coverage by generating viewpoint in high fidelity mesh model following point-to-point connecting based on RRT* [150]. The Voronoi-based re-meshing algorithm down-samples the mesh model of the structure to improve the inspection path with guaranteed coverage. Glorieux *et al.* [15] presented a targeted viewpoint sampling strategy by combining both SCP and TSP. The self-adaptive differential evolution algorithm could optimize the best next viewpoint, following the implementation of RRT for collision avoidance. The results showed the reduction of inspection cycle-time and travel costs by up to 23.8% and 22.7% as compared to the greedy approximation method. However, most of the existing sampling algorithms cannot generate accurate maps in high-dimensional search space. Thus, Hou *et al.* [151] use the Gibbs sampling technique (Markov Chain Monte Carlo) to produce accurate occupancy maps by decomposing the sample space using the NBV algorithm to estimate the conditional probability of each voxel for 3D surface reconstruction. The coverage ratio could be further enhanced by using the CPP algorithm as well as NBV, which could be planned in real-time to maximize the information gain by applying a Monte Carlo tree search [152].

There are many prior works concerning the optimization problem in viewpoint planning and coverage planning to improve the coverage efficiency and to ensure the quality of viewpoint planners. The high demand for high geometric accuracy also results in the high computation complexity of the algorithm. Hence, it is still challenging to have a balance between the model quality (completeness and accuracy) and the computation time. Moreover, the feasibility of real-time applications with an implementation in large-scale space is a complicated task worthy of future study.

G. GREEDY SEARCH AND GRAPH SEARCH ALGORITHMS

The greedy algorithm is the well-known heuristic approach used to solve optimization problems by constructing a solution through a sequence of choices available without changing on subsequent steps once the choice is made at every step [153]. The algorithm often looks for the best choice by making a locally optimal choice to obtain a globally optimal solution. The greedy algorithm, (i.e. Dijkstra's algorithm) is simple, easy to implement, and generally fast but the algorithm does not guarantee to find the globally optimal solution due to the short-term solution [154]. The graph search algorithms such as A* algorithm, D* algorithm, and Theta* algorithm typically combine the boustrophedon motion or spiral pattern to plan and optimize the coverage path. The search algorithm finds the shortest path between a pair of nodes in a graph to move from the current blind position to the uncovered area when the robot falls into the dead zone or encounters an obstacle, re-planning the path to identify the next position of the robot to escape the blind nodes; otherwise, the robot continually follows the zigzag or spiral path if no obstacles are detected. The tasks repeat until the ROI is fully covered. Hence, the search algorithms are important to address the CPP problem and improve search efficiency. However, it is still challenging for path searching in the large grid map due to the large computation cost.

1) DEPTH-FIRST SEARCH AND BREADTH-FIRST SEARCH ALGORITHMS

The depth-first search (DFS) or breadth-first search (BFS) is the recursive algorithm for searching the nodes based on the graph data structure [155]. Both algorithms provide good performance in terms of time complexity, but each algorithm has its drawbacks. The DFS fails in infinite depth spaces and does not guarantee to find an optimal solution (shortest coverage path), whereas the BFS consumes large memory space due to the high branching factor in the search space. The DFS optimizes the sequence path with the benefit of minimum overlapped and several turns for CPP [156]–[158]. Kabir *et al.* [159] utilized the DFS technique to create a cleaning trajectory by generating a sequence of setups. However, the robot is relatively complex with heavy computing due to the multiple degrees of freedom. Barrientos *et al.* [160] suggested a waveform planner based on the BFS technique that can be applied over the grid-based workspace to generate the coverage path with a minimum number of turns. Wang *et al.* [161] proposed a CPP method to reduce the uncovered area by employing the BFS algorithm. However, this approach causes an uncovered edge, and the robot may fail to operate in the corner. In [2], a knowledge reasoning for robot CPP combines with the BFS to avoid the dynamic obstacles under an uncertain environment, lowering repetition rate and computation time. Miao *et al.* [162] proposed a distribution technique by using sub-map decomposition and BFS methods. This technique decomposes an unknown map into several sub-areas, distributes each robot to select the nearest sub-areas to be covered by using a spiral pattern. Both DFS and BFS algorithms can effectively optimize the coverage paths in the case of a small graph.

2) DIJKSTRA'S ALGORITHM

Dijkstra's algorithm applies a generalized graph searching technique for solving a single source shortest path issue with non-negative costs for all the edges [163]. The algorithm obtains the shortest path tree by visiting vertices from the starting node according to the cost function in each neighbor vertex. Almadhoun *et al.* [164] presented an efficient path coverage by employing Dijkstra's algorithm to explore and visit all the nodes with minimum cost in an indoor environment. Yehoshua *et al.* [165] introduced a spiral STC approach to optimize coverage path, following with Dijkstra's algorithm to find the minimum weighted path. Then, an approximation algorithm builds each pair of the connected area to solve the TSP, obtaining a higher expected percentage coverage path. Cheng *et al.* [84] used Dijkstra's algorithm to calculate the shortest path between the stripe layer subgraphs (fast path searching), reducing the total action cost to achieve maximum area coverage within the strip layer in the attempt to minimize the revisited nodes. Rosa *et al.* [166] presented the task planning of a multi-robot system by using Dijkstra's algorithm with a honeybee (hexagonal) structure. Zhang *et al.* [167] improved Dijkstra's algorithm by considering the cost function of turning times and angles. Nevertheless, the search path is not optimal in terms of travel distance [166], [167].

3) A* ALGORITHM

The A* algorithm determines a neighbor vertex by estimating the cost of the path from the current vertex towards the goal according to the heuristic function [168]. The algorithm chooses the best node to find the shortest path instead of searching the whole map. The algorithm based on the cost function has been used to minimize the number of turns and reduce the processing time of the path search [169], [170]. Viet *et al.* [171] implemented CPP by utilizing the A* algorithm with a backtracking approach to obtain optimal coverage, albeit large memory is needed to store the backtracking points. Cai *et al.* [172] described the concept of the A* algorithm to search the shortest path from escaping the dead node to an uncovered area. However, it finds difficulty in covering the cells around the obstacles if the robot moves in a diagonal path. Also, the robot revisits the cell at a high overlapping rate without covering the other cells during obstacle avoidance. Thus, Le *et al.* [173] proposed a modified A* algorithm for CPP by determining the boundary waypoints and obstacle waypoints, reducing the revisiting ratio by 7.01%, and increasing the coverage ratio by 6.4% as compared to traditional A*. The A* algorithm can outperform DFS and BFS algorithm if the location of the target is known.

4) D* ALGORITHM

The D* algorithm is effective for pathfinding in a dynamic environment [174]. The algorithm is a variant of the optimal A* algorithm capable of re-planning the path by applying the cost path optimization solution when the robot encounters the obstacle. Dakulovic *et al.* [175] computed the cost value in the D* algorithm to avoid revisiting nodes and reduce the overlapping path in the path re-planning process. Maurovic *et al.* [176] implemented an active SLAM to explore a dynamic environment by modifying the D* algorithm with negative edge weights. The D* lite algorithm improved path re-planning efficiency by obtaining the information from a previous search (shorter than D*) [177]. Luo *et al.* [178] employed the D* lite re-planning algorithm as a global path planner to generate a collision-free path in an unknown environment and used the ant colony optimization (ACO) to plan the sequence of the waypoint path to address the TSP, minimizing the overall distance along the planned trajectory in exploring a terrain. In [179], an improved version of the D* lite algorithm, namely the AD* algorithm could find the optimal path through online re-planning for dynamic obstacle avoidance. In general, the D* Lite algorithm is more efficient than the A* algorithm in the path re-planning process when obstacles exist because the D* lite algorithm having previous information data during the first search but the A* algorithm needs to re-plan the path from the beginning. Thus, the selection of the algorithm is dependent on different requirements in the specific task.

5) THETA* ALGORITHM

The A* and D* algorithms discrete search methods cannot find the shortest path in continuous space since the generated paths are created by grid edges. Thus, the Theta* algorithm is based on any angle pathfinding solver [180], and the Lazy Theta* algorithm can address the limitation. The shortest path generation is based on a pair of points on a grid map that follows the vertex parent to be any vertex instead of the vertex parent having to be a neighbor of the vertex (A* algorithm). Choi *et al.* [181] presented an online CPP of the cleaning robot using the Theta* algorithm and boustrophedon motion to optimize the local backtracking path. The recalling pass knowledge determines the backtracking points when the robot reaches an ending point after performing a boustrophedon motion before planning the shortest backtracking path to the next starting point. Similarly, the cost and goal selection functions could reduce the coverage time of multi-robot CPP in an unknown environment [182]. However, the algorithm failed to generate a global optimization solution in terms of path length. In the case of 3D space, Lazy Theta* algorithm is more suitable to perform on cubic grids due to the high number of neighbors per node as compared to 2D space (square grids). Faria *et al.* [183] implemented frontier cell exploration with Lazy Theta* algorithm to explore and avoid the obstacle in the 3D Octomap framework. Meanwhile, [184] improved the efficiency of the Lazy Theta* algorithm by reducing the number of generated neighbors to reduce the computation cost with a fewer number of line-of-sight checks. Faria *et al.* [185] added the flyby sampling technique in the exploration system, including frontier and Lazy Theta* planner for global searching, CPP, and target inspection to produce a smooth path and cover the region without overlapped albeit the path length is not guaranteed to be optimal.

H. EVOLUTIONARY ALGORITHMS

Evolutionary algorithms (EAs) are based on natural or genetic evolution in which the algorithms tend to find a better solution for solving optimization problems [186]. EAs consist of variation operators (crossover and mutation) and evaluation of the fitness function. The fitness function determines the qualities of individuals' solutions by giving a corresponding score value to everyone. The calculation of the fitness function can be expressed as an objective function for solving optimization problems to minimize or maximize the value of the fitness function [187]. EAs play an important role in building genetic searching more efficiently for solving real-world CPP optimization problems in mobile robots.

1) GENETIC EVOLUTION

The GA is a meta-heuristic population-based stochastic algorithm inspired by the idea of natural laws of biogenetics [188] as well as survival and breeding of the fittest for solving search problems [189]. GA can produce a near-optimal solution to solve path planning problems rapidly with parallel processing implementation. The GA algorithm is an ideal way that has been introduced by Wang and Bo [190] to solve the TSP in CPP. Hameed *et al.* [191], [192] presented a GA by optimizing the selection of driving direction and sequence of track from the perspective of less overlap path and minimum cost. Shen *et al.* [193] used the GA to optimize the energy efficiency based on the order of path connection between multiple fields. Ellefsen *et al.* [194] employed a multi-objective planner with EA in AUV to plan a coverage trajectory for underwater surface inspection with non-dominated sorting GA to generate the collision path on purpose, establishing the planner with penalizing strategy. This method could provide a better balance in terms of coverage and energy usage compared to circling and sampling-based CPP. In [195], the computation time of the GA-based approach for TCP-CPP is faster than DP-based when the free space is decomposed into many cells. Due to the limitation of power usage and communication distance, Sun *et al.* [22] applied the GA for multiple robots to solve task allocation problems with the multi-TSP model.

The GA has a good global search capability in an area coverage but has poor stability due to large search space complexity, requiring high computation time [196]. Hence, Sadek *et al.* [197] introduced multi-objective GA combining with DP for online CPP, improving the speed of convergence toward the optimal value when a deterministic crossover process replaces the randomized crossover process in GA [198]. Batista and Zampirolli [199] described the implementation of the GA with a near-optimal sequence of CPP for pool cleaning. The double fitness function could compute the chromosome's efficiency to reduce the energy consumption of the robot. In [200], the simulated annealing algorithm and the GA algorithm could generate the global and multiple local area coverage paths, respectively. Both algorithms are processed in parallel to reduce the computation cost. The simulation result proved that the algorithm has good stability in finding the shortest path after the 37th iteration. Liu *et al.* [201] applied the optimization algorithm to combine the GA and neural network to generate a cooperative path. The GA optimizes the weights and thresholds of the neural network through the learning process, providing a 93.74% coverage rate and a 4.25% repetition rate. Still, there is room for improvement on the convergence efficiency of GA and the combination of algorithms is a very promising solution.

Differential evolution (DE) is an EA that alternative to GA [202]. In every iteration, the trial vector generation is an important step in the DE process to solve optimization problems, including differential mutation, recombination, and selection [203]. The performance is dependent on the selection of the control parameter and the mutation strategy. DE has several advantages, such as quick convergence and robustness [204]. Vesterstrom *et al.* [205] conducted the experiments over numerical benchmarks and demonstrated the DE has a better performance compared to GA and PSO. For the robot task planning problem, Xiao *et al.* [206] modified the DE algorithm by combining the roulette and multi-neighborhood operations (to solve local optimal solution), the de-crossover strategy (to increase the convergence speed), and the multi-population integration strategy (to get high computing resources). The DE optimal path model could provide good performance as compared to the shortest path model under limited energy usage. Gonzalez *et al.* [207] utilized the DE algorithm to optimize the coverage path (zig-zag path) by reducing the distance cost. The combination of DE and fast matching square could generate a smooth trajectory concerning turning radius while avoiding collision with obstacles at minimum distance cost in four different 3D environments.

2) SWARM INTELLIGENCE

Swarm intelligence is introduced by Beni and Wang [208], inspired by the collective social behavior of living organisms [209]. It refers to the collective intelligence that emerges from the cooperation of swarm agents [210]. The objective of swarm intelligence is to develop a probability-based search algorithm in optimization problems. Therefore, swarm intelligence algorithms have been used to solve global and non-linear optimization problems in the real world due to the advantage of flexible ability and high efficiency [211]. There are several classes of optimization algorithms in CPP, namely, PSO [212], ACO [213], and bee colony optimization (BCO) [214]. The CPP-based swarm intelligence algorithm utilizes particle population movements to find the shortest path or reach a target with minimum duration to provide the optimal coverage solution.

The PSO is a meta-heuristic algorithm based on the social behavior patterns of organisms involving the swarming of the natural population [215]. Lee *et al.* [216] conducted an online CPP based on PSO to provide a smooth coverage path in a high-resolution grid map. In [217], the clustering distribution factor and PSO algorithm could cover the area in each division map. Sahu and Choudhury [218] used PSO to generate a trajectory for covering the targets globally. Y. H. Lin applied single-objective PSO [219] and multi-objective PSO [220] to optimize dynamic route planning. Wang *et al.* [221] demonstrated that the CPP based on the PSO approach has less redundant coverage as compared to the cattle method. Overall, the PSO has global searchability in the initial stage, but the swarm can easily trap in local minima, leading to a slow convergence rate during the lately searching process. Couceiro *et al.* [222] used the Darwinian PSO algorithm to divide the swarm into several small cooperative swarms (sub-groups) to provide the ability for escaping locally optimal solutions based on reward and punishment mechanisms. In [223], a collection of the sampled paths feed into the PSO framework could optimize the cost function in terms of the quality and the efficiency of a coverage path. Then, the global best particle updates the particle exploration with minimal cost selected from the camera view, overcoming the limitation of premature convergence. However, the computation time is still huge on the different model sizes. Besides, the performance of the PSO algorithm has a possibility of rapid deterioration when it deals with multi-dimensional search space [224]. Thus, [225] proposed a cooperatively coevolving particle swarm optimization (CCPSO2) technique for solving large-scale optimization problems. Sun *et al.* [226] proposed a combined approach (CCPSO2 and modified GA) to find the optimal solution sensor deployment problem and solve the TSP, respectively, achieving better coverage and obstacle avoidance in all the respective sub-regions, albeit lacks experimental results.

The ACO is a probabilistic technique that bio-mimics the behavior of ants and the process of searching foods by searching the optimal path route to solve the complex optimization problem [227]. Implementing the ACO algorithm for solving the path optimization problems has several advantages, such as strong robustness [228], [229] and parallel computation [230], [231]. However, the algorithm could easily trap in the local optimum as well as slow convergence speed [232], [233]. Thus, [234] proposed an improved ACO algorithm using a pheromone updating rule to avoid trapping into the local minimum. Chibin *et al.* [235] used the ACO algorithm to optimize the coverage of the sub-area following the distance matrix. Zhou *et al.* [265] introduced an ACO algorithm by optimizing block sequence to solve TSP. Whilst [237] presented a global inspection routing optimization based on the ACO algorithm. Max-Min Ant System (MMAS) is another improved ACO algorithm to solve the local optimum problem by bounding the pheromone value between the maximum and minimum value [238]. Karakaya [239] applied MMAS for UAVs to plan the desired paths for target coverage. Tewolde and Sheng [240] compared the CPP performance in spray painting between GA and ACO algorithms and showed that the ACO algorithm can reduce the coverage path length by 13% relative to the GA algorithm. Chen *et al.* [13] improved the accuracy of the spraying path by using an exponential mean Bézier curve and trajectory optimization based on ACO or GA, further enhancing the smooth path by optimizing the trajectory on the Bézier surface [241]. Gao *et al.* [242] proposed an improved ACO algorithm to optimize the coverage performance by reducing the number of turns in multi-robot CPP in simulated 2D grid space. Ye *et al.* [12] improved the algorithm by randomly calculating the transition probability and updating the pheromone besides the acceleration factor, improving the global searchability despite the randomness of the algorithm could induce failure. Dentler *et al.* [243] utilized a waypoint follower based on ACO combined with a chaotic solution of a dynamical to enhance the coverage efficiency. However, high-risk crash scenarios might occur due to poor localization precision. Le *et al.* presented the cleaning robot (hTetro) [244] and tiling robots (hTetrakis [245] and hTrihex [246]) for CPP by using GA and ACO algorithms to reduce energy consumption. Also, each robot type can change shape to provide high efficiency of coverage in a given workspace. Han *et al.* [247] used the glider to glide through the navigation points with back-and-forth motion to cover the sea level with the ACO algorithm to find the shortest path to avoid obstacles, which is challenging with the influence of a thermocline that changed the communication radius.

The BCO is another swarm intelligence based on a bio-inspired machine learning algorithm similar to ACO and PSO. Caliskanelli *et al.* [248] introduced a pheromone signaling algorithm based on BCO [249] for multi-robot coverage as well as a hybrid BCO-ACO pheromone signaling technique to solve the loss of communication network problems in multiple robots [250]. Firefly algorithm (FA) is a nature-inspired optimization algorithm [251] that has been widely used in coverage and exploration of the unknown area, especially mine disarming tasks [252], [253]. The goal of multi-robots is to explore and cover the area for mining as well as finding the optimal path for obstacle avoidance. Palmeiri *et al.* [254] compared the performance of FA, PSO, and BCO in the coordination of the swarm robotics system in terms of energy consumption. FA also has better performance to globally cover all the nodes than the ACO algorithm, reducing the computation time by 7.2% and decreasing the coverage path length by 2.5% in the case of grid size 10×10 of dynamic sloped terrain [255]. Nevertheless, there is no significant improvement in the path length if it increases the robot density. Henrio *et al.* [256] suggested the hyper-parameters tuning based on Bayesian optimization to apply on the FA for addressing the optimization problems. The grey wolf optimizer (GWO) is one of the recent meta-heuristic algorithms that mimic the hunting behavior and social leadership of grey wolves [257], whereby alpha, beta, delta, and omega are the categories of the moving of wolves [258], [259]. Kamalova and Lee [258] used the coordinated multi-robot exploration (CME) and GWO algorithm for multi-robot exploration to achieve optimal coordination and optimize the coverage area effectively, achieving better performance compared to the deterministic CME algorithm. Although the average coverage is 97.98% in four different obstacle maps, the obstacle avoidance constraint remains a challenge. Meanwhile, [260] conducted a similar experiment based on a multi-objective GWO algorithm to demonstrate the robot coverage capability, but the robots kept revisiting the previously explored area, leading to a long executing time. Besides, the GWO algorithm finds difficulties in obtaining global optimal solutions and dealing with dynamic obstacles due to step size mechanisms. Thus, Ge *et al.* [261] improved the local optimal solution by combining GWO and fruit fly optimization algorithm. Also, Dewangan *et al.* [259] proved that the improved GWO algorithm has better exploration ability and local optimal avoidance. Kamalova *et al.* [262] implemented the global waypoints control method in frontier-based exploration to generate the frontier points that lie on the open regions of uncertainties (the sensor does not receive any transmitted signal) and create the global waypoint based on the input parameters of the array of frontier points. The GWO algorithm could estimate the next global waypoint by calculating the average of three distances from the current robot position to the frontier point positions (mean alpha points, mean beta points, and mean delta points), thus, achieving high searching actions compared to the PSO algorithm. The robot has a high capability to avoid the obstacle, although it ultimately results in long-distance traveled.

3) ECOLOGY

The ecological algorithm is a bio-inspired algorithm from nature, and it has been used in engineering and robotics as an optimization method. Invasive weed optimization (IWO) is a well-known algorithm that utilizes an ecological behavior based on the colonizing property and distribution of weed in nature [263]. IWO algorithm has better global convergence and robustness in terms of optimization search capacity [264], [265]. The algorithm transforms the weed individuals into a positive integer by an encoder to reform its population (the set of all weeds) to solve TSP problems [266]. Ghalenoei *et al.* [267] employed the discrete IWO algorithm in a centralized manner for multiple task assignments, resulting in less computation time relative to GA. Zhuang *et al.* [268] presented the local and global coverage holes' detection and healing in the wireless sensor by using IWO and DE algorithms. Sandamurthy and Ramanujam [269] proposed the CPP based on a discrete IWO algorithm with an improved 2-Opt operator for harvesting robots. The IWO algorithm optimizes the collecting path (or TSP) according to the distribution patterns of spreading invasive weed whilst the partition strategy uses the Mahalanobis distance method, effectively optimizing the path and providing maximum coverage of 76% compared to existing graph traversal techniques. The performance of the generated path could be further improved using online methods in terms of coverage.

I. HUMAN-INSPIRED ALGORITHMS

The human-inspired algorithm is one of the sub-intelligence algorithms that mimic the human brain for learning to optimize decision-making in path planning. In recent years, the algorithm has been studied in the field of exploration tasks especially addressing coverage planning in a large dynamic environment. The algorithm can avoid collision with obstacles along the trajectory but still involves the considerable computation burden and local minima problem.

1) NEURAL NETWORK

The neural network is a well-known model and one of the most important in the field of robotics. It has been widely used and applied to robot motion planning and control of the robotic system. Besides, it also plays a crucial part in enhancing the performance of CPP. Yang and Luo [270] presented a non-learning neural network-based CPP approach for cleaning robots to avoid obstacles while planning a collision-free coverage path, but the environment is assumed to be off-line. Thus, [271] proposed a biologically inspired neural network (BINN) for real-time CPP under a dynamic environment. The BINN structure gives better performance in the CPP of mobile robots since the learning process is not needed (less computation). This approach has been further improved to reduce the path planning time and provide a low overlapping coverage area [272], [273]. However, the model is not suitable for long-term online planning due to high energy consumption. Yan *et al.* [274] introduced a neuro-dynamics model in the real-time 2D grid map building that could be applied to robot coverage through the neural activity landscape, building a dynamic map and solving the CPP in an unknown environment effectively. Meanwhile, [275] presented a workspace model and guidance of multiple robots using the neural dynamics method. Although the multi-robot system increases the time efficiency of the area coverage, the system has a high deployment cost. Yang *et al.* [276] employed the BINN approach with pedestrian and obstacle avoidance strategy to optimize the collision-free CPP trajectory. Singha *et al.* [277] applied the BINN algorithm by modifying the backtracking technique to improve the computing efficiency of neural activities, overcoming the deadlock issue.

In CPP based on the BINN approach, the algorithm requires high complexity and large calculation that leads to the high cost. Besides, the robot must wait at the current blind location until the neural activity value of the deadlock is smaller than the neighboring locations (or decay) to escape from the deadlock. Consequently, low-efficiency problems may occur in the mobile robot, and it is not suitable for long-term online planning. Thus, a Glasius bio-inspired neural network (GBNN) is an improved algorithm to decrease the time taken of CPP, especially in escaping from the deadlock situation. Zhu *et al.* [278] proposed the GBNN model to deal with CPP in building the 2D grid map. Whilst [279] further built on the 3D grid map in static and dynamic environments based on the GBNN approach. Although the model has high computation cost, the robot could plan the path to cover the area under a 2D or 3D environment without collision. Sun *et al.* [280] introduced the cooperative multiple robot system using the GBNN algorithm with the centralized planning for CPP in the 2D static environment, dramatically decreasing time complexity and reducing the repeated coverage in the region by 13.4% compared to the BINN method. Kwon and Thangavelautham [281] presented the artificial neural tissue control algorithm (sparse and variable topology neural network with adaptive activation functions) to address the coverage task. The advantages of using the controller are the non-central controller and no communication between agents of limited onboard sensors. Samarakoon *et al.* [282] enhanced the area coverage by using a reconfigurable robot and compared two similar performance techniques (feed-forward neural network and adaptive neuro-fuzzy inference system). Meanwhile, [283] investigated the tradeoff between energy usage and area coverage using a fuzzy inference system. The neural network algorithm has high computation cost and time complexity, especially in focusing on CPP in a large-scale environment, which still has the potential to be optimized in the future.

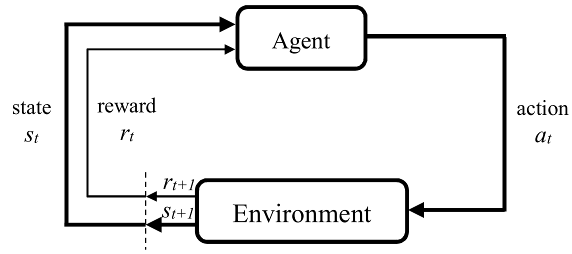

FIGURE 5. **The agent-environment interaction in reinforcement learning.**

2) REINFORCEMENT LEARNING AND DEEP LEARNING

Reinforcement learning (RL) is one of machine learning where the agent learns to reach the desired goal by dealing with sequential decision-making [284]. RL is neither supervised learning nor unsupervised learning but instead learns from the experience by trial-and-error rule. Markov's decision process (MDP) is the framework for describing RL problems. The basic concept of RL is illustrated in Fig. 5. The agent takes the possible action, by interacting with an uncertain environment under the given state, s_t at each of a sequence of time steps, t. As a result, the environment will provide feedback to the agent while changing into a new state, s_{t+1} and the agent receives the reward, r_t from the environment. By providing new data (s_t, a_t, r_t, s_{t+1}), the agent can learn to self-optimize through iterations to generate the policy, π from a training process.

RL is widely used in robotic applications [285], especially in recent CPP work. Although the classical DP can solve the optimal planning problem, it has difficulty solving large-scale Markov decision problems due to the computation of the transition probability matrices. Thus, RL has been developed to generate near-optimal solutions for solving complex and large MDPs [286]. A model-free approach based on RL has recently been successfully applied in real-world problems even if the environment model is incomplete [287]. Shakeri *et al.* [288] highlighted that RL could be utilized for CPP. Jing *et al.* [289] proposed 3D surface inspection on a production line using an MDP and ε-greedy forward tree search (FTS) method to generate an online-based inspection planning policy, proving that the ε-greedy FTS performed better than the NBV method by reducing 24% of the average cycle time among eight target models. The proximal policy optimization (PPO) algorithm is the policy gradient method that can be implemented in industrial coverage spray painting [14]. Le *et al.* [290] used the PPO algorithm based on the RL reward function to solve the TSP optimization problem in finding a low-cost path. In [291], the PPO algorithm with intrinsic rewards could provide a high coverage ratio and prevent a frequent collision. However, the coverage efficiency could decrease due to environmental change. Piardi *et al.* [292] presented a Q-learning algorithm by employing a grid map to optimize the CPP trajectory. Meanwhile, [293] deployed a distributed Q-learning algorithm for cooperative multi-agent with an information map to enhance the coverage efficiency, providing stable local optimal coverage solution in limited communication distance.

In real-world problems, the larger state-action (knowledge) space may lead to the issue of retrieving the value for all state-action pairs because the size of the table that stores related knowledge is limited. Hence, deep RL replaces the tabular function as function approximation to avoid collecting large-scale data, such as Deep Q-network (DQN) approach in mobile robot exploration and path planning [294], [295]. Sometimes, trained DQN tends to be unstable because the deep Q-learning overestimates the action value. Thus, Luis *et al.* [296] designed a double deep Q-learning CPP to perform patrolling tasks effectively. Piciarelli and Foresti [297] fed a bi-dimensional relevance map into a convolutional layer in which the network is trained by employing double DQN for optimizing the area coverage of relevant zones according to the observation. The results indicated that the RL approach is better than the zig-zag path [296], [297], but only a single form of the result is presented. Chen *et al.* [298] combined the n-step Q-learning and fitted Q-iteration without using the replay buffer to train the network for solving the CPP problem, reducing the path length and number of turns by 21.8% and 38.6%, respectively, but it is hard to deal with online CPP with the high search cost of planners.

Typically, DQN suffers from slow convergence speed and excessive randomness during training. Hence, actor-critic methods were developed to accelerate the optimization and training processes such as deep deterministic policy gradient (DDPG) algorithm and asynchronous advantage actor-critic (A3C) network. The DDPG model relies on architecture with experience replay that frequently uses each sample from the environment and separates the target network. Whereas an A3C network utilizes the gradient descent algorithm to optimize network controllers. The algorithm leverages deep learning in continuous action spaces. Based on the DDPG algorithm (the combination of policy gradient and DQN), [299] proposed multi-AUVs using the online and offline RL to perform coverage within the field of interest and the communication range. Both RL approaches have high efficiency as compared to the RW method and both have similar performance, albeit the total traveled angle of the online RL approach is more than off-line RL. The cost of exploration in a complex environment could be high, especially in dealing with obstacles. Hence, Hu *et al.* [300] enhanced the learning speed of DDPG by integrating it with a prioritized experience replay algorithm. Niroui *et al.* [301] developed the A3C network with frontier exploration to generate a robot path in an unknown map. Meanwhile, Cao *et al.* [302] used a similar algorithm with the dual-stream Q-learning technique for target search to explore the unknown environment, but the task allocation is a problem. The cleaning robot (hTetro) uses an actor-critic with an experience replay algorithm (off-policy implementation of A3C) to enhance the coverage time and energy efficiency, reducing the coverage time by 25.88% and 29.11% as compared to ACO and GA methods, respectively [303]. Kyaw *et al.* [304] addressed the TSP on decomposed cells by using a long short-term memory network (building units for layers of a recurrent neural network), slightly reducing the path length and overlapping rate. [290], [303], [304] demonstrated the efficiency of the RL approach (or deep RL approach) for finding the solution of the TSP. However, the model is only best suited for a self-reconfigurable robot in a 2D workspace, otherwise, it could significantly increase the number of turns, leading to a costly path in the conventional robot. Hence, the adaptability of the RL approach with suitable robot platforms in a dynamically changing environment is still a big challenge in robotics.

J. OTHER CLASSICAL AND HEURISTIC ALGORITHMS

There are many other classical and heuristic algorithms for exploration and CPP. The boustrophedon motion and the internal spiral algorithm are simple CPP algorithms that are commonly performed in each cell by back and forth (zigzag) pattern and spiral path. Koval *et al.* [305] presented the multi-agent exploration and coverage based on boustrophedon motion with a PRM planner. Balampanis *et al.* [306] created a Delaunay triangulation mesh model to produce coverage waypoint by utilizing the spiral pattern. Meanwhile, [307], [308] proved that the smooth spiral path has better coverage with minimal path length compared to the boustrophedon motion, but less attention to the curvature of the complex surface. A Voronoi partition approach is a common modeling technique that is applied in the distributed coordination for a multi-robot system [309]. Although the Voronoi partition-based coverage using the STC algorithm can cover the area with a non-overlapping path, prior knowledge of the environment is required to complete the task. Brick and Mortar is a heuristic search algorithm for multi-agent exploration to search and cover the area of interest. Ferranti *et al.* [310] presented the idea of using the Brick and Mortar algorithm

TABLE 1. Randomized algorithms.

Contributions (Coverage mission)	Approaches	Ref. No.	Environment modeling	Coverage	Research	Comments	
						Advantages	Disadvantages
1. Robot swarm model for searching task. 2. Unpredictable rapid search to find the explosive or unknown target (rescuing human).	Random walk (RW)	[40] [47]	2D	Off-line	Simulation	1. Simple algorithm 2. No robot localization 3. Less computation	1. Less coverage efficiency in the presence of obstacles 2. High area overlapping rate
		[41]	2D	Online	Experiment		
		[43] [45]	3D	Off-line	Experiment		
		[44]	2D	Off-line	Experiment		
	Chaotic coverage path planning	[53] [54]	2D	Off-line	Simulation	1. Fast scanning versus RW 2. Guarantee coverage in a defined region. 3. Sensitive upon initial conditions 4. No robot localization	1. Same as RW 2. High cost of coverage time
		[57]	3D	Off-line	Simulation		
		[62] [63]	2D	Off-line	Simulation		
		[65] [66]	2D, Cellular decomposition	Off-line	Simulation		

TABLE 2. Spanning tree coverage algorithm.

Contributions (Coverage mission)	Ref. No.	Environment modeling	Coverage	Research	Comments	
					Advantages	Disadvantages
A 'plan and go' based coverage technique, where computes a spanning tree and the robot visits each cell once in a configuration space without backtracking search.	[71] [76]	2D, Cellular decomposition	Off-line	Simulation	1. The cells or grids are guaranteed completely covered 2. Better coverage performance in the presence of the arbitrary obstacles 3. Random initial positions for the robot placement	1. Hard to cover the free space less than four-time of the size of the robot 2. Slightly time-consuming when concerning the number of turns of the robot
	[72]	2D, Grid-based	Online	Simulation		
	[73] [75]	2D, Grid-based	Off-line	Simulation		
	[74]	2D, Hierarchical quadtree	Off-line	Simulation		
	[77] [78] [79]	2D, Cellular decomposition	Online	Experiment		

by thickening the block of visited or wall cells without losing connectivity of explored or unexplored cells. The algorithm marks the visited cells provided that the latter does not block the path between two cells, either explored or unexplored cells in the neighborhood. The algorithm shows better performance in terms of speed and coverage. However, the algorithm might stop executing because the agents strictly avoid the visited terrain instead of finding a way to visit unexplored areas. Becker *et al.* [311] used a multi-agent flood (MAF) algorithm to explore unknown terrain by finding the point of interest. Blatt *et al.* [312] combined the wavefront frontier detection algorithm with the MAF algorithm to increase searching speed as well as using the Bug2 algorithm with edge following technique to bypass the obstacle and find the frontier points along the straight line from the start position to the end position.

Xiao *et al.* [313] proposed an improved CPP method to overcome the drawbacks of hierarchical clustering and iterative self-organizing field planning algorithms in terms of computation and overlap rate. The local search and the cost path could be improved by utilizing the nearest neighbor insertion algorithm and variable neighborhood strategy. Meaclem *et al.* [314] and Ding *et al.* [315] used the k-means clustering method and the density-based spatial clustering algorithm, respectively to partition the regions and assign the robots in each region for area coverage. Azpurua *et al.* [32] segmented the environment into sub-hexagonal cells and divided them into sub-regions by the k-means algorithm. Although the robot can execute the planned path, the wind disturbance could significantly influence the robot's performance. Tang *et al.* [316] used CCPSO2, k-means clustering with a feedback mechanism, and GA combined with A* algorithm for sensor deployment, area partition, and CPP. Miao *et al.* [317] proposed a map decomposition and sub-map cleaning according to the types of edge corners (concave or convex) around the boundaries of the wall and obstacles for multi-robot distribution. Each distributed robot can cover the area in different assigned tasks and cover the whole map in a large environment but lacks experiment results [316], [317].

Ma *et al.* [318] presented the CPP algorithms to deal with the area coverage issues, especially in the dead zone and obstacle boundaries. A quadtree segmentation method could build a neuron map to split the map into different levels of sub-blocks before the Hilbert curve traversal algorithm traversed each mode to obtain the path. Liang *et al.* [319] applied the path generator strategy with the Hilbert curve techniques for data collection to maximize area coverage. A supervisory control-based algorithm has also been implemented in a multi-robot system to enhance exploration efficiency [320]. Song and Gupta [321] introduced the ε^* algorithm using an Exploratory Turing Machine (ETM) to supervise the robot for performing the CPP. The waypoint initiation is based on

TABLE 3. Artificial potential field algorithm.

Contributions (Coverage mission)	Ref. No.	Environment modeling	Coverage	Research	Comments: Advantages	Comments: Disadvantages
1. A simple math model to develop obstacle avoidance strategies. 2. Avoid collision between the multiple robots (formation control).	[89]	2D	Online	Simulation	1. Quick response 2. Random initial positions for the robot placement 3. Effective in avoiding the local obstacles 4. Real-time planning	1. Easy to fall into local optimum 2. Hard to implement in a real-world situation 3. Poor to cover the area near obstacles
	[90] [93]	2D	Off-line	Simulation		
	[91]	3D OctoMap	Off-line	Simulation		
	[92]	2D, Grid-based	Off-line	Simulation		
	[94]	3D	Online	Experiment		

TABLE 4. Sampling-based planning algorithms.

Contributions (Coverage mission)	Approaches	Ref. No.	Environment modeling	Coverage	Research	Comments: Advantages	Comments: Disadvantages
1. Handle the CPP problem to find the uncovered area especially in large state space (i.e., complex, 3D, or narrow corridor). 2. Find a collision-free connection. 3. Sample a set of points of interest (POI) or best views in the environment, following the shortest path to visit all the POIs or selected best viewpoints by dealing with the problem of TSP (i.e., meta-heuristic algorithm).	Probabilistic roadmap (PRM)	[97]	3D Octomap	Off-line	Simulation	1. Full coverage modeling 2. Probabilistically completeness	1. High cost in computation 2. Hard to deal with dynamic obstacles
		[101]	3D Octomap, Grid-based	Off-line	Simulation		
		[102]	3D	Off-line	Simulation		
	Rapidly exploring random tree (RRT) and RRT*	[109]	2D, Space decomposition	Off-line	Simulation	1. Same as PRM 2. Low memory 3. Rapid search 4. Suitable in high dimensional search space	1. Strong randomness 2. Easily get stuck (fast greedy) 3. RRT* has a slow convergence speed 4. Lack of handling dynamic obstacles
		[110]	2D, Graph layers	Online (Assume)	Simulation		
		[111]	3D, Hypothetical cubes	Online	Simulation		
		[112]	3D, Grid-based	Online	Experiment		
		[114] [115]	3D, Triangular mesh	Off-line	Experiment		
	Next best view (NBV) and receding horizon NBV planning (RHNBV)	[131]	3D Octomap, Volumetric	Off-line	Experiment	1. High-quality modeling 2. Real-time exploration 3. Guarantee coverage in high dimensional search space 4. Good performance in an unknown environment	1. Relatively expensive in computation 2. Inaccurate structure models 3. May get stuck in local minima 4. Require high accuracy of the sensor for positioning 5. Long execution time for acquiring a complete view
		[132]	3D, Triangular mesh	Off-line	Simulation		
		[133] [134]	3D, Clustering	Off-line	Experiment		
		[135]	3D, Voxel discretization	Off-line	Simulation		
		[139] [140]	3D Octomap, Volumetric	Online	Experiment		
		[141]	3D Octomap, Volumetric	Off-line	Simulation		
	Combined RHNBV and frontier-based exploration (RHNBV-FE)	[136] [144]	3D Octomap, Volumetric	Online	Experiment	1. Same as NBV 2. Local minima optimization without getting stuck 3. Maximize the information gain (optimal frontier)	1. Computationally expensive (slightly lower than RHNBV) 2. Make decisions greedily in the exploration strategies [324]
		[143] [147]	3D Octomap with signed distance field	Online	Experiment		
		[145] [152]	3D, Volumetric	Online	Simulation		

multi-scale potential surfaces then forms 2D multi-level tape to enable adaptive decision-making. The algorithm forms the baseline coverage based on a resilience approach in multi-robots [24]. An implementation in a re-planning algorithm according to the game-theoretic framework, whereby each robot is supervised by a discrete event system, holding the promise of resilience if robot failure occurred. Although the ε^* algorithm has low computation complexity, the robot can trap in the local optimum. Hence, Shen *et al.* [322] deployed the onboard sensor to update the map information by using ETM with Dubins path, avoiding trapped near the obstacles.

IV. DISCUSSION AND FUTURE RESEARCH DIRECTION

The review compared the CPP technique of various algorithms and described the robot deployment methodology depending on the environment modeling involved in the CPP of a known or unknown environment. Table 1 to 7 shows the summary of the CPP methods by analyzing each technique's benefits and limitations, and their main contributions in tackling the coverage tasks. Notably, most studies have been conducted in the simulated environment, if not deployed in off-line mode due to the constraint within the respective field of research, such as hardware platforms and environmental conditions. Hence, some of the researchers have made assumptions for online deployment in a dynamic environment. However, existing works still lack a robust solution for inefficiency, unreliability (task execution), and instability in the real environment. Some of the CPP algorithms are not well developed, leading to poor optimization for coverage efficiency and obstacle avoidance. Table 8 listed the detailed

TABLE 5. Greedy search and graph search algorithms.

Contributions (coverage mission)	Approaches	Ref. No.	Environment modeling	Coverage	Research	Comments: Advantages	Comments: Disadvantages
1. Plan the shortest return path from the target source back to the initial position after the completion of the coverage. 2. Find the shortest backtracking path from the current blind position to the next uncovered region when the robot falls into the dead zone (or detects an ending point). 3. Re-plan the path from the current position to the newly uncovered area when the robot encounters a static or dynamic obstacle. 4. Find the charging station with the shortest path and return to continuously cover the area. 5. Distribute the robots to select the nearest target area.	Depth-first search (DFS)	[157]	2D, Cellular decomposition	Off-line	Simulation	1. Simple to implement 2. Optimal searching to find the hidden target (uninformed) and all pathways in an unknown maze environment	1. BFS consume large memory 2. Not guarantee to find an optimal solution (shortest path) in a large search space and fail in unbounded depth
		[158]	2D, Morse cell decomposition	Off-line	Simulation		
		[159]	3D	Online	Experiment		
	Breadth-first search (BFS)	[160]	3D, Grid-based	Off-line	Experiment		
		[161]	2D, Cellular decomposition	Online	Experiment		
		[2]	2D, Grid-based	Off-line	Simulation		
		[162]	2D, Cellular decomposition	Online	Simulation		
	Dijkstra's algorithm	[164]	2D, Grid-based	Online	Simulation	1. Sub-optimize in the closest path (when there is multiple target area) 2. Greedy expand to cover a large area of the graph	1. Large memory 2. Fail on the negative edge weights 3. Performance degradation over the large distance
		[165]	2D, Cellular decomposition	Off-line	Simulation		
		[166]	3D, Hexagonal cell pattern	Online	Simulation		
		[167]	2D, Sub-cell decomposition	Off-line	Simulation		
	A* algorithm	[168] [170] [172]	2D, Grid-based	Off-line	Simulation	1. Least cost of computation time 2. High efficiency to deal with a single target destination	1. Not guarantee to provide the shortest path (if overestimating) 2. Hard to handle the dynamic environment
		[171]	2D, Cellular decomposition	Online	Simulation		
		[173]	2D, Grid-based pre-built map	Off-line/ Online	Simulation		
	D* algorithm and D* lite	[175]	2D, Cellular decomposition	Online	Simulation	1. Capable of handling in the dynamic complex environment 2. Fast re-planning	1. Hard to implement in a large number of obstacles 2. High cost in large search space
		[176] [179]	2D, Grid-based	Online	Experiment		
		[177]	2D, Grid-based	Online	Simulation		
		[178]	2D, Cellular decomposition	Online	Simulation		
	Theta* algorithm and Lazy Theta* (or any angle path planning Theta*)	[181]	2D, Cellular decomposition	Online	Experiment	1. Shortest collision-free path 2. Smooth post-processing and speed up planning [183, 184] 3. Lazy Theta* is capable of finding the shortest path in the continuous space	1. High computation time in large search space (environment complexity) 2. Ignore the kinematic constraints of the robot
		[182]	2D, Binary cell map	Online	Simulation		
		[183]	3D, Regular and sparse grid	Online	Experiment		
		[184]	3D Octomap, Sparse grid	Online	Experiment		
		[185]	3D Octomap, Sparse grid	Online	Simulation		

descriptions of the technical properties of the motion planning problem, whilst Figure 6 shows their respective features comparison of different algorithms in a typical grading scale. Table 9 shows computational complexities in big O-notation by analyzing each kind of algorithm. Table 10 illustrates a performance comparison of the seven algorithms regarding the coverage efficiency, optimization criteria, and future trends.

Randomized algorithms (e.g., random walk and chaotic CPP) are well-known for random or unpredictable trajectories in the motion plan. They are widely used in low-end swarm robotics without the need for map information, effectively searching and exploring an unknown environment. They provide a very simple random motion, running in $O(\log n)$, which only records the current vertex, n, and count the number of steps taken. Some works (i.e., the searching efficiency in terms of step length, the number of visited cells, and coverage time) have been considered key aspects in futures steps.

In any case, the STC algorithm could optimize a covering path in each area, addressing the single robot coverage problem and provide the least coverage repetition to cover all accessible grids. Other improvement methodologies such as spiral STC, full-STC, and smooth STC could achieve a maximum coverage rate over the original STC. Those STC methodologies compute the coverage path in linear time, $O(n)$, where n is the number of grid cells (sub-cells). An extension of multi-robot STC follows a proper selection from various cellular decomposition techniques to shorten the coverage time in a large area. STC is simple, responsive to change in the environment, but only suitable to operate under no circumstances of the dynamic obstacles due to the path generated is predetermined. The new spanning tree could be

TABLE 6. Evolutionary algorithms (Metaheuristic).

Contributions (coverage mission)	Approaches	Ref. No.	Environment modeling	Coverage	Research	Comments	
						Advantages	Disadvantages
1. Optimally assign a group of robots to execute the number of tasks in cooperation to achieve the overall system goals (multi-robot task allocation problem). 2. Generate an optimal coverage sequence (shortest path) in each sub-region (to solve the TSP optimization problem). 3. Optimize the objective (or multiple objectives) to minimize cost functions (i.e., coverage path length, obstacle avoidance, energy consumption, path smoothness, and vehicle kinematic constraints).	Genetic algorithm (GA)	[190] [195]	2D, Cellular decomposition	Off-line	Simulation	1. Global search capability 2. Provide a smooth and time-optimal path 3. Support parallel computing or multi-objective optimization 4. Effective in solving time extended task allocation	1. Poor stability 2. High computing power and computation time 3. High model complexity 4. Uncertainty convergence speed
		[192]	2D, Clustering	Off-line	Experiment		
		[193]	2D, Grid-based	Off-line	Simulation		
		[194]	3D, Triangular mesh	Off-line	Experiment		
		[22]	2D, Grid, and segmentation	Off-line	Experiment		
		[197] [200]	2D, Cellular decomposition	Online	Simulation		
		[199]	2D, Non-Euclidean grid	Off-line	Simulation		
	Differential evolution (DE)	[206]	3D, Grid-based	Off-line	Experiment	1. Same as GA 2. More effective than GA	1. Same as GA 2. Many parameters adjustment
		[207]	2D, Cellular decomposition	Online	Experiment		
	Particle swarm optimization (PSO)	[216]	2D, Grid-based	Online	Simulation	1. Easy to implement 2. Global and local searchability 3. Few parameters can be adjusted to obtain the global best solution	1. Premature convergence 2. Easily trap in local minima 3. Slow convergence speed in the late phase
		[217]	2D, Grid-based	Off-line	Simulation		
		[220]	3D	Online	Simulation		
		[221]	2D, Irregular polygon region	Off-line	Experiment		
		[222]	2D	Off-line	Simulation		
		[223]	3D, Triangular surface	Online	Simulation		
	Ant colony optimization (ACO)	[235] [237]	2D, Cellular decomposition	Off-line	Simulation	1. Global optimization 2. Distributed computation to avoid premature convergence 3. Fast convergence speed in the late phase 4. Effective in solving TSP 5. Optimal task allocation	1. Uncertain time to convergence and slow convergence speed in the initial speed 2. Easily trap in local minima 3. Poor performance in the large search space
		[236]	2D, Cellular decomposition	Online	Experiment		
		[13]	3D, Triangular surface	Off-line	Experiment		
		[242]	2D, Grid-based DARP	Off-line	Simulation		
		[12]	2D, Mesh division	Online	Experiment		
		[243]	3D, Grid-based	Off-line	Simulation		
		[247]	3D	Online (Assume)	Simulation		
	Firefly algorithm (FA)	[252] [253]	2D, Grid-based	Online (Assume)	Simulation	1. Local searchability 2. High efficiency in solving TSP 3. Fast convergence speed with parameter tuning	1. Low accuracy 2. Easily to fall into local minima
		[254]	2D, Grid-based	Online (Assume)	Simulation		
		[255]	2D, Grid-based	Online	Simulation		
		[256]	2D, Grid-based	Online (Assume)	Simulation		
	Grey wolf optimization (GWO)	[258] [260]	2D, Grid-based	Online	Simulation	1. Derivation-free mechanism 2. Free from the initialization of input parameters 3. Optimal search capability	1. Slow convergence 2. Poor local searchability
		[259]	3D, Grid-based	Online	Simulation		
		[261]	2D	Off-line	Simulation		
		[262]	2D	Online	Experiment		
	Invasive weed optimization (IWO)	[267]	2D	Off-line	Simulation	1. Simple to implement 2. Global search capability 3. Well-adapted to change in the environment	1. Premature convergence 2. Easily trap in local minima 3. Poor exploitation ability
		[268]	2D, Multi-constrained circumstances	Off-line	Simulation		
		[269]	2D, Grid-based	Off-line	Simulation		

constructed based on the remaining uncovered grid cells by using a path re-planning algorithm to cope with dynamically changing in the environment, yielding additional computation time.

APF algorithm is a simple calculation that provides fast planning speed of obstacle avoidance path by building a model according to attraction and repulsive forces analogy. APF does not need global information; thus, the robots can effectively avoid the obstacle in real-time and coordination control of multi-robot. However, the robots can easily fall into the local optimum if large or arbitrary obstacles approach the target point. This is due to no movement takes place when

TABLE 7. Human-inspired algorithms (Neural network).

Contributions (coverage mission)	Approaches	Ref. No.	Environment modeling	Coverage	Research	Comments: Advantages	Comments: Disadvantages
1. Obtain knowledge through the exploration of the environment via sensors to learn a model of the environment in the coverage task. 2. Embed in a self-organizing map to assign the next robot location (or the robot takes an action) according to the neural activity value (or action value). 3. Provide higher objective functions on visual coverage, covering the AOI, avoid the obstacle or select the best backtrack point. 4. Optimize the SCP or TSP solution in an extremely large workspace.	Biologically inspired neural network (BINN)	[270] [271]	2D, Cellular decomposition	Online	Simulation	1. Simple structure 2. No learning process 3. Good real-time performance 4. Good in dealing with an unknown static and dynamic environment 5. High efficiency to avoid obstacles	1. Issue of overconfidence 2. Time-consuming to escape from deadlock 3. Slightly less efficient in a fast-changing environment
		[272]	2D, Fictitious frontier	Online	Simulation		
		[273] [274] [275] [276] [282]	2D, Grid-based	Online	Simulation		
		[277] [281]	2D, Grid-based	Off-line	Simulation		
	Glasius bio-inspired neural network (GBNN)	[278]	2D, Grid-based	Online	Simulation	1. Same as BINN 2. High efficiency to escape from deadlock instead of waiting for the neural activity process	The model is slightly less efficient in a fast-changing environment
		[279]	3D with 2D planning, Grid-based	Off-line	Simulation		
		[280]	2D, Grid-based, Voronoi	Online	Simulation		
	Deep reinforcement learning	[294]	3D	Online	Simulation	1. Completed independent from human labeling (strong self-learning) 2. Parallel computing 3. Fast training speed [300, 301, 302] 4. Good in dealing with an unknown static and dynamic environment 5. High efficiency to avoid obstacles	1. Slow convergence speed in the training phase 2. Sparse reward problem 3. Overestimate the action value [294] 4. High computation cost 5. Long learning time 6. Hard to deal with continuous state space
		[296]	2D	Off-line	Simulation		
		[297]	2D	Online	Simulation		
		[298] [303] [328]	2D, Cellular decomposition	Off-line	Simulation		
		[299]	2D, Field posterior distribution	Online	Simulation		
		[300]	2D, Grid-based	Online	Simulation		
		[301]	2D, Grid-based	Online	Experiment		
		[302]	2D, Grid-based	Off-line	Experiment		

TABLE 8. Detailed description about the properties of motion planning problem.

Types	Features	Descriptions	Grading scale 1	2	3	4	5	Best grade
A	Searching time	Rapid search to find unexplored area or target	Very slow	Slow	Moderate	Fast	Very fast	5
B	Path optimality	Shortest coverage path or sequence	Poor	Fair	Average	Good	Excellent	5
C	Dynamic performance	Capable of handling in a dynamic environment	Poor	Fair	Average	Good	Excellent	5
D	Computational complexities	Time and space complex calculation	Very low	Low	Medium	High	Very high	1
E	Convergence speed	Rate of convergence of a sequence	Very slow	Slow	Moderate	Fast	Very fast	5

the amounts of repulsive and attractive forces, acting on the robot are equal. Besides, the planned path is not an optimal path and the adaptability in handling dynamic obstacles is relatively poor, leading the robots to easily collide with moving obstacles. Although the APF approach only validates local obstacle avoidance and is hard to meet high-speed robots' requirements, it is still most suitable for low-end swarm robots by combining with the randomized algorithm.

DP is a classical exact-based approach to solve the TSP optimization problem. It guarantees to choose the best solution within an acceptable time in finding the global optimum. Nevertheless, the time complexity increases to address the largest tour, leading to high computation power. Hence, approximation approaches had gained attention to solve large-scale TSP, i.e., metaheuristic evolutionary algorithms. Evolutionary algorithms were proven to be effective to deal with single or multiple objective optimization problems. For example, GA often finds the best solution to address the combinational optimization problems (i.e., task allocation). PSO requires few parameters and takes less time to reach the target with computationally simple. On this account, it is sensitive to control parameters, directly influencing the performance. Whereas, the ACO algorithm has high efficiency in finding the shortest TSP but is not practical in performing real-time planning due to large memory to store in a pheromone matrix. FA achieves fast convergence speed and simplicity due to minimum parameters adjustment. Albeit the metaheuristic algorithms have robust global or local searchability, they tend to fail into local minima.

Recent trending includes hybrid algorithm (the combination of local search heuristic and evolutionary algorithm, i.e., 2-opt algorithm and IWO [269] or two heuristic algorithms, i.e., GA and PSO [226]) in optimizing the CPP solution. The hybrid algorithm incurred a high computation time but

TABLE 9. Computational complexities of each type of algorithm.

Algorithms	Time complexity, C_T	Space complexity, C_S	Grid size
Randomized	$O(\log n) \le C_T \le O(n)$	$O(\log n) \le C_S \le O(1)$	$n \times n$
STC	$O(n)$	$O(n) \le C_S \le O(1)$	$n \times n$
APF	$O(\log n) \le C_T \le O(n)$	$O(\log n) \le C_S \le O(n)$	$n \times n$
DP	$O(n^2 * 2^n)$	$O(n^2)$	$n \times n$
PRM	$O(n \log n) \le C_T \le O(n)$	$O(\log n) \le C_S \le O(n)$	$n \times n$
RRT/RRT*	$O(n \log n) \le C_T \le O(n^2)$	$O(\log n) \le C_S \le O(n)$	$n \times n$
RHNBV	$O\left(n \log n + n \log\left(V / r^3\right)\left(NM / r^4 + 1 / r^3\right)\right)$	$O(n) \le C_S$	V
RHNBV-FE	$O\left(n \log n + nV\left(NM / r + 1 / r^3\right)\right)$	$O(n) \le C_S$	V
DFS	$O(m * n)$	$O(m * n)$	$m \times n$
BFS	$O(m * n)$	$O(m + n)$	$m \times n$
Dijkstra	$O(n \log n) \le C_T \le O(n^2)$	$O(\log n) \le C_S \le O(n^2)$	$n \times n$
A*	$O(\log n) \le C_T \le O(n^2)$	$O(\log n) \le C_S \le O(n^2)$	$n \times n$
D*/D* lite	$O(\log n) \le C_T \le O(n^2)$	$O(\log n) \le C_S \le O(n^2)$	$n \times n$
Theta*/ Lazy Theta*	$O(n^2 \log n) \le C_T \le O(n^3 \log n)$	$O(n) \le C_S \le O(n^2)$	$n \times n$
Evolutionary	$O(n^2) \le C_T$	$O(n) \le C_S$	$n \times n$
BINN/ GBNN	$O(n^2)$	$O(n) \le C_S \le O(n^2)$	$n \times n$

TABLE 10. Comparison of various coverage path planning algorithms: Performance and analysis.

Performance metrics	Types of coverage path planning algorithms						
	Randomized algorithms	Spanning tree coverage	Artificial potential field	Sampling-based planning	Graph search algorithms	Evolutionary algorithms	Human-inspired algorithms
Fast searching time	✓		✓				
Collision avoidance			✓	✓	✓	✓	✓
Complete coverage		✓		✓			✓
The shortest path between two points				✓	✓		✓
Non-backtracking		✓					
TSP optimization						✓	✓
Large-scale structural coverage (SCP optimization)				✓			✓
Good real-time performance	✓		✓	✓			✓
Low computation cost	✓	✓	✓		✓		
Dynamic environment			✓	✓	✓	✓	✓
Global path		✓		✓	✓	✓	✓
Local path	✓	✓	✓	✓			
Experimentally sufficient	✓	✓			✓		
Maturity level	✓	✓	✓	✓	✓	✓	
Potential future research				✓	✓	✓	✓

delivers better coverage efficiency in terms of TSP optimization. The selection of best-suited hybrid algorithms from various metaheuristic algorithms for a specific CPP problem is still uncertain due to lacking benchmarks or satisfactory solutions.

In graph theory, search algorithms are probably the most widely used in shortest path finding between two nodes. BFS and DFS algorithms are the basic graph search techniques to get the shortest path due to their blind search strategy (without information about the environment). They can provide better searching in small problems but are often inefficient in terms of time and memory. Their time complexities are $O(m^*n)$, where n is the number of vertices (nodes), and m is the number of edges. In contrast, informed search algorithms, i.e., Dijkstra's algorithm, A* algorithm, and D* algorithm, are highly efficient heuristic search techniques to find the solution. Dijkstra's algorithm is a classical backtracking solution for tackling the CPP problem, a logical choice for indoor CPP implementation for a small distance when the robot escapes from the dead zone. Whereas A* and D* algorithms are the fastest approaches known so far to speed up the search in the large and complex search space, but they do not often

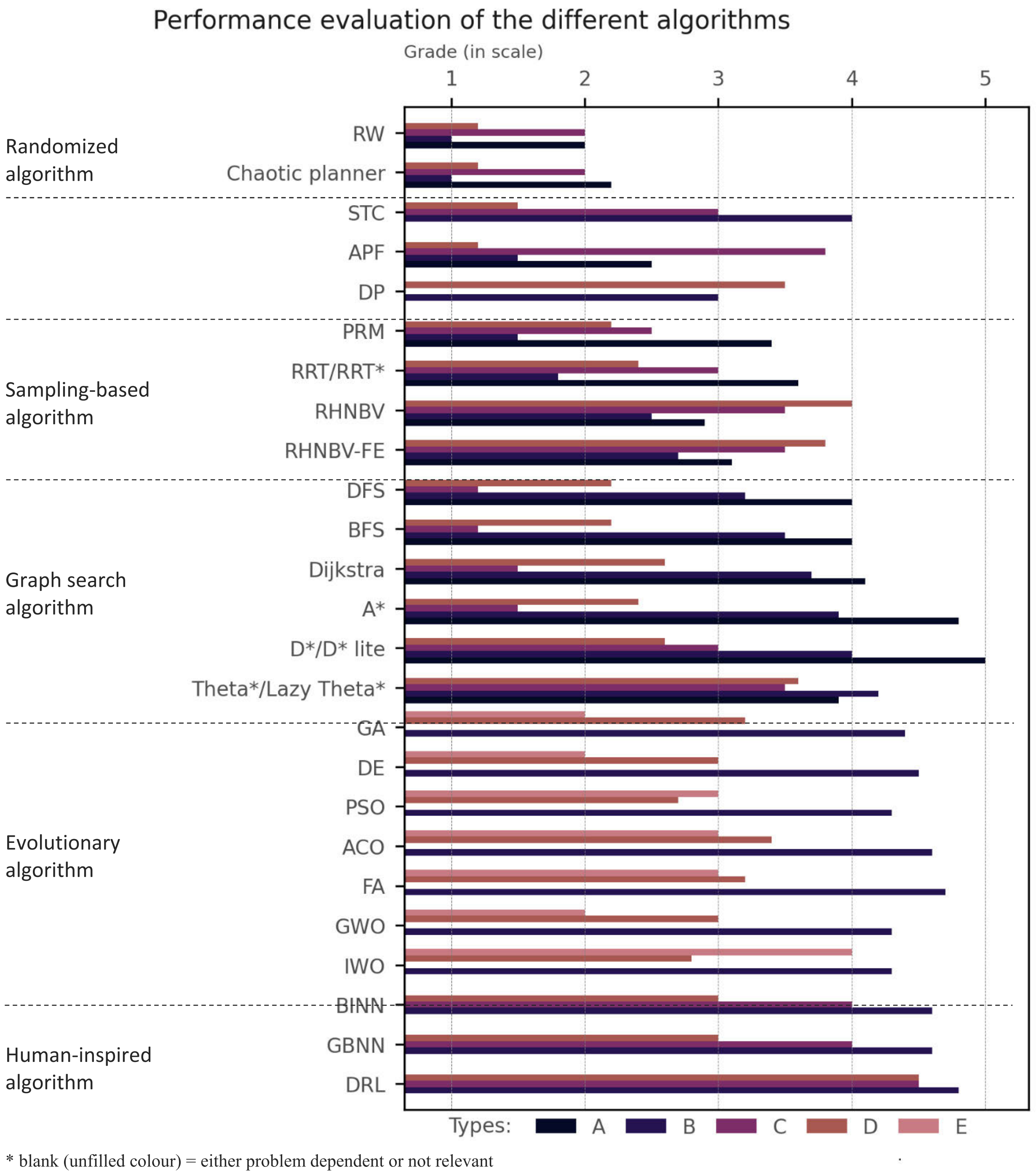

FIGURE 6. The performance evaluation of the different algorithms based on five key features (as described in Table 8).

guarantee to provide the least-cost path due to their heuristic strategy.

The time and space complexities of Dijkstra's algorithm are $O(n \log n)$ and $O(n)$, respectively, in finding the single source shortest path. In most cases, the $O(n^2)$ is the best possible solution to compute the shortest distance for all pairs of vertices for dense graphs. The complexities of A* and D* are highly dependent upon their heuristic functions (estimate the cost from the given vertex to goal), reducing the complexity to a lower degree, i.e., $O(\log n)$, which enable for online implementation, yielding to $O(n^2 \log n)$ or $O(n^2)$ if utilizing a binary heap to implement the priority queue. The only difference between them is the capability to meet the requirement of mobile robots in a dynamic environment: A* relays on the nodes with the lowest value of the summation of the cost path from the start node to any given vertex and the heuristic function, whereas D* relays on the nodes with the lowest value of estimated cost by comparing the goal node and current node (A* - forward search; D* - backward search). Thus, the D* algorithm has a better solution to address a complex problem, i.e., dynamic environment, as it can handle this situation based on updating the reverse search process (incremental search) to re-plan the path.

Alternatively, D* lite (based on lifelong planning A*) is preferable as it is simple to implement (shorter than D*), utilizing one tie-breaking criterion when comparing priorities (simplifying maintenance). However, the complexity of the D* or D* lite could dramatically increase when the search space is relatively large due to many re-planning executions. Also, an unrealistic distance could be produced if there is a lot of moving obstacles. Overall, the A* algorithm has a high search efficiency in a static environment (i.e., shortest backtracking path for a mobile robot). Whereas D* lite algorithm is better suited for dealing with changes in obstacle features (i.e., industrial robotic manipulator for inspection).

In most cases, the paths are constrained to the edges in dealing with a discrete grid-based map (regular patterns), leading to the generated path is not being the best shortest path. The theta* algorithm overcomes this shortcoming with an any-angle search method based on the utilization of a line-of-sight check (LoS-Check). It is best suited for large-scale coverage in an unknown environment, mainly deployed by holonomic aerial robots to find the next starting point since the planned path is fast and smooth. Alternatively, sampling-based planning algorithms such as PRM and RRT could specifically deal with the motion planning problem for non-holonomic constraints. RRT algorithm (single-query planner) is preferable in solving single start and goal states, but it fails to converge to the optimal solution. Hence, the RRT*, a variant of RRT, eventually claims to reach convergence towards the optimal solution by employing local rewiring operations.

Although it is more promising to solve the shortest path problem in a significantly large search space and the unknown cluttered environment with narrow corridors, it requires additional smoothing and re-planning algorithms to follow the shortest path and avoid the dynamic obstacles, respectively. This is due to the elimination of unnecessary waypoints in the path pruning process, generating a linear piecewise path, resulting in not being feasible for a robot with kino-dynamic constraint. Similarly, theta* algorithm might deal with the same issue, it needs to further implement the post-processing technique to achieve a kinematically-feasible path. Despite both RRT* and Theta* algorithms have been improved into several variants based on LoS-Check to obtain the trade-off between the solution quality and the planning time in tackling the coverage task, there is still lacking clear solution and performance comparison of each algorithm. Due to LoS-Check and online collision checking, the time complexity of the RRT* and Theta* could reach to $O(n^2)$ and $O(n^3 \log n)$, respectively.

For high-quality structural coverage, view planning is the top priority for accurate surface modeling. Based on the previous studies' findings, the NBV planning approach could gain the most informative view that considers the unknown area from a given partial model. Nevertheless, this approach does not consider the global route of the environment, leading to the overlapping path with the previous known views that might exist. Some features like holes and sparse surfaces might get ignored, resulting in less completeness of the constructed model. The receding horizon NBV technique achieves higher performance in local exploration, but it is prone to local minima due to poor global coverage. It also requires a relatively expensive to explore in a large workspace because it tends to terminate the exploration prematurely when the robot is not closer to the nearest frontier (low-cost function). Thus, the computational complexity of this technique mainly depends on RRT tree construction, gain estimation (using ray casting) and collision checking, giving the overall complexity as $O\left(n \log n + n \log\left(V/r^3\right)\left(NM/r^4 + 1/r^3\right)\right)$, where N is the number of horizontal rays, M is the number of vertical rays, r is the map resolution and V is the volume of the environment to be explored [122]. In the current research review, the combination of fusion-based algorithms provides better solutions, utilizing various algorithms' advantages. For instance, sampling-based planning with frontier-based exploration methods could optimize local and global searchability [149], [323]. In addition, the combined receding horizon NBV and frontier-based exploration approach could reduce the computational complexity of gain estimation from inversely quartic growth to inversely linear growth, providing the overall complexity as $O\left(n \log n + nV\left(NM/r + 1/r^3\right)\right)$ [144]. There are still many limitations as the performance might be degraded due to localization drift and high computation requirement for online operation, as well as the algorithm is highly dependent on the sensor used and map resolution. Hence, there is an endless opportunity for a fusion CPP algorithm with high-quality optimization and a correct model in a real-world situation for future work.

Recently, human-inspired approaches have received more attention in addressing the CPP problem. The BINN and GBNN are the most effective techniques to deal with real-time coverage tasks as they do not need a learning process. It utilizes the neural activity landscape for generating an optimal path without prior knowledge of the environment and no explicit searching procedures in the neural network model. Thus, they have a high capability to handle an unknown static and dynamic environment. Instead of waiting for the long decay time in the neural activity process, the GBNN provides a better model in rapidly escaping from deadlock to overcome the shortcoming of the BINN. Both models could achieve obstacle avoidance in real-time and the complexity is squarely proportional to the degree of discretization, $O(n^2)$, where n is the number of neurons in the system. The BINN has also been utilized to deal with the problem of multi-robot formation control in coverage planning tasks [325]. Still, somehow the optimal path is planned close enough to the obstacles or multi-robot near-collision situation, leading to difficulty avoiding the fast-moving obstacles. The robot might eventually fail when moves along the edge of the obstacles, leaving many rooms for improvement in planning the strategy in a rapidly changing environment. Most studies assume the location of the robot is known with prior knowledge of the environment, high precision sensing, and ideal communication network due to the experimental is often involved high-cost hardware with expensive sensor and safety hazard in the workspace. Notably, the adaptability of the BINN approach to work in real-world applications is still uncertain as there is a big gap between the simulation environment and practical experimental.

More recently, deep and RL started to gain importance in addressing the CPP problem, allowing experience-driven learning to tackle real-world problems. Several studies based on RL have been made to accomplish the coverage task, i.e., avoiding collision [291], balancing the coverage ratio and energy usage [326], and is beneficial for view planning in solving SCP optimization [327]. Metaheuristic algorithm is superior in solving small workspace but can get stuck in local minima and the computational complexity exponentially increases when the workspace expands. Conversely, the deep RL approach is an alternative to solve the optimization problem under a large and complex environment, such as disinfection tasks in reducing the spread of COVID-19 in workspaces [328]. Although deep RL has relatively better performance, it is not preferable to tackle small workspaces due to the model computation complication of huge agent training time and hyperparameter tuning. Performance comparison between the deep RL approach and metaheuristic algorithms for holonomic and non-holonomic robots in solving TSP are lacking. It is challenging to use deep RL to deal with multi-robot CPP tasks in an unknown environment. Nevertheless, deep RL offers a promising future direction for addressing CPP problems despite being relatively immature.

Many CPP algorithms and methodologies have been presented in the field of robotics research. However, there revolves many constraints and technical issues awaiting to be explored and addressed. Future research should focus on the following directions:

A. COVERAGE COMPLETENESS AND TIME-EFFICIENCY TRADEOFF

The robot's number of turns dramatically influences the total coverage time. CPP techniques widely adopt back-and-forth motion due to simple path design compared to spiral motion. However, in a large-scale unknown environment, seeking coverage completeness often results in a longer path and more turning, increasing coverage time and reducing efficiency. In a 3D complex structure, the existing algorithm is limited to handling the target with hidden parts, which is considered a non-interest region and obstacle in most previous research, leading to significant time-consuming for a complete coverage plan. Therefore, the right balance between the coverage completeness and execution time is required to optimize the overall coverage efficiency.

B. ROBOT ADAPTABILITY VERSUS COST-EFFICIENCY

Dynamic environmental characteristics might influence the robot's motion and lead to unnecessary performance degradation. Robots might lack flexibility and easily get trapped in common dead-lock situations. The robot with the ability to change operating behavior over time is essential to seek a collision-free path under an unknown environment with uncertain obstacles. Besides increasing the number of onboard sensors in handling complex environments, the computation cost might be high. Evolutionary algorithms are typically not suitable in low-cost robots due to large memory requirements and computationally expensive. Therefore, the computation cost factors must be considered for designing a suitable environment model for CPP. The hybrid algorithm is an exciting development to manage the change in the environment with minimum cost.

C. PATH SMOOTHNESS

The coverage and connectivity are crucial in wireless sensor networks. With limited communication and sensing capabilities, the robot cannot regenerate the best path if the unexpected occurs, degrading the effective coverage ratio. A kinematic constraint of the robot, such as path curvature, is also one of the challenges that must be addressed. For fast-moving robots such as drones, trajectory smoothing on a sharp turn helps to provide the robot with an efficient inertia motion transfer to minimize power consumption and prevent premature mechanical damage. Hence, there is a need to project a smooth path while following the CPP route [329].

V. CONCLUSION

Comprehensive knowledge of the CPP algorithms based on classical algorithms and heuristic-based algorithms was summarized in this paper. All the elements were listed and compared by analyzing the merits and demerits of each technique. The challenges that exist in the CPP were critically evaluated, involving coverage efficiency and collision avoidance in terms of several typical features such as area coverage, path length, travel time, repetition rate, and energy usage. Most of the approaches were shown the capability of the robot to avoid obstacles effectively and cover the area in a static and dynamic environment with the highest coverage percentage and low overlapped paths. Each algorithm can perform well in practice, but still has the limitation in the CPP literature. The optimization algorithms may still not well develop in solving CPP problems. As such, the SCP, TSP, and local minima escaping problems are necessary to be tackled. The connection between local and global coverage paths could solve the integrated TSP and CPP problems. Still, it is limited to handling the target with hidden parts. The issue of adaptability in the complex unknown environment still not well solving. Deep RL has been applied in various

CPP with great achievement in recent development. However, the current RL techniques are still immature, thus, many challenges need to be addressed before carrying out the CPP in the dynamic environment. In a multi-robot scenario, issues such as robot distribution and structure of the environment should be considered for improving the efficiency of CPP. Even though multi-robot can cover the AOI collaboratively, transferring data online is still challenging [330]. In future work, the performance of the CPP could be improved by combining other algorithms to reduce the shortcoming of the existing classical algorithms. The hybrid algorithm should be the direction of CPP development. Lastly, the researchers believed that the experimental results could be conducted from real scenarios with the verification of the simulation model.

REFERENCES

[1] H. H. Viet, V.-H. Dang, M. N. U. Laskar, and T. Chung, ''BA*: An online complete coverage algorithm for cleaning robots,'' *Int. J. Speech Technol.*, vol. 39, no. 2, pp. 217–235, Sep. 2013.

[2] H. V. Pham and T. N. Lam, ''A new method using knowledge reasoning techniques for improving robot performance in coverage path planning,'' *Int. J. Comput. Appl. Technol.*, vol. 60, no. 1, pp. 57–64, Apr. 2019.

[3] I. A. Hameed, ''Coverage path planning software for autonomous robotic lawn mower using Dubins' curve,'' in *Proc. IEEE Int. Conf. Real-time Comput. Robot. (RCAR)*, Okinawa, Japan, Jul. 2017, pp. 517–522.

[4] E. Galceran, R. Campos, N. Palomeras, D. Ribas, M. Carreras, and P. Ridao, ''Coverage path planning with real-time replanning and surface reconstruction for inspection of three-dimensional underwater structures using autonomous underwater vehicles,'' *J. Field Robot.*, vol. 32, no. 7, pp. 952–983, Oct. 2015.

[5] C. Peng and V. Isler, ''Visual coverage path planning for urban environments,'' *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5961–5968, Oct. 2020.

[6] I. A. Hameed, ''Intelligent coverage path planning for agricultural robots and autonomous machines on three-dimensional terrain,'' *J. Intell. Robotic Syst.*, vol. 74, nos. 3–4, pp. 965–983, Jun. 2014.

[7] K. Wang, Z. Meng, L. Wang, Z. Wu, and Z. Wu, ''Practical obstacle avoidance path planning for agriculture UAVs,'' in *Advances and Trends in Artificial Intelligence. From Theory and Practice* (Lecture Notes in Computer Science), vol. 11606, F. Wotawa, G. Friedrich, I. Pill, R. Koitz-Hristov, and M. Ali, Eds. Cham, Switzerland: Springer, 2019, pp. 196–203.

[8] N. Basilico and S. Carpin, ''Deploying teams of heterogeneous UAVs in cooperative two-level surveillance missions,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Hamburg, Germany, Sep. 2015, pp. 610–615.

[9] D. Jia, M. Wermelinger, R. Diethelm, P. Krusi, and M. Hutter, ''Coverage path planning for legged robots in unknown environments,'' in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot. (SSRR)*, Lausanne, Switzerland, Oct. 2016, pp. 68–73.

[10] E. Galceran and M. Carreras, ''Efficient seabed coverage path planning for ASVs and AUVs,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Vilamoura, Portugal, Oct. 2012, pp. 88–93.

[11] S. Kalburgi, V. G. Nair, and K. R. Guruprasad, ''Application of coverage path planning algorithm for milling operations,'' in *Intelligent Systems, Technologies, and Applications* (Advances in Intelligent Systems and Computing), vol. 1148, S. M. Thampi, L. Trajkovic, S. Mitra, P. Nagabhushan, E.-S. M. El-Alfy, Z. Bojkovic, and D. Mishra, Eds. Singapore: Springer, 2020, pp. 213–220.

[12] X. Ye, L. Luo, L. Hou, Y. Duan, and Y. Wu, ''Laser ablation manipulator coverage path planning method based on an improved ant colony algorithm,'' *Appl. Sci.*, vol. 10, no. 23, pp. 1–19, Dec. 2020.

[13] W. Chen, J. Liu, Y. Tang, J. Huan, and H. Liu, ''Trajectory optimization of spray painting robot for complex curved surface based on exponential mean Bézier method,'' *Math. Problems Eng.*, vol. 2017, pp. 1–10, Nov. 2017.

[14] J. C. Kiemel, P. Yang, P. Meißner, and T. Kröger, ''PaintRL: Coverage path planning for industrial spray painting with reinforcement learning,'' in *Proc. Conf. Robot., Sci. Syst.*, Freiburg, Germany, Jun. 2019, pp. 1–3.

[15] E. Glorieux, P. Franciosa, and D. Ceglarek, ''Coverage path planning with targetted viewpoint sampling for robotic free-form surface inspection,'' *Robot. Comput.-Integr. Manuf.*, vol. 61, pp. 1–11, Feb. 2020.

[16] A. Khan, C. Mineo, G. Dobie, C. Macleod, and G. Pierce, ''Vision guided robotic inspection for parts in manufacturing and remanufacturing industry,'' *J. Remanufacturing*, vol. 11, no. 1, pp. 49–70, Apr. 2021.

[17] H. Choset, ''Coverage for robotics—A survey of recent results,'' *Ann. Math. Artif. Intell.*, vol. 31, no. 1, pp. 113–126, Oct. 2001.

[18] E. Galceran and M. Carreras, ''A survey on coverage path planning for robotics,'' *Robot. Auton. Syst.*, vol. 61, no. 12, pp. 1258–1276, Dec. 2013.

[19] R. Almadhoun, T. Taha, L. Seneviratne, and Y. Zweiri, ''A survey on multi-robot coverage path planning for model reconstruction and mapping,'' *Social Netw. Appl. Sci.*, vol. 1, no. 8, pp. 1–24, Aug. 2019.

[20] T. M. Cabreira, L. B. Brisolara, and P. R. Ferreira, Jr., ''Survey on coverage path planning with unmanned aerial vehicles,'' *Drones*, vol. 3, no. 1, pp. 1–38, Jan. 2019.

[21] P. M. M. M. Falaki, A. Padman, V. G. Nair, and K. R. Guruprasad, ''Simultaneous exploration and coverage by a mobile robot,'' in *Control Instrumentation Systems* (Lecture Notes in Electrical Engineering), vol. 581, C. Shreesha and R. Gudi, Eds. Singapore: Springer, 2020, pp. 33–41.

[22] R. Sun, C. Tang, J. Zheng, Y. Zhou, and S. Yu, ''Multi-robot path planning for complete coverage with genetic algorithms,'' in *Intelligent Robotics and Applications* (Lecture Notes in Computer Science), vol. 11744, H. Yu, J. Liu, L. Liu, Z. Ju, Y. Liu, and D. Zhou, Eds. Cham, Switzerland: Springer, 2019, pp. 349–361.

[23] 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* (Studies in Computational Intelligence), vol. 604, A. Koubaa and J. R. M. Dios, Eds. Cham, Switzerland: Springer, 2015, pp. 31–51.

[24] J. Song and S. Gupta, ''CARE: Cooperative autonomy for resilience and efficiency of robot teams for complete coverage of unknown environments under robot failures,'' *Auto. Robots*, vol. 44, nos. 3–4, pp. 647–671, Mar. 2020.

[25] W. He, Z. Li, and C. L. P. Chen, ''A survey of human-centered intelligent robots: Issues and challenges,'' *IEEE/CAA J. Autom. Sinica*, vol. 4, no. 4, pp. 602–609, Oct. 2017.

[26] Z. Zeng, L. Lian, K. Sammut, F. He, Y. Tang, and A. Lammas, ''A survey on path planning for persistent autonomy of autonomous underwater vehicles,'' *Ocean Eng.*, vol. 110, pp. 303–313, Dec. 2015.

[27] J. Kim, D. E. Lee, and J. Seo, ''Task planning strategy and path similarity analysis for an autonomous excavator,'' *Autom. Construct.*, vol. 112, pp. 1–12, Apr. 2020.

[28] A. Ghaddar, A. Merei, and E. Natalizio, ''PPS: Energy-aware grid-based coverage path planning for UAVs using area partitioning in the presence of NFZs,'' *Sensors*, vol. 20, no. 13, pp. 1–32, Jul. 2020.

[29] S. Xing, R. Wang, and G. Huang, ''Area decomposition algorithm for large region maritime search,'' *IEEE Access*, vol. 8, pp. 205788–205797, 2020.

[30] Y. Choi, Y. Choi, S. Briceno, and D. N. Mavris, ''Energy-constrained multi-UAV coverage path planning for an aerial imagery mission using column generation,'' *J. Intell. Robotic Syst.*, vol. 97, no. 1, pp. 125–139, Jan. 2020.

[31] E. M. Arkin, S. P. Fekete, and J. S. B. Mitchell, ''Approximation algorithms for lawn mowing and milling,'' *Comput. Geometry*, vol. 17, nos. 1–2, pp. 25–50, Oct. 2000.

[32] H. Azpúrua, G. M. Freitas, D. G. Macharet, and M. F. M. Campos, ''Multi-robot coverage path planning using hexagonal segmentation for geophysical surveys,'' *Robotica*, vol. 36, no. 8, pp. 1144–1166, Aug. 2018.

[33] J. Xie, L. R. G. Carrillo, and L. Jin, ''An integrated traveling salesman and coverage path planning problem for unmanned aircraft systems,'' *IEEE Control Syst. Lett.*, vol. 3, no. 1, pp. 67–72, Jan. 2019.

[34] J. Xie, L. R. G. Carrillo, and L. Jin, ''Path planning for UAV to cover multiple separated convex polygonal regions,'' *IEEE Access*, vol. 8, pp. 51770–51785, 2020.

[35] B. Englot and F. S. Hover, ''Sampling-based coverage path planning for inspection of complex structures,'' in *Proc. 22nd Int. Conf. Automated Planning Scheduling*, Sao Paulo, Brazil, Jun. 2012, pp. 29–37.

[36] C. Dornhege, A. Kleiner, A. Hertle, and A. Kolling, "Multirobot coverage search in three dimensions," *J. Field Robot.*, vol. 33, no. 4, pp. 537–558, Jun. 2016.

[37] F. Bartumeus, M. G. E. da Luz, G. M. Viswanathan, and J. Catalan, "Animal search strategies: A quantitative random-walk," *Ecology*, vol. 86, no. 11, pp. 3078–3087, Nov. 2005.

[38] C. Dimidov, G. Oriolo, and V. Trianni, "Random walks in swarm robotics: An experiment with Kilobots," in *Swarm Intelligence* (Lecture Notes in Computer Science), vol. 9882, M. Dorigo, M. Birattari, X. Li, M. L. Ibanez, K. Ohkura, C. Pinciroli, and T. Stutzle, Eds. Cham, Switzerland: Springer, 2016, pp. 185–196.

[39] M. Kegeleirs, D. G. Ramos, and M. Birattari, "Random walk exploration for swarm mapping," in *Towards Autonomous Robotic Systems* (Lecture Notes in Computer Science), vol. 11650, K. Althoefer, J. Konstantinova, and K. Zhang, Eds. Cham, Switzerland: Springer, 2019, pp. 211–222.

[40] K. M. Hasan, Abdullah-Al-Nahid, and K. J. Reza, "Path planning algorithm development for autonomous vacuum cleaner robots," in *Proc. Int. Conf. Informat., Electron. Vis. (ICIEV)*, Dhaka, Bangladesh, May 2014, pp. 1–6.

[41] Y. Liu, X. Lin, and S. Zhu, "Combined coverage path planning for autonomous cleaning robots in unstructured environments," in *Proc. 7th World Congr. Intell. Control Autom.*, Chongqing, China, Jun. 2008, pp. 8271–8276.

[42] I. A. Wagner, M. Lindenbaum, and A. M. Bruckstein, "Robotic exploration, Brownian motion, and electrical resistance," in *Randomization and Approximation Techniques in Computer Science* (Lecture Notes in Computer Science), vol. 1518, M. Luby, J. D. P. Rolim, and M. Serna, Eds. Berlin, Germany: Springer, 1998, pp. 116–130.

[43] D. Sutantyo, P. Levi, C. Moslinger, and M. Read, "Collective-adaptive Lévy flight for underwater multi-robot exploration," in *Proc. IEEE Int. Conf. Mechtron. Autom.*, Takamatsu, Japan, Aug. 2013, pp. 456–462.

[44] B. Pang, Y. Song, C. Zhang, and R. Yang, "Effect of random walk methods on searching efficiency in swarm robots for area exploration," *Int. J. Speech Technol.*, vol. 51, no. 7, pp. 5189–5199, Jul. 2021.

[45] Y. Katada, A. Nishiguchi, K. Moriwaki, and R. Watakabe, "Swarm robotic network using Lévy flight in target detection problem," *Artif. Life Robot.*, vol. 21, no. 3, pp. 295–301, Sep. 2016.

[46] F. Martinez, E. Jacinto, and D. Acero, "Brownian motion as exploration strategy for autonomous swarm robots," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Guangzhou, China, Dec. 2012, pp. 2375–2380.

[47] R. Fujisawa and S. Dobata, "Lévy walk enhances efficiency of group foraging in pheromone-communicating swarm robots," in *Proc. IEEE/SICE Int. Symp. Syst. Integr.*, Kobe, Japan, Dec. 2013, pp. 808–813.

[48] G. Li, C. Chen, C. Geng, M. Li, H. Xu, and Y. Lin, "A pheromone-inspired monitoring strategy using a swarm of underwater robots," *Sensors*, vol. 19, no. 19, pp. 1–25, Oct. 2019.

[49] A. Schroeder, S. Ramakrishnan, M. Kumar, and B. Trease, "Efficient spatial coverage by a robot swarm based on an ant foraging model and the Lévy distribution," *Swarm Intell.*, vol. 11, no. 1, pp. 39–69, Mar. 2017.

[50] A. Sekiguchi and Y. Nakamura, "The chaotic mobile robot," *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 898–904, Dec. 2001.

[51] C.-H. Li, C. Fang, F.-Y. Wang, B. Xia, and Y. Song, "Complete coverage path planning for an Arnold system based mobile robot to perform specific types of missions," *Frontiers Inf. Technol. Electron. Eng.*, vol. 20, no. 11, pp. 1530–1542, Dec. 2019.

[52] C. K. Volos, I. M. Kyprianidis, I. N. Stouboulos, H. E. Nistazakis, and G. S. Tombras, "Cooperation of autonomous mobile robots for surveillance missions based on hyperchaos synchronization," *J. Appl. Math. Bioinf.*, vol. 6, no. 3, pp. 125–143, Dec. 2016.

[53] M. J. M. Tavera, O. Lengerke, and M. S. Dutra, "Implementation of chaotic behavior on a fire fighting robot," in *Mechatronics Series I-Intelligent Transportation Vehicles*, O. Lengerke, and M. S. Dutra, Eds. Dubai: Bentham Science, 2011, pp. 170–182.

[54] C. Li, Y. Song, F. Wang, Z. Wang, and Y. Li, "A bounded strategy of the mobile robot coverage path planning based on Lorenz chaotic system," *Int. J. Adv. Robotic Syst.*, vol. 13, no. 3, pp. 1–9, May 2016.

[55] P. Sooraska and K. Klomkarn, "'No-CPU' chaotic robots: From classroom to commerce," *IEEE Circuits Syst. Mag.*, vol. 10, no. 1, pp. 46–53, Mar. 2010.

[56] A. Anwar and H. Khammari, "An investigation on patrol robot coverage performance based on chaotic and non-chaotic guiding signals," *Int. Trans. J. Eng., Manage. Appl. Sci. Technol.*, vol. 2, no. 4, pp. 405–421, Sep. 2011.

[57] C. H. Pimentel-Romero, J. M. Munoz-Pacheco, O. Felix-Beltran, L. C. Gomez-Pavon, and C. K. Volos, "Chaotic planning paths generators by using performance surfaces," in *Fractional Order Control and Synchronization of Chaotic Systems* (Studies in Computational Intelligence), vol. 688, A. Azar, S. Vaidyanathan, and A. Ouannas, Eds. Cham, Switzerland: Springer, 2017, pp. 805–832.

[58] S. Nasr, H. Mekki, and K. Bouallegue, "A multi-scroll chaotic system for a higher coverage path planning of a mobile robot using flatness controller," *Chaos, Solitons Fractals*, vol. 118, pp. 366–375, Jan. 2019.

[59] C. K. Volos, I. M. Kyprianidis, and I. N. Stouboulos, "Experimental investigation on coverage performance of a chaotic autonomous mobile robot," *Robot. Auto. Syst.*, vol. 61, no. 12, pp. 1314–1322, Dec. 2013.

[60] C. K. Volos, N. Doukas, I. M. Kyprianidis, I. N. Stouboulos, and T. G. Kostis, "Chaotic autonomous mobile robot for military missions," in *Proc. 17th Int. Conf. Commun., Recent Adv. Telecommun. Circuit Design*, Rhodes Island, Greece, Jul. 2013, pp. 197–202.

[61] C. Li, F. Wang, L. Zhao, Y. Li, and Y. Song, "An improved chaotic motion path planner for autonomous mobile robots based on a logistic map," *Int. J. Adv. Robotic Syst.*, vol. 10, no. 1, pp. 1–9. Jan. 2013.

[62] E. K. Petavratzis, C. K. Volos, L. Moysis, I. N. Stouboulos, H. E. Nistazakis, G. S. Tombras, and K. P. Valavanis, "An inverse pheromone approach in a chaotic mobile robot's path planning based on a modified logistic map," *Technologies*, vol. 7, no. 4, pp. 84–99, Dec. 2019.

[63] L. S. Martins-Filho and E. E. N. Macau, "Patrol mobile robots and chaotic trajectories," *Math. Problems Eng.*, vol. 2007, pp. 1–13, Apr. 2007.

[64] C. Li, Y. Song, F. Wang, Z. Liang, and B. Zhu, "Chaotic path planner of autonomous mobile robots based on the standard map for surveillance missions," *Math. Problems Eng.*, vol. 2015, pp. 1–11, Aug. 2015.

[65] C. Li, Z. Wang, C. Fang, Z. Liang, Y. Song, and Y. Li, "An integrated algorithm of CCPP task for autonomous mobile robot under special missions," *Int. J. Comput. Intell. Syst.*, vol. 11, no. 1, pp. 1357–1368, Aug. 2018.

[66] C.-H. Li, Y. Song, F.-Y. Wang, Z.-Q. Wang, and Y.-B. Li, "A chaotic coverage path planner for the mobile robot based on the Chebyshev map for special missions," *Frontiers Inf. Technol. Electron. Eng.*, vol. 18, no. 9, pp. 1305–1319, Sep. 2017.

[67] K. Sridharan and Z. N. Ahmadabadi, "A multi-system chaotic path planner for fast and unpredictable online coverage of terrains," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5275–52683, Oct. 2020.

[68] Y. Gabriely and E. Rimon, "Spanning-tree based coverage of continuous areas by a mobile robot," *Ann. Math. Artif. Intell.*, vol. 31, nos. 1–4, pp. 77–98, Oct. 2001.

[69] N. Hazon and G. A. Kaminka, "Redundancy, efficiency and robustness in multi-robot coverage," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, Apr. 2005, pp. 735–741.

[70] Y. Gabriely and E. Rimon, "Competitive on-line coverage of grid environments by a mobile robot," *Comput. Geometry*, vol. 24, no. 3, pp. 197–224, Apr. 2003.

[71] N. Agmon, N. Hazon, and G. A. Kaminka, "Constructing spanning trees for efficient multi-robot coverage," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Orlando, FL, USA, May 2006, pp. 1698–1703.

[72] K. S. Senthilkumar and K. K. Bharadwaj, "Spanning tree based terrain coverage by multi robots in unknown environments," in *Proc. Annu. IEEE India Conf.*, Kanpur, India, Dec. 2008, pp. 120–125.

[73] A. C. Kapoutsis, S. A. Chatzichristofis, and E. B. Kosmatopoulos, "DARP: Divide areas algorithm for optimal multi-robot coverage path planning," *J. Intell. Robotic Syst.*, vol. 86, nos. 3–4, pp. 663–680, Jun. 2017.

[74] 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.

[75] G.-Q. Gao and B. Xin, "A-STC: Auction-based spanning tree coverage algorithm formotion planning of cooperative robots," *Frontiers Inf. Technol. Electron. Eng.*, vol. 20, no. 1, pp. 18–31, Jan. 2019.

[76] T. D. Ranjitha and K. R. Guruprasad, "Pseudo spanning tree-based complete and competitive robot coverage using virtual nodes," *IFAC-PapersOnLine*, vol. 49, no. 1, pp. 195–200, Feb. 2016.

[77] H. V. Pham, P. Moore, and D. X. Truong, "Proposed smooth-STC algorithm for enhanced coverage path planning performance in mobile robot applications," *Robotics*, vol. 8, no. 2, pp. 1–19, Jun. 2019.

[78] 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, Jul. 2021.

[79] W. Dong, S. Liu, Y. Ding, X. Sheng, and X. Zhu, "An artificially weighted spanning tree coverage algorithm for decentralized flying robots," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 4, pp. 1689–1698, Oct. 2020.

[80] R. Wu, "The application of dynamic programming in production planning," *AIP Conf. Proc.*, vol. 1839, no. 1, pp. 1–3, May 2017.

[81] P. Zhou, Z.-M. Wang, Z.-N. Li, and Y. Li, "Complete coverage path planning of mobile robot based on dynamic programming algorithm," in *Proc. 2nd Int. Conf. Electron. Mech. Eng. Inf. Technol.*, Shenyang, China, Sep. 2012, pp. 1837–1841.

[82] M. Morin, I. Abi-Zeid, Y. Petillot, and C.-G. Quimper, "A hybrid algorithm for coverage path planning with imperfect sensors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Tokyo, Japan, Nov. 2013, pp. 5988–5993.

[83] A. A. Altahir, V. S. Asirvadam, N. H. B. Hamid, P. Sebastian, N. B. Saad, R. B. Ibrahim, and S. C. Dass, "Optimizing visual sensor coverage overlaps for multiview surveillance systems," *IEEE Sensors J.*, vol. 18, no. 11, pp. 4544–4552, Jun. 2018.

[84] M. Coombes, T. Fletcher, W. H. Chen, and C. Liu, "Optimal polygon decomposition for UAV survey coverage path planning in wind," *Sensors*, vol. 18, no. 7, pp. 1–28, Jul. 2018.

[85] K. P. Cheng, R. E. Mohan, N. H. K. Nhan, and A. V. Le, "Graph theory-based approach to accomplish complete coverage path planning tasks for reconfigurable robots," *IEEE Access*, vol. 7, pp. 94642–94657, 2019.

[86] A. Ghaddar and A. Merei, "EAOA: Energy-aware grid-based 3D-obstacle avoidance in coverage path planning for UAVs," *Future Internet*, vol. 12, no. 2, pp. 1–20, Feb. 2020.

[87] S. G. Tzafestas, *Introduction to Mobile Robot Control*. Boca Raton, NY, USA: Elsevier, 2013.

[88] D. K. Sutantyo, S. Kernbach, P. Levi, and V. A. Nepomnyashchikh, "Multi-robot searching algorithm using Lévy flight and artificial potential field," in *Proc. IEEE Saf. Secur. Rescue Robot.*, Bremen, Germany, Jul. 2010, pp. 1–6.

[89] M. Nykolaychuk and F. Ortmeier, "Coverage path re-planning for processing faults," in *Intelligent Robotics and Applications* (Lecture Notes in Computer Science), vol. 9245, H. Liu, N. Kubota, X. Zhu, R. Dillmann, and D. Zhou, Eds. Cham, Switzerland: Springer, 2015, pp. 358–368.

[90] C. Wei, F. Zhang, C. Yin, Y. Liu, L. Liu, Z. Li, and W. Wang, "Research on UAV intelligent obstacle avoidance technology during an inspection of transmission line," in *Applied Mechanics* (Mechatronics and Intelligent Systems), S. Qin and X. Li, Eds. Singapore: World Scientific, 2016, pp. 319–327.

[91] C. Wang, L. Meng, T. Li, C. W. D. Silva, and M. Q.-H. Meng, "Towards autonomous exploration with information potential field in 3D environments," in *Proc. 18th Int. Conf. Adv. Robot. (ICAR)*, Hong Kong, Jul. 2017, pp. 340–345.

[92] J. Yan, Y. Li, Z. Song, and Q. Zhang, "Research on UAV coverage path planning algorithm based on improved artificial potential field method," *Oper. Res. Fuzziol.*, vol. 9, no. 4, pp. 264–270, Nov. 2019.

[93] C. Huang, W. Li, C. Xiao, B. Liang, and S. Han, "Potential field method for persistent surveillance of multiple unmanned aerial vehicle sensors," *Int. J. Distrib. Sensor Netw.*, vol. 14, no. 1, pp. 1–13, Jan. 2018.

[94] X. Jiang and Y. Deng, "UAV track planning of electric tower pole inspection based on improved artificial potential field method," *J. Appl. Sci. Eng.*, vol. 24, no. 2, pp. 123–132, Apr. 2021.

[95] T. Danner and L. E. Kavraki, "Randomized planning for short inspection paths," in *Proc. IEEE Int. Conf. Robot. Automat. Symposia Millennium Conf. (ICRA)*, San Francisco, CA, USA, Apr. 2000, pp. 971–976.

[96] J. Faigl, M. Kulich, and L. Přeučil, "A sensor placement algorithm for a mobile robot inspection planning," *J. Intell. Robotic Syst.*, vol. 62, nos. 3–4, pp. 329–353, Jun. 2011.

[97] Á. Madridano, A. Al-Kaff, D. Martín, and A. D. L. Escalera, "3D trajectory planning method for UAVs swarm in building emergencies," *Sensors*, vol. 20, no. 3, pp. 1–20, Feb. 2020.

[98] T. Zeng and B. Si, "Mobile robot exploration based on rapidly-exploring random trees and dynamic window approach," in *Proc. 5th Int. Conf. Control, Autom. Robot. (ICCAR)*, Beijing, China, Apr. 2019, pp. 51–57.

[99] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.

[100] O. Salzman, "Sampling-based robot motion planning," *Commun. ACM*, vol. 62, no. 10, pp. 54–63, Sep. 2019.

[101] A. Dias, T. Fernandes, J. Almeida, A. Martins, and E. Silva, "3D path planning methods for unmanned aerial vehicles in search and rescue scenarios," in *Human-Centric Robotics*, M. F. Silva, G. S. Virk, M. O. Tokhi, B. Malheiro, and P. Guedes, Eds. Singapore: World Scientific, 2017, pp. 213–220.

[102] M. Ulrich, G. Lux, L. Jürgensen, and G. Reinhart, "Automated and cycle time optimized path planning for robot-based inspection systems," *Procedia CIRP*, vol. 44, pp. 377–382, May 2016.

[103] S. M. Lavalle and J. J. Kuffner, "Rapidly-exploring random trees: Progress and prospects," in *Algorithmic Comput. Robotics: New Directions*, B. Donald, K. Lynch, and D. Rus, Eds. Boca Raton, FL, USA: CRC Press, 2001, pp. 293–308.

[104] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: A review," *IEEE Access*, vol. 2, pp. 56–77, 2014.

[105] S. Zaheer, J. M, and T. Gulrez, "Performance analysis of path planning techniques for autonomous mobile robots," in *Proc. IEEE Int. Conf. Electr., Comput. Commun. Technol. (ICECCT)*, Coimbatore, India, Mar. 2015, pp. 1–5.

[106] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. Millennium Conf. IEEE Int. Conf. Robot. Automat. Symposia (ICRA)*, San Francisco, CA, USA, Apr. 2000, pp. 995–1001.

[107] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, Jun. 2011.

[108] A. Bircher, K. Alexis, U. Schwesinger, S. Omari, M. Burri, and R. Siegwart, "An incremental sampling-based approach to inspection planning: The rapidly exploring random tree of trees," *Robotica*, vol. 35, no. 6, pp. 1327–1340, Jun. 2017.

[109] Y. Bouzid, Y. Bestaoui, and H. Siguerdidjane, "Quadrotor-UAV optimal coverage path planning in cluttered environment with a limited onboard energy," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Vancouver, BC, Canada, Sep. 2017, pp. 979–984.

[110] M. Fu, A. Kuntz, O. Salzman, and R. Alterovitz, "Toward asymptotically-optimal inspection planning via efficient near-optimal graph search," in *Proc. Robot., Sci. Syst. Conf.*, Freiburg im Breisgau, Germany, Jun. 2019, pp. 1–12.

[111] S. Faghihi, S. Tavana, and A. D. Ruiter, "Autonomous on-orbit inspection of complex space structures in deep space environment," in *Proc. 71st Int. Astron. Congr.-Cyberspace Ed.*, Dubai, United Arab Emirates, Oct. 2020, pp. 1–10.

[112] H. Liu, J. Ma, and W. Huang, "Sensor-based complete coverage path planning in dynamic environment for cleaning robot," *CAAI Trans. Intell. Technol.*, vol. 3, no. 1, pp. 65–72, Mar. 2018.

[113] K. Yu, C. Guo, and J. Yi, "Complete and near-optimal path planning for simultaneous sensor-based inspection and footprint coverage in robotic crack filling," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Montreal, QC, Canada, May 2019, pp. 8812–8818.

[114] A. Bircher, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, and R. Siegwart, "Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Seattle, WA, USA, May 2015, pp. 6423–6430.

[115] A. Bircher, M. Kamel, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, and R. Siegwart, "Three-dimensional coverage path planning via viewpoint resampling and tour optimization for aerial robots," *Auto. Robots*, vol. 40, no. 6, pp. 1059–1078, Aug. 2016.

[116] W. Jing, J. Polden, P. Y. Tao, C. F. Goh, W. Lin, and K. Shimada, "Model-based coverage motion planning for industrial 3D shape inspection applications," in *Proc. 13th IEEE Conf. Autom. Sci. Eng. (CASE)*, Xi'an, China, Aug. 2017, pp. 1293–1300.

[117] X. Zhou, Z. Yi, Y. Liu, K. Huang, and H. Huang, "Survey on the path and view planning for UAVs," *Virtual Reality Intell. Hardw.*, vol. 2, no. 1, pp. 56–69, Feb. 2020.

[118] R. Zeng, Y. Wen, W. Zhao, and Y.-J. Liu, "View planning in robot active vision: A survey of systems, algorithms, and applications," *Comput. Vis. Media*, vol. 6, no. 3, pp. 225–245, Sep. 2020.

[119] Y. B. Sebbane, *Intelligent Autonomy of UAVs: Advanced Missions and Future Use*. Boca Raton, FL, USA: CRC Press, 2018.

[120] W. Jing, J. Polden, W. Lin, and K. Shimada, "Sampling-based view planning for 3D visual coverage task with unmanned aerial vehicle," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Daejeon, South Korea, Oct. 2016, pp. 1808–1815.

[121] C. Connolly, "The determination of next best views," in *Proc. IEEE Int. Conf. Robot. Autom.*, St. Louis, MO, USA, Mar. 1985, pp. 432–435.

[122] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, ''Receding horizon 'next-best-view' planner for 3D exploration,'' in *Proc. IEEE Int. Conf. Robot. Autom.*, Stockholm, Sweden, May 2016, pp. 1462–1468.
[123] S. Song and S. Jo, ''Surface-based exploration for autonomous 3D modeling,'' in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Brisbane, QLD, Australian, May 2018, pp. 4319–4326.
[124] R. Tinos, K. Helsgaun, and D. Whitley, ''Efficient recombination in the Lin-Kernighan-Helsgaun traveling salesman heuristic,'' in *Parallel Problem Solving from Nature—PPSN XV* (Lecture Notes in Computer Science), vol. 11101, A. Auger, C. M. Fonseca, N. Lourenco, P. Machado, L. Paquete, and D. Whitley, Eds. Cham, Switzerland: Springer, 2018, pp. 95–107.
[125] N. Palomeras, N. Hurtos, E. Vidal, and M. Carreras, ''Autonomous exploration of complex underwater environments using a probabilistic next-best-view planner,'' *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1619–1625, Apr. 2019.
[126] S. Osswald, P. Karkowski, and M. Bennewitz, ''Efficient coverage of 3D environments with humanoid robots using inverse reachability maps,'' in *Proc. IEEE-RAS 17th Int. Conf. Humanoid Robot. (Humanoids)*, Birmingham, U.K., Nov. 2017, pp. 151–157.
[127] I. Ardiyanto and J. Miura, ''Visibility-based viewpoint planning for guard robot using skeletonization and geodesic motion model,'' in *Proc. IEEE Int. Conf. Robot. Autom.*, Karlsruhe, Germany, May 2013, pp. 660–666.
[128] I. Ardiyanto and J. Miura, ''Time-space viewpoint planning for guard robot with chance constraint,'' *Int. J. Autom. Comput.*, vol. 16, no. 4, pp. 475–490, Aug. 2019.
[129] Y. Wang, S. James, E. K. Stathopoulou, C. Beltran-Gonzalez, Y. Konishi, and A. Del Bue, ''Autonomous 3-D reconstruction, mapping, and exploration of indoor environments with a robotic arm,'' *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3340–3347, Oct. 2019.
[130] R. Border, J. D. Gammell, and P. Newman, ''Surface edge explorer (see): Planning next best views directly from 3D observations,'' in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Brisbane, QLD, Australia, May 2018, pp. 1–8.
[131] C. Wu, R. Zeng, J. Pan, C. C. L. Wang, and Y.-J. Liu, ''Plant phenotyping by deep-learning-based planner for multi-robots,'' *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3113–3120, Oct. 2019.
[132] F. Paus, P. Kaiser, N. Vahrenkamp, and T. Asfour, ''A combined approach for robot placement and coverage path planning for mobile manipulation,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Vancouver, BC, Canada, Sep. 2017, pp. 1–8.
[133] S. S. Mansouri, C. Kanellakis, E. Fresk, D. Kominiak, and G. Nikolakopoulos, ''Cooperative coverage path planning for visual inspection,'' *Control Eng. Pract.*, vol. 74, pp. 118–131, May 2018.
[134] C. Kanellakis, S. S. Mansouri, E. Fresk, D. Kominiak, and G. Nikolakopoulos, ''Cooperative UAVs as a tool for aerial inspection of large scale aging infrastructure,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, vol. 5, M. Hutter and R. Siegwart, Eds. Cham, Switzerland: Springer, Oct. 2018, pp. 177–189.
[135] S. Lindner, C. Garbe, and K. Mombaur, ''Optimization based multi-view coverage path planning for autonomous structure from motion recordings,'' *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3278–3285, Oct. 2019.
[136] Z. Meng, H. Qin, Z. Chen, X. Chen, H. Sun, F. Lin, and M. H. Ang, ''A two-stage optimized next-view planning framework for 3-D unknown environment exploration, and structural reconstruction,'' *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1680–1687, Jul. 2017.
[137] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, ''OctoMap: An efficient probabilistic 3D mapping framework based on octrees,'' *Auton. Robot.*, vol. 34, no. 3, pp. 189–206, Apr. 2013.
[138] R. Almadhoun, T. Taha, J. Dias, L. Seneviratne, and Y. Zweiri, ''Coverage path planning for complex structures inspection using an unmanned aerial vehicle (UAV),'' in *Intelligent Robotics and Applications* (Lecture Notes in Computer Science), vol. 11744, H. Yu, J. Liu, L. Liu, Z. Ju, Y. Liu, and D. Zhou, Eds. Cham, Switzerland: Springer, 2019, pp. 243–266.
[139] C. Papachristos, F. Mascarich, S. Khattak, T. Dang, and K. Alexis, ''Localization uncertainty-aware autonomous exploration and mapping with aerial robots using receding horizon path-planning,'' *Auto. Robots*, vol. 43, no. 8, pp. 2131–2161, Dec. 2019.
[140] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, ''Receding horizon path planning for 3D exploration and surface inspection,'' *Auto. Robots*, vol. 42, no. 2, pp. 291–306, Feb. 2018.
[141] C. Papachristos, M. Kamel, M. Popovic, S. Khattak, A. Bircher, H. Oleynikova, T. Dang, F. Mascarich, K. Alexis, and R. Siegwart, ''Autonomous exploration and inspection path planning for aerial robots using the robot operating system,'' in *Robot Operating System (ROS)* (Studies in Computational Intelligence), vol. 778, A. Koubaa, Ed. Cham, Switzerland: Springer, 2019, pp. 67–111.
[142] S. Jung, S. Song, P. Youn, and H. Myung, ''Multi-layer coverage path planner for autonomous structural inspection of high-rise structures,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Madrid, Spain, Oct. 2018, pp. 1–9.
[143] H. Oleynikova, Z. Taylor, R. Siegwart, and J. Nieto, ''Safe local exploration for replanning in cluttered unknown environments for microaerial vehicles,'' *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1474–1481, Jul. 2018.
[144] M. Selin, M. Tiger, D. Duberg, F. Heintz, and P. Jensfelt, ''Efficient autonomous exploration planning of large-scale 3-D environments,'' *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1699–1706, Apr. 2019.
[145] T. Dang, S. Khattak, F. Mascarich, and K. Alexis, ''Explore locally, plan globally: A path planning framework for autonomous robotic exploration in subterranean environments,'' in *Proc. 19th Int. Conf. Adv. Robot. (ICAR)*, Belo Horizonte, Brazil, Dec. 2019, pp. 9–16.
[146] R. Almadhoun, A. Abduldayem, T. Taha, L. Seneviratne, and Y. Zweiri, ''Guided next best view for 3D reconstruction of large complex structures,'' *Remote Sens.*, vol. 11, no. 20, pp. 1–20, Oct. 2019.
[147] L. Schmid, M. Pantic, R. Khanna, L. Ott, R. Siegwart, and J. Nieto, ''An efficient sampling-based method for online informative path planning in unknown environments,'' *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1500–1507, Apr. 2020.
[148] S. Song, D. Kim, and S. Jo, ''Online coverage and inspection planning for 3D modeling,'' *Auto. Robots*, vol. 44, no. 8, pp. 1431–1450, Aug. 2020.
[149] W. Jing, D. Deng, Z. Xiao, Y. Liu, and K. Shimada, ''Coverage path planning using path primitive sampling and primitive coverage graph for visual inspection,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Macau, China, Nov. 2019, pp. 1472–1479.
[150] K. Alexis, C. Papachristos, R. Siegwart, and A. Tzes, ''Uniform coverage structural inspection path–planning for micro aerial vehicles,'' in *Proc. IEEE Int. Symp. Intell. Control (ISIC)*, Sydney, NSW, Australia, Sep. 2015, pp. 59–64.
[151] L. Hou, X. Chen, K. Lan, R. Rasmussen, and J. Roberts, ''Volumetric next best view by 3D occupancy mapping using Markov chain Gibbs sampler for precise manufacturing,'' *IEEE Access*, vol. 7, pp. 121949–121960, 2019.
[152] M. Corah, C. O'Meadhra, K. Goel, and N. Michael, ''Communication-efficient planning and mapping for multi-robot exploration in large environments,'' *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1715–1721, Apr. 2019.
[153] Y. Wu and J. Wang, *Algorithm Design Practice for Collegiate Programming Contests and Education*. Boca Raton, FL, USA: CRC Press, 2018.
[154] P. Joshi, *Artificial Intelligence With Python*. Birmingham, U.K.: Packt, 2017.
[155] A. Zdesar, S. Blazic, G. Klancar, and I. Skrjanc, *Wheeled Mobile Robotics: From Fundamentals Towards Autonomous Systems*. Oxford, U.K.: Butterworth-Heinemann, 2017.
[156] G. Zuo, P. Zhang, and J. Qiao, ''Path planning algorithm based on sub-region for agricultural robot,'' in *Proc. 2nd Int. Asia Conf. Informat. Control, Autom. Robot. (CAR)*, Wuhan, China, Mar. 2010, pp. 197–200.
[157] X. Wang and D. Li, ''Coverage path planning for UAVs in unknown directional regions,'' *Int. J. Wireless Mobile Comput.*, vol. 8, no. 3, pp. 285–293, May 2015.
[158] P. S. Pratama, J.-W. Kim, H.-K. Kim, S.-M. Yoon, T.-K. Yeu, S. Hong, S.-J. Oh, and S.-B. Kim, ''Path planning algorithm to minimize an overlapped path and turning number for an underwater mining robot,'' in *Proc. 15th Int. Conf. Control, Autom. Syst. (ICCAS)*, Busan, South Korea, Oct. 2015, pp. 499–504.
[159] A. M. Kabir, K. N. Kaipa, J. Marvel, and S. K. Gupta, ''Automated planning for robotic cleaning using multiple setups and oscillatory tool motions,'' *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 3, pp. 1364–1377, Jul. 2017.
[160] A. Barrientos, J. Colorado, J. D. Cerro, A. Martinez, C. Rossi, D. Sanz, and J. Valente, ''Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots,'' *J. Field Robot.*, vol. 28, no. 5, pp. 667–689, 2011.

[161] W. Wang, P. Zhang, C. Liang, and Y. Shi, "Design, path planning improvement and test of a portable massage robot on the human back," *Int. J. Adv. Robotic Syst.*, vol. 15, no. 4, pp. 1–11, Jul. 2018.
[162] X. Miao, H.-S. Lee, and B.-Y. Kang, "Multi-cleaning robots using cleaning distribution method based on map decomposition in large environments," *IEEE Access*, vol. 8, pp. 97873–97889, 2020.
[163] M. Zhou and N. Gao, "Research on optimal path based on Dijkstra algorithms," in *Proc. 3rd Int. Conf. Mechtron. Eng. Inf. Technol. (ICMEIT)*, Dalian, China, Mar. 2019, pp. 884–892.
[164] I. Wieser, A. V. Ruiz, M. Frassl, M. Angermann, J. Mueller, and M. Lichtenstern, "Autonomous robotic SLAM-based indoor navigation for high-resolution sampling with complete coverage," in *Proc. IEEE/ION Position, Location Navigat. Symp.*, Monterey, CA, USA, May 2014, pp. 945–951.
[165] R. Yehoshua, N. Agmon, and G. A. Kaminka, "Robotic adversarial coverage of known environments," *Int. J. Robot. Res.*, vol. 35, no. 12, pp. 1419–1444, Oct. 2016.
[166] R. Rosa, T. Brito, A. I. Pereira, J. Lima, and M. A. Wehrmeister, "Using multi-UAV for rescue environment mapping: Task planning optimization approach," in *CONTROLO 2020* (Lecture Notes in Electrical Engineering), vol. 695, J. A. Gonçalves, M. Braz-César, and J. P. Coelho, Eds. Cham, Switzerland: Springer, 2020, pp. 507–517.
[167] X. Zhang, S. Liu, and Z. Xiang, "Optimal inspection path planning of substation robot in the complex substation environment," in *Proc. Chin. Autom. Congr. (CAC)*, Hangzhou, China, Nov. 2019, pp. 5064–5068.
[168] X. Liu and D. Gong, "A comparative study of A-star algorithms for search and rescue in perfect maze," in *Proc. Int. Conf. Electr. Inf. Control Eng.*, Wuhan, China, Apr. 2011, pp. 24–27.
[169] A. M. Chaudhari, M. R. Apsangi, and A. B. Kudale, "Improved A-star algorithm with least turn for robotic rescue operations," in *Computational Intelligence, Communications, and Business Analytics* (Communications in Computer and Information Science), vol. 776, J. Mandal, P. Dutta, and S. Mukhopadhyay, Eds. Singapore: Springer, 2017, pp. 614–627.
[170] S. Dogru and L. Marques, "A*-based solution to the coverage path planning problem," in *Proc. 3rd Iberian Robot. Conf.* (Advances in Intelligent Systems and Computing), vol. 693, A. Ollero, A. Sanfeliu, L. Montano, N. Lau, and C. Cardeira, Eds. Cham, Switzerland: Springer, 2018, pp. 240–248.
[171] H. H. Viet, V.-H. Dang, S. Choi, and T. C. Chung, "BoB: An online coverage approach for multi-robot systems," *Int. J. Speech Technol.*, vol. 42, no. 2, pp. 157–173, Mar. 2015.
[172] Z. Cai, S. Li, Y. Gan, R. Zhang, and Q. Zhang, "Research on complete coverage path planning algorithms based on A* algorithms," *Open Cybern. Syst. J.*, vol. 8, no. 1, pp. 418–426, Dec. 2014.
[173] A. V. Le, V. Prabakaran, V. Sivanantham, and R. E. Mohan, "Modified A-star algorithm for efficient coverage path planning in Tetris inspired self-reconfigurable robot with integrated laser sensor," *Sensors*, vol. 18, no. 8, pp. 1–27, Aug. 2018.
[174] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, San Diego, CA, USA, May 1994, pp. 3310–3317.
[175] M. Dakulović, S. Horvatić, and I. Petrović, "Complete coverage D* algorithm for path planning of a floor-cleaning mobile robot," *IFAC Proc. Volumes*, vol. 44, no. 1, pp. 5950–5955, Jan. 2011.
[176] I. Maurovic, M. Seder, K. Lenac, and I. Petrovic, "Path planning for active SLAM based on the D* algorithm with negative edge weights," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 8, pp. 1321–1331, Aug. 2018.
[177] A. T. Le and T. D. Le, "Search-based planning and replanning in robotics and autonomous systems," in *Advanced Path Planning for Mobile Entities*, R. Roka, Ed. London, U.K.: IntechOpen, 2018, pp. 63–89.
[178] C. Luo, H. Mo, F. Shen, and W. Zhao, "Multi-goal motion planning of an autonomous robot in unknown environments by an ant colony optimization approach," in *Advances in Swarm Intelligence* (Lecture Notes in Computer Science), vol. 9713, Y. Tan, Y. Shi, and L. Li, Eds. Cham, Switzerland: Springer, 2016, pp. 519–527.
[179] M.-K. Ng, Y.-W. Chong, K.-M. Ko, Y.-H. Park, and Y.-B. Leau, "Adaptive path finding algorithm in dynamic environment for warehouse robot," *Neural Comput. Appl.*, vol. 32, no. 17, pp. 13155–13171, Sep. 2020.
[180] K. Daniel, A. Nash, S. Koenig, and A. Felner, "Theta*: Any-angle path planning on grids," *J. Artif. Intell. Res.*, vol. 39, pp. 533–579, Oct. 2010.
[181] S. Choi, S. Lee, H. H. Viet, and T. Chung, "B-Theta*: An efficient online coverage algorithm for autonomous cleaning robots," *J. Intell. Robotic Syst.*, vol. 87, no. 2, pp. 265–290, Aug. 2017.
[182] Y.-C. Huang and H.-Y. Lin, "Development and implementation of a multi-robot system for collaborative exploration and complete coverage," in *Proc. 14th Int. Conf. Signal-Image Technol. Internet-Based Syst. (SITIS)*, Las Palmas, Spain, Nov. 2018, pp. 472–479.
[183] M. Faria, I. Maza, and A. Viguria, "Applying frontier cells based exploration and Lazy Theta* path planning over single grid-based world representation for autonomous inspection of large 3D structures with an UAS," *J. Intell. Robotic Syst.*, vol. 93, nos. 1–2, pp. 113–133, Feb. 2019.
[184] M. Faria, R. Marín, M. Popovic, I. Maza, and A. Viguria, "Efficient Lazy Theta* path planning over a sparse grid to explore large 3D volumes with a multirotor UAV," *Sensors*, vol. 19, no. 1, pp. 1–21, Jan. 2019.
[185] M. Faria, A. S. Ferreira, H. P. Leon, I. Maza, and A. Viguria, "Autonomous 3D exploration of large structures using a UAV equipped with a 2D LIDAR," *Sensors*, vol. 19, no. 22, pp. 1–24, Nov. 2019.
[186] S. Mirjalili, *Evolutionary Algorithms, and Neural Networks: Theory and Applications*. Cham, Switzerland: Springer, 2019.
[187] B. M. Wilamowski and J. D. Irwin, *The Industrial Electronics Handbook: Intelligent Systems*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2011.
[188] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence*. Cambridge, U.K.: MIT Press, 1992.
[189] C. García-Martínez, F. J. Rodriguez, and M. Lozano, "Genetic algorithms," in *Handbook of Heuristics*, R. Martí, P. Pardalos, and M. Resende, Eds. Cham, Switzerland: Springer, 2018, pp. 431–464.
[190] Z. Wang and Z. Bo, "Coverage path planning for mobile robot based on genetic algorithm," in *Proc. IEEE Workshop Electron., Comput. Appl.*, Ottawa, ON, Canada, May 2014, pp. 732–735.
[191] I. A. Hameed, D. D. Bochtis, and C. G. Sorensen, "Driving angle and track sequence optimization for operational path planning using genetic algorithms," *Appl. Eng. Agricult.*, vol. 27, no. 6, pp. 1077–1086, Nov. 2011.
[192] I. A. Hameed, D. D. Bochtis, and C. G. Sorensen, "An optimized field coverage planning approach for navigation of agricultural robots in fields involving obstacle areas," *Int. J. Adv. Robotic Syst.*, vol. 10, no. 5, pp. 1–9, May 2013.
[193] M. Shen, S. Wang, S. Wang, and Y. Su, "Simulation study on coverage path planning of autonomous tasks in hilly farmland based on energy consumption model," *Math. Problems Eng.*, vol. 2020, pp. 1–15, Aug. 2020.
[194] K. O. Ellefsen, H. A. Lepikson, and J. C. Albiez, "Multiobjective coverage path planning: Enabling automated inspection of complex, real-world structures," *Appl. Soft Comput.*, vol. 61, pp. 264–282, Dec. 2017.
[195] W.-C. Tung and J.-S. Liu, "Genetic algorithm with modified operators for an integrated traveling salesman and coverage path planning problem," in *Proc. 16th Int. Conf. Appl. Comput.*, Cagliari, Italy, Nov. 2019, pp. 205–216.
[196] L. Qiu, "Research on a hierarchical cooperative algorithm based on genetic algorithm and particle swarm optimization," in *Computational Intelligence and Intelligent Systems* (Communications in Computer and Information Science), vol. 874, K. Li, W. Li, Z. Chen, and Y. Liu, Eds. Singapore: Springer, 2018, pp. 16–25.
[197] M. G. Sadek, A. E. Mohamed, and A. M. El-Garhy, "Augmenting multi-objective genetic algorithm and dynamic programming for online coverage path planning," in *Proc. 13th Int. Conf. Comput. Eng. Syst. (ICCES)*, Cairo, Egypt, Dec. 2018, pp. 475–480.
[198] C.-T. Cheng, K. Fallahi, H. Leung, and C. K. Tse, "A genetic algorithm-inspired UUV path planner based on dynamic programming," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 1128–1134, Nov. 2012.
[199] V. R. Batista and F. A. Zampirolli, "Optimising robotic pool-cleaning with a genetic algorithm," *J. Intell. Robotic Syst.*, vol. 95, no. 2, pp. 443–458, Aug. 2019.
[200] X. L. Hu and Z. Lin, "Coverage path planning of Penaeus vannamei feeding based on global and multiple local areas," in *Data Science* (Communications in Computer and Information Science), vol. 1179, J. He, P. S. Yu, Y. Shi, X. Li, Z. Xie, G. Huang, J. Cao, and X. Fu, Eds. Singapore: Springer, 2020, pp. 687–697.
[201] C. Liu, W. Xie, P. Zhang, Q. Guo, and D. Ding, "Multi-UAVs cooperative coverage reconnaissance with neural network and genetic algorithm," in *Proc. 3rd High Perform. Comput. Cluster Technol. Conf.*, Guangzhou, China, Jun. 2019, pp. 81–86.
[202] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, Dec. 1997.

[203] D. Alvarez, J. V. Gomez, S. Garrido, and L. Moreno, "Geometrically constrained path planning with fast marching square," in *Proc. 23rd Medit. Conf. Control Autom. (MED)*, Torremolinos, Spain, Jun. 2015, pp. 1014–1019.

[204] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Berlin, Germany: Springer, 2005.

[205] J. Vesterstrom and R. Thomsen, "A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems," in *Proc. Congr. Evol. Comput.*, Portland, OR, USA, Jun. 2004, pp. 1980–1987.

[206] P. Xiao, H. Ju, Q. Li, and F. Chen, "Task planning of space-robot clusters based on modified differential evolution algorithm," *Appl. Sci.*, vol. 10, no. 14, pp. 1–24, Jul. 2020.

[207] V. Gonzalez, C. A. Monje, S. Garrido, L. Moreno, and C. Balaguer, "Coverage mission for UAVs using differential evolution and fast marching square methods," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 35, no. 2, pp. 18–29, Feb. 2020.

[208] G. Beni and J. Wang, "Swarm intelligence in cellular robotics system," in *Robots and Biological Systems: Towards a New Bionics?* (NATO ASI Series), vol. 102, P. Dario, G. Sandini and P. Aebischer, Eds. Berlin, Germany: Springer, 1993, pp. 703–712.

[209] E. Cuevas, F. Fausto, and A. Gonzalez, *New Advancements in Swarm Algorithms: Operators and Applications* (Intelligent Systems Reference Library). Cham, Switzerland: Springer, 2020.

[210] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*. London, U.K.: Oxford Univ. Press, 1999.

[211] C. Blum and X. Li, "Swarm intelligence in optimization," in *Swarm Intelligence* (Natural Computing Series), C. Blum, and D. Merkle, Eds. Berlin, Germany: Springer, 2008, pp. 43–85.

[212] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, Perth, WA, Australia, Nov. 1995, pp. 1942–1948.

[213] M. Dorigo and T. Stützle, *Ant Colony Optimization*. Cambridge, MA, USA: MIT Press, 2004.

[214] D. Teodorović, "Bee colony optimization (BCO)," in *Innovations in Swarm Intelligence* (Studies in Computational Intelligence), vol. 248, C. Lim, L. Jain, and S. Dehuri, Eds. Berlin, Germany: Springer, 2009, pp. 39–60.

[215] J. Kennedy, "The particle swarm: Social adaptation of knowledge," in *Proc. IEEE Int. Conf. Evol. Comput.*, Indianapolis, IN, USA, Apr. 1997, pp. 303–308.

[216] T.-K. Lee, S.-H. Baek, Y.-H. Choi, and S.-Y. Oh, "Smooth coverage path planning and control of mobile robots based on high-resolution grid map representation," *Robot. Auto. Syst.*, vol. 59, no. 10, pp. 801–812, Oct. 2011.

[217] S. Sharma, C. Sur, A. Shukla, and R. Tiwari, "Multi-robot area exploration using particle swarm optimization with the help of CBDF-based robot scattering," in *Computational Vision and Robotics* (Advances in Intelligent Systems and Computing), vol. 332, I. K. Sethi, Ed. New Delhi, India: Springer, 2015, pp. 113–123.

[218] S. Sahu and B. B. Choudhury, "PSO based path planning of a six-axis industrial robot," in *Computational Intelligence in Data Mining* (Advances in Intelligent Systems and Computing), vol. 990, H. Behera, J. Nayak, B. Naik, and D. Pelusi, Eds. Singapore: Springer, 2020, pp. 213–220.

[219] Y.-H. Lin, S.-M. Wang, L.-C. Huang, and M.-C. Fang, "Applying the stereo-vision detection technique to the development of underwater inspection task with PSO-based dynamic routing algorithm for autonomous underwater vehicles," *Ocean Eng.*, vol. 139, pp. 127–139, Jul. 2017.

[220] Y. H. Lin, L. C. Huang, S. Y. Chen, and C. M. Yu, "The optimal route planning for inspection task of autonomous underwater vehicle composed of MOPSO-based dynamic routing algorithm in currents," *Appl. Ocean. Res.*, vol. 75, pp. 178–192, Jun. 2018.

[221] S. Wang, Y. Bai, and C. Zhou, "Coverage path planning design of mapping UAVs based on particle swarm optimization algorithm," in *Proc. Chin. Control Conf. (CCC)*, Guangzhou, China, Jul. 2019, pp. 8236–8241.

[222] M. S. Couceiro, R. P. Rocha, and N. M. F. Ferreira, "A novel multi-robot exploration approach based on particle swarm optimization algorithms," in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot.*, Kyoto, Japan, Nov. 2011, pp. 327–332.

[223] Z. Shang, J. Bradley, and Z. Shen, "A co-optimal coverage path planning method for aerial scanning of complex structures," *Expert Syst. Appl.*, vol. 158, pp. 1–16, Nov. 2020.

[224] F. van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 225–239, Jun. 2004.

[225] X. Li and X. Yao, "Cooperatively co-evolution particle swarms for large scale optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 2, pp. 210–224, Apr. 2012.

[226] 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, pp. 1–28, May 2019.

[227] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, Nov. 2006.

[228] H. Duan, "Ant colony optimization: Principle, convergence and application," in *Handbook of Swarm Intelligence* (Adaptation, Learning, and Optimization), vol. 8, B. K. Panigrahi, Y. Shi, and M. H. Lim, Eds. Berlin, Germany: Springer, 2011, pp. 373–388.

[229] J. Ning, Q. Zhang, C. Zhang, and B. Zhang, "A best-path-updating information-guided ant colony optimization algorithm," *Inf. Sci.*, vols. 433–434, pp. 142–162, Apr. 2018.

[230] A. Borisenko and S. Gorlatch, "Optimizing a GPU-parallelized ant colony metaheuristic by parameter tuning," in *Parallel Computing Technologies* (Lecture Notes in Computer Science), vol. 11657, V. Malyshkin, Ed. Cham, Switzerland: Springer, 2019, pp. 151–165.

[231] M. Starzec, G. Starzec, A. Byrski, and W. Turek, "Distributed ant colony optimization based on actor model," *Parallel Comput.*, vol. 90, pp. 1–13, Dec. 2019.

[232] C. Pan, H. Wang, J. Li, and M. Korovkin, "Path planning of mobile robot based on an improved ant colony algorithm," in *Convergent Cognitive Information Technologies* (Communications in Computer and Information Science), vol. 1140, V. Sukhomlin and E. Zubareva, Eds. Cham, Switzerland: Springer, 2020, pp. 132–141.

[233] B. Li and T. Li, "Vehicle path optimization with time window based on improved ant colony algorithm," in *Data Processing Techniques and Applications for Cyber-Physical Systems* (Advances in Intelligent Systems and Computing), vol. 1088, C. Huang, Y. W. Chan, and N. Yen, Eds. Singapore: Springer, 2020, pp. 167–175.

[234] W. Zhang, X. Gong, G. Han, and Y. Zhao, "An improved ant colony algorithm for path planning in one scenic area with many spots," *IEEE Access*, vol. 5, pp. 13260–13269, 2017.

[235] Z. Chibin, W. Xingsong, and D. Yong, "Complete coverage path planning based on ant colony algorithm," in *Proc. 15th Int. Conf. Mechtron. Mach. Vis. Pract.*, Auckland, New Zealand, Dec. 2008, pp. 357–361.

[236] K. Zhou, A. L. Jensen, C. G. Sørensen, P. Busato, and D. D. Bothtis, "Agricultural operations planning in fields with multiple obstacle areas," *Comput. Electron. Agricult.*, vol. 109, pp. 12–22, Nov. 2014.

[237] S.-H. Huang, Y.-H. Huang, C. A. Blazquez, and G. Paredes-Belmar, "Application of the ant colony optimization in the resolution of the bridge inspection routing problem," *Appl. Soft Comput.*, vol. 65, pp. 443–461, Apr. 2018.

[238] T. Stützle and H. H. Hoos, "MAX-MIN ant system," *Future Generat. Comput. Syst.*, vol. 16, no. 8, pp. 889–914, Jun. 2000.

[239] M. Karakaya, "UAV route planning for maximum target coverage," *Comput. Sci. Eng., Int. J.*, vol. 4, no. 1, pp. 27–34, Feb. 2014.

[240] G. S. Tewolde and W. Sheng, "Robot path integration in manufacturing processes: Genetic algorithm versus ant colony optimization," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 38, no. 2, pp. 278–287, Mar. 2008.

[241] W. Chen, J. Liu, Y. Tang, and H. Ge, "Automatic spray trajectory optimization on Bézier surface," *Electronics*, vol. 8, no. 2, pp. 1–16, Feb. 2019.

[242] C. Gao, Y. Kou, Z. Li, A. Xu, Y. Li, and Y. Chang, "Optimal multirobot coverage path planning: Ideal-shaped spanning tree," *Math. Problems Eng.*, vol. 2018, pp. 1–10, Sep. 2018.

[243] J. Dentler, M. Rosalie, G. Danoy, P. Bouvry, S. Kannan, M. A. Olivares-Mendez, and H. Voos, "Collision avoidance effects on the mobility of a UAV swarm using chaotic ant colony with model predictive control," *J. Intell. Robotic Syst.*, vol. 93, nos. 1–2, pp. 227–243, Feb. 2019.

[244] A. V. Le, P. C. Ku, T. T. Tun, N. H. K. Nhan, Y. Shi, and R. E. Mohan, "Realization energy optimization of complete path planning in differential drive-based self-reconfigurable floor cleaning robot," *Energies*, vol. 12, no. 6, pp. 1–23, Mar. 2019.

[245] A. V. Le, N. H. K. Nhan, and R. E. Mohan, "Evolutionary algorithm-based complete coverage path planning for tetriamond tiling robots," *Sensors*, vol. 20, no. 2, pp. 1–14, Jan. 2020.
[246] A. V. Le, R. Parween, R. E. Mohan, N. H. K. Nhan, and R. E. Abdulkader, "Optimization complete area coverage by reconfiguring hTrihex tiling robot," *Sensors*, vol. 20, no. 11, pp. 1–20, Jun. 2020.
[247] G. Han, Z. Zhou, T. Zhang, H. Wang, L. Liu, Y. Peng, and M. Guizani, "Ant-colony-based complete-coverage path-planning algorithm for underwater gliders in ocean areas with thermoclines," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8959–8971, Aug. 2020.
[248] I. Caliskanelli, B. Broecker, and K. Tuyls, "Multi-robot coverage: A bee pheromone signaling approach," in *Artificial Life and Intelligent Agents* (Communications in Computer and Information Science), vol. 519, C. J. Headleand, W. J. Teahan, and L. A. Cenydd, Eds. Cham, Switzerland: Springer, 2015, pp. 124–140.
[249] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *J. Global Optim.*, vol. 39, no. 3, pp. 459–471, Apr. 2007.
[250] B. Broecker, I. Caliskanelli, K. Tuyls, E. I. Sklar, and D. Hennes, "Hybrid insect-inspired multi-robot coverage in complex environments," in *Towards Autonomous Robotic Systems* (Lecture Notes in Computer Science), vol. 9287, C. Dixon and K. Tuyls, Eds. Cham, Switzerland: Springer, 2015, pp. 56–68.
[251] X. S. Yang, *Nature-Inspired Metaheuristic Algorithms*. Frome, U.K.: Luniver Press, 2008.
[252] F. De Rango, N. Palmieri, X. S. Yang, and S. Marano, "Bio-inspired exploring and recruiting tasks in a team of distributed robots over mined regions," in *Proc. Int. Symp. Perform. Eval. Comput. Telecommun. Syst. (SPECTS)*, Chicago, IL, USA, Jul. 2015, pp. 1–8.
[253] N. Palmieri, F. de Rango, X. She Yang, and S. Marano, "Multi-robot cooperative tasks using combined nature-inspired techniques," in *Proc. 7th Int. Joint Conf. Comput. Intell.*, Lisbon, Portugal, Nov. 2015, pp. 74–82.
[254] N. Palmieri, X.-S. Yang, F. D. Rango, and S. Marano, "Comparison of bio-inspired algorithms applied to the coordination of mobile robots considering the energy consumption," *Neural Comput. Appl.*, vol. 31, no. 1, pp. 263–286, Jan. 2019.
[255] M. M. Gangadharan and A. Salgaonkar, "Ant colony optimization and firefly algorithms for robotic motion planning in dynamic environments," *Eng. Rep.*, vol. 2, no. 3, pp. 1–23, Mar. 2020.
[256] J. Henrio, T. Deligne, T. Nakashima, and T. Watanabe, "Route planning for multiple surveillance autonomous drones using a discrete firefly algorithm and a Bayesian optimization method," *Artif. Life Robot.*, vol. 24, no. 1, pp. 100–105, Mar. 2019.
[257] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
[258] K. Albina and S. G. Lee, "Hybrid stochastic exploration using Grey Wolf optimizer and coordinated multi-robot exploration algorithms," *IEEE Access*, vol. 7, pp. 14246–14255, 2019.
[259] R. K. Dewangan, A. Shukla, and W. W. Godfrey, "Three dimensional path planning using grey wolf optimizer for UAVs," *Int. J. Speech Technol.*, vol. 49, no. 6, pp. 2201–2217, Jun. 2019.
[260] A. Kamalova, S. Navruzov, D. Qian, and S. G. Lee, "Multi-robot exploration based on multi-objective grey wolf optimizer," *Appl. Sci.*, vol. 9, no. 14, pp. 1–19, Jul. 2019.
[261] F. Ge, K. Li, W. Xu, and Y. Wang, "Path planning of UAV for oilfield inspection based on improved grey wolf optimization algorithm," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Nanchang, China, Jun. 2019, pp. 3666–3671.
[262] A. Kamalova, K. D. Kim, and S. G. Lee, "Waypoint mobile robot exploration based on biologically inspired algorithms," *IEEE Access*, vol. 8, pp. 190342–190355, 2020.
[263] A. R. Mehrabian and C. Lucas, "A novel numerical optimization algorithm inspired from weed colonization," *Ecol. Inform.*, vol. 1, no. 4, pp. 355–366, Dec. 2006.
[264] P. K. Mohanty and D. R. Parhi, "A new efficient optimal path planner for mobile robot based on invasive weed optimization algorithm," *Frontiers Mech. Eng.*, vol. 9, no. 4, pp. 317–330, Aug. 2014.
[265] P. K. Mohanty, S. Kumar, and D. R. Parhi, "A new ecologically inspired algorithm for mobile robot navigation," in *Proc. 3rd Int. Conf. Frontiers Intell. Comput. Appl.*, in Advances in Intelligent Systems and Computing, vol. 327, S. Satapathy, B. Biswal, S. Udgata, and J. Mandal, Eds. Cham, Switzerland: Springer, 2014, pp. 755–762.
[266] Y. Zhou, Q. Luo, H. Chen, A. He, and J. Wu, "A discrete invasive weed optimization algorithm for solving traveling salesman problem," *Neurocomputing*, vol. 151, no. 3, pp. 1227–1236, Mar. 2015.
[267] M. R. Ghalenoei, H. Hajimirsadeghi, and C. Lucas, "Discrete invasive weed optimization algorithm: Application to cooperative multiple task assignment of UAVs," in *Proc. 48h IEEE Conf. Decis. Control (CDC), 28th Chin. Control Conf.*, Shanghai, China, Dec. 2009, pp. 1665–1670.
[268] Y. Zhuang, C. Wu, H. Wu, H. Chu, Y. Gao, and L. Li, "The repair strategy for event coverage holes based on mobile robots in wireless sensor and robot networks," *Sensors*, vol. 19, no. 22, pp. 1–21, Nov. 2019.
[269] K. Sandamurthy and K. Ramanujam, "A hybrid weed optimized coverage path planning technique for autonomous harvesting in cashew orchards," *Inf. Process. Agricult.*, vol. 7, no. 1, pp. 152–164, Mar. 2020.
[270] S. X. Yang and C. Luo, "A neural network approach to complete coverage path planning," *IEEE Trans. Syst., Man, Cybern.*, vol. 34, no. 1, pp. 718–725, Jan. 2004.
[271] C. Luo and S. X. Yang, "A bioinspired neural network for real-time concurrent map building and complete coverage robot navigation in unknown environments," *IEEE Trans. Neural Netw.*, vol. 19, no. 7, pp. 1279–1298, Jul. 2008.
[272] G. Hu, Z. Hu, and H. Wang, "Complete coverage path planning for road cleaning robot," in *Proc. Int. Conf. Netw., Sens. Control (ICNSC)*, Chicago, IL, USA, Apr. 2010, pp. 643–648.
[273] K. Chen and Y. Liu, "Optimal complete coverage planning of wall-climbing robot using improved biologically inspired neural network," in *Proc. IEEE Int. Conf. Real-time Comput. Robot. (RCAR)*, Okinawa, Japan, Jul. 2017, pp. 587–592.
[274] M. Z. Yan, D. Q. Zhu, and S. X. Yang, "Complete coverage path planning in an unknown underwater environment based on D-S data fusion real-time map building," *Int. J. Distrib. Sensor Netw.*, vol. 8, no. 10, pp. 1–11, Oct. 2012.
[275] C. Luo, S. X. Yang, X. Li, and M. Q.-H. Meng, "Neural-dynamics-driven complete area coverage navigation through cooperation of multiple mobile robots," *IEEE Trans. Ind. Electron.*, vol. 64, no. 1, pp. 750–760, Jan. 2017.
[276] C. Yang, Y. Tang, L. Zhou, and X. Ma, "Complete coverage path planning based on bioinspired neural network and pedestrian location prediction," in *Proc. IEEE 8th Annu. Int. Conf. Technol. Autom., Control, Intell. Syst. (CYBER)*, Tianjin, China, Jul. 2018, pp. 528–533.
[277] A. Singha, A. K. Ray, and A. B. Samaddar, "Neural dynamics-based complete coverage of grid environment by mobile robots," in *Proc. Int. Conf. Frontiers Comput. Syst.*, in Advances in Intelligent Systems and Computing, vol. 1255, D. Bhattacharjee, D. K. Kole, N. Dey, S. Basu, and D. Plewczynski, Eds. Singapore: Springer, 2021, pp. 411–421.
[278] D. Zhu, C. Tian, B. Sun, and C. Luo, "Complete coverage path planning of autonomous underwater vehicle based on GBNN algorithm," *J. Intell. Robotic Syst.*, vol. 94, no. 1, pp. 237–249, Apr. 2019.
[279] B. Sun, X. Zhu, W. Zhang, D. Zhu, and Z. Chu, "Three dimensional AUV complete coverage path planning with Glasius bio-inspired neural network," in *Intelligent Robotics and Applications* (Lecture Notes in Artificial Intelligence), vol. 10985, Z. Chen, A. Mendes, Y. Yan, and S. Chen, Eds. Cham, Switzerland: Springer, Aug. 2018, pp. 125–136.
[280] B. Sun, D. Zhu, C. Tian, and C. Luo, "Complete coverage autonomous underwater vehicles path planning based on Glasius bio-inspired neural network algorithm for discrete and centralized programming," *IEEE Trans. Cogn. Develop. Syst.*, vol. 11, no. 1, pp. 73–84, Mar. 2019.
[281] B. Kwon and J. Thangavelautham, "Autonomous coverage path planning using artificial neural tissue for aerospace applications," in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, USA, Mar. 2020, pp. 1–10.
[282] S. M. B. P. Samarakoon, M. A. V. J. Muthugala, A. V. Le, and M. R. Elara, "hTetro-Infi: A reconfigurable floor cleaning robot with infinite morphologies," *IEEE Access*, vol. 8, pp. 69816–69828, 2020.
[283] M. A. V. J. Muthugala, S. M. B. P. Samarakoon, and M. R. Elara, "Trade-off between area coverage and energy usage of a self-reconfigurable floor cleaning robot based on user preference," *IEEE Access*, vol. 8, pp. 76267–76275, 2020.
[284] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.
[285] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1238–1274, Sep. 2013.
[286] A. Gosavi, "Solving Markov decision processes via simulation," in *Handbook of Simulation Optimization: International Series in Operations Research & Management Science*, vol. 216, M. C. Fu, Ed. New York, NY, USA: Springer, 2015, pp. 341–379.

[287] E. Barrett and S. Linder, "Autonomous HVAC control, a reinforcement learning approach," in *Machine Learning and Knowledge Discovery in Databases* (Lecture Notes in Computer Science), vol. 9286, A. Bifet, M. May, B. Zadrozny, R. Gavalda, D. Pedreschi, F. Bonchi, J. Cardoso, and M. Spiliopoulou, Eds. Cham, Switzerland: Springer, 2015, pp. 3–19.

[288] R. Shakeri, M. A. Al-Garadi, A. Badawy, A. Mohamed, T. Khattab, A. K. Al-Ali, K. A. Harras, and M. Guizani, "Design challenges of multi-UAV systems in cyber-physical applications: A comprehensive survey and future directions," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3340–3385, Jun. 2019.

[289] W. Jing, C. F. Goh, M. Rajaraman, F. Gao, S. Park, Y. Liu, and K. Shimada, "A computational framework for automatic online path generation of robotic inspection tasks via coverage planning and reinforcement learning," *IEEE Access*, vol. 6, pp. 54854–54864, 2018.

[290] A. V. Le, R. Parween, P. T. Kyaw, R. E. Mohan, T. H. Q. Minh, and C. S. C. S. Borusu, "Reinforcement learning-based energy-aware area coverage for reconfigurable hRombo tiling robot," *IEEE Access*, vol. 8, pp. 209750–209761, 2020.

[291] X. Xia, T. Roppel, J. Y. Hung, J. Zhang, S. C. G. Periaswamy, and J. Patton, "Balanced map coverage using reinforcement learning in repeated obstacle environments," in *Proc. 29th IEEE Int. Symp. Ind. Electron.*, Delft, The Netherlands, Jun. 2020, pp. 41–48.

[292] L. Piardi, J. Lima, A. I. Pereira, and P. Costa, "Coverage path planning optimization based on Q-learning algorithm," in *Proc. CENTRAL Eur. Symp. Thermophys. (CEST)*, Rhodes, Greece, Sep. 2019, pp. 1–4.

[293] J. Xiao, G. Wang, Y. Zhang, and L. Cheng, "A distributed multi-agent dynamic area coverage algorithm based on reinforcement learning," *IEEE Access*, vol. 8, pp. 33511–33521, 2020.

[294] L. Tai and M. Liu, "Mobile robots exploration through CNN-based reinforcement learning," *Robot. Biomimetics*, vol. 3, no. 1, pp. 1–8, Dec. 2016.

[295] J. Xin, H. Zhao, D. Liu, and M. Li, "Application of deep reinforcement learning in mobile robot path planning," in *Proc. Chin. Autom. Congr. (CAC)*, Jinan, China, Oct. 2017, pp. 7112–7116.

[296] S. Y. Luis, D. G. Reina, and S. L. T. Marin, "A deep reinforcement learning approach for the patrolling problem of water resources through autonomous surface vehicles: The Ypacarai lake case," *IEEE Access*, vol. 8, pp. 204076–204093, 2020.

[297] C. Piciarelli and G. L. Foresti, "Drone patrolling with reinforcement learning," in *Proc. 13th Int. Conf. Distrib. Smart Cameras*, Trento, Italy, Sep. 2019, pp. 1–6.

[298] X. Chen, T. M. Tucker, T. R. Kurfess, and R. Vuduc, "Adaptive deep path: Efficient coverage of a known environment under various configurations," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Macau, China, Nov. 2019, pp. 3549–3556.

[299] C. Wang, L. Wei, Z. Wang, M. Song, and N. Mahmoudian, "Reinforcement learning-based multi-AUV adaptive trajectory planning for under-ice field estimation," *Sensors*, vol. 18, no. 11, pp. 1–19, Nov. 2018.

[300] J. Hu, H. Niu, J. Carrasco, B. Lennox, and F. Arvin, "Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 14413–14423, Oct. 2020.

[301] F. Niroui, K. Zhang, Z. Kashino, and G. Nejat, "Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 610–617, Apr. 2019.

[302] X. Cao, C. Sun, and M. Yan, "Target search control of AUV in underwater environment with deep reinforcement learning," *IEEE Access*, vol. 7, pp. 96549–96559, 2019.

[303] A. K. Lakshmanan, R. E. Mohan, B. Ramalingam, A. V. Le, P. Veerajagadeshwar, K. Tiwari, and M. Ilyas, "Complete coverage path planning using reinforcement learning for Tetromino based cleaning and maintenance robot," *Autom. Construct.*, vol. 112, pp. 1–11, Apr. 2020.

[304] 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.

[305] A. Koval, S. S. Mansouri, and G. Nikolakopoulos, "Online multi-agent based cooperative exploration and coverage in complex environment," in *Proc. 18th Eur. Control Conf. (ECC)*, Naples, Italy, Jun. 2019, pp. 3964–3969.

[306] F. Balampanis, I. Maza, and A. Ollero, "Spiral-like coverage path planning for multiple heterogeneous UAS operating in coastal regions," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Miami, FL, USA, Jun. 2017, pp. 617–624.

[307] M. Hassan and D. Liu, "A deformable spiral based algorithm to smooth coverage path planning for marine growth removal," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Madrid, Spain, Oct. 2018, pp. 1913–1918.

[308] M. Hassan, D. Liu, and X. Chen, "Squircular-CPP: A smooth coverage path planning algorithm based on squircular fitting and spiral path," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechtron. (AIM)*, Boston, MA, USA, Jul. 2020, pp. 1075–1081.

[309] V. G. Nair and K. R. Guruprasad, "GM-VPC: An algorithm for multi-robot coverage of known spaces using generalized Voronoi partition," *Robotica*, vol. 38, no. 5, pp. 845–860, May 2020.

[310] E. Ferranti, N. Trigoni, and M. Levene, "Brick& mortar: An on-line multi-agent exploration algorithm," in *Proc. IEEE Int. Conf. Robot. Autom.*, Roma, Italy, Apr. 2007, pp. 761–767.

[311] M. Becker, F. Blatt, and H. Szczerbicka, "A multi-agent flooding algorithm for search and rescue operations in unknown terrain," in *Multiagent System Technologies* (Lecture Notes in Computer Science), vol. 8076, M. Klusch, M. Thimm, and M. Paprzycki, Eds. Berlin, Germany: Springer, 2013, pp. 19–28.

[312] F. Blatt and H. Szczerbicka, "Combining the multi-agent flood algorithm with frontier-based exploration in search & rescue applications," in *Proc. Int. Symp. Perform. Eval. Comput. Telecommun. Syst. (SPECTS)*, Seattle, WA, USA, Jul. 2017, pp. 1–7.

[313] Z. Xiao, Z. Wang, D. Liu, Z. Niu, and D. Cao, "A PCB-oriented path planning for AOI full coverage field of view," in *Proc. IEEE 3rd Adv. Inf. Manage., Communicates, Electron. Autom. Control Conf. (IMCEC)*, Chongqing, China, Oct. 2019, pp. 586–592.

[314] C. V. Meaclem, X. Q. Chen, S. Gutschmidt, C. Hann, and R. Parker, "K-means partitioned space path planning (KPSPP) for autonomous robotic harvesting," *Int. J. Adv. Robotic Syst.*, vol. 12, no. 11, pp. 1–10, Nov. 2015.

[315] Y. Ding, R. Cao, S. Liang, F. Qi, Q. Yang, and W. Yan, "Density-based optimal UAV path planning for photovoltaic farm inspection in complex topography," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Hefei, China, Aug. 2020, pp. 3931–3936.

[316] Y. Tang, R. Zhou, G. Sun, B. Di, and R. Xiong, "A novel cooperative path planning for multirobot persistent coverage in complex environments," *IEEE Sensors J.*, vol. 20, no. 8, pp. 4485–4495, Apr. 2020.

[317] X. Miao, J. Lee, and B.-Y. Kang, "Scalable coverage path planning for cleaning robots using rectangular map decomposition on large environments," *IEEE Access*, vol. 6, pp. 38200–38215, 2018.

[318] Y. Ma, H. Sun, P. Ye, and C. Li, "Mobile robot multi-resolution full coverage path planning algorithm," in *Proc. 5th Int. Conf. Syst. Informat. (ICSAI)*, Nanjing, China, Nov. 2018, pp. 120–125.

[319] H. Liang, W. Gao, J. H. Nguyen, M. F. Orpilla, and W. Yu, "Internet of Things data collection using unmanned aerial vehicles in infrastructure free environments," *IEEE Access*, vol. 8, pp. 3932–3944, 2020.

[320] X. Dai, L. Jiang, and Y. Zhao, "Cooperative exploration based on supervisory control of multi-robot systems," *Int. J. Speech Technol.*, vol. 45, no. 1, pp. 18–29, Jul. 2016.

[321] J. Song and S. Gupta, "ε^*: An online coverage path planning algorithm," *IEEE Trans. Robot.*, vol. 34, no. 2, pp. 526–533, Apr. 2018.

[322] Z. Shen, J. P. Wilson, and S. Gupta, "An online coverage path planning algorithm for curvature-constrained AUVs," in *Proc. OCEANS MTS/IEEE SEATTLE*, Seattle, WA, USA, Oct. 2019, pp. 1–5.

[323] E. Vidal, N. Palomeras, K. Istenič, N. Gracias, and M. Carreras, "Multi-sensor online 3D view planning for autonomous underwater exploration," *J. Field Robot.*, vol. 37, no. 6, pp. 1123–1147, Sep. 2020.

[324] B. Zhou, Y. Zhang, X. Chen, and S. Shen, "FUEL: Fast UAV exploration using incremental frontier structure and hierarchical planning," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 779–786, Apr. 2021.

[325] S. Godio, S. Primatesta, G. Guglieri, and F. Dovis, "A bioinspired neural network-based approach for cooperative coverage planning of UAVs," *Information*, vol. 12, no. 2, pp. 1–17, Jan. 2021.

[326] M. Theile, H. Bayerlein, R. Nai, D. Gesbert, and M. Caccamo, "UAV coverage path planning under varying power constraints using deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Las Vegas, NV, USA, Oct. 2020, pp. 1–6.

[327] S. G. Potapova, A. V. Artemov, S. V. Sviridov, D. A. Musatkina, D. N. Zorin, and E. V. Burnaev, ''Next best view planning via reinforcement learning for scanning of arbitrary 3D shapes,'' *J. Commun. Technol. Electron.*, vol. 65, no. 12, pp. 1484–1490, Dec. 2020.
[328] B. Nasirian, M. Mehrandezh, and F. Janabi-Sharifi, ''Efficient coverage path planning for mobile disinfecting robots using graph-based representation of environment,'' *Frontiers Robot. AI*, vol. 8, pp. 1–19, Mar. 2021.
[329] X. Kan, H. Teng, and K. Karydis, ''Online exploration and coverage planning in unknown obstacle-cluttered environments,'' *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5969–5976, Oct. 2020.
[330] L. Shi, S. Xu, H. Liu, and Z. Zhan, ''QoS-aware UAV coverage path planning in 5G mmWave network,'' *Comput. Netw.*, vol. 175, pp. 1–10, Jul. 2020.

CHEE SHENG TAN received the B.Eng. and M.S. degrees in mechatronics engineering from Universiti Sains Malaysia, Pulau Pinang, Malaysia, in 2016 and 2019, respectively. He is currently pursuing the Ph.D. degree with the Underwater Control Robotics Research Group. He is a Graduate Research Assistant with the School of Electrical and Electronic Engineering, Universiti Sains Malaysia. His research interests include robot positioning, path planning, image processing, and underwater robotics technology.

ROSMIWATI MOHD-MOKHTAR (Senior Member, IEEE) received the B.Eng. (Hons.) and M.Sc. degrees in electrical and electronic engineering from Universiti Sains Malaysia (USM), in 1999 and 2002, respectively, and the Ph.D. degree in electrical engineering from the Royal Melbourne Institute of Technology University, Australia, in 2008. She joined the School of Electrical and Electronic Engineering, USM, in October 2008, where she is currently an Associate Professor and the Mechatronic Program Chairman with the School of Electrical and Electronic Engineering. Her research interests include system identification, advanced control system design, process modeling, process optimization, mechatronics applications, and underwater system applications. She is a Chartered Engineer (C.Eng.) of the Institution of Engineering and Technology (IET), a Professional Engineer (P.Eng.) of the Board of Engineers Malaysia (BEM), and a member of the Institution of Engineers Malaysia (IEM).

MOHD RIZAL ARSHAD (Senior Member, IEEE) received the B.Eng. degree in medical electronics and instrumentation from the University of Liverpool, U.K., in July 1994, the M.Sc. degree in electronic control engineering from the University of Salford, in 1995, and the Ph.D. degree in electronic engineering from the University of Liverpool, in January 1999. He is currently a Full Professor and the Deputy Vice-Chancellor in academic and international with Universiti Malaysia Perlis (UniMAP), Malaysia. His research interests include ocean robotics and instrumentation, control, and intelligent system for industrial applications.