Online Multi-Agent Based Cooperative Exploration and Coverage in Complex Environment

Anton Koval¹, Sina Sharif Mansouri² and George Nikolakopoulos²

Abstract—In this article, a novel online collaborative task assignment method is proposed for the exploration and coverage of unknown complex indoor environment with multiple agents. The proposed method is a sensor based approach and it is able to take under consideration the battery lifetime limitation of the agents and assign feasible, from a realistic mission planning perspective, tasks to each of them, including the capability of providing reachable, collision free paths. The efficacy of the proposed scheme will be extensively evaluated by multiple simulation results in a complex unknown environment.

I. INTRODUCTION

Nowadays, exploration and complete coverage of unknown areas is receiving a considerable high research attention. In the following survey articles [1], [2] numerous approaches have been proposed for solving this demanding task, while in general the task of exploration could be divided into online and offline approaches. The offline algorithms typically requires prior knowledge of the environment that in many real life use-cases is hardly achievable. In contrary, online algorithms or sensor-based algorithms can operate without prior knowledge of the environment, however the outcome in this case is reaching sub-optimal solutions.

Towards the second direction, one of the most known online algorithm for achieving a complete area coverage is based on the Boustrophedon motion [3]. For example, in [4] this algorithm was utilized together with the dynamic programming for a single mobile root. The authors in [5] proposed an algorithm that incrementally decomposed the unknown environment into cells and the corresponding online path planning relied on data from a laser scanner. In [6] it was suggested the worst case scenario model for planning coverage paths that could be applicable for GPS enabled robots, while the work in [7] suggested to use a genetic algorithm for achieving a coverage based path planning. In [8], a path planning algorithm was proposed that was able to minimize the number of turns or repetitions in order to increase the overall algorithmic efficiency. However, one of the main shortcomings of these works was the fact that the physical limitations of the utilized robots, from a power consumption point of view, have not been taken under consideration, while at the same time was focusing

on the single robot use-case. In contrary, studies as those in [9], [10], [11] suggested approaches that considered limited robot resources, but still were solving the coverage problem only for the case of a single agent.

In the field of collaborative area coverage, significant contributions have been reported in [12]–[15], where it was suggested an online coverage algorithm and a cost function was proposed based on distance measurements, however the reality of a limited battery life has not been taken under consideration. In [16] it was proposed a cooperative mapping algorithm, based on heterogeneous robots and the established algorithm in [3], [17], [18] with an explorer-coverer architecture. As it was presented, during the stage of the cooperative exploration, the agents discover the unknown region boundaries, while the task finishes when the Reeb Graph [19] is complete. In this approach, the covers execute a back-and-forth motion (sweeping motion). For this scenario an auctioning mechanism is used for the task re-allocation. But in both works the authors do not consider a limited battery life. Moreover, the approach suggested in [20] uses multiple heterogeneous unmanned aerial vehicles for the area partitioning, while in this case the proposed computational algorithm requires future optimizations and is an offline approach.

The aim of this article is to tackle the objective to explore at a maximum coverage and with multiple agents a complex 2D environment with no a priori knowledge and by taking under consideration the limited battery life time of the agents. This contribution can be directly applied on multiple real life use cases, such as the ones in underground mine inspection [21], earthquake-damaged building inspection [22] and subterranean exploration [23]. Thus, the main contribution of this article stems from the deployment of multiple agents in complex indoor environment with a sensor based approach, where the exploration method is inspired from the Boustrophedon Motion (BM) proposed in [13] and originally presented in [3] and expanded by considering multiple agents and with a limited battery life. Additionally, to determine backtracking points, the detection conditions in [24] have been modified to meet the overall mission requirements, while the overall swarm approach is based on the work in [25]. As it will be demonstrated the proposed method is evaluated in simulation scenarios inspired from real-world underground mine areas with multiple branches.

The rest of the article is structured as it follows. Initially, the notations and preliminaries are presented in Section II, while the proposed multi-agent based cooperative exploration and coverage scheme is established in Section III. The

*This work has been partially funded by the European Unions Horizon 2020 Research and Innovation Programme under the Grant Agreement No. 730302 SIMS and the SI VISBY programme

¹Department of Automation and Computer Integrated Technologies, Zhytomyr State Technological University, Ukraine, Email: koval.anton@gmail.com

²Robotics Team, Department of Computer, Electrical and Space Engineering, Luleå University of Technology, Luleå, Sweden, Emails: {sinsha,geonik}@ltu.se

simulation results are presented in Section IV and finally the article is concluded in Section V.

II. NOTATION AND PRELIMINARIES

In the proposed method, each agent is able to cover a squared area of same size and with a $l \in \mathbb{R}_{>0}$ length, where it is assumed that the agent is placed in the middle of the defined area and thus the result of the exploration task is a grid map, obtained with the same resolution of the square size. Furthermore and without a loss of generality, it is assumed that the movement of the agent is defined with four directions, specifically North, South, East and West and rotation around its geometric center. In each step, the agents can move in a fixed distance l and rotate by $\alpha = \{\pm\frac{\pi}{2}, \pm\pi\}$. The pose of the $i \in \mathbb{N}$ agent can be defined by the $q_{i,k} = [x_{i,k}, y_{i,k}, \theta_{i,k}]^T$ vector, where $x, y \in \mathbb{R}^2$ are the coordinates and θ is the heading at the time instant $k \in \mathbb{N}$, with the corresponding kinematic equations denoted by:

$$q_{i,k+1} = \begin{bmatrix} x_{i,k} \\ y_{i,k} \\ \theta_{i,k} \end{bmatrix} + \begin{bmatrix} l \sin \theta \\ l \cos \theta \\ \alpha \end{bmatrix} \quad (1)$$

The workspace W and its coverage model M are in $\mathbb{R}^2$, while the sets C within the workspace are denoted as $\mathbb{R}^{2 \times m}$ and $C \subset W$ and C_k^f , C_k^o and C_k^{global} denote free space, obstacles and global backtracking points sets respectively. The transpose of a matrix $M \in \mathbb{R}^{n \times m}$ is denoted as $M^T \in \mathbb{R}^{m \times n}$. The superscripts *chp*, *BTP*, *min*, *total* and *acc* denote a charging point, a backtracking point, a minimum, a total and an accumulated value correspondingly. The energy $E \in \mathbb{R}$ is denoted as a scalar value, while $E_{i,k}$ is the energy level of the agent i and at the k time instant, $E_{i,k}^{tr}$ is the energy spent for the translation from $q_{i,k}$ to $q_{i,k+1}$. Furthermore, a lost agent is defined as $q_{i,k} = \emptyset$, a blocked position as $q_{i,k} = \infty$ and an unblocked position as $q_{i,k} \neq \infty$. The path from $q_{i,k}$ to $q_{i,k+1}$ is defined as $P_{i,k} = \{p_{i,k} | p_{i,k} = (x_{i,k}, y_{i,k}) \in \mathbb{R}^2\}$, where $p_{i,k}$ is the position of the agent and $P_{i,k}$ is set of all the agents positions. Furthermore, $d_{i,k}$ denotes the Euler distance between the position $q_{i,k}$ and $q_{i,k+1}$. The positions of each discovered branch in the evaluation areas will be denoted as p_l^b and the corresponding distances to them as d_l^b and with $n \in \mathbb{N}$ to denote the number of agents. Finally, $s_j, j \in \{1, 2, \dots, 8\}$ is the sensor model and operator $\setminus$ denotes set subtraction.

III. COOPERATIVE EXPLORATION AND COVERAGE SCHEME

In the proposed methodology, it is assumed that the environment has only one entrance, which is considered as the starting point and might contain undiscovered multiple branches. It is also assumed that all the discovered branches within the environment have at least the same width as the agent's sweep step. At the starting point all agents are looking towards the main tunnel and in each iteration the agent completely covers the next cell, while the planner presented in [26] is used to generate collision free trajectories.

Generally, the proposed exploration scheme contains three main parts: a) an exploration task, b) a staying alive policy, and c) a path planning part, while these parts will be analyzed in the sequel.

A. Exploration Method

In the proposed methodology, the unknown areas of the workspace W are discovered incrementally through the Boustrophedon Motion [13], so that the agent translates forward or backward, exploring the surrounding space, until reaches a blocked position and rotates to alter the direction of the translation. During exploration, each agent and at each step, performs sensing of the surrounding environment, denoted by $s_{i,k}$, according to the sensor model defined as:

$$s_j = \begin{cases} 1, & \text{if } s_j \in C_{i,k}^f \\ 0, & \text{if } s_j \in C_{i,k}^o \end{cases} \quad (2)$$

where $j \in [1, 2, \dots, 8]$ - corresponds to each adjacent area around the agent. After the completion of the overall unknown area it will be represented as a grid map (global map, coverage model) M [24] with discovered free space $C_{i,k}^f$ and obstacles $C_{i,k}^o$, with the overall architecture depicted in Figure 1.

Fig. 1: General scheme of the proposed approach, where q denotes the agents, C_k^f denotes the free area, C_k^{btp} denotes the backtracking points, C_k^o denotes the obstacles, and the undiscovered area is presented in a white color.

B. Staying Alive Policy

Relying on the $s_{i,k}$, the agents perform the BM according to the algorithm in 1 and detect the corresponding Back Tracking Points (BTP) that in generally can be any agent's neighbouring point that satisfies the following statement $C_{i,k}^{BTP} \not\subset (C_{i,k}^f \cup C_{i,k}^o)$, while it should be mentioned that these points are utilized for the detection of the map branches if existing. Without a proper selection mechanism for the BTPs, the total agent's path can have a lot of unnecessary transitions that will increase the overall distance and computation time. This article proposes the novel idea to reduce the number of the utilized BTPs, as appeared in [3], [24], where data from all the neighbouring points are used to detect a BTP. In this research direction, sensor measurements are used to detect BTPs, but novel detection conditions are proposed as depicted at Figure 2, an approach that requires less computations time than the previous approaches [13],

Algorithm 1 Boustrophedon motion (BM) algorithm

Require: $q_{i,k}, M$

- 1: **Outputs:** $M \cup C_{i,k}^o \cup C_{i,k}^f, C_{i,k}^{BTP}, q_{i,k+1}$
 - 2: **Step 1.** Find North, South, East or West direction from $s_{i,k} = s_j$
 - 3: **Step 2.** Motion according to (1)
 - 4: **Step 3.** Update $M \cup C_{i,k}^o \cup C_{i,k}^f$
 - 5: **Step 4.** Obtain $C_{i,k}^{BTP}, q_{i,k+1}$
 - 6: **Step 5.** Go to Step 1.
-

[15], [24]. Thus, for detecting BTPs, data from 3 surrounding cells instead of 8, is utilized, which directly reduces the number of computations needed. In Figure 2, the black tiles denote obstacles, the red tiles correspond to BTPs

Fig. 2: The conditions of BTP detection

and the gray tiles are free areas that are not considered for detection, while the green arrows indicate the agent's motion direction, when the BTP can be detected, with the agent placed in the middle (white tile) and the neighbouring cells are sensor measurements, unlike the approach proposed in [3], [24] where the backtracking point is in the middle. For making the overall approach more clear, let's consider left condition in the middle row of Figure 2 which uses 3 out of the 8 measured points to calculate the BTP. In this case the green arrows indicate that the BTP can be detected, while moving to the North or to the South. After obtaining the sensor measurements s_j , it results that $s_7 \in C_{i,k}^o$ and $\{s_j \setminus s_7\} \in C_{i,k}^f$. To obtain the motion direction, the s_2 should be considered and if the forward or backward direction is not blocked then $s_8 \in C_{i,k}^{BTP}$. Depending on the complexity of the unknown area the number of backtracking points could be proportional to the number of agent steps, which in turn will require additional computational power. Therefore, in the proposed work we consider that while the agent performs BM it discovers and covers BTPs, while the already visited BTPs are removed from the agent's BTP stack, an action that can be defined as:

$$C_{i,k}^{BTP} = C_{i,k}^{BTP} \setminus C_{i,k}^f \quad (3)$$

At the same time, each agent locally operates based on his own BTP stack ($C_{i,k}^{BTP}$) and globally uses data from the global BTP stack shared among the agents as C_k^{global}

if $C_{i,k}^{BTP} = \emptyset$) is the overall map knowledge, and relies on its sensor measurements $s_{i,k}$ to provide non overlapping coverage paths. In case there are more agents than BTPs, a free agent is waiting for the first available BTP in the global stack. The global map M (coverage model) is updated each iteration from all agents, where the smallest area to be discovered corresponds to agent's cover area per one step, unless it makes a turn. The whole map model M of the unknown workspace W is discovered when agents have achieved a complete coverage of all the unknown areas, so the $C_k^{global} = \emptyset$.

In the proposed approach, it is considered that while performing BM, the agents never enter again in the discovered areas unless they reach a blocked position $q_{i,k} = \infty$, a case that is depicted in Figure 3. As an example, several conditions are depicted on the Figure 3, where it is required to generate the path for the agent to go to the next available backtracking point.

Fig. 3: The blocked conditions for the agent with forward motion. Green arrow depicts motion direction, white tiles denote agent's discovered path, gray tiles denote free space. The left image describes the case when the agent is blocked by obstacles (black tiles), the middle image depicts the cases when the agent is blocked with obstacles on the left and the neighbouring areas are covered by other agents. The image on the right depicts the case when all neighbouring regions were discovered by other agents.

In the proposed scheme, it is considered that the agents are powered by batteries with a limited battery life time [27] and thus mission planning without this consideration can cause loss of multiple agents [28] and even result in a global mission failure. Thus, in this work, the policy for staying alive is introduced to minimize this risk, while in the presented approach the remained energy is defined as $E_{i,k}^{tr}$ and it is evaluated at each BM step, as:

$$E_{i,k}^{tr} = e_i d_{i,k}, \quad (4)$$

where $e_i \in [1, 100]$ corresponds to the battery energy level $E_{i,k}^{total}$, where a value of 100 denotes a full charge and below 20 denotes the case of an empty battery or a lost agent (cannot be recovered). The minimum energy level required to return to the charging station is denoted by E_i^{min} , while the energy consumption for each agent can be defined as:

$$E_{i,k+1} = E_{i,k} - E_{i,k}^{tr}, \quad (5)$$

The overall algorithm is shown in 2, where the energies $E_{i,k}^{chp}$ and $E_{i,k}^{BTP}$ are calculated using (4). Thus, in every iteration and for the agent that requires charging $q_{i,k} = q_i^{chp}$, the proposed scheme will generate a path to the charging

Algorithm 2 Staying alive policy algorithm

Require: $d_{i,k}$, E_i^{min} , $q_{i,k-1}$, q_i^{chp}

```

1: for  $d_{i,k} = \{d_{i,k}^{chp}, d_{i,k}^{BTP}\}$  do
2:   if  $q_{i,k-1} \neq \infty$  then
3:     if  $E_{i,k} \leq E_{i,k}^{chp}$  then
4:        $q_{i,k} = q_i^{chp}$ 
5:     else if  $E_{i,k} < E_i^{min}$  then
6:        $q_{i,k} = \emptyset$ 
7:     end if
8:   else
9:     if  $E_{i,k} < E_{i,k}^{chp} + E_{i,k}^{BTP}$  then
10:       $q_{i,k} = q_i^{chp}$ 
11:    else
12:       $q_{i,k} = q_{i,k}^{BTP}$ 
13:    end if
14:   end if
15: end for
  
```

point, by using the path planner presented in the following subsection.

C. Path Planner

In general, the agents perform an exploration task based on BM. However, when the battery level of an agent is low, which is denoted from $E_{i,k} < E_i^{min}$ or the agent is blocked $q_{i,k} = \infty$, the agent should navigate to the charging station $q_{i,k+1} = q_i^{chp}$ or the closest BTP $q_{i,k+1} = C_{i,k}^{BTP}$ respectively. Thus, a Probabilistic Roadmap (PRM) planner [26] is implemented to generate collision free paths $P_{i,k}$ from $q_{i,k}$ to $q_{i,k+1}$ based on the discovered map M , as depicted in Figure 4.

Fig. 4: Example of the PRM planner, applied on a time instant of the explored area, where free (discovered) space is in white, undiscovered area is in gray, obstacles are in black, generated path is in orange. In this Figure, the graph with all the PRM based potential paths is also indicated by blue.

The PRM is based on non directed graph that contains all possible trajectories that are derived from the occupancy grid. Generally the PRM path planner has two phases: learning and query. During first phase it constructs a roadmap of a free space as an non directed graph, while in the second phase

the method tries to connect the $q_{i,k}$ and the $q_{i,k+1}$ positions by a corresponding path.

Furthermore, the path planner algorithm is presented in 3, the path planner requires the current position of the agent $q_{i,k}$, the position of charging station q_i^{chp} and the set of BTPs $C_{i,k}^{BTP}$. Then, in case of $q_{i,k} = \infty$ all paths to BTPs are calculated, while the BTP with a minimum length of the path is selected for the agent to visit. However, if the agent battery level is not sufficient, for reaching to $q_{i,k+1}$ based on Algorithm 2, the agent should navigate to the charging station. In case of $q_{i,k+1} = q_i^{chp}$, only one path from $q_{i,k}$ to the charging station is obtained.

Algorithm 3 Path planner

Require: $q_{i,k}$, q_i^{chp} , $C_{i,k}^{BTP}$

```

1:  $d_{i,k}^{min} = \inf$ 
2: if  $q_{i,k} == \infty$  then
3:   for each  $C_{i,k}^{BTP}$  do
4:      $(P_{i,k}, d_{i,k}) = PRM$ 
5:     if  $d_{i,k} \leq d_{i,k}^{min}$  then
6:        $d_{i,k}^{min} = d_{i,k}$ 
7:        $P_{i,k}^{min} = P_{i,k}$ 
8:     Check Algorithm 2
9:   end if
10:  end for
11: end if
12: if  $q_{i,k+1} == q_i^{chp}$  then
13:    $(P_{i,k}, d_{i,k}) = PRM$ 
14: end if
  
```

Overall, the final cooperative exploration and coverage scheme, by compining all the presented modules and considering the multi-agent case can be summarized in 4, where “==” denotes equality and with Figure 5 to represent an overview of the suggested scheme in a block diagram approach.

Fig. 5: The entire system design for multi-agent cooperative exploration and coverage

IV. SIMULATION RESULTS

The proposed methodology will be evaluated in simulation studies, where all the algorithms have been implemented in MATLAB R2018a, (in a single thread) on a platform with Intel Core i5-8250U and 16GB of RAM. Furthermore, it is assumed that the charging time for each agent was negligible compared to the exploration time, while all simulations were conducted for teams of $\{1, 3, 5, 7, 9, 11\}$ agents with same

Algorithm 4 Cooperative exploration and coverage algorithm with multiple agents

Require: $q_{i,k}, q_i^{chp}, M, n$

```

1: while  $C_k^{global} \neq \emptyset$  do
2:   Find path according to Algorithm 3
3:   Estimate charge level according to Algorithm 2
4:   if  $q_{i,k} \neq \{\emptyset, q_i^{chp}\}$  then
5:     Motion according to BM Algorithm 1
6:     Update BTP stacks according to (3)
7:     if  $q_{i,k} \in C_k^f$  then
8:       if  $C_k^{global} = \emptyset$  then
9:         End of mission.
10:      end if
11:      for  $i \in n$  do
12:        if  $C_{i,k}^{BTP} = \emptyset$  then
13:           $C_{i,k}^{BTP} = C_k^{global}$ 
14:        end if
15:        Find path according to Algorithm 3
16:      end for
17:    else if  $q_{i,k} = \infty$  then
18:      if  $C_k^{global} = \emptyset$  then
19:        End of mission.
20:      else if  $C_{i,k}^{BTP} = \emptyset$  then
21:         $C_{i,k}^{BTP} = C_k^{global}$ 
22:      end if
23:      Find path according to Algorithm 3
24:    end if
25:  else if  $q_{i,k} == \emptyset$  then
26:    Agent is lost.
27:    Launch new agent.
28:  else if  $q_{i,k} == q_i^{chp}$  then
29:    Find path according to Algorithm 3
30:    Go to Step 4.
31:  end if
32: end while

```

configuration and without any prior knowledge about the surrounding area in the same complex environment and as it will be presented in all the scenarios, the suggested cooperative exploration and coverage algorithms, succeeded to provide a complete area coverage.

The scenario shown in Figure 6 was based on a team of 5 agents. All agents were deployed from the same starting location (blue square at the bottom of the figures), which also stands as a charging point, while the discovered obstacles are shown in black. The red squares correspond to backtracking points with white faces, until are covered with an agent and the face color changes to the corresponding agent's color, in order to denote which parts of the area have been explored by a specific agent. The BTPs in the middle of the exploration area (points that are not adjacent with the obstacles) depict that the agent went to the charging point according to the staying alive policy.

Moreover, Figure 6 provides the evolution of the discovered area over the time with five agents, where it can be

observed that each agent uses its own BTP stack until its empty, then uses the BTP from the global BTP stack. As an example the $agent_4$ in 25% of the simulation execution time utilized its own local BTP (red squares), however, in 50% of the simulation time and after completing the local BTP, it moves to the global BTPs. For evaluating the overall impact of the number of agent on the coverage time, the exploration time, required for the full area coverage, with the multi-agent teams is presented on Figure 7.

As it can be observed from the obtained results, the larger the team is, it requires less iterations to cover the same area. So the exploration time reduces dramatically with more number of agents as it is straightforwardly expected. As an example, the 5 agent team, overtook the 3 agent team with 39% less coverage time. However, the case of an exploration with 11 agents is close to the time need to complete the same exploration with 9 agents and this indicates that there should be also an optimization towards the optimal number of utilized agents for inspecting a specific unknown area.

The average time required to execute one step at the overall Algorithm 4 of the proposed method is presented in Figure 8. As one can see, the average computational time required for each iteration is around 1 sec, with an linear fit of the obtained data denoted by $0.12n - 0.01$. Furthermore, it should be noted that the overall method is $O(n)$, which means that the rate of the computation time grows with the number of utilized agents.

On the Figure 9 is depicted the variation of agent's charge level over the iteration process. The lower bound of the agent's battery level E^{min} is depicted as solid horizontal line and in red color. As it can be observed at the top and the middle images, the charge level $E_{i,k}$ goes below the stated bound. This means that due to the large traveling distance from the charging point, the battery level dropped below 20%. Thus, the introduced staying alive policy proved its viability in all conducted simulations by guaranteeing 0% of agent loss. Further analysis revealed that the total number of charges per agent were reduced with a growth of the utilized agents.

On Figure 10 it is depicted a comparison of the total number of charges required to explore the current simulation scenario, where for the case of 1 agent, the total and average number of charges are the same. But for larger teams, we observe fluctuations e.g. in some cases the utilized team will have more charges than a single agent but in average it will be more efficient and will require less time for the entire exploration objective. Also the average number of charges per agent varies, so for example on Figure 9 $agent_1$ made 5 charges and $agent_3$ made 3 charges.

V. CONCLUSION

In this study we introduced a novel online coverage approach for multi-agent systems in unknown environment. The proposed algorithm had also the novelty to couple the agent's limited battery life with the path planner to guarantee a staying alive path planning for the multi-agent team. Additionally, a novel BTP point detection scheme for

Fig. 6: Area discovered with 5 agents, launched from the same location. The black areas are obstacles, while the red squares are BTPs.

Fig. 7: Explored area with teams of $\{1, 3, 5, 7, 9, 11\}$ agents

Fig. 8: Average computation time of the proposed Algorithm 4 per one step for teams of $\{1, 3, 5, 7, 9, 11\}$ agents

Fig. 9: Charge level time evolution for 1, 3 and 5 agents

Fig. 10: The total and average number of charges for teams of $\{1, 3, 5, 7, 9, 11\}$ agents

exploration task, while performing Boustrophedon motion, has been proposed. The overall proposed online algorithm was successfully evaluated in simulations in a complex environment. Trajectories provided by PRM planner were not always optimal and it requires more investigation towards local optimal path planning, while the implemented staying alive policy in all the simulations achieved a zero agent loss. The number of total charges required per agent for simulated scenario is generally decreasing with a larger team but there could be no guarantee that the total number of charges will be

less than those for the one agent case. Part of the future work includes the enhancement and fusion of the presented staying alive policy by the path planning algorithm based on a real discharging model and the full experimental verification of the suggested scheme with either mobile or aerial robots.

REFERENCES

- [1] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

- [2] A. Khan, I. Noreen, and Z. Habib, "On complete coverage path planning algorithms for non-holonomic mobile robots: Survey and challenges," *J. Inf. Sci. Eng.*, vol. 33, no. 1, pp. 101–121, 2017.
- [3] I. Rekleitis, A. P. New, E. S. Rankin, and H. Choset, "Efficient boustrophedon multi-robot coverage: an algorithmic approach," *Annals of Mathematics and Artificial Intelligence*, vol. 52, no. 2-4, pp. 109–142, 2008.
- [4] P. Zhou, Z.-m. Wang, Z.-n. Li, and Y. Li, "Complete coverage path planning of mobile robot based on dynamic programming algorithm," in *2nd international conference on electronic and mechanical engineering and information technology*, 2012, pp. 1837–1841.
- [5] D. Batsaikhan, A. Janchiv, and S.-G. Lee, "Sensor-based incremental boustrophedon decomposition for coverage path planning of a mobile robot," in *Intelligent Autonomous Systems 12*. Springer, 2013, pp. 621–628.
- [6] T. Bretl and S. Hutchinson, "Robust coverage by a mobile robot of a planar workspace," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 4582–4587.
- [7] W. Zhongmin and Z. Bo, "Coverage path planning for mobile robot based on genetic algorithm," in *Electronics, Computer and Applications, 2014 IEEE Workshop on*. IEEE, 2014, pp. 732–735.
- [8] X. Yu and J. Y. Hung, "Coverage path planning based on a multiple sweep line decomposition," in *Industrial Electronics Society, IECON 2015-41st Annual Conference of the IEEE*. IEEE, 2015, pp. 004 052–004 058.
- [9] J. Broderick, D. Tilbury, and E. Atkins, "Energy usage for uavs executing coverage tasks," in *Unmanned Systems Technology XIV*, vol. 8387. International Society for Optics and Photonics, 2012, p. 83871A.
- [10] —, "Maximizing coverage for mobile robots while conserving energy," in *ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2012, pp. 791–798.
- [11] G. P. Strimel and M. M. Veloso, "Coverage planning with finite resources," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 2950–2956.
- [12] H. H. Viet, S. Choi, and T. Chung, "An online complete coverage approach for a team of robots in unknown environments," in *Control, Automation and Systems (ICCAS), 2013 13th International Conference on*. IEEE, 2013, pp. 929–934.
- [13] H. H. Viet, V.-H. Dang, M. N. U. Laskar, and T. Chung, "Ba*: an online complete coverage algorithm for cleaning robots," *Applied intelligence*, vol. 39, no. 2, pp. 217–235, 2013.
- [14] H. H. Viet, V.-H. Dang, S. Choi, and T. C. Chung, "Bob: an online coverage approach for multi-robot systems," *Applied Intelligence*, vol. 42, no. 2, pp. 157–173, 2015.
- [15] S. Choi, S. Lee, H. H. Viet, and T. Chung, "B-theta*: an efficient online coverage algorithm for autonomous cleaning robots," *Journal of Intelligent & Robotic Systems*, vol. 87, no. 2, pp. 265–290, 2017.
- [16] E. Masehian, M. Jannati, and T. Hekmatfar, "Cooperative mapping of unknown environments by multiple heterogeneous mobile robots with limited sensing," *Robotics and Autonomous Systems*, vol. 87, pp. 188–218, 2017.
- [17] I. Rekleitis, A. P. New, and H. Choset, "Distributed coverage of unknown/unstructured environments by mobile sensor networks," in *Multi-Robot Systems. From Swarms to Intelligent Automata Volume III*. Springer, 2005, pp. 145–155.
- [18] C. S. Kong, N. A. Peng, and I. Rekleitis, "Distributed coverage with multi-robot system," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 2423–2429.
- [19] D. B. West *et al.*, *Introduction to graph theory*. Prentice hall Upper Saddle River, 2001, vol. 2.
- [20] F. Balampanis, I. Maza, and A. Ollero, "Area partition for coastal regions with multiple uas," *Journal of Intelligent & Robotic Systems*, vol. 88, no. 2-4, pp. 751–766, 2017.
- [21] C. Kanellakis, S. S. Mansouri, G. Georgoulas, and G. Nikolakopoulos, "Towards Autonomous Surveying of Underground Mine Using MAVs," in *International Conference on Robotics in Alpe-Adria Danube Region*. Springer, 2018, pp. 173–180.
- [22] N. Michael, S. Shen, K. Mohta, V. Kumar, K. Nagatani, Y. Okada, S. Kiribayashi, K. Otake, K. Yoshida, K. Ohno *et al.*, "Collaborative mapping of an earthquake damaged building via ground and aerial robots," in *Field and Service Robotics*. Springer, 2014, pp. 33–47.
- [23] J. G. Rogers, R. E. Sherrill, A. Schang, S. L. Meadows, E. P. Cox, B. Byrne, D. G. Baran, J. W. Curtis, and K. M. Brink, "Distributed subterranean exploration and mapping with teams of uavs," in *Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR VIII*, vol. 10190. International Society for Optics and Photonics, 2017, p. 1019017.
- [24] A. Khan, I. Noreen, H. Ryu, N. L. Doh, and Z. Habib, "Online complete coverage path planning using two-way proximity search," *Intelligent Service Robotics*, vol. 10, no. 3, pp. 229–240, 2017.
- [25] J. Hudack, "Risk-aware planning for sensor data collection," 2016.
- [26] P. Svestka, J. Latombe, and L. Overmars Kavraki, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.
- [27] S. S. Mansouri, P. Karvelis, G. Georgoulas, and G. Nikolakopoulos, "Remaining useful battery life prediction for UAVs based on machine learning," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 4727–4732, 2017.
- [28] H. Wei, B. Wang, Y. Wang, Z. Shao, and K. C. Chan, "Staying-alive path planning with energy optimization for mobile robots," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3559–3571, 2012.