Development and Review of Group Rescue Robots Based on Artificial Intelligence Technology

JunHan Hu

University Of New South Wales, Sydney,2032, Australia

Abstract. Rescue robots can perform rescue missions in dangerous and complex environments, protect humans from harm, and improve the efficiency and effectiveness of rescue, thus playing an increasingly important role in disaster management and rescue. This article reviews the technologies and methods required to apply artificial intelligence to rescue robot teams. Firstly, the feasibility of motion control for swarm robots was explored from the perspective of biomimetic robots. Through the analysis of animal biomimetics and the comparison of commonly used topological structures, the nature of team rescue robot rescue is emphasized, and based on this, a scheme for optimizing topological networks by combining environmental intelligence is proposed. Secondly, several existing micro robots were introduced and their data loading capabilities were evaluated. On this basis, the process of robot vision and motion commands was outlined. At the meanwhile, researchers focus on the current mainstream robot motion trajectory algorithms, and study the algorithm optimization process from extending the motion path planning of a single robot to group coordinated motion. This includes traditional cell decomposition algorithms and algorithms combined with machine learning to improve path planning efficiency. Finally, the above methods were summarized, and the impact of other possible feasible methods in the field of artificial intelligence was explored and analyzed.

Keywords: Group Rescue Robots, Regional Exploration, Artificial Intelligence.

1. Introduction

When a disaster occurs, it is difficult to find survivors in the ruins promptly solely by manpower. This is because the rescue environment for different disasters is different. However, efficiency can be improved by designing specialized rescue robots. In actual disasters, the efficiency of team rescue is significantly higher than that of individual rescue. Because in complex environments, team robots have better investigative and rescue capabilities than individual multifunctional robots [1].

However, the exploration space of the ruins also means the need to design small or even microrobots to adapt to the needs of narrow environments. Although microrobots can flexibly explore, their load capacity has to decrease, which can lead to performance damage and low-range issues. Therefore, a trade-off should be made between the size and function of the robot to ensure the efficiency of search and rescue tasks as much as possible.

At present, the mainstream design of microrobots only retains perception and motion modules, and wirelessly obtains instructions from a central server and executes them. Alers et al. [2] introduced two different types of robots, and for resource-constrained robots, he used the electronic punk platform as an example. Although it only has a few hardware and micro cameras, it is very suitable for the research field of swarm intelligence. This proves the feasibility and development potential of microrobots. Although robots with additional resources can meet various task requirements, they are limited by their size and cannot cope with smaller exploration spaces and unknown dangers. Alers also studied how to label some visual features to help visually constrained microrobots distinguish between environmental markers and the distance of nearby teammates. This technology can help robots strengthen their team collaboration ability and serve as the foundation for coordinated team actions.

Once the design of micro-robots breaks through the limitations of traditional technology or adopts a more reasonable coordination architecture, the application prospects of team robots will not be limited to the rescue field. Koes et al. [3] provided expressive language to describe current environmental constraints and designed a multi-robot team collaboration protocol to quickly optimize solutions. At present, the repository is not yet complete, but with the continuous deepening of research, the system can be used in more and more task environments.

This article will analyze the design concept of team robots from the perspective of biomimetics. Analyze and compare the advantages and disadvantages of virtual force field, optimal reciprocity collision avoidance, interaction speed obstacle method, centralized merge avoidance method, and multi-sensor-based merge avoidance method under different topological systems. Next, summarize the current robot path planning methods from different cell decomposition algorithms. Finally, integrates artificial intelligence into the current algorithm and attempts to optimize existing pathfinding algorithms through machine learning.

2. Design concept

Examples of multi-instance division of labor include bees and ants. From a biomimetic perspective, similar topological structures can be applied to the control system of team robots. Both bees and ants have a certain degree of autonomy in their actions, but when the queen releases a task, it is used as the highest criterion. Given this, Vaishnav et al. [4] designed a swarm-like robot team in a biomimetic manner, with each team member receiving signals from the same host through MCU modules. This means that the robot team needs the 'queen bee' as the main brain to issue commands and heavily relies on the network connection between the central processor and various members. If the signal is difficult to spread in the ruins of some concrete buildings after the disaster, it will inevitably have an impact on rescue efforts.

Therefore, current topology research is often divided into two types: one tends to focus on the full motion control of the central processing unit, which is a star structure, and the other is a distributed algorithm that relies more on individual computing power. Depending on different algorithms, they can adopt methods such as mesh or tree topology to improve connection efficiency. The biggest problem with star topology is that the connection distance between each individual and the central server is limited, but its advantage is that it can increase the computing power of the server separately outside the field to reduce the load cost of the members themselves. Distributed networks are increasingly becoming the mainstream in current multi-robot system design. Yi et al. [5] has improved the connection of networks in multi-robot systems, using topology correction controllers to flexibly allocate available intermediate connection points for the entire system. Its core is to use taskless robots as signal nodes to extend network depth. This structure can still be further optimized, and in future improvements, a portion of micro robots can be differentiated into robots that only carry signal relays and a stable network can be formed using specialized deployment algorithms. The algorithm designed by Kiyohiko Hattori [6] can coordinate the movement of robots through Radio Signal Strength Indicators and has passed the evaluation of simulated disaster area environments. The mobile robot networking algorithm should also have the ability to autonomously adjust transition points, and control the number of nodes in the current area based on the number of robots responsible for detection. This concept can optimize the system network throughput to strengthen connectivity [7]. So it is obvious that the topology network construction of the entire system should be flexibly adjusted according to the current environment, dispersing robots as much as possible in large spaces to increase search width, and prioritizing extending signal propagation distance in narrow environments to ensure search depth.

However, topology systems can only serve as the command mode for pathfinding algorithms. For multi-robot rescue systems, the distance between individuals should also be considered to avoid repeated searches. Unlike the strong purposefulness of pathfinding algorithms, controlling distance means that robots need to avoid collisions with each other, which should be an embedded protocol that is executed passively by each team member independently rather than actively through processor commands. This predetermined rule enables robot teams to organize in an orderly manner to save communication costs and coordinate action routes in a unified manner.

Hu et al.[8] designed a robust formation protocol that can ensure the orderliness of the formation through adjacent communication even when disturbed. Taking the Mona robot as an example, it is a small robot equipped with low-cost sensors that can still execute formation commands and move groups based on guidance signals despite communication delays.

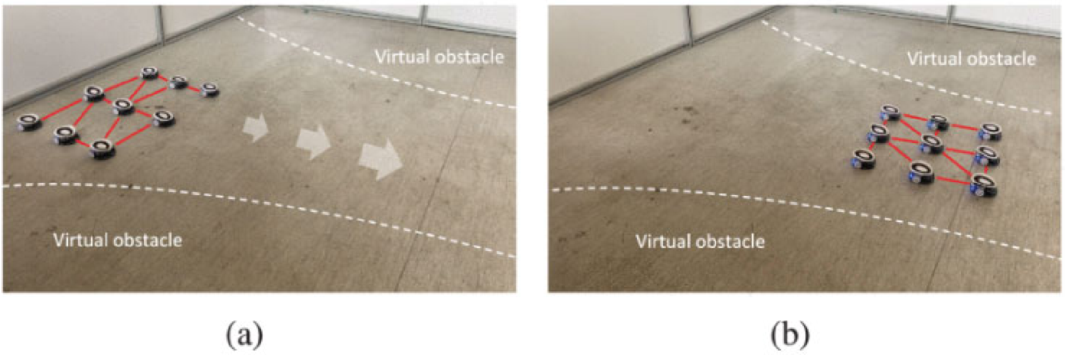

Figure 1 Demonstration of collaboration among multi-robot in a virtual environment

As shown in Figure 1, this experiment proves the feasibility of the protocol and lays the foundation for multi-robot coordination in real-time scenarios. The exploration algorithm designed on this basis can optimize its perception space, such as avoiding search space conflicts by controlling distance.

Liu et al.[9] introduced five methods to support collision avoidance algorithms in her article. For example, a Virtual Force Field can avoid collisions by setting high attraction to robot exploration targets while also setting repulsive forces on teammates. This star-based topology algorithm should be optimal in a smooth environment, but if it is in a distributed robot group, Optimal Reciprocal Collision Avoidance should be more suitable because it disperses the "collision prevention" task to each robot. However, this also requires the hardware capability of robot sensors, as it heavily relies on the robot's accurate perception of the nearby environment. In addition, there are also the Reciprocal Velocity Obstacles method, Centralized consolidation avoidance method, and Multi-sensor-based consolidation avoidance method. However, these algorithms pose certain requirements for the hardware configuration of the robot itself. They often require these micro robots to carry items such as GPS, camera modules, and infrared sensors to ensure their visual accuracy, but this may also lead to short range issues for micro robots.

Finally, before designing the path planning algorithm, we need to design an architecture that can describe the constraints of the current disaster environment and the algorithm execution capabilities of the robot team. If the system can select the optimal algorithm based on the current environment and combine it with the local protocol of the robot team, it can reduce algorithm costs. Koes et al. [10] provided an expressive language to describe current environmental constraints, while their multi-robot coordination architecture can quickly optimize solutions. By developing different environmental benchmarks, their team evaluated the performance of different algorithms under different benchmarks. At present, this repository is not yet perfect, but with more and more research deepening, this system can be used in more and more rescue environments. This language description can help the central server better understand the characteristics and constraints of the current environment, and reduce a certain amount of training time when using machine learning and other algorithms to optimize the system.

3. Regional Exploration Algorithms

The path design of robots determines their search space. For swarm robots, the first step is to control the path planning of individual robots to meet the needs of the movement. In a space filled with obstacles, robots should immediately plan a feasible path to their destination from their current position.

3.1 Cell-Based Decomposition

The most commonly used approach for obstacle avoidance planning is Cell Decomposition. It abstracts complex environmental spaces into continuous two-dimensional planes and divides them into grids of the same size. By determining whether there are obstacles in the grid, we can replace them with binarization, which sets the grid containing obstacles to 1 and the rest to 0.

Under the guidance of this method, different specific algorithms have also been continuously explored. For spatial decomposition, there are Trapezoidal Cell Decomposition (TCD), Boustrophedon Cell Decomposition (BCD), and Exact Cell Decomposition (ECD).

TCD locates obstacles within the area, finds their vertex positions, and creates parallel lines along the X or Y axes. Each vertex will have a line intersecting the boundary of the region, and the space will be divided into multiple convex regions. This method is very convenient when dealing with regular obstacle boundaries. However, for concave obstacles, it is necessary to make a good decision on drawing lines at the vertices. There is still room for further optimization in this decomposition method, which is BCD.

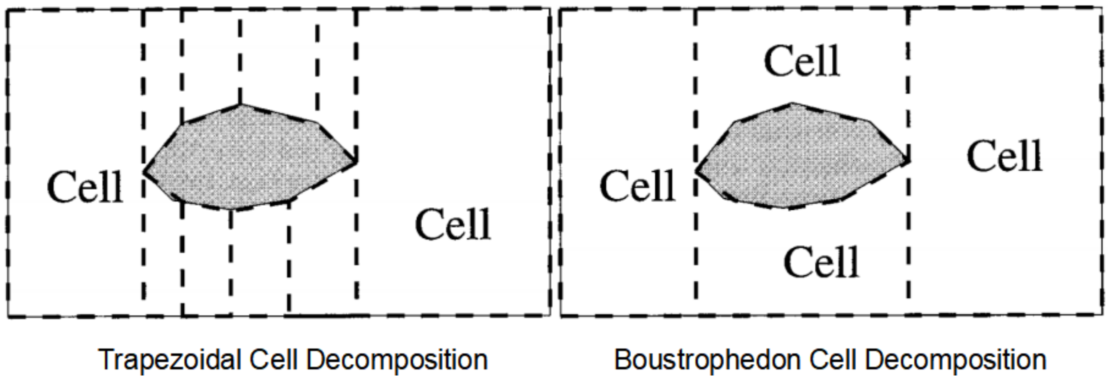

Figure 2 Schematic diagram of Boustrophedon Cell Decomposition and Trapezoidal Cell Decomposition

The difference between BCD and TCD lies in the rules for drawing lines. From Figure 2, it can be seen that in TCD, parallel lines are drawn for each vertex of the obstacle and intersect with the boundary. However, BCD reduces the redundant decomposition of TCD, making it simpler and more effective. It was proposed by Choset [11] in 2000 as a generalization of TCD, which is better at decomposing irregularly shaped obstacles, such as circles and ellipses. The feasibility of practical application in robots has also been demonstrated in Choset's article.

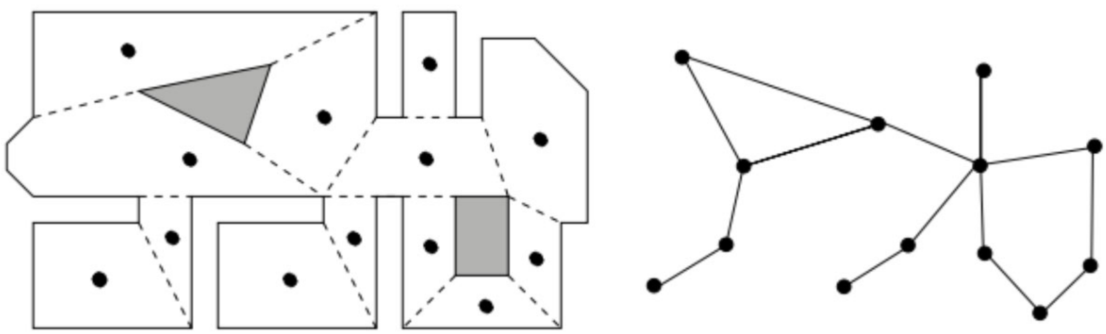

Figure 3 Minimizing cell decomposition in ECD

Latombe et al.[12] provided a detailed introduction to ECD, also known as Exact Cell Decomposition. It only limits the cell to a convex polygon without internal holes. The decomposition of each cell allows the entire search area to be mapped into a graph of point line combinations. As shown in Figure 3, this is a method of minimizing cell decomposition to minimize the number of cells, that is, the number of vertices in the graph.

The opposite is the Approximate Cell Decomposition, also known as ACD. It focuses more on outlining the general outline rather than accurately covering all free spaces. And ACD will use smaller cubes to segment the overall space. However, this cell can be set to an adaptive size, as in a sparse space, larger granularity cells are preferred for segmentation. As depicted in Figure 4, the edge open area will be divided into larger cells. Use smaller cells at the edges of obstacles to achieve precise segmentation.

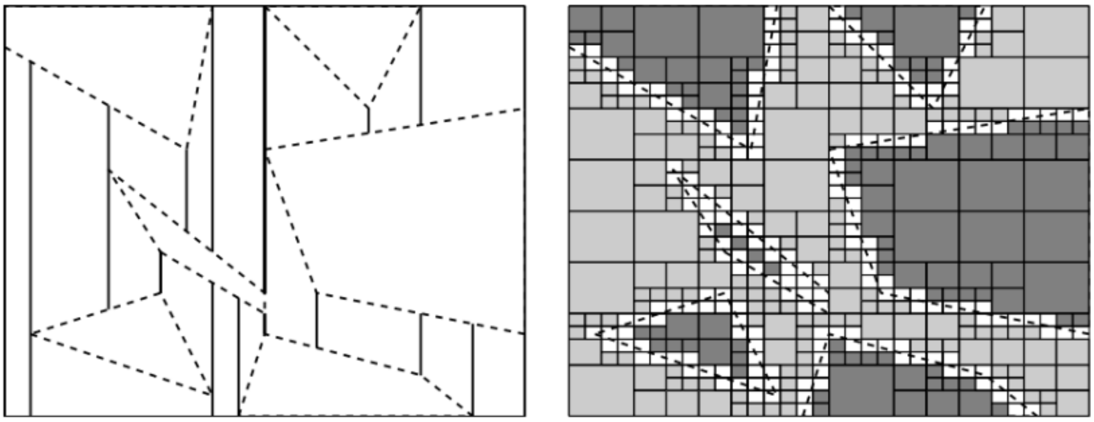

Figure 4 Adaptive Size Cell Decomposition

Woo et al. [13] exploited the maximum volume decomposition. He solved the global effect of face extension and the heavy computational load for cell collection. This optimization scheme has made ACD play a significant role in solid modeling.

Different decomposition methods have their strengths. In different obstacle environments, priority should be given to handling. Debnath et al. [14] studied and discussed cell decomposition techniques under different standards, and classified them. Gonzalez et al. compared the performance of different planning schemes. Through these technologies, we can flexibly choose specific algorithms that are currently applicable, thereby saving robot exploration costs.

3.2 Optimal coverage

The collision prevention protocol that comes with rescue robots is a passive area coverage method, which can only ensure that the distance between robots is neither close nor scattered. However, in practical applications, the maximization of the overall search scope of the team should also be considered. This requires the use of proactive area coverage algorithms, whose main idea is to consider the detection range of the robot itself as a solid volume.

From the simple algorithms and comprehensive models introduced above, it can be seen that they can all play a role in practical rescue to some extent, but they always cannot cover all aspects. It can be seen that it is necessary to establish a comprehensive system rescue task platform. The system needs to judge the current environment and select the most suitable path planning scheme for this task, and update the exploration route in real time based on the environmental information transmitted by the forward detection robot. Of course, it is also necessary to prepare backup human-machine joint solutions to increase the robustness of the overall planning.

In this system, when the robot enters the designated search space, the sensor returns environmental information. At this point, the system can choose the optimal cell decomposition method to cut the space. After the cutting is completed, the next exploration location can be specified to the edge position and tasks can be assigned to the robot. Subsequently, the movement path of the robot population can be planned based on the CNN training results. This system can meet the needs of the vast majority of rescue scenarios.

During the movement of a group of robots, each robot itself can be seen as a small movable obstacle. Yang et al. [15] pointed out that when using Deep Reinforcement Network (DQN) algorithm, the dynamic changes in these environments can increase the complexity of computation. However, by integrating the motion and position information of all robots, the resulting state vector ensures that the entire system makes stable decisions. Considering the slow convergence speed of the DQN algorithm, it is necessary to ensure the stability of the training results as much as possible. Yang et al. designed a conflict strategy to allow the robot to update Q-values based on the algorithm when it conflicts with its teammates. Experiments have shown that this can significantly improve the efficiency of the algorithm.

In addition, multi-robot exploration also means assigning tasks to each robot. At this point, it is necessary to consider the possible coverage redundancy caused by robot group question searches. Only by minimizing the repeated exploration area can we maximize the efficiency of group search. Rekleitis et al. [16] proposed a set of multi-robot coverage algorithms for this which use BCD for cell decomposition. More specifically, corresponding strategies were designed for task allocation for each robot. Unlike other swarm robot path algorithms, Rekleitis's scheme is determined based on the current communication type. This also means that this algorithm can adapt to different environments and has higher practical application value.

At the same time, the path coverage of multiple robots has also been proven to be robust. Hazon and Kaminka [17] used Approximate Cell Decomposition(ACD) as the decomposition method for the region coverage algorithm, and analysis shows that the algorithm has strong execution ability in the worst case. Even if only one robot can work, it can also complete the coverage of the entire area. However, if a non-redundant version of the algorithm is used, it cannot guarantee performance improvement. This also means that the rescue efficiency of swarm robots will not be affected by rough targets, as redundant algorithms can reduce search time by half even in the worst environmental conditions.

3.3 Optimization by Machine Learning

In the field of swarm robot rescue, the most important thing is to calculate the optimal solution in real time. Therefore, more and more AI algorithms are being applied in this field. Compared to common algorithms, AI's powerful computing power can ensure automatic analysis and identification of environmental variables, and intelligent planning of the robot's movement path and search range.

Firstly, Luna [18] proposed an efficient method to address the issue of exchanging positions during robot movement, which eliminates the high time complexity of composite search in the system. This optimization plan can help large rescue teams with robot path planning. According to Luna's simulation experiments, compared to the traditional centralized and decoupled planners, their team can solve path planning problems for up to 100 robots in a very short time. This time consumption is beyond the reach of ordinary algorithms.

Once the deadline task is set, the entire algorithm will be limited by linear time. In this way, the original problem can be solved by linear Integer programming. Wang [19] considered this model and extended it to the area coverage problem of swarm robots. His algorithm mainly solves the problems of obstacle avoidance, punctuality, and overtime penalties. Although the feasibility and completeness of this algorithm have only been proven through theory, it can calculate the local optimal solution within time constraints. This method can serve as a backup solution for the rescue team to avoid errors or bugs in the main system. Overall, this model can increase the robustness of rescue team solutions and provide secondary solutions that meet the deadline for rescue implementation.

In the field of reinforcement learning, CNN can be used to process pixel information. The reward mechanism for training mentioned in [20] is to be as close to the target as possible, but negative values will be assigned when collisions occur. However, it was found from the content of the article that the movement movements used in the experiment were limited to four directions, and the movement strategy of the swarm robot was not specifically considered. Therefore, there are certain limitations. Sartoretti et al. [21] has considered a new distributed algorithm to enable each robot to perform actions that are beneficial to the overall strategy. This implicit collaboration is similar to centralized training but with distributed execution. Similar to the reward mechanism of Bae, it adopts the methods of plus and minus points and increases the penalty points for the robot when stationary. In fact, as a classic multi-robot path planning method, Sartoretti's method has been iteratively upgraded to the PRIMAL2 version, with higher accuracy and training result accuracy.

However, these two optimization methods still have limitations. When training robots to move, the more directions to choose, the better, to meet the degrees of freedom of robot movement in the real world. In Qiu's experiment [22], the robot movement method he used was achieved through Unity3D simulation, with each robot having sensors in 45 directions. By using the $A*$ algorithm to select a path, each step forward is set as a negative reward, and there are more negative rewards when stopping and colliding. Positive values are only given when the robot reaches the endpoint. Such training results can meet the pathfinding needs of large-scale robot teams, and the selection of multiple directions makes the experimental results more convincing.

Other AI algorithms can also be used to train the model. The decision tree can be used here to determine whether the robot can take the next action, such as first determining its battery level, then determining the distance of obstacles ahead, and then determining whether it can reach the target within the given time. If it cannot be reached, return to the original location. And because each robot's environment and its state are unique, it is not easy to have overfitting problems

The clustering algorithm is not used here to group robots by distance. When connecting to the central network, small machine people can compare their current environment by comparing the established information models in the database. This way, they can determine which situation they belong to, whether to return, continue exploring, or choose to support their teammates.

For CNN, it’s possible to simulate the optimal solution in the current environment through a large number of known rescue cases, which may only be a locally optimal solution. Compared with dynamic programming, it is more suitable for very complex actual environments. Because when the iteration pool of dynamic programming is too deep, its calculation will be very heavy. However, as mentioned earlier, although some training time has been sacrificed, the already-trained results can be directly used in each practical application. After each task is completed, the system can add the actual case to the training set. Over time, the training results will become more and more perfect.

4. Conclusion

The special working environment of rescue robots determines their high work difficulty. Planning a low collision (or no collision) path within a limited time also imposes certain requirements on the performance of the central computer. In actual rescue environments, there is a risk of secondary deformation in the exploration space at any time. Overall, the issues that need to be addressed are signal propagation, obstacle avoidance, range coverage, task time constraints, and timeout remediation in a constrained environment.

From the simple algorithms and comprehensive models introduced above, it can be seen that they can all play a role in practical rescue to some extent, but they always cannot cover all aspects. It can be seen that it is necessary to establish a comprehensive system rescue task platform. The system needs to judge the current environment and select the most suitable path planning scheme for this task, and update the exploration route in real time based on the environmental information transmitted by the forward detection robot. Of course, it is also necessary to prepare backup human-machine joint solutions to increase the robustness of the overall planning.

The signal control strategy should be optimized as a hybrid model. Flexible topology selection can improve the robustness of the improved system. Regardless of the signal under current environmental constraints, the machine population can always continue to operate using the built-in protocol, rather than stopping the search when the signal is lost. Therefore, the current direction of topology improvement is to optimize built-in protocols, enhance the system's simple communication capabilities, and optimize the central control algorithm logic in star topology networks.

So an expected system should meet the requirement that sensors can promptly return environmental information when the robot enters the designated search space. At this point, the system can choose the optimal unit decomposition method to cut the space. After the cutting is completed, the next exploration position can be assigned to the edge position, and the task assigned to the robot. Subsequently, the motion path of the planned robot population is selected based on whether machine learning optimization schemes are adopted. This system can meet the needs of the vast majority of rescue plans and has room for optimization in the future.

With the development of artificial intelligence, we can also adopt more algorithms. Artificial neural networks may have better fitting ability than other models, and their main training direction should be set as the optimal action-solving method under constraint conditions. Overall, supervised algorithms may have stronger analytical capabilities than unsupervised algorithms in such disaster rehearsals, as overly complex environmental information may make the model difficult to fit or overfit.

Reference

[1] Cai Luo, A. P. Espinosa, D. Pranantha and A. De Gloria, "Multi-robot search and rescue team," 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics, Kyoto, Japan, 2011, pp. 296-301, doi: 10.1109/SSRR.2011.6106746.

[2] Sjriek Alers, Karl Tuyls, Bijan Ranjbar-Sahraei, Daniel Claes, Gerhard Weiss; July 30–August 2, 2014. "Insect-Inspired Robot Coordination: Foraging and Coverage." Proceedings of the ALIFE 14: The Fourteenth International Conference on the Synthesis and Simulation of Living Systems. ALIFE 14: The Fourteenth International Conference on the Synthesis and Simulation of Living Systems. Manhattan, New York. (pp. pp. 761-768). ASME. https://doi.org/10.1162/978-0-262-32621-6-ch123.

[3] M. Koes, I. Nourbakhsh and K. Sycara, "Constraint optimization coordination architecture for search and rescue robotics," Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006., Orlando, FL, USA, 2006, pp. 3977-3982, doi: 10.1109/ROBOT.2006.1642312.

[4] J. Vaishnav, A. B. Uday and T. Poulose, "Pattern Formation in Swarm Robotic Systems," 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2018, pp. 1466-1469, doi: 10.1109/ICOEI.2018.8553906.

[5] S. Yi, W. Luo and K. Sycara, "Distributed Topology Correction for Flexible Connectivity Maintenance in Multi-Robot Systems," 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 2021, pp. 8874-8880, doi: 10.1109/ICRA48506.2021.9561721.

[6] Hattori, K., Tatebe, N., Kagawa, T. et al. Deployment of wireless mesh network using RSSI-based swarm robots. Artif Life Robotics 21, 434–442 (2016). https://doi.org/10.1007/s10015-016-0300-y

[7] Nguyen CQ, Min B-C, Matson ET, Smith AH, Dietz JE, Kim D. Using Mobile Robots to Establish Mobile Wireless Mesh Networks and Increase Network Throughput. International Journal of Distributed Sensor Networks. 2012;8(8). doi:10.1155/2012/614532

[8] J. Hu, A. E. Turgut, B. Lennox and F. Arvin, "Robust Formation Coordination of Robot Swarms With Nonlinear Dynamics and Unknown Disturbances: Design and Experiments," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 69, no. 1, pp. 114-118, Jan. 2022, doi: 10.1109/TCSII.2021.3074705.

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

[10] M. Koes, I. Nourbakhsh and K. Sycara, "Constraint optimization coordination architecture for search and rescue robotics," Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006., Orlando, FL, USA, 2006, pp. 3977-3982, doi: 10.1109/ROBOT.2006.1642312.

[11] Choset, H. Coverage of Known Spaces: The Boustrophedon Cellular Decomposition. Autonomous Robots 9, 247–253 (2000). https://doi.org/10.1023/A:1008958800904

[12] Latombe, JC. (1991). Exact Cell Decomposition. In: Robot Motion Planning. The Springer International Series in Engineering and Computer Science, vol 124. Springer, Boston, MA. https://doi.org/10.1007/978-1-4615-4022-9_5

[13] Yoonhwan Woo, Fast cell-based decomposition and applications to solid modeling, Computer-Aided Design, Volume 35, Issue 11, 2003, Pages 969-977, ISSN 0010-4485, https://doi.org/10.1016/S0010-4485(02)00144-6

[14] Debnath, S.K. et al. (2021). Different Cell Decomposition Path Planning Methods for Unmanned Air Vehicles-A Review. In: , et al. Proceedings of the 11th National Technical Seminar on Unmanned System Technology 2019 . NUSYS 2019. Lecture Notes in Electrical Engineering, vol 666. Springer, Singapore. https://doi.org/10.1007/978-981-15-5281-6_8

[15] Yang, Y., Juntao, L. and Lingling, P. (2020), Multi-robot path planning based on a deep reinforcement learning DQN algorithm. CAAI Trans. Intell. Technol., 5: 177-183. https://doi.org/10.1049/trit.2020.0024

[16] Rekleitis, I., New, A.P., Rankin, E.S. et al. Efficient Boustrophedon Multi-Robot Coverage: an algorithmic approach. Ann Math Artif Intell 52, 109–142 (2008). https://doi.org/10.1007/s10472-009-9120-2

[17] N. Hazon and G. A. Kaminka, "Redundancy, Efficiency and Robustness in Multi-Robot Coverage," Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 2005, pp. 735-741, doi: 10.1109/ROBOT.2005.1570205.

[18] R. Luna and K. E. Bekris, "Efficient and complete centralized multi-robot path planning," 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 2011, pp. 3268-3275, doi: 10.1109/IROS.2011.6095085.

[19] H. Wang and W. Chen, "Multi-Robot Path Planning With Due Times," in IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 4829-4836, April 2022, doi: 10.1109/LRA.2022.3152701.

[20] Bae H, Kim G, Kim J, et al. Multi-robot path planning method using reinforcement learning[J]. Applied sciences, 2019, 9(15): 3057.

[21] G. Sartoretti et al., "PRIMAL: Pathfinding via Reinforcement and Imitation Multi-Agent Learning," in IEEE Robotics and Automation Letters, vol. 4, no. 3, pp. 2378-2385, July 2019, doi: 10.1109/LRA.2019.2903261.

[22] Qiu H. Multi-agent navigation based on deep reinforcement learning and traditional pathfinding algorithm[J]. arXiv preprint arXiv:2012.09134, 2020.