NAME or ID:	oj22395 Runze Yuan		**MARK (%):** 71
	To *pass*, you must	**For *merit*, you must also**	**For *distinction*, you must also:**
Aims and Objectives	Write some sensible aims and/or objectives ✓	Write a good number of clear aims and/or objectives ✓	Neatly define project with concise aims and/or objectives ✓
Motivation	Sensibly justify your project ✓	Provide an evidence base for your project ✓ good	Make a clear, concise case for your project ✓
Literature Review	Show that you have read appropriate papers ✓	Provide a good summary of relevant literature ✓	Clearly relate your project to the state of the art nice connections
Impact Assessment	Identify some potential impacts of robotics ✓	Highlight significant impacts of the proposed project	Clearly relate project to wider context
Risk Register	Identify sensible project risks ✓	Identify and rank all significant risks OK	Effectively manage all significant risks
Timeline	Divide project into feasible sequence of activities and milestones ✓	Define clear, efficient sequence of activities and milestones ✓	Propose a usable, robust plan in appropriate level of detail ✓

Dissertation Project Research Proposal

Self-organised Resource Sharing for Decentralised Multi Agents on 2D plane Based on STDMA

By

Runze Yuan

Department of Engineering Mathematics
University of Bristol
&
Department of Engineering Design and Mathematics
University of the West of England

March 30, 2023

Contents

1 Aims and Objectives

1.1 Aims

Based on the self-organised channel resource allocation principle of the STDMA(Self-organised Time Division Multiple Access) [1] communication protocol, develop a method for agents to achieve collision-free moving on a 2D plane, make it self-organised, decentralised and scalable.

1.2 Objectives

- Develop a multi agent communication scenario with STDMA in ROS2. In this scenario, multiple identical agents will share the same channel and all agents have both receiving and transmitting capabilities. The goal is to enable agents to autonomously organise/join the communication process after start up.
- On the basis above, a simple grid world is implemented: a 2D map composed of grids, each grid representing a part of space. Agents would move from one grid to another on the map.
- Using the principle of STDMA [1] to implement simple decentralized conflict-free space(which is a kind of resource) sharing and simulate implementation in the above scenario.
- STDMA is designed for 1D resource sharing (slots in time frames), and directly applying STDMA for 2D space sharing will definitely lead to some deadlock and inefficient scenarios [2,3]. Determine the cause of this problem by observing the experiment, and make corresponding improvements to this problem.
- Use appropriate performance metrics (makespan, sum of cost, and agent capacity when algorithm running in widely used benchmark maps [3]) to evaluate the advantages and disadvantages of algorithm performance.

nice plan

← good

2 Motivation

The motivation of this project is to answer a question inspired by [4]. Which is:

> **What will happen if use STDMA for 2D resource sharing?**

The detailed explanation of the motivation is as follows.

2.1 Why STDMA?

What is STDMA? The *Self-organised Time Divided Multi Access* (STDMA) is an channel access technique for communication. It is based on a set of policies dictating how agents ought to apply for slots in repeating frames, as elaborated in Section 3.1.

The reason for using STDMA is its **characteristics** [5]:

1. **Deterministic**: Agents arrange their data transmission based on a determined timetable.

2. **Decentralised**: Agents listen to the channel first, then find free slots in the timetable for themselves to use.

These characteristics could be useful for multiple agents to achieve collision-free (use free slots only), self-organised (find slots on its own) resource sharing.

There are also two **reasons that making this challenging**:

Interesting question

1. **Dimensional Difference**: STDMA is designed for sharing discrete time slots(1D), and cannot be directly applied to resource sharing in a 2D plane. Modification is needed for 2D space application.

2. **Moving by Grids**: In the communication scenario, agents don't have destinations in the timetable, i.e., don't need to move to a specific slot in the table. But that's different for 2D space moving, where agents have their destinations and need to move grid by grid to reach their goals. Agents could easily be trapped in situations of inefficiency or even deadlock situations [2, 3].

These characteristics and challenges are making this topic interesting and need further experiment to investigate.

2.2 Why 2D resource sharing?

2D resource sharing is a name for better understanding and is not the accurate name for this problem. This problem is actually a ***Multi-Agent Path Finding*** (**MAPF**) [3] problem.

Here, the term "finding" could be somewhat confusing. In fact, he MAPF problem combines both **collision-free movement** of multiple agents and **path efficiency optimising**.

Overall, the problem of *Multi-Agent Path Finding* (MAPF) concerns **the movement of multiple agents in a grid world**.

To easily get a better understanding of this problem, please refer to this website [1]. The website presents the MAPF problems and their solutions through a simple and engaging animation (estimated time required: 1∼2 min). For detailed explanation, please see Section 3.2.

MAPF algorithms have broad prospects for use.

- **Warehouse Automation**: Pick-pack-and-ship system in Amazon warehouse (like Fig 1.) [6]. Delivering items in sorting station [7, 8].

- **Intersection Management**: Coordinate autonomous vehicle movement through intersections [9].

- **Robot Fleet**: Automating fleets of autonomous robots like forklift fleets. [10, 11].

- **Agents in video games and CGIs**: Flock simulating and animating [12, 13].

[1] https://primalgrid.netlify.app/primal

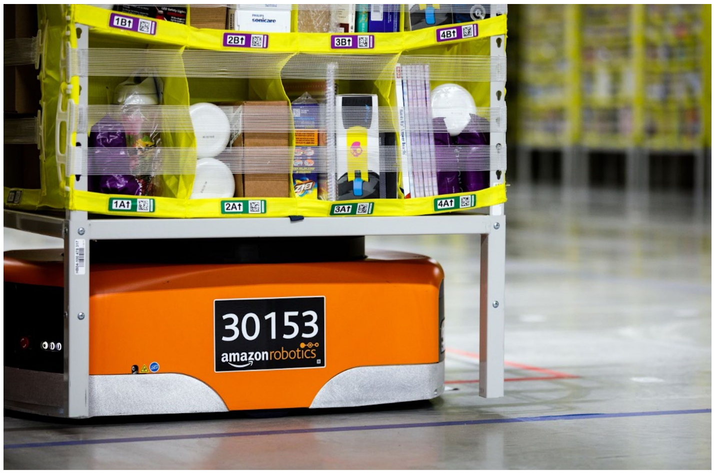


Figure 1: Kiva system operating in Amazon warehouse.[1]

- **Swarm Robots**: Controlling self-organised robot swarms [14].
- ... ?

In general, the application of this algorithm is very broad. Furthermore, in the context of Industry 4.0 and flexible manufacturing, it has even better prospects for application [15–17], as centralized control is gradually becoming inadequate to meet new production needs.

good evidence

3 Literature Review

This section will introduce the details of the STDMA [1] protocol and summarize the development of relevant MAPF algorithms.

3.1 STDMA Explained

As previously mentioned in Section 2.1, STDMA [1] allows multiple devices to share the same channel for communication without centralised control. The main assumption of this protocol is that all devices have synchronized clocks. In practice, this is achieved through GPS [18].

The core idea of STDMA can be summarized as follows: Represent continuous time with repeated frames, each frame has several discrete slots. While devices are always listening to channel information, they look for free slots to occupy for their broadcasting their own data.

Devices using STDMA have **four phases**, which are arranged in chronological order as follows:

1. **Initialisation**: Devices in this phase have not yet joined the network. The device listens to an entire frame and determines the current slot allocation.

[1] https://spectrum.ieee.org/interview-brad-porter-vp-of-robotics-at-amazon

2. **Network Entry**: Randomly choose an unallocated slot to broadcast their existence and reserve one slot for the next phase.

3. **First Frame**: Use the slot reserved in the previous phase to reserve more slots for themselves. The number of reserved slots depends on the size of the data packet that the device needs to send.

4. **Continuous Operation**: Use the previously reserved slot to work normally. If some slots are released or more slots are needed, reapply for slots.

Although the above description omits some details (such as slot choosing methods, calculation of required number of slots, etc.), it is clear that **the core of STDMA is the strategy of finding and reserving unallocated slots**.

This protocol also has some limitations, such as: **(1)** Collision: In Network Entry phase, multiple devices may accidentally choose the same unallocated slot to broadcast their existence. **(2)** Capacity: When the slots are not enough, conflicts will inevitably occur.There are some studies [19, 20] that proposed improvements to the above limitations, but this is not the focus of this project.

3.2 MAPF Algorithm Review

The classical definition of the MAPF problem is: **Multiple agents moving step by step in an undirected graph in discrete time** [2].

For the MAPF problem with n agents, its **inputs** are:

1. **Environment**: The environment is represented as an undirected graph $G = (V, E)$, where vertices V represent space in the environment and edges E describe the connection of vertices.

2. **Start**: The starting vertices of n agents are represented with $s : [1, \ldots, n] \rightarrow V_s$, which maps n agents to the source vertices V_s.

3. **Target**: The target vertices of n agents are represented with $t : [1, \ldots, n] \rightarrow V_t$, which maps n agents to the source vertices V_t.

For the MAPF problem with n agents, it **requires**:

1. **Single-agent Plan**: A sequence of actions. By executing this sequence of actions, an agent can reach the target from the start.

2. **Solution**: A set of single-agent plans for all n agents in the system. By executing the sequence of actions in this set simultaneously, all agents should be able to reach the destination from the starting point. A good solution should be both efficient and collision-free.

The MAPF algorithm can be divided into **several categories: classical, heuristic, rule-based, and machine learning combined**. Among these categories, **the rule-based algorithms are the most relevant to this project**. However, I would also briefly introduce other types of algorithms.

- **Classical**: The most popular classic MAPF algorithm is the *Artificial Potential Field* (APF) algorithm, which put force fields around obstacles and goals to generate a force to guide the agents. APF methods does not guarantee collision-free and is prone to fall into local minima. There are still some research today dedicated to improving the performance of the APF algorithm, with a primary focus on optimizing the potential field function [21–23].

- **Heuristic**: Heuristic algorithms mainly include A* and its variants (like D* [24]). Heuristic algorithms mainly operate on the principle of searching for the optimal solution. Many research combine A* with other algorithms to achieve better performance and goes beyond the original A* [25, 26].

- **ML Combined**: The machine learning algorithm mainly used in this field is reinforced learning, and usually combined with other non-RL techniques [27–30].

- **Rule-based**: Most of the rule-based methods are based on the idea that each agent follows a set of identical local decision-making principles (which means it is decentralized) and usually improve algorithm performance by modifying the rules. Many of these algorithms are bio-inspired.

  - **PSO**: The idea of *Particle Swarm Optimization* is to simulate social behaviour of animals. [31] improved the algorithm performance with swarm dynamics and designed a obstacle-avoidance algorithm for multi-UAV path finding. [32] designed an algorithm for patrol robot and presented an interval update law for best position estimation, then applied an iterative procedure for obstacle avoiding.

  - **GA**: *Genetic Algorithm* is inspired by the process of natural selection that is popular for solving optimizing problems. [33] proposed an algorithm for UAV transmission which uses GA and RL to generate incentive rewards for agents. And [34] designed an algorithm for parking slot allocation with similar principle.

  - **ACO**: *Ant Colony Optimisation* is widely used for all questions that can be reduced to finding path to food through graphs. These algorithm usually are built with pheromone-based communication between agents. [35] proposed an AGO-based algorithm for AGV that could do path finding and task allocating at the same time. [36] proposed an adaptive waypoints-repair method with reinitialisation mechanism integrated for multiple unmanned ground vehicle path planning.

  - **PIO**: *Pigeon-Inspired Optimisation* is inspired by the movement of pigeon flock going home. [37] developed a path finding method for UAVs with PIO based on time stamp segmentation technique. [38] improved the UAV path planning algorithm performance in plateau narrow area with cauchy mutant PIO algorithm.

  - **GWO**: *Grey Wolf Optimisation* [39] is based on the leadership hierarchy and hunting mechanism of grey wolves. Agents are divided into four types to mimic the nature of grey wolf leadership hierarchy and applicable for problems with unknown possibility spaces. [40] proposed a UAV path planning good study & review

algorithm which is a hybrid of GWO and *Whale Optimiser Algorithm* and is capable of avoiding dynamic obstacles. [41] using GWO to solve multi-UAV path planning model with energy constraint, and improved the optimisation ability of GWO.

- **Push-and-Swap**: *Push-and-Swap* algorithms [42] are based on two types of moves: push (move towards the goal) and swap (exchange position between two agents). It have many variants, and algorithms in this category are usually complete and fast. However, there is no guarantee that the solution given is optimal. [43] proposed a solution to pebble-motion on graphs with Push-and-Swap algorithm. [44] built a MAPF algorithm with the principle of Push-and-Swap which provides theoretical completeness and optimality, but cannot deal with deadlocks on narrow passages.

In general, there are many different solutions to the MAPF problem, but **all algorithms are trade-offs between completeness, computational complexity, and optimality**. There is currently no perfect algorithm for this question.

4 Impact Assessment

The main goal of this project is to help readers gain a better understanding of the MAPF problem by answering a question (see Section 2), and the proposed algorithm is mainly based on simulation rather than practical system operation. Therefore, **the impacts of this project are mostly potential rather than immediate**.

Positives:

- Academical Impact: Help readers to get better understanding of MAPF problems and may provide inspiration to developers.
- Improve Production: MAPF could help agents to better navigate and accomplish tasks, improve efficiency, flexibility, and reduce costs.
- Increased Safety: Reduce the accidents and fatalities caused by human error.

Neutral:

- Energy Efficiency: MAPF algorithms helps the robot swarm to improve their energy efficiency by optimizing its route and reduce idle time. But there is also a risk of increased energy consumption due to the overuse of robots in production processes and accurately estimating the impact of this part is beyond my knowledge and ability.

Negatives:

- Impact on Workforce: Implementation of MAPF algorithms could impact the workforce, potentially leading to job displacement or changes in job. Although it may also bring more R&D personnel and technical personnel positions, considering the difference in the number of bases, the number of affected workers is overwhelming.

ok.

- Cultural Resistance: Application of MAPF algorithms could face resistance from cultural or social groups who value human labor and human-based decision making.
- Digital Security: Applications of MAPF algorithms (like AGVs) may introduce risks of cyber attack and private data leakage.

5 Risk Register

From a risk management perspective, your project can be described as:

> **I** simulate and optimize **algorithms** in a **computer** with normal laptop-level computing power and **ROS2** environment.

The possible risks are represented by the table below(Table 1.). For the likelihood and impact columns, levels 1∼4 are used for measuring, with 1 representing the least likely/least impact and 4 representing the most likely/most impact.

PC Malfunction: Due to some mysterious software malfunction, my computer has been performing very unstably. In fact, during the process of writing this proposal, I was forced to reinstall the operating system once. For this risk, I will try my best to solve it before the project enters the full-day stage. At the same time, I will also pay attention to synchronising progress to the cloud, so I could continue my work on shared computer if PC under repairing.

Underestimated Complexity: My understanding of this topic still does not allow me to estimate the true complexity of the problem, and I may have underestimated the difficulty of certain parts when planning, which may disrupt my plans. For this risk, I will try to arrange the uncertain parts at the end so that other parts do not depend too much on the parts where I am uncertain about the complexity.

ROS2 Capability: It is unclear whether ROS2 can support a certain number of agent simulations. For ROS2, simulating with more than 4-5 agents could be quite slow, but the specific performance depends on the project and the programmer, so it cannot be determined at the beginning. For this risk, I will try to simulate large-scale swarms only when necessary. And I will pay attention to performance issues when writing codes and try to avoid migrating the project to other platforms due to computing power reasons, because code migration is not part of my research topic.

Progress Loss: A classical, serious but constantly occurring problem for all research projects. For this problem, progress should be saved and cloud synchronization should be kept in mind at all times.

Risk	Likelihood(A)	Impact(B)	Risk Score(A×B)
PC Malfunction	2	4	8
Underestimated Complexity	3	2	6
ROS2 Capability	2	2	4
Progress Loss	1	4	4

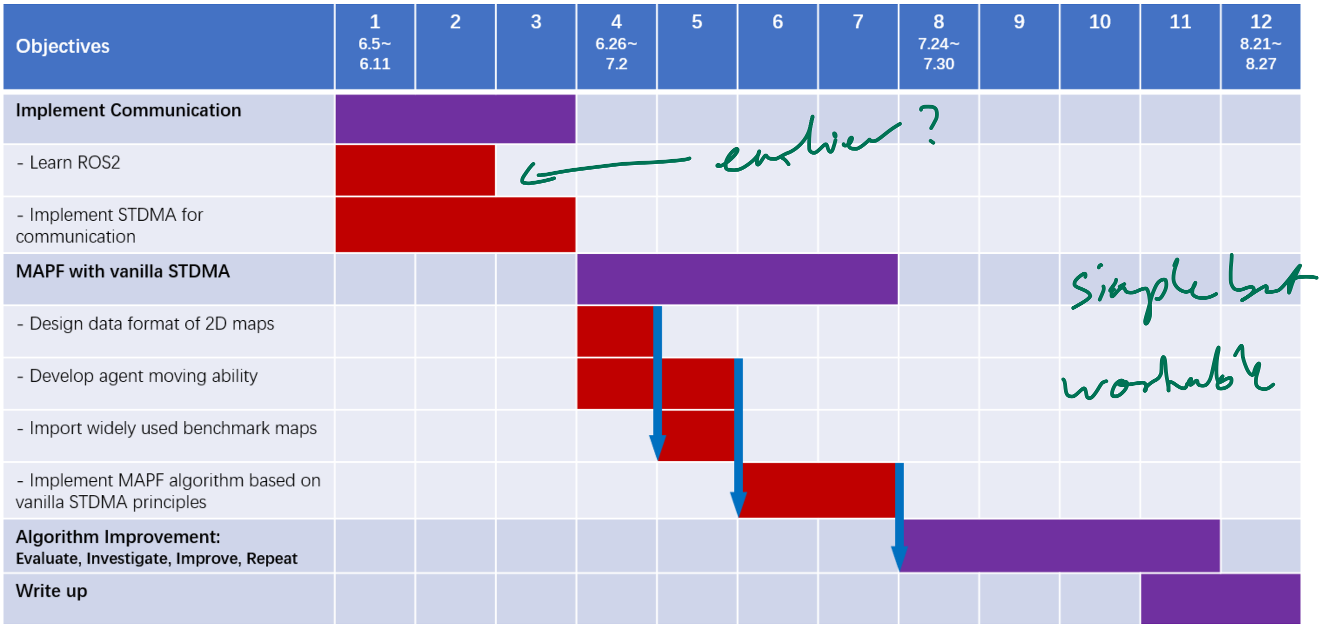
Table 1: Project Risk Register.

bit stretched – but project is quite low-risk.

Need lenovo computer?

6 Timeline

The project's timeline is illustrated in Figure 2, where downward arrows indicate the inter-dependencies among different sub-tasks



earlier?

simple but workable

Figure 2: Project Gantt Chart.

Note: The actual available time is slightly longer than 12 weeks.

7 References

[1] Tristan Gaugel, Jens Mittag, Hannes Hartenstein, Stylianos Papanastasiou, and Erik G. Ström. In-depth analysis and evaluation of self-organizing tdma, 2013.

[2] Roni Stern. Multi-agent path finding–an overview. *Artificial Intelligence: 5th RAAI Summer School, Dolgoprudny, Russia, July 4–7, 2019, Tutorial Lectures*, pages 96–115, 2019.

[3] Roni Stern, Nathan Sturtevant, Ariel Felner, Sven Koenig, Hang Ma, Thayne Walker, Jiaoyang Li, Dor Atzmon, Liron Cohen, TK Kumar, et al. Multi-agent pathfinding: Definitions, variants, and benchmarks. In *Proceedings of the International Symposium on Combinatorial Search*, volume 10, pages 151–158, 2019.

[4] Elham Fatemeh Asadi and Arthur Richards. Scalable distributed model predictive control for constrained systems. *Automatica*, 93:407–414, 2018.

[5] Jae-Han Lim. Understanding stdma via computer simulation: feasibility to vehicular safety applications, configurations, and time synchronization errors. *EURASIP Journal on Wireless Communications and Networking*, 2016:1–14, 2016.

[6] Peter R Wurman, Raffaello D'Andrea, and Mick Mountz. Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI magazine*, 29(1):9–9, 2008.

[7] Ngai Meng Kou, Cheng Peng, Hang Ma, TK Satish Kumar, and Sven Koenig. Idle time optimization for target assignment and path finding in sortation centers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9925–9932, 2020.

[8] Wolfgang Hönig, Scott Kiesel, Andrew Tinka, Joseph W Durham, and Nora Ayanian. Persistent and robust execution of mapf schedules in warehouses. *IEEE Robotics and Automation Letters*, 4(2):1125–1131, 2019.

[9] Kurt Dresner and Peter Stone. A multiagent approach to autonomous intersection management. *Journal of artificial intelligence research*, 31:591–656, 2008.

[10] Federico Pecora, Henrik Andreasson, Masoumeh Mansouri, and Vilian Petkov. A loosely-coupled approach for multi-robot coordination, motion planning and control. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 28, pages 485–493, 2018.

[11] João Salvado, Robert Krug, Masoumeh Mansouri, and Fedorico Pecora. Motion planning and goal assignment for robot fleets using trajectory optimization. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7939–7946, 2018.

[12] Reza Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on automatic control*, 51(3):401–420, 2006.

[13] Hang Ma, Jingxing Yang, Liron Cohen, TK Kumar, and Sven Koenig. Feasibility study: Moving non-homogeneous teams in congested video game environments. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 13, pages 270–272, 2017.

[14] Wolfgang Hönig, James A Preiss, TK Satish Kumar, Gaurav S Sukhatme, and Nora Ayanian. Trajectory planning for quadrotor swarms. *IEEE Transactions on Robotics*, 34(4):856–869, 2018.

[15] Heiner Lasi, Peter Fettke, Hans-Georg Kemper, Thomas Feld, and Michael Hoffmann. Industry 4.0. *Business & information systems engineering*, 6:239–242, 2014.

[16] Daniel Alejandro Rossit, Fernando Tohmé, and Mariano Frutos. Industry 4.0: smart scheduling. *International Journal of Production Research*, 57(12):3802–3813, 2019.

[17] Matthias De Ryck, Mark Versteyhe, and Frederik Debrouwere. Automated guided vehicle systems, state-of-the-art control algorithms and techniques. *Journal of Manufacturing Systems*, 54:152–173, 2020.

[18] Zaloa Fernández, Iñaki Val, Mikel Mendicute, and Elisabeth Uhlemann. Analysis and evaluation of self-organizing tdma for industrial applications. In *2019 15th IEEE International Workshop on Factory Communication Systems (WFCS)*, pages 1–8, 2019.

[19] Hamed Mosavat-Jahromi, Yue Li, Yuanzhi Ni, and Lin Cai. Distributed and adaptive reservation mac protocol for beaconing in vehicular networks. *IEEE Transactions on Mobile Computing*, 20(10):2936–2948, 2020.

[20] Takeru Terauchi, Katsuya Suto, and Masashi Wakaiki. Harvest-then-transmit-based tdma protocol with statistical channel state information for wireless powered sensor networks. In *2021 IEEE 93rd vehicular technology conference (VTC2021-Spring)*, pages 1–5. IEEE, 2021.

[21] Tao Zhao, Haodong Li, and Songyi Dian. Multi-robot path planning based on improved artificial potential field and fuzzy inference system. *Journal of Intelligent & Fuzzy Systems*, 39(5):7621–7637, 2020.

[22] Chenyuan He, Yan Wan, Yixin Gu, and Frank L Lewis. Integral reinforcement learning-based multi-robot minimum time-energy path planning subject to collision avoidance and unknown environmental disturbances. *IEEE Control Systems Letters*, 5(3):983–988, 2020.

[23] Junxiao Xue, Xiangyan Kong, Bowei Dong, and Mingliang Xu. Multi-agent path planning based on mpc and ddpg. *arXiv preprint arXiv:2102.13283*, 2021.

[24] Haodong Li, Tao Zhao, and Songyi Dian. Prioritized planning algorithm for multi-robot collision avoidance based on artificial untraversable vertex. *Applied Intelligence*, 52(1):429–451, 2022.

[25] Haodong Li, Tao Zhao, and Songyi Dian. Prioritized planning algorithm for multi-robot collision avoidance based on artificial untraversable vertex. *Applied Intelligence*, 52(1):429–451, 2022.

[26] Zhiheng Yuan, Zhengmao Yang, Lingling Lv, and Yanjun Shi. A bi-level path planning algorithm for multi-agv routing problem. *Electronics*, 9(9):1351, 2020.

[27] Hamid Shiri, Hyowoon Seo, Jihong Park, and Mehdi Bennis. Attention-based communication and control for multi-uav path planning. *IEEE Wireless Communications Letters*, 11(7):1409–1413, 2022.

[28] Daqi Zhu, Bei Zhou, and Simon X Yang. A novel algorithm of multi-auvs task assignment and path planning based on biologically inspired neural network map. *IEEE Transactions on Intelligent Vehicles*, 6(2):333–342, 2020.

[29] Tingxiang Fan, Pinxin Long, Wenxi Liu, and Jia Pan. Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios. *The International Journal of Robotics Research*, 39(7):856–892, 2020.

[30] Tingxiang Fan, Pinxin Long, Wenxi Liu, and Jia Pan. Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios. *The International Journal of Robotics Research*, 39(7):856–892, 2020.

[31] EN Mobarez, A Sarhan, and MM Ashry. Obstacle avoidance for multi-uav path planning based on particle swarm optimization. In *IOP Conference Series: Materials Science and Engineering*, volume 1172, page 012039. IOP Publishing, 2021.

[32] Zhihuan Chen, Huaiyu Wu, Yang Chen, Lei Cheng, and Binqiao Zhang. Patrol robot path planning in nuclear power plant using an interval multi-objective particle swarm optimization algorithm. *Applied Soft Computing*, 116:108192, 2022.

[33] Jiancheng Song, Liang Liu, Yulei Liu, Jie Xi, and Wenbin Zhai. Path planning for multi-vehicle-assisted multi-uavs in mobile crowdsensing. *Wireless Communications and Mobile Computing*, 2022:1–21, 2022.

[34] Raza Abdulla Saeed, Diego Reforgiato Recupero, and Paolo Remagnino. The boundary node method for multi-robot multi-goal path planning problems. *Expert Systems*, 38(6):e12691, 2021.

[35] Guohong Yi, Zhili Feng, Tiancan Mei, Pushan Li, Wang Jin, and Siyuan Chen. Multi-agvs path planning based on improved ant colony algorithm. *The Journal of Supercomputing*, 75:5898–5913, 2019.

[36] Jing Liu, Sreenatha Anavatti, Matthew Garratt, and Hussein A Abbass. Modified continuous ant colony optimisation for multiple unmanned ground vehicle path planning. *Expert Systems with Applications*, 196:116605, 2022.

[37] Daifeng Zhang and Haibin Duan. Social-class pigeon-inspired optimization and time stamp segmentation for multi-uav cooperative path planning. *Neurocomputing*, 313:229–246, 2018.

[38] Bo Hang Wang, Dao Bo Wang, and Zain Anwar Ali. A cauchy mutant pigeon-inspired optimization–based multi-unmanned aerial vehicle path planning method. *Measurement and Control*, 53(1-2):83–92, 2020.

[39] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. Grey wolf optimizer. *Advances in engineering software*, 69:46–61, 2014.

[40] Meng Zhou, Zihao Wang, Jing Wang, and Zhe Dong. A hybrid path planning and formation control strategy of multi-robots in a dynamic environment. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 26(3):342–354, 2022.

[41] Gewen Huang, Yanguang Cai, Jianqi Liu, Yuanhang Qi, and Xiaozhou Liu. A novel hybrid discrete grey wolf optimizer algorithm for multi-uav path planning. *Journal of Intelligent & Robotic Systems*, 103:1–18, 2021.

[42] Ryan Luna and Kostas E Bekris. Push and swap: Fast cooperative path-finding with completeness guarantees. In *IJCAI*, pages 294–300, 2011.

[43] Miroslav Kulich, Tomáš Novák, and Libor Přeucil. Push, stop, and replan: An application of pebble motion on graphs to planning in automated warehouses. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 4456–4463, 2019.

[44] Hanlin Wang and Michael Rubenstein. Walk, stop, count, and swap: Decentralized multi-agent path finding with theoretical guarantees. *IEEE Robotics and Automation Letters*, 5(2):1119–1126, 2020.