Journal of Physics: Conference Series

PAPER • **OPEN ACCESS**

Neighbor Reward with Optimal Reciprocal Collision Avoidance for Swarm Agents*

To cite this article: Linlin Du *et al* 2022 *J. Phys.: Conf. Ser.* **2216** 012082

Neighbor Reward with Optimal Reciprocal Collision Avoidance for Swarm Agents*

Linlin Du[1*], Hao Tang[2], Pengfei Li[3], Tao Ma[4], Shuangquan Ge[5], Kang Cao[6]

[1]Computer Application Institute China Academy of Engineering Physics, Mianyang, 621900, China

[2]Computer Application Institute China Academy of Engineering Physics, Mianyang, 621900, China

[3]Computer Application Institute China Academy of Engineering Physics, Mianyang, 621900, China

[4]Computer Application Institute China Academy of Engineering Physics, Mianyang, 621900, China

[5]Computer Application Institute China Academy of Engineering Physics, Mianyang, 621900, China

[6]Computer Application Institute China Academy of Engineering Physics, Mianyang, 621900, China

*dll.00@qq.com

ABSTRACT: Navigating in an unknown area safely is counted as the underlying work which can support swarm agents for more complex tasks. When available information of search regions are lacking, agents make real-time action decisions according to surrounding environments they have perceived. For swarm agent system, connectivity maintenance and collision avoidance are both essential. Based on optimal Reciprocal Collision Avoidance (ORCA) algorithm, we proposed a method that agents can provide assistances to surrounding agents by spreading the status information of themselves, which is the neighbor reward method (NRM). This kind of status information contains ambient information and perceptions of the task which are transferred to reward data for convenient and uniform distributions. In other words, individuals utilize inter-neighbor interactions to achieve the same high-level goal, as well as result in an intelligent independent swarm agents system.

This method solves the velocity selection problem of ORCA and optimizes the obstacle avoidance of the original NRM. The algorithm has been integrated in ROS framework and simulated on GAZEBO. In the tested scenario, our method is efficient for swarm agents collision avoidance in decentralized way.

1.INTRODUCTION

The increasing need for safe, inexpensive and quick search, combined with the development of unmanned equipment, has made search missions by a team of agents in the spotlight. Without a previous knowledge of the searching area, a fleet of robotic agents cannot rely on a pre-planning action. Unified plan by one central control node is also hard for the reason, that real-time task allocation and collision avoid between agents are hard to balance [1]. When a fleet of robotic agents navigates in a shared workspace, there arose another risk that robots may frequently block each other's ways [2]. As for distributed control model, agents' cooperation and collision avoidance rely on local real-time perception, decision and planning. Thus, frequent and massive interactions with neighbor agents and environments for collision-free collective behaviors are inevitable which may cause great amount of calculation.

Cooperation and collision avoidance are of equal importance for the agent swarm. There is a great deal of research focusing on decentralized flocking of agent swarm which derived from swarm intelligence in natural systems [3]. Three rules of flocking agent swarm are flock centering, collision avoidance and velocity matching [4]. In research by Olfati [5], protocol unified all three rules most likely leads to regular fragmentation instead of flocking behavior. Olfati proposed multi-species framework for construction of collective potentials that leads to flocking and obstacle avoidance. In many previous studies [6, 7], the ability of identifying relative orientation of neighbors is indispensable for agents to achieve flocking behaviors or collision avoidance. To further simplify the agent intelligence required for collective behaviors, a minimal control multi-agent model (MCMA) is proposed by Chen and Zhang [8]. MCMA is rigorously proved to be collision avoidance and velocity alignment. Agents employ relative positions and maneuverable heading to realize collision avoidance. The guidance of the two walls and a constant speed of all agents form velocity alignment.

For agent swarm, the collision avoidance with obstacles and neighbors are different. Most studies in the field of agent swarm have only focused on the formation of agents, which adopt simple obstacles avoidance strategy. In studies [6-9], formation maintenance guarantees the safe distance between agents, artificial potential filed method and virtual force field keep swarm agents away from obstacles as a whole. Put the coordination problem aside, there are a lot of studies about collision avoidance. Fox et.al [11] proposed dynamic window approach. Van den Berg et al. [10] proposed Optimal Reciprocal Collision Avoidance (ORCA) algorithm which avoid collision by providing pairwise maneuvers. Deep reinforcement learning has received extensive attention and study in multi-agent collision avoidance area [12, 13].

In this paper, we address the above issues by going beyond geometrical cohesion and combine the unified goal. All agents are informed with the unified goal direction and equipped with sensors that could perceive obstacles and other agents. On the one hand, we introduce ORCA algorithm to avoid collision. On the other hand, agent- agent interactions are modeled through neighbor reward model (NRM).

The rest of the paper is organized as follows. Section 2 reviews previous approaches as well as the problem to swarm agents collision-free navigation. Section 3 presents the details of our approach. Then the effectiveness of our proposed model is verified by simulated situations in Section 4. The conclusion and perspective are given in the last section.

2.BACKGROUND

2.1.Related Work

Collision avoidance is one of the essential tasks of swarm agents. Artificial potential field [14] is very common in collision avoidance for agents because of its mathematical elegance and implementation for real-time path planning.

In our previous work [15], the obstacle avoidance strategy is wall following, based on artificial potential field. According to the distance from obstacles and goal of agent A, we get the repulsion/attractions which could deduce the velocity for agent A avoiding collisions [16-18]. Nevertheless, with the potential function, agents' movements tend to be zigzag due to the kinematics and dynamics of agents. To solve the problem above, agents ignore attractions of goal when there are obstacles on the goal direction. Therefore, agents follow obstacles' contours when they encounter obstacles.

Regarding of multi-agent system, the velocity-based approach for avoiding collisions such as RVO [19] and ORCA [10] are applied to agents. In [10], each agent selects its velocity from its own 2-D velocity space and dose not communicate with other agents. In this method agents are nothing more than moving obstacles among one another, which cannot generalize well for swarm cooperation.

Speak of agent-agent interactions, swarm agents can collect information from the whole and communicate as team. Nonetheless, the decision of which message should be sent and how to move is a complex process. Nature-inspired methods are being used in several works [20-24]. In our work, we introduce the NRM to help ORCA select velocity.

2.2.Problem Formulation

In this work, we consider a swarm agents navigating through an unknown environment with a set of obstacles. All agents move toward a unified goal. Our approach uses ORCA, a velocity-based method to solve collision avoidance, and NRM to work on cohesion. We assume agents are moving in the plane E^2 . This algorithm can easily be extended to higher dimensions. For each agent A has a current position $\mathbf{p}_A$, a current velocity $\mathbf{v}_A$ and a preferred velocity $\mathbf{v}_A^{pref}$. Each agent are the same, which means they have the same velocity constraints (maximum speeds - $\mathbf{v}_{max}$) and a safe distance radius $\mathbf{d}_{safe}$ for avoiding collisions.

The objective of swarm agents is to cross barriers safely. At a preset amount of time δ , each agent A independently selects a new velocity $\mathbf{v}_A^{new}$ for swarm agents to avoid collision. Simultaneously, the advantage of swarm agents sharing the same goal is applied to mutual help to find the goal. Thus, for each agent has state $\mathbf{s}_A$ and reward $\mathbf{r}_A$. The reward $\mathbf{r}_A$ denotes the state of agent A and surroundings around it, that can be divided into velocity-reward $\mathbf{g}_A$ and obstacle-reward $\mathbf{o}_A$. In order to arise awareness of surroundings and come to rational new velocity $\mathbf{v}_A^{new}$, Agent A compares reward $\mathbf{r}_A$ with neighbor agents' rewards.

3.APPROACH

As we have explained previously, the purpose of this paper is to present a real-time support approach for distributed collision avoidance and connectivity. The NRM separates the collision avoidance between neighbor agents and obstacles, which requires precise discernment (NRM). This motivates us to optimize the collision avoidance strategy by combining ORCA. We name this model NRM-O (Neighbor Reward Model with ORCA). Note that the neighbor reward communications between agents are not central coordination.

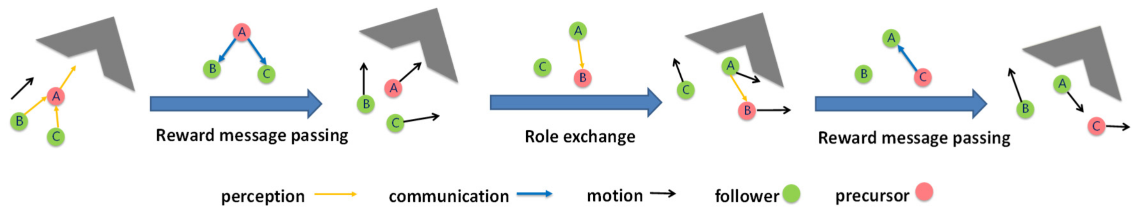


Figure 1. Three robots A, B and C navigate through a gray obstacle. The role of each robot is decided independently.

Figure 1 demonstrates the feedback process of reward model. Firstly, when agents haven't encounter obstacles, they move toward the same goal, which is replaced by the same direction here. We treat agent A as precursor, agent B and C as follower on the basis of their positions and orientations. The front agent A detects the obstacle first and affects the reward $\mathbf{r}_A$. Agent B and C subscribe the reward $\mathbf{r}_A$. As soon as agent B and C perceive the change of reward $\mathbf{r}_A$, they compare the reward $\mathbf{r}_A$ with their own and carry out obstacle avoidance strategy in advance. Even agent B and C haven't perceived the obstacle. The role change of agents can be seen when agent A turns right to stay away from the gray obstacle and then perceive the reward $\mathbf{r}_C$ of agent C. In this case, $\mathbf{r}_C$ is higher than $\mathbf{r}_A$. Thus, agent A will learn and follow the behavior of agent C.

$$ratio_A = \left|2 \times \left(\frac{1}{1+e^{\mathbf{o}_A}} - \frac{1}{2}\right)\right| \quad (1)$$
$$\mathbf{r}_A = ratio_A \times \mathbf{o}_A + (1 - ratio_A) \times \mathbf{g}_A$$

As shown in Equation 1, to combine the velocity-reward $\mathbf{g}_A$ and obstacle-reward $\mathbf{o}_A$. On one hand, for $\mathbf{g}_A$ and $\mathbf{o}_A$, when the value of $\mathbf{o}_A$ is high, the reference value of $\mathbf{g}_A$ is low, because of the crowded condition. On the other hand, when the $\mathbf{o}_A$ is low, $\mathbf{g}_A$ should be highlight. Thus, $ratio_A$ is used to normalize $\mathbf{r}_A$.

The velocity-reward $\mathbf{g}_A$ and obstacle-reward $\mathbf{o}_A$ of agent A are defined as follow:

$$\mathrm{g_A} = \frac{2 \times \mathrm{Gaussian}\left(\mathrm{a1}, \mathrm{v_A^{pref}}, \mathrm{c1}\right)}{\mathrm{a1}} - 1 \quad (2)$$
$$\mathrm{o_A} = \sum \frac{1}{\mathrm{p_A}}$$

The p_A is the distance from agent A to obstacles.

For agent A and B, the velocity obstacle $\mathrm{VO}^{\delta}_{A|B}$ is the set of all relative velocities of A with respect to B that will result in a collision between A and B within time horizon $[0, \delta]$[19], as is shown below:

$$\mathrm{VO}^{\delta}_{A|B} = \{v \mid \exists t \in [0, \delta] :: tv \in D\left(\mathbf{p}_B - \mathbf{p}_A, \mathbf{d}_{safe} + \mathbf{d}_{safe}\right)\} \quad (3)$$

where $D\left(\mathbf{p}_B - \mathbf{p}_A, \mathbf{d}_{safe} + \mathbf{d}_{safe}\right)$ denotes an open disc with radius $\mathbf{d}_{safe} + \mathbf{d}_{safe}$ centered at the relative position $\mathbf{p}_B - \mathbf{p}_A$.

The ORCA set is composed of all possible permitted velocity vectors, and the set can be geometrically interpreted as a half-plane region.

$$\mathrm{ORCA}^{\delta}_{A|B} = \{v \mid \left(v - (\mathbf{v}_A^{opt} + \frac{1}{2}u)\right) \cdot n \geq 0\} \quad (4)$$

Here introduces the optimization velocities $\mathbf{v}_A^{opt}$ and the vector from $\mathbf{v}_A^{opt} - \mathbf{v}_B^{opt}$ to the closest point on the boundary of the velocity obstacle u, as shown in Fig. 2.

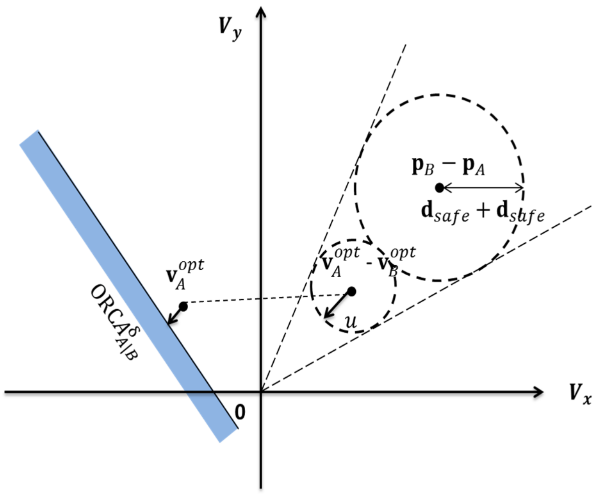


Figure 2. The geometrical representation of ORCA.

How to choose $\mathbf{v}_A^{opt}$ is one issue for each agent A [10]. In our agent swarm, the preferred velocity can be observed by other agents, due to the unified goal of all agents. Besides, neighbor reward implies current surroundings of other agents, that can guide the next time step action of agent A. Thus, we can get the locally optimal solution of the new velocity $\mathbf{v}_A^{new}$.

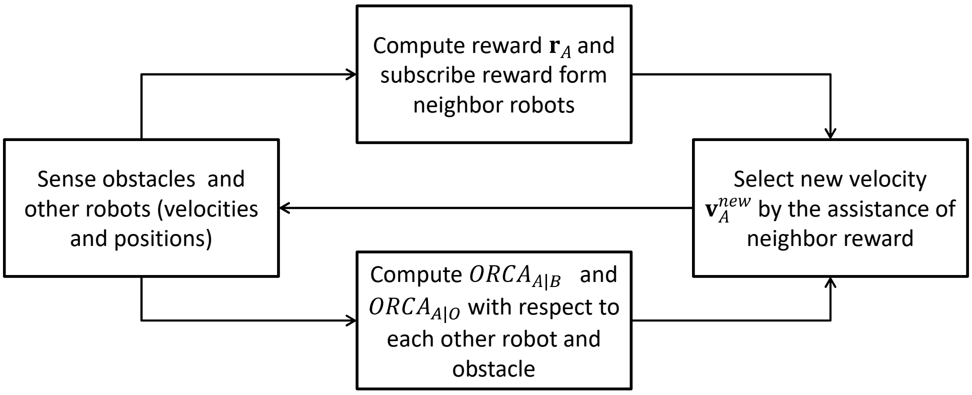


Figure 3. Procedure of NRM-O (neighbor reward modal with optimal reciprocal collision avoidance).

In the following subsections, we present the architecture and formulations of NRM-O. The general procedure for the NRM-O approach is shown in Fig. 3. NRM and ORCA are two methods calculated by positions and velocity of other agents and obstacles. Agent A compares $\mathbf{r}_A$ with neighbor agents' reward. There are two possibilities:

1. $\mathrm{r_A}$ is the best one. Agent A selects $\mathrm{v_A^{new}}$ which is closest to $\mathrm{v_A^{pref}}$ amongst all velocities inside the region of permitted velocities.

$$\mathbf{v}_A^{new} = \underset{v \in \mathrm{ORCA}_A^{\delta}}{\operatorname{argmin}} \left\|\mathbf{v} - \mathbf{v}_A^{pref}\right\| \tag{5}$$

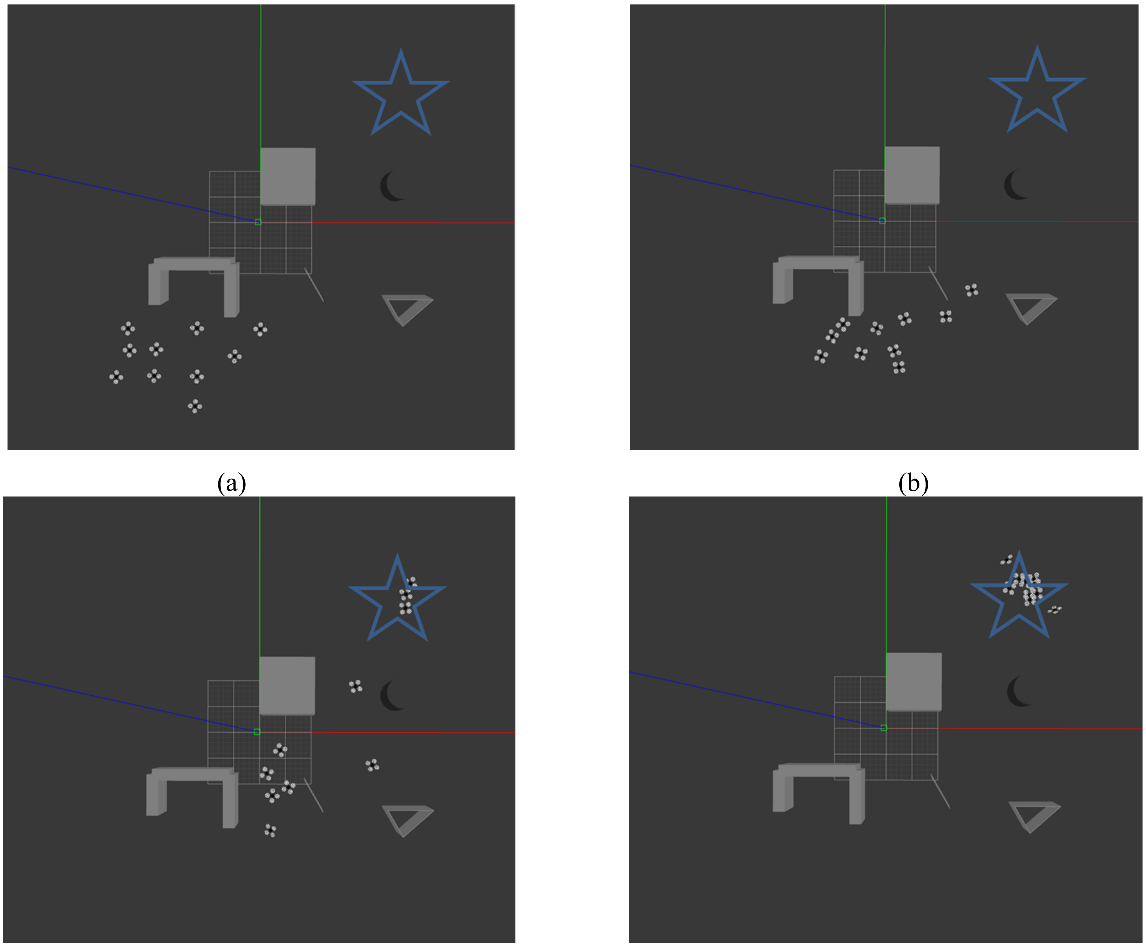
2. $\mathrm{r_A}$ is not the best one. Agent A finds the best reward neighbor agent with the current velocity $\mathrm{v_{best}}$. Then agent A selects $\mathrm{v_A^{new}}$ which is closest to $\mathrm{v_{best}}$ amongst all velocities inside the region of permitted velocities.

$$\mathbf{v}_A^{new} = \underset{v \in \mathrm{ORCA}_A^{\delta}}{\operatorname{argmin}} \left\|\mathbf{v} - \mathbf{v}_{best}\right\| \tag{6}$$

$\mathbf{v}_A^{new}$ is a result that deals with the balance of collision avoidance and coordination with neighbor agents in an area with barriers. We utilized collective efforts of agent swarm which agents improve environmental perception through communications between neighbor agents for better decision.

4.EXPERIMENTAL RESULTS

We built a simulation environment in ROS and GAZEBO for swarm robots (UAVs) navigation in an area with obstacles. Movements of UAVs are controlled by PID in this simulation, which makes the kinematics and dynamics of UAVs in touch with facts. We incorporated our neighbor reward modal with optimal reciprocal collision avoidance formulation into the existing obstacles simulation framework of [15]. All UAVs are randomly positioned at the lower left of the scenario. A unified goal for all UAVs locates at the upper right corner of the scenario. In the central of the scenario, some obstacles are placed in. The simulation scenario can be seen in Fig. 4.



(a) (b)

(c) (d)

Figure 4. Navigation scenario (top side). In the scenario tested, the neighbor reward method (NRM) combined with ORCA shows the ability to navigate in an area with obstacles safely.

As we have seen in this Fig. 4(a), shapes of obstacles are square, triangle, linear, round and concave obstacles, which all of them can formed by line. Thus, ORCA for static obstacles can be easily used. To test NRM-O, the open-source multi-agent simulation system: the RVO2 Library [10] is incorporated. RVO2 Library provides a global goal for all agents and can realize collision-free navigation. NRM ensures UAVs reflect their own states and surroundings information in real time. Through comparison of neighbor reward, next action orientation is decided. We replace the real-time preferred velocity which responds to the neighbor reward, instead of the preferred velocity based on its own.

The goal positioned at (30,30). 10 UAVs with the same speed (4m/s) and head to goal. When there are no obstacles and neighbor on the way of agent, the preferred velocity is orientating to goal position. Once the agent detected collision potential, it takes action to avoid collision. Fig. 4(b) shows no agent is trapped in the concave obstacle, even its right in front of the way to the goal. This shows the neighbor reward guides agent to avoid obstacles in advance, that also generate cohesion of swarm naturally. Along the flight UAVs keep required safe separation. As soon as UAVs confirm a safe situation, they take action to head to their goal positon, as shown in Fig. 4(c) and Fig. 4(d). It illustrates that the unknown scenario with obstacles and cohesion with neighbor situations can be handled successfully.

5.CONCLUSION

In this work, we tackle the swarm agents collision-free navigation problem by combination of NRM and ORCA. We first model the state and surrounding of agents into velocity reward and obstacle reward and then integrate them as neighbor reward to realize interactions among swarm agents. Neighbor reward helps agents make better decisions by learning from neighbors. Meanwhile, we solve the preferred velocity selection problem of ORCA by guidance of neighbor reward. Our approach ensures collision avoidance task accomplishments and cohesion of swarm agents while remaining a distributed system. The results of the simulations indicate that the proposed approach is both valid and efficient.

The deep reinforcement learning techniques show the power to implicitly encode the interactions and cooperation among agents. In future work, we would like to introduce the reinforcement learning frameworks to learn cooperative polices between agent swarm. The reward formulation in present NRM will be extended to the reinforcement learning frameworks deployments.

REFERENCES

[1] Chaimowicz, L., & Kumar, V.. (2007). Aerial shepherds: coordination among uavs and swarms of robots. Springer Japan.
[2] Standley, T. S., & Korf, R. E.. (2011). Complete Algorithms for Cooperative Pathfinding Problems. International Joint Conference on Artificial Intelligence. AAAI Press.
[3] Khaldi, B., & Cherif, F.. (2015). An overview of swarm robotics: swarm intelligence applied to multi-robotics. International Journal of Computer Applications, 126(2), 31-37.
[4] Reynolds, C. W.. (1987). Flocks, herds and schools: a distributed behavior model. Proc. SIGGRAPH '87, 25.
[5] Olfati-Saber, R.. (2006). Flocking for multi-agent dynamic systems: algorithms and theory. IEEE Transactions on Automatic Control, 51(3), 401-420.
[6] Hou-Sheng, S. U.. (2012). Flocking in multi-agent systems with multiple virtual leaders based only on position measurements. Communications in Theoretical Physics, 57(5), 7.
[7] Pei, H., Chen, S., & Lai, Q.. (2015). A local flocking algorithm of multi-agent dynamic systems. International Journal of Control, 88(11), 1-8.
[8] Chen, Z., & Zhang, H. T.. (2017). A minimal control multiagent for collision avoidance and velocity alignment. IEEE Transactions on Cybernetics, 47(8), 2185-2192.
[9] D Zhou, & Schwager, M.. (2016). Assistive collision avoidance for quadrotor swarm teleoperation. IEEE International Conference on Robotics & Automation (pp.1249-1254). IEEE.
[10] Berg, J., Guy, S. J., Ming, L., & Manocha, D.. (2011). Reciprocal n-body collision avoidance. Robotics Research.
[11] Fox, D., & Burgard, W.. (1997). The dynamic window approach to collision avoidance. IEEE Robotics & Automation Magazine, 4(1), P.23-33.
[12] Long, P., Fan, T., Liao, X., Liu, W., Zhang, H., & Pan, J.. (2018). Towards Optimally Decentralized Multi-Robot Collision Avoidance via Deep Reinforcement Learning. 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE.
[13] Everett, M., Yu, F. C., & How, J. P.. (2018). Motion Planning Among Dynamic, Decision-Making Agents with Deep Reinforcement Learning. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE.
[14] Kathib, O.. (1986). Real-time obstacle avoidance for manipulators and mobile robots. Springer New York.
[15] Du, L., Y Huang, Y Tang, & Huang, D. (2019). A Distributed Control Method Based on Neighbor Reward for Robot Swarm. RCAE 2019: 2019 The 2nd International Conference on Robotics, Control and Automation Engineering.
[16] Kim, D. H. , & Shin, S. . (2006). Local path planning using a new artificial potential function composition and its analytical design guidelines. Advanced Robotics, 20(1), 115-135.

[17] Rehmatullah, F. , & Kelly, J. . (2015). Vision-Based Collision Avoidance for Personal Aerial Vehicles Using Dynamic Potential Fields. Computer & Robot Vision. IEEE.
[18] Nieuwenhuisen, M. , Droeschel, D. , Schneider, J. , Holz, D. , & Behnke, S. . (2013). Multimodal Obstacle Detection and Collision Avoidance for Micro Aerial Vehicles. 6th European Conference on Mobile Robots (ECMR). IEEE.
[19] Berg, J. P. V. D. , Lin, M. C. , & Manocha, D. . (2008). Reciprocal Velocity Obstacles for Real-Time Multi-agent Navigation. 2008 IEEE International Conference on Robotics and Automation, ICRA 2008, May 19-23, 2008, Pasadena, California, USA. IEEE.
[20] Jain, L. C., & Fukuda, T.. (1998). Soft computing for intelligent robotic systems. Studies in Fuzziness and Soft Computing, 21.
[21] Stephane, D., Nicolas, B., Jean-Baptiste, M., & Eiben, A. E.. (2015). Evolutionary robotics: what, why, and where to. Frontiers in Robotics and AI, 2.
[22] Schmickl, T., & Hamann, H.. (2010). Beeclust: a swarm algorithm derived from honeybees. Bio.
[23] Pietro Liò, Yoneki, E., Crowcroft, J., & Verma, D. C.. (2008). Bio-inspired computing and communication: first workshop on bio-inspired design of networks, biowire 2007 cambridge, uk, april 2-5, 2007, revised papers. Springer-Verlag.
[24] Mendona, R. M. D., Nedjah, N., & Mourelle, L.. (2013). Efficient distributed algorithm of dynamic task assignment for swarm robotics. Springer Berlin Heidelberg.