New collision avoidance algorithm approach based on social sciences

Shuntaro Tamura[1] and Masahide Horita[1]

[1]University of Tokyo, Japan.

tamura-syuntaro653@g.ecc.u-tokyo.ac.jp, horita@g.ecc.u-tokyo.ac.jp

Abstract -

In future automated construction sites, multiple autonomous agents must navigate safely and efficiently. While velocity obstacle-based algorithms like Optimal Reciprocal Collision Avoidance(ORCA) handle microlevel control, achieving efficiency also requires the management of macrolevel dynamics, similar to human social behavior. The existing C-Nav algorithm, which incorporates the concept of "consideration for others" to promote yielding behavior, shows promise, but its evaluation method can lead to inefficient or nonconsiderate actions. This study found out that insufficient space for yielding causes these issues, and proposes a new evaluation method to encourage social behaviors. The results indicate reduced stacking in narrow areas, although some inefficiencies persist. As a result, the "waiting behind" action functions as an auxiliary mechanism that reinforces the mutual yielding behavior intended by C-Nav, thereby enhancing overall operational efficiency. These insights suggest that further refining the algorithm, guided by a deeper understanding of human social interactions, can lead to more robust and efficient navigation strategies.

Keywords -

Multiagent navigation; Coordination; Robotics

1 Introduction

Owing to the growing shortage of skilled workers in the construction sector, the automation of construction machinery is advancing to improve productivity and safety. A noteworthy example is Taisei Corporation's T-iCraft®, which tests coordinated operations using four types of autonomous machines, namely backhoes, crawlers, dump trucks, and bulldozers.

Path planning for autonomous robots relies heavily on their ability to "pass each other." For instance, in environments in which robots cannot pass each other, all pathways must be one-way or single-track, which leads to inefficient movements.

Figure 1 illustrates a situation in a construction site where passageways are blocked by construction materials, making it difficult for robots to pass each other. Such situations are expected to frequently occur in construction sites, and developing methods to address them is crucial for advancing automation.

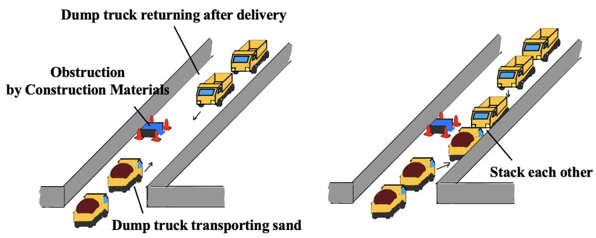

Figure 1. Illustration of the Situation in Which Robots Cannot Pass Each Other

Planning the movements of multiple autonomous construction machines (referred to as "agents") in continuous space to avoid collisions is not easy. Although humans can easily manage passing movements, autonomous machines require advanced algorithms. Velocity obstacle-based algorithms like Optimal Reciprocal Collision Avoidance (ORCA) [1], Reciprocal Velocity Obstacle (RVO) [2], Hybrid Reciprocal Velocity Obstacle (HRVO) [3], and Extended Velocity Obstacle (EVO) [4] treat other agents as dynamic obstacles and calculate safe velocities to prevent collisions.

Velocity obstacle-based algorithms use distributed systems in which each agent independently calculates its movement based on predefined rules. . This ability is aligned well with construction sites, where obstacles frequently appear, as illustrated in Figure 1. In addition, construction machines are typically large and lack maneuverability, making detours less feasible compared to waiting for clearance. This constraint is unique to construction sites, and under this premise, the avoidance-focused algorithm proposed in this study is expected to be highly effective.

However, the algorithms often leads to overall efficiency reductions such as traffic jams (referred to as "stacks"). By drawing inspiration from human interactions, such as crowd dynamics and mutual respect, and incorporating a social modeling framework into these algorithms, these types of problems may be mitigated. The literature on game theory and other related disciplines in the social sciences has a crucial role in developing these enhanced algorithms.

C-Nav [5] is a collision avoidance algorithm that incorporates the concept of "empathy," voluntarily balancing agents' progression toward their goals in consideration of the freedom of others. While C-Nav improves movement efficiency in narrow passages, challenges remain in diverse site conditions and parameter dependencies. Strengthening the generalizability and robustness of such algorithms is essential for their practical deployment.

The purpose of this study is to analyze the effects of enhancing sociality, by introducing more social parameters within the existing C-Nav algorithm, on resulting group performances. It aims to apply analogies with human social behaviors to agent interactions and examine whether they improve the algorithm's performance. Methodologically, we conduct herein computer simulations based on existing ORCA and C-Nav codes, visualize agent behaviors, propose improvements, and compare results to evaluate enhancements. The novelty of this study is its emphasis of algorithmic robustness and strengthening of this robustness through the incorporation of new social behavior concepts.

2 Related Work

2.1 Overview of Collision Avoidance Algorithms

Various approaches exist for collision avoidance algorithms in a continuous space. The mainstream approach is that associated with the use of velocity obstacle-based algorithms. In these algorithms, agents calculate the velocity (to be adopted next) at each time step based on information about their surrounding environment. One of the advantages of this approach is its distributed nature. Instead of relying on a central decision-maker overseeing the entire environment, individual players plan their paths based on their judgments. This improves the computational efficiency and allows flexible responses to dynamic environments.

2.2 Overview of ORCA

Various formulations have been proposed to determine "safe" velocities, aiming to ensure safety while maintaining the maximum possible freedom for velocity selection. Among these, ORCA is currently one of the most extensively used methods. ORCA calculates the "collision velocity range" based on the velocities and positions of the surrounding agents. By "sharing" velocity vector adjustments to escape from this range with other agents, ORCA can estimate a "safe velocity range" with reasonable accuracy. Within this "safe velocity range," agents are free to select velocities (e.g., by choosing the direction and speed that brings them closest to their goal).

This method has been demonstrated to enable collision avoidance among thousands of agents, with the computation time scaling linearly with the number of agents. However, as ORCA focuses on guaranteeing safe velocities at each time step, there is room for improvement in motion planning from a macroscopic perspective. For example, in narrow passages, simply directing groups of agents toward their goals can lead to stacking when the two groups intersect.

Since the introduction of ORCA, research on velocity-based collision-avoidance algorithms in environments that guarantee accurate and complete communication has primarily focused on addressing these types of inefficiencies. Examples include the VGVO [6], proxemics [7], and GrpAvoid [8]. The mainstream approach proposes improvements in velocity decisions at a slightly higher macrolevel than ORCA. The same approach was adopted in this study.

2.3 Limitations of the ORCA Algorithm and the C-Nav Algorithm

When proposing algorithms to address the aforementioned limitations of ORCA, many researchers have drawn inspiration from human social behaviors and game-theoretic considerations. Among these, the C-Nav algorithm yields particularly interesting results. C-Nav is an algorithm that incorporates the advanced social behavior concept of "yielding." Explicitly integrating the concept of "empathy" from the related disciplines in social sciences, it predicts actions several steps ahead and rewards movements that reduce constraints on the actions of others. Consequently, it enables construction machines to pass each other in narrow passages, where only one machine can pass at a time.

3 Problem Formulation

3.1 Definition of the Velocity Obstacle Problem

According to Van Den Berg et al. [1], the velocity obstacle (VO) problem is formulated as described below. We consider an environment shared by n agents. Each agent was modeled as a disk moving in a two-dimensional plane. The following information is provided for agent A:

- Current position ($\mathbf{p}_A$): The agent's current location in the 2D plane
- Current velocity ($\mathbf{v}_A$): The agent's current speed and direction
- Radius ($\mathbf{r}_A$): The radius of the agent assumed to be a two-dimensional circle
- Maximum velocity ($v_A^{\max}$): The upper limit of the agent's speed

- Preferred velocity ($\mathbf{v}_A^{\text{pref}}$): The desired velocity in the absence of obstacles, typically directed toward the goal with magnitude $\mathbf{v}_A^{\max}$

The symbols $\mathbf{p}_A$, $\mathbf{v}_A$, and $\mathbf{r}_A$ are considered as external states that are observable through environmental sensing. By contrast, $\mathbf{v}_A^{\max}$ and $\mathbf{v}_A^{\text{pref}}$ are internal states that are not directly observable by the other agents. With a time-step interval τ, each agent selects a new velocity based on the parameters at each time step.

3.2 ORCA Algorithm Details

The primary objective of ORCA is for each agent to select a velocity that ensures that there are no collisions with other agents or obstacles in the subsequent time step. In addition, as a secondary objective, the agent may choose a velocity that is as close as possible to the preferred velocity v_{pref}^A. ORCA was designed to satisfy these requirements.

The collision velocity range refers to the range of velocities based on two agents that may collide within a specific timeframe. Selecting a velocity within this range introduces a risk of collision. The collision avoidance set is the set of velocities that avoids this range in the subsequent time step. By selecting the velocity from this set, we guaranteed that no collisions would occur. Based on this idea, each agent calculates the minimum necessary velocity adjustments to avoid collisions with all surrounding agents. From the convex set of safe velocities obtained through this process, Agent A selects the velocity closest to v_{pref}^A.

3.3 C-Nav Algorithm Details

In this section, we describe the setting of v_{pref}^A, which is a unique feature of C-Nav, according to the explanation of Godoy et al. [5].

At each time step, the agent first simulates hypothetical actions over several future steps for various possible actions it can take in the subsequent time step based on observations of the positions and velocities of the surrounding agents. Examples of possible actions include movements in eight directions based on the direction directly toward the goal.

Each action is then scored according to its degree of goal achievement and those of the surrounding agents. Herein, the agent's degree of goal achievement is defined as the distance moved closer to the goal. The degree of goal achievement for the surrounding agent j (more precisely, the extent to which the agent's action affects the agent's goal achievement) is given by the value $P_{a,j}$

$$P_{a,j} = v^{\max} - \|\mathbf{v}_j^{\text{intent}} - \mathbf{v}_j^{\text{new}}\| \tag{1}$$

where v_j^{intent} is assumed to be communicated by an agent j via a signal.

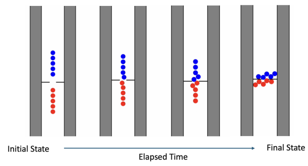

Figure 2. Behavior in the Case of Optimal Reciprocal Collision Avoidance

Consequently, the reward for each action is calculated using the following equations:

$$R_a = (1-\gamma)\cdot R_a^g + \gamma \cdot R_a^c \tag{2}$$

$$R_a^g = \frac{\sum_{t=0}^{T-1}\left(\frac{\mathbf{v}_i^{\text{new}}\cdot(\mathbf{g}_i-\mathbf{p}_i)}{\|\mathbf{g}_i-\mathbf{p}_i\|}\right)}{(T\cdot v^{\max})} \tag{3}$$

$$R_a^c = \frac{\sum_{t=1}^{T-1}\sum_{j\in C_{\text{rank}}} P_{a,j}}{(T-1)\cdot k \cdot v^{\max}} \tag{4}$$

where γ is a parameter that balances the agent's goal achievement and the consideration of other agents. A value of γ closer to zero places more emphasis on the agent's goal achievement, whereas a value closer to one places more emphasis on the consideration of other agents. This γ is known as the coordination factor.

Finally, the action with the highest R_a value is selected and passed to ORCA as v_{pref}^A to determine the final velocity.

4 Observation and Analysis

In this section, we determine what needs to be improved regarding the behavior of agent groups resulting from the existing C-Nav algorithm based on observations of our simulations.

4.1 C-Nav's Superiority over ORCA in Narrow Passages

In this section, we explain how C-Nav overcomes the challenges presented by ORCA and what concrete improvements it achieves in practice. In section 4.2, we present examples of undesirable behaviors in which C-Nav deviates from its intended principles, thereby clarifying directions for further refinement.

First, in the case of a simple ORCA, the two agents are set to move straight toward their respective goals; thus, they both enter a narrow passage. Because they collide, the leading agents of each line try to avoid each other by moving to the left and right, as calculated by the ORCA algorithm. However, because the narrow passage is adequately broad for only one agent, they cannot pass each other and become stuck in the passage (Figure 2).

We now explain the ideal avoidance behavior intended to be used in the original paper [5] in the case of C-Nav. In

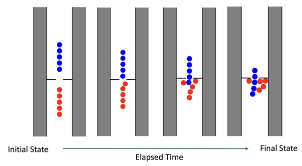

Figure 3. Behavior in the Case of C-Nav

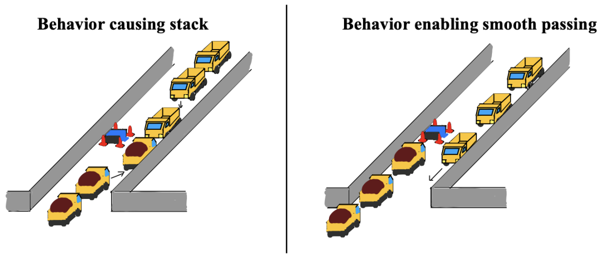

Figure 4. Illustration of the Real-world Narrow Passage Scenarios

Figure 3, the red agent determines that proceeding straight ahead would obstruct the blue agent, and will thus avoid moving to the right. Meanwhile, the blue agent notices that the red agent has moved to the right and proceeds straight ahead, resulting in "passing each other" behavior.

These simulated behaviors correspond to real-world movements observed at construction sites, as shown in Figure 4, specifically as these pertain to the passing of dump trucks heading to transport soil and those returning from transportation.

4.2 Unintended Stacking Caused by C-Nav in Narrow Passages

We also observed the behavior after modifying the width of the narrow passage to allow the two agents to pass through each other side-by-side. The ideal behavior in this field is for the agents to shift to the left and right directions and pass each other while maintaining their formation, as shown in Figure 5. However, when we conducted repeated trials under the same conditions, the following behavior often occurred (Figure 6). In their efforts to predict several

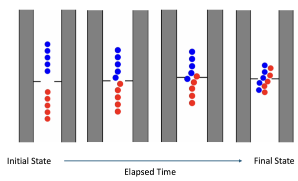

Figure 5. Smooth Behavior Attained in the Case of C-Nav

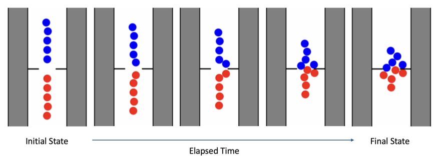

Figure 6. Stacking in the Case of C-Nav

steps ahead of their current positions in cases of straight motion trajectories, both the leading agents of the red and blue groups obstruct each other and fail to achieve their own goals; accordingly, they try to change their courses by moving to the right or left. However, the directions they chose to avoid were the same, resulting in a situation where they blocked each other's paths. In this case, the two leading agents that are "hesitating" restrict the width of the narrow passage, and the agents which follow can only pass through one at a time, reducing the efficiency of the crossing.

The fact that this inefficient behavior occurs often, presents a challenge when considering applications to actual construction sites and indicates a lack of robustness in the current algorithm. Therefore, modifications are required to the algorithm to avoid these types of inefficiencies.

4.3 Analysis of the Cause of Stacking

4.3.1 Changes in Behavior When Switching to Sequential Turns

First, let us consider an analogy to human behavior. In the narrow passage situation presented in Section 4.2.3, it is common to experience congestion in reality because the avoidance directions of oncoming individuals overlap. The essential cause in this example is that "the choice of avoidance direction is made simultaneously." If you knew (from your perspective) that another person would avoid it, you would not avoid it. Believing that this phenomenon also occurs in C-Nav, we modified the algorithm. We changed the evaluation of the actions in each direction, which was previously performed simultaneously based on the desired velocities of each agent at the start of the for-loop—to be performed sequentially. With this change, each agent performs evaluations while recognizing the preferred velocities of surrounding agents.

However, even when this algorithm was used, avoidance actions in the same direction occurred (Figure 7).

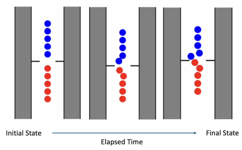

Figure 7. Same-direction Avoidance in the Case of Sequential Turns

Therefore, we concluded that this behavior was not due to the order of turns.

4.3.2 Changes in Behavior When the Number of Agents was Altered

We verified this by changing the number of agents and their intervals.

First, in the one-on-one case, the results were as follows: out of 1000 trials, the shortest and longest completion times were 98 and 112. In other words, severe stacks, such as those described in Section 4.2.3, did not occur.

In another case, the following trajectories were observed: The red and blue lines represent the agents moving upward and downward, respectively. If we focus on the points where each agent first changes its direction of movement, we can observe that both avoid the right side of the screen. However, after several adjustments in direction, these two agents eventually passed through (Figure 8).

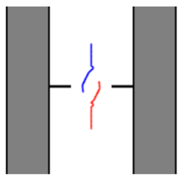

Figure 8. Elimination of "Hesitation" Observed in the 1 on 1 Case

Based on the results presented earlier, we can infer that when there is sufficient space around oneself and the target agent that needs to be avoided, repeated directional adjustments are made; as a result, stacking due to mutual avoidance does not occur.

4.3.3 Specific Mechanism of Stacking in C-Nav and Opportunities for Algorithm Improvement

Based on the results of Sections 4.3.1, and 4.3.2, it can be inferred that the presence of the following agents influences the occurrence of stacks owing to mutual avoidance. Reinterpreting the behavior based on this, the sequence is as follows:

1. In Figure 6, both the red and blue agents first move to the right side of the screen to avoid obstacles.
2. The following agents proceeded straight into the space that had opened up.
3. The leading red and blue agents, who had initially avoided each other by moving to the right, lose the freedom to readjust their direction of movement as in the one-on-one case owing to the entry of the agents that followed; these two agents are continually pushed further to the right by these incoming agents.

It is considered that the stacks occurred because of the aforementioned mechanism.

Considering the meaning of this behavior, it implies that "even though the leading agents are trying to yield the way to the opposing agents, the following agents attempt to overtake and proceed." This contradicts one of the major themes of the C-Nav algorithm, which is "consideration for others."

Furthermore, this behavior was observed in cases where the

1. coordination factor was increased
2. reward target was limited to only the nearest agent
3. agents were allowed to select low-speed actions or stop at desired velocities

These observations indicate that with the current reward function, it is difficult to restrict "overtaking."

4.4 Proposed Improvements

As pointed out by Godoy et al. [5], an attempt to vary the way rewards are given, depending on whether the agents are moving in the same or opposite directions, is considered useful for constructing a more efficient algorithm. Based on this, we will revisit the current method of assigning rewards by focusing on the "direction of movement of the agent being considered."

The current method for assigning consideration rewards is as follows:

$$P_{a,j} = v^{\max} - \|v_j^{\text{intent}} - v_j^{\text{new}}\| \tag{5}$$

In other words, this method values "not hindering the movement of others."

Based on our observations thus far (e.g., passing through narrow passages in one-on-one situations), it has been confirmed that this is effective for agents moving in opposite directions. However, for agents moving in the same direction—especially the agent immediately ahead who is stalled—the possibility that one's actions will directly block the agent's desired path is low. Consequently, differences in the rewards for each of one's actions were less likely to occur. Moreover, as we have observed, blocking the space necessary for avoidance actions between that agent and agents facing it becomes a "hindrance" when viewed over a longer number of time steps. Therefore, a more appropriate method for assigning rewards is required to avoid this.

We can draw an analogy with human social behavior. Suppose that at a place where only one person can pass—such as an automatic door at a building entrance—a few people from inside and another group from outside arrive and stumble across each other. In this situation, the leader of the group (coming from inside or outside) will step aside by the door and yield the way to the other group. At this time, the highest priority for the other members of the group that is yielding (assuming everyone in this group is considerate) is "not to step ahead of the person who is yielding."

Based on the above, a more desirable way of consideration is to "ensure freedom of action for agents moving in the opposite direction as in the existing method, but for agents moving in the same direction, if that agent yields the way or its movement stalls, to line up behind and wait."

In the existing method case, the reward calculation is performed for a specified number of agents in order from those with the largest speed change, regardless of the distances from each other. However, this method of assigning rewards depends on how the neighborhood is defined (e.g., the setting of the neighborhood radius); correspondingly, robustness against the problem situation is not guaranteed. In Godoy et al. [5], there is limited discussion on the order of consideration. When introducing the movement of "lining up behind a stalled agent ahead," it is more intuitive to consider agents as targets for this reward concept in order of increasing proximity to oneself.

Based on this, we propose changing the method for assigning consideration rewards as follows.

Existing reward assignment methods

Without distinguishing the agents, we calculate $P_{a,j}$ for each agent using Eq. 5, and estimate the average of a specified number of agents in order from the ones with the largest $P_{a,j}$ as a consideration reward.

Proposed method for reward assignment

We define whether the "flow directions are the same or opposite" based on whether the inner product of one's desired velocity vector and that of the other agent is positive or negative. In addition, we define an agent's movement as "slow" if its current speed is less than half its maximum speed. In this case,

- For agents j moving in the opposite direction, calculate $P_{a,j}$ using Eq. 5.
- For agents j moving in the same direction, calculate $P_{a,j}$ using the following equation:

$$P_{a,j} = \alpha \times \frac{(p_j - 4.0 \times r_j \times v_j^{\text{intent}}) - p_i}{\|(p_j - 4.0 \times r_j \times v_j^{\text{intent}}) - p_i\|} \cdot v_a \tag{6}$$

In this study, we refer to this method of providing rewards as "Q-Virtue," inspired by the Japanese queuing culture (the virtue of standing in line).

5 Experiments

We conducted experiments to verify whether the proposed changes in the reward rules led to a reduction in stacking in narrow passages.

5.1 Evaluation Method

Using C-Nav as a comparison point, we evaluated the C-Nav algorithm using the proposed reward allocation method.

In this study, we focused on actions to determine whether it is possible to avoid the phenomenon according to which the movement completion time becomes considerably longer than when ideal crossing occurs, owing to stacks resulting from the existing C-Nav. Therefore, we evaluated the method by performing 100 trials and comparing the frequency distribution of the movement completion time (the average time required for 10 agents to reach their goals).

Regarding the parameters of C-Nav, we fixed them at the following values, which showed the best performance in the existing algorithm as a result of the experiments; coordination factor (γ) = 0.9, and number of agents considered in reward = 1.

5.2 Results

For each scenario, we present the following for both the existing C-Nav and the proposed assignment method:

1. A distribution plot with the average completion time (in s) on the horizontal axis and the frequency on the vertical axis
2. A sequence of representative snapshots of observed behavior (the left image shows the initial state and the right image shows the state after some time has elapsed)

5.2.1 Narrow Passage Scenario

First, we consider a situation in which agents can pass through two narrow passages simultaneously.

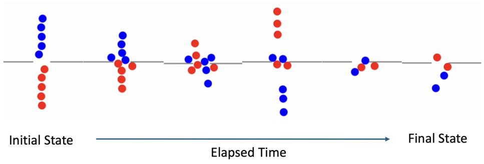

Figure 9. Behavior in the Case of C-Nav

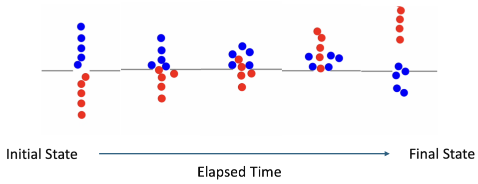

Figure 10. Behavior in the Case of Q-Virtue

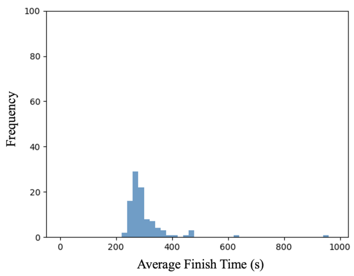

Figure 11. Completion Times in the Case of C-Nav

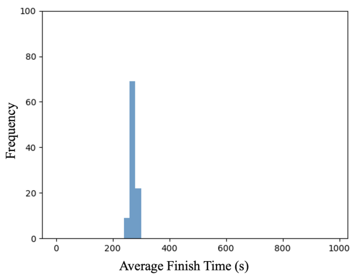

Figure 12. Completion Times in the Case of Q-Virtue

In contrast to the existing C-Nav method, where Figure 9 illustrates numerous instances of agents blocking each other's paths in narrow sections, the Q-Virtue demonstrates movements where groups on the avoiding side align in orderly rows, as shown in Figure 10. Additionally, virtually no time was spent obstructing each other's paths. Consequently, compared with the existing C-Nav (Figure 11), the distribution of movement completion times for the Q-Virtue (Figure 12) is skewed to the left. This shift led to a decreased number of trials in which crossing completion required a long time.

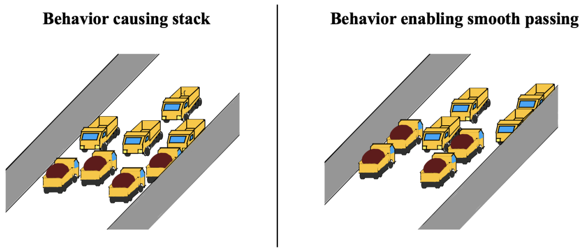

Figure 13. Illustration of the Real-world Situations Associated With the Group Crossing Scenario

5.2.2 Group Crossing Scenario

We focus next on the crossing of two groups consisting of multiple agents in a crowded situation and compare the behaviors resulting from each method handled so far. We verified whether Q-Virtue, which focuses on improving behavior in narrow passages, is also efficient in other situations. These simulated behaviors correspond to the real-world movements observed at construction sites, as shown in Figure 13.

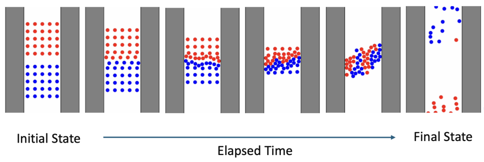

Figure 14. Behavior in the Case of C-Nav

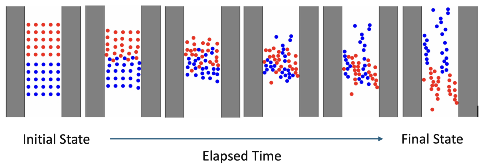

Figure 15. Behavior in the Case of Q-Virtue

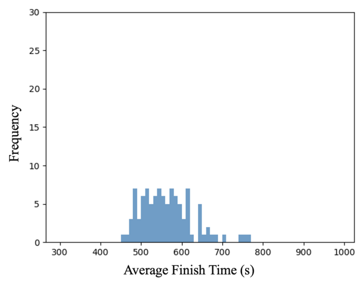

Figure 16. Completion Times in the Case of C-Nav

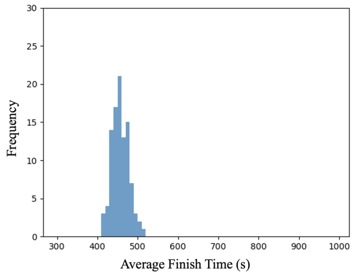

Figure 17. Completion Times in the Case of Q-Virtue

When comparing the existing C-Nav method (Figure 14) with the Q-Virtue (Figure 15), the latter exhibits formations at an earlier stage, which consequently alleviates crowding during crossings. Consequently, the distribution of movement completion times for the Q-Virtue (Figure 17) is skewed to the left compared with the existing C-Nav method (Figure 16). This shift leads to a decreased number of trials in which crossing completion requires a long time.

5.3 Summary

In all situations, the distribution of the average movement completion time shifted to a side with a smaller length because of the Q-Virtue.

6 Conclusion and future work

The existing C-Nav yielded interesting results by designing the algorithm to incorporate information related to the "consideration for others," generating yielding behaviors. However, it was found to be parameter-dependent, especially concerning behaviors during crossings in narrow passages, revealing robustness issues as areas for improvement in practical applications. To elicit the unique performance intended by C-Nav, which is "mutual yielding," we proposed the addition of auxiliary processes and consequently achieved an improvement in robustness. These findings provide new insights into refining velocity obstacle-based collision avoidance algorithms and can help alleviate congestion and facilitate more efficient passage among agents in future automated construction sites.

In this study, we demonstrated that for C-Nav (and for many other collision avoidance algorithms proposed as improvements of ORCA), the action of "waiting behind" is an important factor in achieving efficient movement. However, discussions on methodologies to induce the behavior of "waiting behind" and the conditions under which it becomes effective are still insufficient, and more detailed analyses and improvements are required.

Due to limitations in research duration and equipment, this study did not conduct experiments in actual construction sites. Future work will focus on testing the algorithm in real environments to assess the impact of construction-specific physical and social factors.

Acknowledgement

This study was supported by JSPS KAKENHI Grant Number JP22H01561. We would like to thank Editage (www.editage.jp) for English language editing.

References

[1] M. Lin J. Van Den Berg, S. J. Guy and D. Manocha. Reciprocal n-body collision avoidance. In *Robotics Research: The 14th International Symposium ISRR*, pages 3–19. Springer, 2011. doi:10.1007/978-3-642-19457-3_1.

[2] M. Lin J. Van Den Berg and D. Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In *2008 IEEE international conference on robotics and automation*, pages 1928–1935. Ieee, 2008. doi:10.1109/ROBOT.2008.4543489.

[3] S. J. Guy J. Snape, J. Van Den Berg and D. Manocha. The hybrid reciprocal velocity obstacle. *IEEE Transactions on Robotics*, 27(4):696–706, 2011. doi:10.1109/TRO.2011.2120810.

[4] S. Engel A. Levy, C. Keitel and J. McLurkin. The extended velocity obstacle and applying orca in the real world. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16–22. IEEE, 2015. doi:10.1109/ICRA.2015.7138974.

[5] M. Gini J. Godoy, S. J. Guy and I. Karamouzas. C-nav: Distributed coordination in crowded multi-agent navigation. *Robotics and Autonomous Systems*, 133: 103631, 2020. doi:10.1109/ROBOT.2008.4543489.

[6] V. G. Santos and L. Chaimowicz. Cohesion and segregation in swarm navigation. *Robotica*, 32(2):209–223, 2014. doi:10.1017/S0263574714000563.

[7] W. Wang L. He, J. Pan and D. Manocha. Proxemic group behaviors using reciprocal multi-agent navigation. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 292–297. IEEE, 2016. doi:10.1109/ICRA.2016.7487147.

[8] J. Ma L. Luo, X. Wang and Y.-S. Ong. Gr-pavoid: Multigroup collision-avoidance control and optimization for uav swarm. *IEEE Transactions on Cybernetics*, 53(3):1776–1789, 2021. doi:10.1109/TCYB.2021.3132044.