Lateral Reciprocal Collision Avoidance: A Probabilistic Social-Norm-Inspired Strategy for Deadlock-Free Multi-Robot Navigation

Siyi Lu[1 [0000−0003−4637−844X]] and Sipu Ruan*[1 [0000−0001−5075−6728]]

Beihang University, Beijing, China
{siyilu,ruansp}@buaa.edu.cn

Abstract. Safe and efficient obstacle avoidance in multi-robot navigation is a challenging problem, with deadlock being a key issue due to strict collision-free constraints. This paper proposes a novel Lateral Reciprocal Collision Avoidance (LRCA) strategy based on velocity obstacle theory to mitigate deadlock among multiple robots. Inspired by pedestrian collision avoidance, LRCA incorporates symmetric lateral displacement to resolve deadlock. Unlike Optimal Reciprocal Collision Avoidance (ORCA) algorithm, which computes velocity constraints based on the minimal adjustment across the full velocity obstacle, our proposed method restricts the velocity changes of the agents to one randomly selected side of the relative velocity. This randomized directional selection strategy effectively prevents deadlock while preserving collision avoidance. This approach avoids conflicting velocity changes that could lead to mutual trapping. Theoretical analysis shows how ORCA cause deadlock using quadratic programming, Lagrangian functions, and KKT conditions, and how LRCA effectively prevents this. Extensive simulations across four benchmark multi-robot navigation scenarios show that LRCA outperforms existing algorithms in success rate, time to goal, path length, and computational efficiency.

Keywords: Collision avoidance · Multi-robot navigation · Velocity obstacles.

1 Introduction

Safe and efficient multi-robot navigation is a fundamental problem in robotics and plays a critical role in a wide range of service and industrial applications [1]. In these task, multiple robots operate in a shared environment and are required to reach their respective goals while avoiding collisions with each other robots [2]. Designing navigation strategies that guarantee collision-free behavior while maintaining efficiency remains a challenging problem. Existing approaches to multi-robot navigation can be broadly categorized into centralized and decentralized methods. Centralized methods compute control actions for all robots jointly, which allows them to provide strong guarantees on safety, completeness, and near-optimality [3–5]. However, these methods typically suffer from high

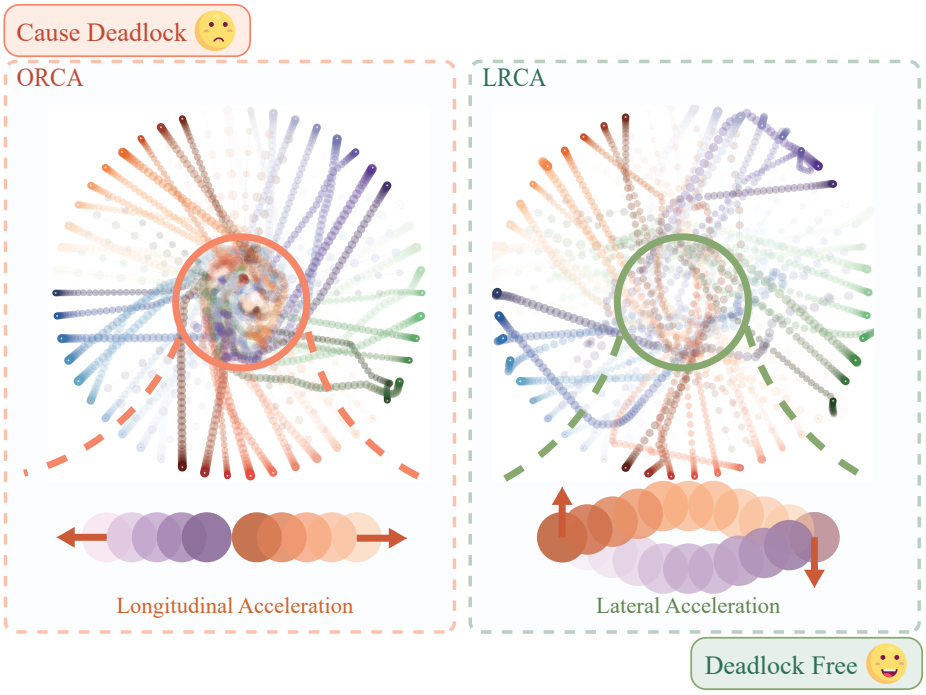

Fig. 1: The left side shows 50 robots navigating with the ORCA, while the right side shows 50 robots navigating with the LRCA. The LRCA method utilizes lateral acceleration, rather than relying solely on longitudinal acceleration as in the ORCA method, which is the primary cause of deadlock.

computational costs and rely on reliable, synchronized communication among all robots. Consequently, they are difficult to scale to large systems and are vulnerable to single-point failures. In contrast, decentralized methods allow each robot to independently determine its control actions based only on locally observable information, such as the positions, velocities, and shapes of neighboring robots [6–10]. These methods are computationally lightweight, more robust to failures, and exhibit better generalization across different environments, making them more suitable for large-scale and real-world multi-robot systems.

Among decentralized approaches, the Velocity Obstacle (VO) [11] framework has become a widely used tool for collision detection, as it enables efficient prediction of potential collisions between pairs of robots within a given time horizon. Building on the VO framework, Reciprocal Collision Avoidance (RCA) [12] methods have been proposed to share collision avoidance responsibility among robots, resulting in efficient and scalable navigation strategies. Despite their practical success, RCA-based methods suffer from a critical limitation: deadlock [13]. Deadlock significantly degrades navigation success rates and efficiency and may occur even in simple scenarios involving only two robots, posing a serious chal-

lenge for decentralized multi-robot navigation. From an optimization perspective, RCA methods can be interpreted as VO-based quadratic programs(QPs). In multi-robot navigation, two objectives naturally coexist: reaching the goal and avoiding collisions. In VO-based QPs, collision avoidance is typically enforced as a hard constraint, while progress towards the goal is treated as the optimization objective. By prioritizing safety over task completion, this formulation guarantees collision-free behavior but may result in deadlock, in which robots remain safe yet make no progress toward their goals. In this paper, we analyze the emergence of deadlock in RCA frameworks from the perspective of VO-based QPs and provide a theoretical characterization of deadlock scenarios using the Karush–Kuhn–Tucker (KKT) conditions.

In contrast to robotic systems, human crowds exhibit remarkably robust collision avoidance behavior and rarely experience deadlock, even in dense environments [14]. Pedestrian avoidance behavior is inherently decentralized, as individuals make decisions based on local observations of each other and shared conventions, which is owe commonly referred to as "social norms". These social norms correspond to an implicit assumption in decentralized robot navigation: *all agents are aware that others follow the same avoidance strategy.* One particularly effective social norm observed in pedestrian dynamics is symmetric lateral displacement, where individuals consistently sidestep in a coordinated manner to avoid collisions [15]. This shared convention plays a crucial role for human crowds to prevent deadlock. Motivated by this observation, we propose to incorporate pedestrian social norms into the RCA framework to mitigate deadlock effects in multi-robot navigation. Specifically, as shown in Fig. 1, we introduce "Lateral Reciprocal Collision Avoidance (LRCA)", a novel RCA-based strategy that integrates symmetric lateral displacement as a shared consensus among robots. By embedding this social norm into the VO-based decision-making process, LRCA significantly reduces the occurrence of deadlock while preserving the decentralized nature of the navigation strategy. Furthermore, we provide a theoretical analysis based on VO-based QPs and KKT conditions to show that LRCA effectively avoids deadlock. The main contributions of this paper are as follows.

- We formalize the RCA framework as a VO-based QPs and provide a theoretical analysis of deadlock scenarios using KKT conditions.
- We propose LRCA, a novel RCA framework inspired by pedestrian lateral avoidance behavior.
- We prove and validate LRCA's ability to avoid deadlock in several scenarios prone to ORCA deadlock.

2 Literature Review

This section reviews prior work related to multi-robot collision avoidance. We first summarize centralized and decentralized collision avoidance methods, with particular emphasis on cooperative decentralized approaches, and then review research on social norms in decentralized collision avoidance.

2.1 Centralized and Decentralized Collision Avoidance

Collision avoidance is essential for collision-free motion planning in multi-robot systems. Approaches are typically categorized as centralized and decentralized. Centralized methods, such as Conflict-Based Search (CBS) [17], plan trajectories using complete system information, including unobservable attributes like goal locations [16]. Variants like explicit estimation CBS (EECBS) [18], lazy constraints addition search for MAPF (LaCAM) [19], and priority inheritance with backtracking (PIBT) [20] improve efficiency and scalability but rely on synchronized communication, limiting scalability in large systems. Decentralized methods use only local states for control, without robot communication. These methods are further divided into non-cooperative and cooperative approaches. Non-cooperative methods, like SARL [23] and T-MPC [24, 25], rely on behavior prediction, while cooperative methods assume shared collision avoidance strategies, improving efficiency and safety [26, 27].

Among cooperative methods, velocity obstacle (VO)-based approaches [11] are widely adopted. The Reciprocal Velocity Obstacle (RVO) [28] extends VO by assuming reciprocal responsibility for collision avoidance, with ORCA [10] defining admissible velocity spaces via linear constraints. While VO-based methods are computationally efficient, they can lead to deadlock in certain scenarios, highlighting the need for further theoretical investigation [13]. Our proposed LRCA algorithm can be categorized as a cooperative method and specifically mitigates the deadlock phenomena inherent in RVO-based methods. While retaining the computational efficiency of VO-based approaches, it achieves safety and navigation performance comparable to, or even exceeding, learning-based methods.

2.2 Social Norms in Decentralized Collision Avoidance

A crowd consists of many individuals coexisting in the same space and time, whose motions are primarily governed by local interactions over extended periods. Pedestrians typically perform operational movements toward tentative destinations that may change over time and are not bound to arrive at specific locations at fixed times [29]. Empirical studies show that pedestrian collision avoidance emerges through early trajectory adjustments made several meters before potential conflict points, reducing interactions and preventing collisions. Since pedestrians do not have access to each other's destinations, crowd navigation can be viewed as a decentralized multi-agent collision avoidance problem.

Despite the absence of centralized coordination, human crowds achieve efficient collision avoidance through a hierarchical structure in which shared social norms guide tactical side selection and operational control handles speed adjustment [30]. In game-theoretic terms, a convention corresponds to one of multiple stable equilibria in a game with two or more equilibria [31], and such conventions are prevalent in pedestrian dynamics, for example, the tendency to walk on a preferred side [15]. These shared conventions enable crowds to function as a cooperative decentralized navigation system. As an intelligent system, human

crowds provide valuable insights for multi-robot navigation. Motivated by the widely observed "keep-right" rule in pedestrian behavior, this paper leverages lateral social conventions to propose a novel lateral reciprocal collision avoidance strategy that addresses deadlock in multi-robot navigation.

3 Problem Definition

In a multi-robot navigation scenario with N robots, each robot is disc-shaped moving in a two-dimensional plane $\mathbb{R}^2$. The robots' attributes are divided into external and internal properties. External properties $\mathcal{O}_e = \{\boldsymbol{p}, \boldsymbol{v}, r\}$ include position $\boldsymbol{p}$, velocity $\boldsymbol{v}$, disc-shape, and radius r, while internal properties $\mathcal{O}_i = \{\boldsymbol{p}^{goal}, \boldsymbol{v}^{pref}\}$ include the goal location $\boldsymbol{p}^{goal}$ and preferred velocity $\boldsymbol{v}^{pref}$ (the robot's speed in the absence of other influential factors). At each time step τ, each robot independently selects a new velocity $\boldsymbol{v}^{new}$ based on the external attributes of other robots, assuming that all robots follow the same cooperative collision avoidance strategy. The robot adjusts its velocity to avoid collisions while steering towards its target location, moving with its preferred velocity.

The velocity obstacle $VO^{\tau}_{i|j}$ is a set of relative velocities $\boldsymbol{v}_{i|j}$ between two robots, i and j, that would result in a collision within the time interval $[0, \tau]$. Let $D(\boldsymbol{p}, r)$ denote an open disc of radius r centered at $\boldsymbol{p}$, i.e.,

$$D(\boldsymbol{p}, r) \doteq \{\boldsymbol{q} \mid \|\boldsymbol{q} - \boldsymbol{p}\| < r\}, \tag{1}$$

then:

$$VO^{\tau}_{i|j} = \left\{\boldsymbol{v}_{i|j} \mid \exists t \in [0, \tau] :: t\boldsymbol{v}_{i|j} \in D\left(\boldsymbol{p}_j - \boldsymbol{p}_i, r_i + r_j\right)\right\}. \tag{2}$$

In cooperative reciprocal collision avoidance [10], the set of permitted velocities V_i for robot i and V_j for robots j are determined based on their velocity obstacles $VO^{\tau}_{i|j}$:

$$V_i \cap (VO^{\tau}_{i|j} \oplus V_j) = \emptyset, \tag{3}$$

where $\oplus$ represents Minkowski sum. The union of the admissible velocity sets for robot i with respect to all other robots defines the set of velocities that allow robot i to avoid collisions in the next time step.

4 Lateral Reciprocal Collision Avoidance

In the proposed framework, we aim to define sets of permitted velocities V_i for robot i and V_j for robot j, such that these sets ensure reciprocal collision avoidance within a given time step τ. The goal is to maximize the proximity of the admissible velocities to the robots' optimal velocities, while maintaining collision-free motion. To determine the sets of permitted velocities for both robots, we first calculate their relative velocity $\boldsymbol{v}_{i|j}$. Then we compute the smallest adjustment vector $\boldsymbol{u}$ needed to avoid or prevent a collision. This adjustment vector is determined as the vector from the relative velocity $\boldsymbol{v}_{i|j}$ difference to the one point on the boundary of the velocity obstacle. The classic ORCA [10] selects the point on the velocity obstacle $VO^{\tau}_{i|j}$ boundary that is closest in terms of relative velocity $\boldsymbol{v}_{i|j}$, but it can lead to severe deadlocks.

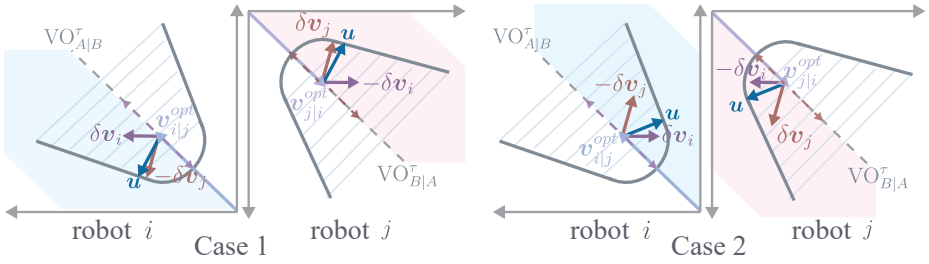

Fig. 2: Two cases ensuring the velocity adjustment difference is not parallel to the relative velocity: Case 1: Both robots' adjustments are counterclockwise; Case 2: Both are clockwise.

4.1 Deadlock-Free Collision Avoidance

Inspired by pedestrian collision avoidance behavior, we constrain the relative velocity adjustment $\delta \boldsymbol{v}$ to avoid being parallel to the current relative velocity, thereby preventing deadlock. Let the updated velocities of robots i and j be

$$\boldsymbol{v}_i^{\text{new}} = \boldsymbol{v}_i^{\text{opt}} + \delta \boldsymbol{v}_i, \qquad \boldsymbol{v}_j^{\text{new}} = \boldsymbol{v}_j^{\text{opt}} + \delta \boldsymbol{v}_j. \tag{4}$$

The new relative velocity on next timestep is given by

$$\boldsymbol{v}_{i|j}^{\text{new}} = \boldsymbol{v}_i^{\text{new}} - \boldsymbol{v}_j^{\text{new}} = \boldsymbol{v}_{i|j}^{\text{opt}} + (\delta \boldsymbol{v}_i - \delta \boldsymbol{v}_j)\,. \tag{5}$$

We impose the following lateral constraint on the relative velocity adjustment:

$$\forall\, k \in \mathbb{R}, \quad \delta \boldsymbol{v}_i - \delta \boldsymbol{v}_j \neq k\, \boldsymbol{v}_{i|j}^{\text{opt}}. \tag{6}$$

Under this constraint, the adjustment term cannot exactly cancel the current relative velocity. Therefore, when $\boldsymbol{v}_{i|j}^{\text{opt}} \neq \mathbf{0}$, the updated relative velocity satisfies

$$\boldsymbol{v}_{i|j}^{\text{new}} \neq \mathbf{0}. \tag{7}$$

A more rigorous and complete derivation process is in Sec. 5.2. This lateral adjustment avoids collision resolution that relies solely on accelerating or decelerating along the relative velocity direction, which is prone to inducing deadlock. To prevent the difference in velocity adjustments between two robots $\delta \boldsymbol{v}_i - \delta \boldsymbol{v}_j$ from being parallel to the relative velocity $\boldsymbol{v}_i^{opt} - \boldsymbol{v}_j^{opt}$, two cases are considered: both robots' velocity adjustments can be in the clockwise direction of the relative velocity, or both can be in the counterclockwise direction, as shown in Fig. 2. Note that for robot j, the relative velocity with respect to robot i is calculated as $\boldsymbol{v}_j^{opt} - \boldsymbol{v}_i^{opt}$.

Since LRCA introduces constraints on the velocity adjustments, requiring them to be either in the clockwise or counterclockwise direction of the relative velocity, continuing to select the minimum velocity adjustment to avoid collisions would shrink the feasible velocity set, potentially even resulting in an empty set

with no valid velocities. This could lead to another form of deadlock. To address this, we select the velocity adjustment with the largest angle relative to the relative velocity, which is the velocity perpendicular to the linear boundary of the $VO^{\tau}_{A|B}$, thereby maximizing the feasible velocity set while ensuring collision avoidance. As shown in Fig. 2, the velocity adjustment can either be perpendicular to the left linear boundary of the $VO^{\tau}_{i|j}$ or perpendicular to the right linear boundary, corresponding to two strategies for avoiding deadlock. Let $\boldsymbol{n}$ be the outward normal of the boundary of $VO^{\tau}_{i|j}$ at point $\boldsymbol{v}^{opt}_{i|j} + \boldsymbol{u}$, the relative velocity adjustment $\boldsymbol{u}$ can be formally expressed as:

$$\boldsymbol{u} = \left(\boldsymbol{n}^{\top}\left(\boldsymbol{v}_i^{\text{opt}} - \boldsymbol{v}_j^{\text{opt}}\right)\right)\boldsymbol{n}, \tag{8}$$

where

$$\begin{aligned} \boldsymbol{n} &= \begin{cases} \mathbf{R}\left(\frac{\pi}{2} + \theta\right)\boldsymbol{e}_{ij}, & \text{(Case 1)}, \\ \mathbf{R}\left(-\frac{\pi}{2} - \theta\right)\boldsymbol{e}_{ij}, & \text{(Case 2)}, \end{cases} \\ \theta &= \arcsin\left(\frac{r_i + r_j}{\|\boldsymbol{p}_j - \boldsymbol{p}_i\|_2}\right), \\ \boldsymbol{e}_{ij} &= \frac{\boldsymbol{p}_j - \boldsymbol{p}_i}{\|\boldsymbol{p}_j - \boldsymbol{p}_i\|_2}, \end{aligned} \tag{9}$$

and $\mathbf{R}\left(\frac{\pi}{2} + \theta\right)$ and $\mathbf{R}\left(-\frac{\pi}{2} - \theta\right)$ are rotation matrices that rotates $\boldsymbol{e}_{ij}$ counterclockwise and clockwise by $\frac{\pi}{2} + \theta$ degrees respectively. Then, the set $\text{LRCA}^{\tau}_{i|j}$ of permitted velocities for robot i can be formally expressed as:

$$\text{LRCA}^{\tau}_{i|j}(k) = \left\{ \boldsymbol{v} \;\middle|\; \begin{array}{l} \left(\boldsymbol{v} - \left(\boldsymbol{v}_i^{\text{opt}} + \frac{1}{2}\boldsymbol{u}\right)\right) \cdot \boldsymbol{n} \geq 0, \\ \text{sign}\left(\left(\boldsymbol{v} - \boldsymbol{v}_i^{\text{opt}}\right) \times \boldsymbol{v}_{i|j}^{\text{opt}}\right) = s_k \end{array} \right\}, \quad k = 1, 2, \tag{10}$$

where $s_1 = 1$ (corresponding to Case 1), $s_2 = -1$ (corresponding to Case 2). The specific case is chosen by the randomized strategy in Sec. 4.2.

4.2 Game-Theoretic Randomization for Distributed Collision Avoidance

To enable robots to reach consensus without explicit communication, selecting either Case 1 or Case 2 as shown in Fig. 2, we introduce a randomized mechanism inspired by human game-theoretic interactions. Each robot i randomly selects whether the velocity adjustment $\delta\boldsymbol{v}_i$ lies on the clockwise or counterclockwise side of the relative velocity with respect to robot j. If both robots i and robot j choose the same side, collision avoidance is achieved without deadlock; otherwise, avoidance is not possible. Alg. 1 outlines an implicit coordination process that enables the robots to reach consensus without communication.

In Algorithm 1, lines 1–2 iterate over all robot pairs (i, j) for distributed pairwise collision checking. Line 3 constructs the velocity obstacle $VO^{\tau}_{i|j}$ and

Algorithm 1: Algorithm process of LRCA

Input: Number of robots n, positions $\mathbb{P}$, velocities $\mathbb{V}^{\text{opt}}$, attributes $\mathbb{A}$
Output: New velocities $\mathbb{V}^{\text{new}}$ for next time step

```
1  for i = 1 to n do
2      for j = 1 to n, j ≠ i do
3          Compute velocity obstacle VO^τ_{i|j} and relative velocity v^opt_{i|j};
4          if v^opt_{i|j} ∈ VO^τ_{i|j} then
5              Calculate velocity changes δv_i and δv_j;
6              if (v^opt_{i|j} × δv_i) · (v^opt_{i|j} × δv_j) < 0 then
7                  if v^opt_{i|j} × δv_i > 0 then
8                      Add Case 1 constraint;
9                  else
10                     Add Case 2 constraint;
11                 end
12             else
13                 Randomly add Case 1 or Case 2 constraint;
14             end
15         end
16     end
17 end
```

computes the nominal relative velocity $\boldsymbol{v}_{i|j}^{\text{opt}}$. Line 4 detects potential collisions by checking whether $\boldsymbol{v}_{i|j}^{\text{opt}}$ lies inside $VO_{i|j}^{\tau}$. Line 5 computes candidate velocity adjustments $\delta\boldsymbol{v}_i$ and $\delta\boldsymbol{v}_j$. Line 6 determines whether the two robots selected the same lateral side using the sign of the cross product. Lines 7–11 add the corresponding deterministic lateral constraint (Case 1 or Case 2) when a consistent side is detected. Line 13 randomly selects between Case 1 and Case 2 according to Eq. (12) when the selections are inconsistent. Let the random variable C denote the selected category, where $C = 1$ corresponds to Case 1 and $C = 2$ corresponds to Case 2. Define the angles:

$$\theta_1 = \angle\left(\boldsymbol{n}_1, -\boldsymbol{v}_{i|j}^{opt}\right), \quad \theta_2 = \angle\left(\boldsymbol{n}_2, -\boldsymbol{v}_{i|j}^{opt}\right), \tag{11}$$

where $\boldsymbol{n}_1 = \mathbf{R}\left(\frac{\pi}{2} + \theta\right)\boldsymbol{e}_{ij}$ and $\boldsymbol{n}_2 = \mathbf{R}\left(-\frac{\pi}{2} - \theta\right)\boldsymbol{e}_{ij}$, $\angle(\boldsymbol{a}, \boldsymbol{b})$ represents the angle between vector $\boldsymbol{a}$ and vector $\boldsymbol{b}$. Then C follows a Categorical distribution with parameters $p_1 = \frac{\theta_2}{\theta_1+\theta_2}$ and $p_2 = \frac{\theta_2}{\theta_1+\theta_2}$, i.e.:

$$C \sim \text{Categorical}\left(p_1, p_2\right). \tag{12}$$

As the relative velocity $\boldsymbol{v}_{i|j}^{opt}$ approaches the left boundary of the $VO_{i|j}^{\tau}$, θ_1 and p_1 increase. Conversely, as the relative velocity $\boldsymbol{v}_{i|j}^{opt}$ approaches the right boundary of the $VO_{i|j}^{\tau}$, θ_2 and p_2 increase.

5 Analysis of the Two-Robot Deadlock

In Sec. 4, we introduced the LRCA method from a set-theoretic perspective. In this section, we reformulate LRCA and ORCA using quadratic programming and present their respective Lagrangian functions and KKT conditions, applying them to a two-robot example. This analysis reveals the underlying characteristics of deadlock in ORCA and demonstrates how LRCA effectively prevents deadlock. These findings can be extended to scenarios involving N robots. The velocity constraints for each robot in the ORCA and LRCA algorithms are given by Eq. (13) and the combination of Eq. (13) and Eq. (14), respectively.

$$\boldsymbol{n}^\top \boldsymbol{v}_i^{new} \geq \boldsymbol{n}^\top \boldsymbol{v}_i^{opt} + \frac{\|\boldsymbol{u}\|_2}{2}, \tag{13}$$

$$\boldsymbol{h}^\top \boldsymbol{v}_i^{new} \leq \boldsymbol{h}^\top \boldsymbol{v}_i^{opt}, \tag{14}$$

where $\boldsymbol{h} = \angle(\boldsymbol{v}_{i|j}^{opt}, \frac{\pi}{2})$ or $\boldsymbol{h} = \angle(\boldsymbol{v}_{i|j}^{opt}, -\frac{\pi}{2})$ according to the randomized decision strategy described in Sec. 4.2. Therefore, any $\boldsymbol{v}_i^{new}$ generated by ORCA or LRCA that satisfy corresponding constraint are guaranteed to ensure collision-free and deadlock-free trajectories for robots i and j in the multi-robot system. Since robot i wants to avoid collisions with $N-1$ robots, there are $N-1$ collision avoidance constraints. To mediate between safety and goal stabilization objective, a QP is posed that computes a controller closest to the target velocity $\hat{\boldsymbol{v}}_i(p_i)$ and satisfies the $N-1$ constraints, as shown:

$$\boxed{\begin{array}{ll} \underset{\boldsymbol{v}_i^{\text{new}}}{\text{minimize}} & \|\boldsymbol{v}_i^{\text{new}} - \hat{\boldsymbol{v}}_i(\boldsymbol{p}_i)\|_2^2 \\ \text{subject to} & \boldsymbol{H}_{ij}\boldsymbol{v}_i^{\text{new}} \leq \boldsymbol{k}_{ij}, \quad j \in \mathcal{N}_i, \end{array}} \tag{15}$$

where $\mathcal{N}_i = [\mathbb{N}] \setminus \{i\}$ denotes the set of all robots except robot i. For ORCA, each neighbor j contributes a single linear constraint, given by

$$\boldsymbol{H}_{ij} = \boldsymbol{a}_{ij}^\top \in \mathbb{R}^{1\times 2}, \quad \boldsymbol{k}_{ij} = b_{ij} \in \mathbb{R}, \tag{16}$$

where $\boldsymbol{a}_{ij} := -\boldsymbol{n}$ and $b_{ij} := -\boldsymbol{n}^\top \boldsymbol{v}_i^{\text{opt}} - \frac{\|\boldsymbol{u}\|_2}{2}$, as derived from Eq. (13). For LRCA, two linear constraints are imposed for each neighbor, leading to

$$\boldsymbol{H}_{ij} = \begin{bmatrix} \boldsymbol{a}_{ij}^\top \\ \boldsymbol{c}_{ij}^\top \end{bmatrix} \in \mathbb{R}^{2\times 2}, \quad \boldsymbol{k}_{ij} = \begin{bmatrix} b_{ij} \\ d_{ij} \end{bmatrix} \in \mathbb{R}^2, \tag{17}$$

where $\boldsymbol{c}_{ij} := \boldsymbol{h}^\top$ and $d_{ij} := \boldsymbol{h}^\top \boldsymbol{v}_i^{\text{opt}}$ follow from Eq. (14). To circumvent Eq. (15), we use duality theory, specifically KKT conditions of to derive an explicit expression for $\boldsymbol{v}_i^{new*}$ as a function of the robots' positions. These conditions are necessary and sufficient for a global optimum of this QP. The Lagrangian is:

$$\mathcal{L}(\boldsymbol{v}_i^{\text{new}}, \boldsymbol{\Lambda}) = \|\boldsymbol{v}_i^{\text{new}} - \hat{\boldsymbol{v}}_i\|^2 + \sum_{j\in\mathcal{N}_i} \boldsymbol{\lambda}_{ij}^\top \left(\boldsymbol{H}_{ij}\boldsymbol{v}_i^{\text{new}} - \boldsymbol{k}_{ij}\right), \tag{18}$$

where $\mathcal{N}_i = [\mathbb{N}] \setminus \{i\}$, and $\boldsymbol{\lambda}$ denotes the Lagrange multipliers. For ORCA $\boldsymbol{\lambda}_{ij} = \lambda_{ij}^{(a)} \in \mathbb{R}$, and for LRCA $\boldsymbol{\lambda}_{ij} = \begin{bmatrix} \lambda_{ij}^{(a)} \\ \lambda_{ij}^{(c)} \end{bmatrix} \in \mathbb{R}^2$. Let $(\boldsymbol{v}_i^{\text{new}*}, \boldsymbol{\Lambda}^*)$ be the optimal primal-dual solutions to Eq. (15), the KKT conditions are:

1. **Stationarity:**

$$\nabla_{\boldsymbol{v}_i^{\text{new}}} \mathcal{L}(\boldsymbol{v}_i^{\text{new}}, \boldsymbol{\Lambda})\big|_{(\boldsymbol{v}_i^{\text{new}*})} = \mathbf{0}$$

$$\implies \quad \boldsymbol{v}_i^{\text{new}*} = \hat{\boldsymbol{v}}_i - \frac{1}{2} \sum_{j \in \mathcal{N}_i} \boldsymbol{\lambda}_{ij}^\top \boldsymbol{H}_{ij}. \tag{19}$$

2. **Primal Feasibility:**

$$\boldsymbol{H}_{ij}^\top \boldsymbol{v}_i^{\text{new}*} \leq \boldsymbol{k}_{ij}, \quad \forall j \in \mathcal{N}_i. \tag{20}$$

3. **Dual Feasibility:**

$$\boldsymbol{\lambda}_{ij}^* \geq 0, \quad \forall j \in \mathcal{N}_i. \tag{21}$$

4. **Complementary Slackness:**

$$\boldsymbol{\lambda}_{ij}^* \left(\boldsymbol{H}_{ij} \boldsymbol{v}_i^{\text{new}*} - \boldsymbol{k}_{ij} \right) = \mathbf{0}, \quad \forall j \in \mathcal{N}_i. \tag{22}$$

Definition 1. *Robot i is in deadlock if $\boldsymbol{v}_i{}^* = 0$ and the prescribed nominal control $\hat{\boldsymbol{v}}_i \neq \mathbf{0} \iff \boldsymbol{p}_i \neq \boldsymbol{p}_{d_i}$ and the speed remains at 0.*

A robot is in deadlock if it receives zero velocity commands from the QP controller indefinitely without having reached its goal. Importantly, deadlock is a system-level phenomenon and cannot occur for a single robot in isolation.

5.1 Deadlock Conditions in Two-Robot ORCA Interactions

According to the deadlock definition, establishing deadlock in ORCA reduces to identifying a two-robot configuration in which, despite nonzero target velocities, ORCA outputs zero velocities at two consecutive time steps. Since ORCA is memoryless and depends only on the current state, zero velocity at two consecutive steps implies that the robot velocity converges to zero thereafter. Assume that a deadlock has occurred. Then there exist two consecutive time steps t_1 and t_2 such that

$$\boldsymbol{v}_i^{t_1} = \boldsymbol{v}_i^{t_2} = \mathbf{0}, \qquad \hat{\boldsymbol{v}}_i^{opt} = \hat{\boldsymbol{v}}_i^{t_1} \neq \mathbf{0}.$$

The equality of the target velocities follows from the fact that the target velocity depends only on the robot's current position and goal position; since $\boldsymbol{v}_i^{t_1} = 0$, the robot position does not change between t_1 and t_2.

We consider a two-robot interaction in which robot i is constrained only by robot j. Hence, at the optimum of the ORCA quadratic program, at most one collision-avoidance constraint is active. From the stationarity condition of the KKT optimality conditions, the ORCA solution satisfies

$$\boldsymbol{v}_i^* = \hat{\boldsymbol{v}}_i - \frac{1}{2} \lambda_{ij}^* \boldsymbol{a}_{ij}.$$

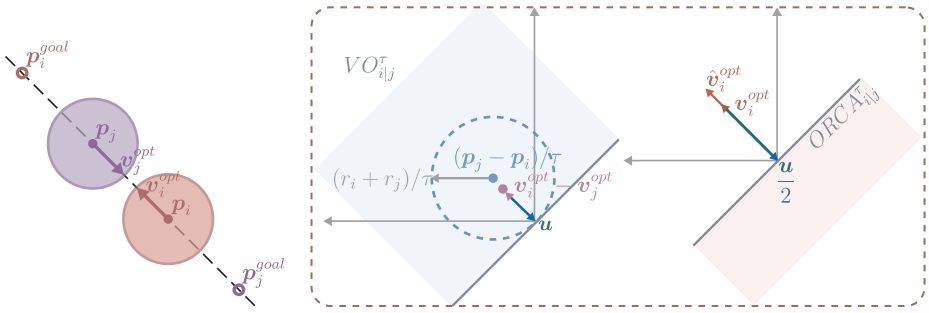

Fig. 3: An example of deadlock in multi-robot navigation with two robots using the ORCA algorithm, along with the ORCA algorithm's computation process.

Substituting $\boldsymbol{v}_i^{t_1} = \mathbf{0}$ yields

$$\hat{\boldsymbol{v}}_i^{opt} = \frac{1}{2}\lambda_{ij}^{opt*}\boldsymbol{a}_{ij}^{opt}, \tag{23}$$

and by the same reasoning at time t_2,

$$\hat{\boldsymbol{v}}_i^{t_1} = \frac{1}{2}\lambda_{ij}^{t_1*}\boldsymbol{a}_{ij}^{t_1}. \tag{24}$$

When the relative velocity satisfies $\boldsymbol{v}_{i|j}^{t_1} = \mathbf{0}$, the ORCA formulation implies that the normal vector $\boldsymbol{n}_{ij}^{t_1}$ is aligned with the relative position vector,

$$\exists\, \alpha > 0 \text{ s.t. } \boldsymbol{n}_{ij}^{t_1} = \alpha(\boldsymbol{p}_i - \boldsymbol{p}_j).$$

Since $\boldsymbol{a}_{ij}^{t_1} = -\boldsymbol{n}_{ij}^{t_1}$ and $\lambda_{ij}^{t_1*} \geq 0$, Eq. (24) implies that the target velocity $\hat{\boldsymbol{v}}_i^{t_1}$ is aligned with relative position vector $\boldsymbol{p}_j - \boldsymbol{p}_i$. Since $\hat{\boldsymbol{v}}_i^{t_1} = \hat{\boldsymbol{v}}_i^{opt}$, combining the above result with Eq. (23) implies that the constraint normal $\boldsymbol{n}_{ij}^{opt}$ is also aligned with the relative position vector $\boldsymbol{p}_j - \boldsymbol{p}_i$. According to the ORCA construction, alignment of the constraint normal with the relative position further implies that the relative velocity $\boldsymbol{v}_{i|j}^{opt}$ is colinear with the same direction. Hence

$$\exists\, \alpha > 0 \text{ s.t. } \boldsymbol{v}_{i|j}^{opt} = \alpha(\boldsymbol{p}_j - \boldsymbol{p}_i).$$

The resulting geometric configuration is illustrated in Fig. 3. Deadlock under ORCA occurs when the relative velocity and relative position of the two robots are parallel, while the target velocities of both robots point toward each other.

5.2 Proving Deadlock-Free Behavior in Two-Robot LRCA

Theorem 1. *Under the constraints of the random obstacle avoidance strategy of LRCA, the robot will not encounter deadlocks.*

Proof. Consider a pair of robots i and j. At a certain moment, if the velocities $\boldsymbol{v}_i^{new}$ and $\boldsymbol{v}_j^{new}$ calculated by the random obstacle avoidance strategy are both 0, then next time step, their relative velocity $\boldsymbol{v}_{i|j}^{opt}$at that moment is also 0 and $(\boldsymbol{v}_{i|j}^{opt} \times \delta \boldsymbol{v}_i) \cdot (\boldsymbol{v}_{i|j}^{opt} \times \delta \boldsymbol{v}_j) = 0$. According to the second type of constraint, the constraint on robot i as:

$$\boldsymbol{n}^\top \boldsymbol{v}_i^{new} \geq 0. \tag{25}$$

According to Lagrangian and KKT conditions, $\boldsymbol{v}_i^{new}$ can be calculated:

$$\boldsymbol{v}_i^{new*} = \hat{\boldsymbol{v}}_i + \frac{1}{2}\lambda_{ij}^* \boldsymbol{n}. \tag{26}$$

According to Eq. (12), there is a one-half probability that $\boldsymbol{n} = \mathbf{R}\left(\frac{\pi}{2} + \theta\right)\boldsymbol{e}_{ij}$, and a one-half probability that $\boldsymbol{n} = \mathbf{R}\left(-\frac{\pi}{2} - \theta\right)\boldsymbol{e}_{ij}$. if $\boldsymbol{n} = \mathbf{R}\left(\frac{\pi}{2} + \theta\right)\boldsymbol{e}_{ij}$, then $\boldsymbol{v}_i^{new} = 0$, then when $\boldsymbol{n} = \mathbf{R}\left(-\frac{\pi}{2} - \theta\right)\boldsymbol{e}_{ij}$, $\boldsymbol{v}_i^{new}$ definitely is not 0. $\boldsymbol{v}_i^{new}$ will be zero in both cases if and only if $\hat{\boldsymbol{v}}_i = 0$. Therefore, under the random obstacle avoidance algorithm, the robot will not get stuck.

6 Simulation Experiments

This section details the experimental setup, showcases representative deadlock scenarios to validate LRCA, and evaluates its performance through both quantitative metrics and qualitative case studies across diverse navigation scenarios.

6.1 Experiments Setup

Implemented as an ORCA extension within the RVO library[1], LRCA sets a maximum robot speed of 1m/s and a control step of 0.1s. Two prediction horizons are used: a short horizon (0.3s) for ORCA-based avoidance and a long horizon (1.5s) for LRCA. Robots' radius are uniformly sampled from $[0.1, 0.15]$ m m. Experiments cover three benchmark scenarios: Cross, Random, and Corridor. In the Cross Scenario, robots move along circle diameters and cross at the center, creating a highly challenging navigation environment. In the Random Scenario, start and goal positions are uniformly sampled in a rectangle. In the Corridor Scenario, robots traverse a narrow passage from opposite ends, simulating constrained navigation. Performance is assessed by success rate, extra distance, extra time, and average speed. Success rate measures robots reaching goals collision-free within a time limit. Extra distance and time measure deviations from the ideal shortest-path distance and minimal travel time, respectively. Average speed is the mean navigation speed. LRCA is compared against four representative circular-robot collision avoidance methods: multi-robot algorithms ORCA [10] and RL-RVO [22], and social navigation algorithms T-MPC [24, 25] and SARL [23]. Although originally designed for ORCA-simulated pedestrians, the latter two are reasonably adapted to multi-robot settings and thus included for comparison.

[1] https://github.com/sybrenstuvel/Python-RVO2

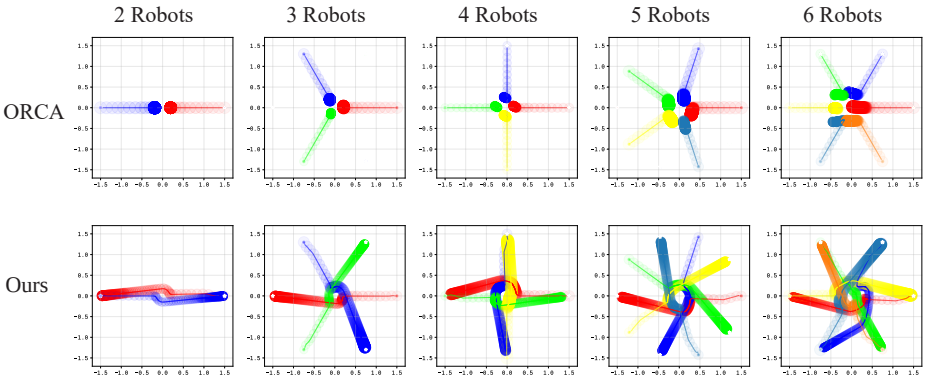

Fig. 4: Comparison of the trajectories of ORCA and Ours in five scenarios prone to deadlock.

6.2 Analysis of Deadlock Cases under Cross Scenarios

In center-symmetrical cross scenarios, the ORCA algorithm is prone to deadlock, as illustrated in Fig. 4. In this symmetric configuration, each robot moves along the straight line connecting its start and goal positions. To avoid collision, ORCA enforces symmetric deceleration, causing both robots to gradually slow down and eventually stop at the safety boundary with zero relative velocity. As a result, the robots fail to reach their goals despite having nonzero target velocities. In contrast, the proposed LRCA strategy resolves this issue by explicitly breaking symmetry through randomized lateral avoidance. As shown in Fig. 4, robots under LRCA perform smooth lateral maneuvers and pass each other without stopping. By restricting velocity adjustments to a randomly selected side of the relative velocity, LRCA induces early asymmetric behavior, thereby preventing convergence to the zero-velocity equilibrium responsible for deadlock. These results indicate that in highly symmetric cross scenarios, obstacle avoidance strategies based on minimal deviation, such as ORCA, are inherently susceptible to system deadlock. Incorporating a social-norm-inspired lateral avoidance mechanism, as in LRCA, effectively eliminates this failure mode.

6.3 Quantitative Analysis of Navigation Performance under Cross Scenarios

In the cross scenario, LRCA consistently outperforms other methods across all metrics. The success rate for LRCA nearly 100% for all group sizes, while ORCA sees a significant decline, dropping from 100% to 70.30% as the number of robots increases. T-MPC and SARL also struggle in larger groups, with SARL's success rate decreasing to 36.4% for 50 robots, and T-MPC's success rate dropping to 0 when the number of robots exceeds 30. For extra distance, LRCA shows the best performance, with values ranging from 0.26 meters for 4 robots to 1.37 meters for 50 robots. ORCA and T-MPC have higher extra distances, especially with

Table 1: Performance metrics (as mean) evaluated for different methods on the cross scenarios with varied scene sizes and different number of robots.

Metrics	Method	#agents (radius of the scene)							
		4 (2.0 m)	6 (2.0 m)	8 (2.5 m)	10 (2.5 m)	20 (3.0 m)	30 (4.0 m)	40 (5.0 m)	50 (6.0 m)
Success Rate(%)	ORCA	**100.00**	**100.00**	**100.00**	**100.00**	79.35	77.27	67.33	70.30
	T-MPC	43.00	16.67	18.09	10.00	7.50	-	-	-
	SARL	**100.00**	**100.00**	**100.00**	99.80	94.00	30.63	69.50	36.40
	RL-RVO	**100.00**	99.83	96.50	99.70	95.85	93.97	91.18	86.67
	Ours	**100.00**	**100.00**	**100.00**	99.60	**97.80**	**97.00**	**95.05**	**95.30**
Extra Distance(m)	ORCA	4.35	0.40	0.66	1.09	1.25	1.46	1.77	2.83
	T-MPC	5.08	4.34	5.07	5.06	4.88	-	-	-
	SARL	15.63	15.77	13.90	13.70	19.95	19.81	6.29	18.25
	RL-RVO	1.59	1.62	1.74	1.97	3.15	3.60	4.04	4.57
	Ours	**0.26**	**0.30**	**0.36**	**0.45**	**0.74**	**1.05**	**1.35**	**1.37**
Extra Time(s)	ORCA	78.00	8.30	6.41	6.83	5.45	6.91	8.07	12.04
	T-MPC	4.98	4.24	4.97	4.96	4.48	-	-	-
	SARL	18.45	18.54	18.39	18.56	26.98	25.35	10.22	26.03
	RL-RVO	6.26	5.80	6.33	7.23	12.00	14.04	15.99	19.22
	Ours	**2.91**	**3.19**	**3.36**	**3.59**	**4.38**	**5.32**	**6.20**	**6.27**
Average Speed(m/s)	ORCA	0.18	0.59	0.65	0.56	0.62	0.62	0.56	0.49
	T-MPC	0.59	0.61	0.63	0.59	0.70	-	-	-
	SARL	**0.85**	**0.86**	**0.83**	**0.80**	**0.79**	**0.83**	**0.80**	**0.79**
	RL-RVO	0.37	0.38	0.27	0.29	0.17	0.20	0.24	0.30
	Ours	0.80	0.78	0.75	0.73	0.66	0.62	0.58	0.58

more robots. Regarding extra time, LRCA again leads, with values ranging from 2.91 seconds for 4 robots to 6.27 seconds for 50 robots. ORCA and RL-RVO experience higher extra times as the robot count increases, while SARL reaches the highest extra time of 26.98 seconds for 20 robots.

Finally, in terms of average speed, LRCA maintains high speeds, from 0.8 m/s for 4 robots to 0.58 m/s for 50 robots. In contrast, ORCA exhibits significantly lower navigation speeds across all scenarios due to the frequent occurrence of deadlock. RL-RVO shows unstable navigation speeds, which can be attributed to the limited interpretability of its learned policy. Although SARL achieves the highest navigation speed in all scenarios, this excessive speed leads to reduced navigation safety and efficiency when other performance metrics are considered. Overall, LRCA demonstrates superior performance in cross scenarios, consistently maintaining high success rates, low distance and time deviations, and robust navigation behavior even as the number of robots increases.

6.4 Qualitative Analysis of Navigation Trajectories

Fig. 5 presents qualitative comparisons of navigation trajectories generated by ORCA, T-MPC, SARL, RL-RVO, and the proposed LRCA in three representative scenarios. Each colored curve denotes the trajectory of an individual robot from its start to its goal. In the cross scenario (first row), ORCA exhibits strong symmetry and congestion near the intersection, leading to inefficient interactions. T-MPC alleviates conflicts but produces long and curved detours. SARL results in irregular and sometimes unstable trajectories, while RL-RVO tends

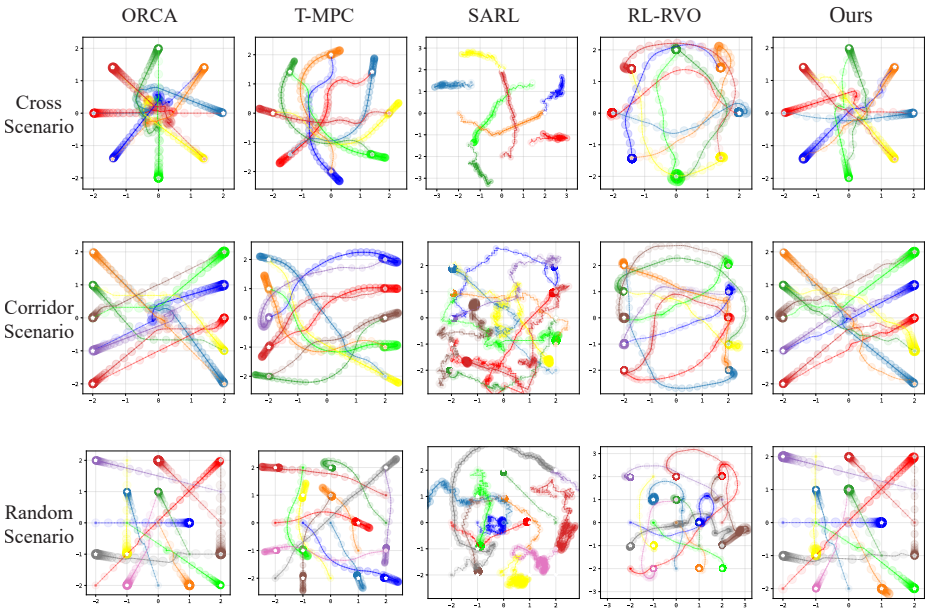

Fig. 5: Navigation trajectories of five methods (ORCA, T-MPC, SARL, RL-RVO, Ours) in cross scenario, corridor scenario, and random scenario.

to generate overly conservative circular motions. In contrast, LRCA produces smooth and well-structured trajectories by introducing consistent lateral avoidance, enabling robots to pass through the intersection efficiently.

In the corridor scenario (second row), ORCA frequently suffers from reciprocal blocking in the narrow passage. Learning-based methods reduce collisions but show oscillatory or overly cautious behaviors. LRCA maintains clear directional separation, allowing robots to traverse the corridor smoothly with minimal interference. In the random scenario (third row), ORCA remains prone to mutual blocking under dense interactions, and SARL and RL-RVO generate scattered or unnecessarily long paths. LRCA achieves compact and well-separated trajectories, demonstrating robust collision avoidance without sacrificing efficiency. Overall, the qualitative results indicate that LRCA effectively breaks symmetric interactions through lateral relative-velocity adjustments, leading to smoother, more efficient, and more reliable navigation across diverse scenarios.

6.5 Computational Efficiency Analysis

Fig. 6 shows the computation time (mean/standard deviation in milliseconds) for different methods with varying numbers of agents. ORCA is the most computationally efficient method, with times increasing from 1.69×10^{-5}ms for 2 agents to 1.76×10^{-3}ms for 60 agents. T-MPC has a higher computational cost, increasing from 7.01×10^{-3}ms to 1.55×10^{-1}ms as the number of agents grows.

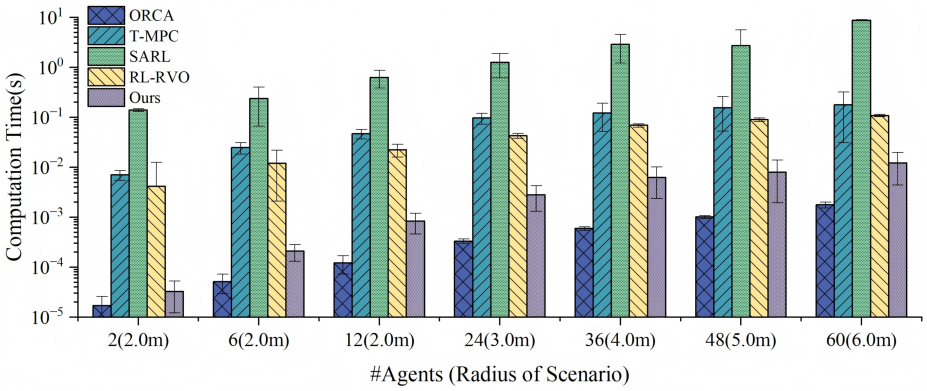

Fig. 6: Computation time (as mean/SD in milliseconds) for different methods with varied number of agents.

SARL exhibits the highest computational cost, rising from 1.39×10^{-1}ms for 2 agents to 8.75 seconds for 60 agents. RL-RVO has moderate computation times, ranging from 3.72×10^{-3}ms to 1.21×10^{-1}ms. Our method, LRCA, provides a good balance, with times increasing from 3.26×10^{-5}ms to 1.21×10^{-2}ms, making it efficient yet slightly slower than ORCA. Overall, LRCA substantially improves navigation safety and efficiency while preserving a level of computational lightness and real-time performance comparable to that of ORCA.

7 Conclusions

This paper proposed the Lateral Reciprocal Collision Avoidance (LRCA) algorithm for multi-robot navigation. Inspired by pedestrian behaviors, LRCA introduces lateral acceleration and randomized lateral velocity selection to achieve deadlock-free collision avoidance. By reformulating ORCA as a quadratic program and analyzing its KKT conditions, we identified deadlock scenarios in two-robot interactions and showed that LRCA effectively avoids such deadlocks. Experimental results demonstrate that LRCA outperforms ORCA and other baseline methods in deadlock-prone scenarios, achieving improved navigability, efficiency, and safety across diverse navigation tasks. A limitation of the current work is that the modeling and algorithm are restricted to two-dimensional space. Future work will extend LRCA to three-dimensional scenarios to enable deadlock-free reciprocal collision avoidance in 3D environments.

Acknowledgments. This work was supported by National Key R&D Program of China (Grant No. 2025YFF0521600) and the National Natural Science Foundation of China (No. 52505001).

References

1. Wang, G., Zhang, G., Zhou, X., Zhang, Y.: Distributed multi-robot task dynamic allocation in digital-twin factory towards industry 5.0. Int. J. Prod. Res. 1-24 (2025)
2. Choset, H., Lynch, K.M., Hutchinson, S., Kantor, G.A., Burgard, W.: Principles of robot motion: theory, algorithms, and implementations. MIT Press (2005)
3. Schwartz, J.T., Sharir, M.: On the piano movers' problem: III. Coordinating the motion of several independent bodies: The special case of circular bodies moving amidst polygonal barriers. Int. J. Robot. Res. 2(3), 46-75 (1983)
4. Yu, J., LaValle, S.M.: Optimal multirobot path planning on graphs: Complete algorithms and effective heuristics. IEEE Trans. Robot. 32(5), 1163-1177 (2016)
5. Tang, S., Thomas, J., Kumar, V.: Hold Or take Optimal Plan (HOOP): A quadratic programming approach to multirobot trajectory generation. Int. J. Robot. Res. 37(9), 1062-1084 (2018)
6. Bareiss, D., van den Berg, J.: Generalized reciprocal collision avoidance. Int. J. Robot. Res. 34(12), 1501-1514 (2015)
7. Claes, D., Hennes, D., Tuyls, K., Meeussen, W.: Collision avoidance under bounded localization uncertainty. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1192-1198. IEEE (2012)
8. Hennes, D., Claes, D., Meeussen, W., Tuyls, K.: Multi-robot collision avoidance with localization uncertainty. In: Proceedings of the International Conference on Autonomous Agents and Multiagent Systems, pp. 147-154 (2012)
9. Snape, J., van den Berg, J., Guy, S.J., Manocha, D.: The hybrid reciprocal velocity obstacle. IEEE Trans. Robot. 27(4), 696-706 (2011)
10. van den Berg, J., Guy, S.J., Lin, M., Manocha, D.: Reciprocal n-body collision avoidance. In: International Symposium on Robotics Research, pp. 3-19. Springer, Berlin (2011)
11. Fiorini, P., Shiller, Z.: Motion planning in dynamic environments using velocity obstacles. Int. J. Robot. Res. 17(7), 760-772 (1998)
12. van den Berg, J., Guy, S.J., Lin, M., Manocha, D.: Optimal reciprocal collision avoidance for multi-agent navigation. In: Proc. of the IEEE International Conference on Robotics and Automation, pp. 1-8. IEEE (2010)
13. Douthwaite, J.A., Zhao, S., Mihaylova, L.S.: A comparative study of velocity obstacle approaches for multiagent systems. In: 2018 UKACC 12th International Conference on Control (CONTROL), pp. 289-294. IEEE (2018)
14. Helbing, D., Molnar, P.: Social force model for pedestrian dynamics. Phys. Rev. E 51(5), 4282 (1995)
15. Moussaïd, M., Helbing, D., Theraulaz, G.: How simple rules determine pedestrian behavior and crowd disasters. Proc. Natl. Acad. Sci. USA 108(17), 6884-6888 (2011)
16. Chen, J., Sun, D.: Resource constrained multirobot task allocation based on leader-follower coalition methodology. Int. J. Robot. Res. 30(12), 1423-1434 (2011)
17. Sharon, G., Stern, R., Felner, A., Sturtevant, N.R.: Conflict-based search for optimal multi-agent pathfinding. Artif. Intell. 219, 40-66 (2015)
18. Li, J., Ruml, W., Koenig, S.: EECBS: A bounded-suboptimal search for multi-agent path finding. In: Proc. of the AAAI Conference on Artificial Intelligence, vol. 35, no. 14, pp. 12353-12362. AAAI Press (2021)
19. Okumura, K.: LaCAM: Search-based algorithm for quick multi-agent pathfinding. In: Proc. of the AAAI Conference on Artificial Intelligence, vol. 37, no. 10, pp. 11655-11662. AAAI Press (2023)

20. Okumura, K., Machida, M., Défago, X., Tamura, Y.: Priority inheritance with backtracking for iterative multi-agent path finding. Artif. Intell. 310, 103752 (2022)
21. Leitmann, G., Skowronski, J.: Avoidance control. J. Optim. Theory Appl. 23(4), 581-591 (1977)
22. Leitmann, G., Skowronski, J.: A note on avoidance control. Optim. Control Appl. Methods 4(4), 335-342 (1983)
23. Chen, C., Liu, Y., Kreiss, S., Alahi, A.: Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning. In: Proc. IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 6015-6022. IEEE (2019)
24. Poddar, S., Mavrogiannis, C., Srinivasa, S.S.: From crowd motion prediction to robot navigation in crowds. In: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6765-6772. IEEE (2023)
25. Mavrogiannis, C., Balasubramanian, K., Poddar, S., Gandra, A., Srinivasa, S.S.: Winding through: Crowd navigation via topological invariance. IEEE Robot. Autom. Lett. 8(1), 121-128 (2023)
26. Stipanović, D.M., Hokayem, P.F., Spong, M.W., Šiljak, D.D.: Cooperative avoidance control for multiagent systems. J. Dyn. Syst. Meas. Control 129(5), 699-707 (2007)
27. Hokayem, P.F., Stipanović, D.M., Spong, M.W.: Coordination and collision avoidance for lagrangian systems with disturbances. Appl. Math. Comput. 217(3), 1085-1094 (2010)
28. van den Berg, J., Lin, M., Manocha, D.: Reciprocal velocity obstacles for real-time multi-agent navigation. In: Proc. IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 1928-1935. IEEE (2008)
29. Duives, D.C., Daamen, W., Hoogendoorn, S.P.: State-of-the-art crowd motion simulation models. Transp. Res. Part C Emerg. Technol. 37, 193-209 (2013)
30. Schultz, M., Rößger, L., Fricke, H., Schlag, B.: Group dynamic behavior and psychometric profiles as substantial driver for pedestrian dynamics. In: Pedestrian and Evacuation Dynamics 2012, pp. 1097-1111. Springer, Cham (2013)
31. Grover, J.S., Liu, C., Sycara, K.: Deadlock analysis and resolution for multi-robot systems. In: International Workshop on the Algorithmic Foundations of Robotics (WAFR), pp. 294-312. Springer, Cham (2020)