Modeling Collision Avoidance Behavior With Zero-Speed Pedestrians

Laura Echeverri, Jean-Michel Auberlet, Jean-Paul Hubert

▶ **To cite this version:**

Laura Echeverri, Jean-Michel Auberlet, Jean-Paul Hubert. Modeling Collision Avoidance Behavior With Zero-Speed Pedestrians. IEEE Transactions on Intelligent Transportation Systems, 2024, 25 (8), pp.9608-9617. 10.1109/TITS.2024.3376077 . hal-04845076

HAL Id: hal-04845076

https://hal.science/hal-04845076v1

Submitted on 18 Dec 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

IEEE Transactions on Intelligent Transportation Systems

Modeling Collision Avoidance Behavior with Zero-speed Pedestrians

Journal:	*Transactions on Intelligent Transportation Systems*
Manuscript ID	T-ITS-22-12-3145.R2
Manuscript Type:	Regular Papers
Date Submitted by the Author:	26-Jan-2024
Complete List of Authors:	Echeverri, Laura C; Université Gustave Eiffel, Laboratoire Ville Mobilité Transport Auberlet, Jean-Michel; Université Gustave Eiffel, COSYS-PICS-L Hubert, Jean-Paul; Université Gustave Eiffel, AME-DEST
Keywords:	Pedestrian dynamics, collision avoidance, Simulation, Virtual agents
Abstract:	In this paper, we present an improved collision avoidance algorithm conceived to deal with zero-speed pedestrians in a crowd. A zero-speed pedestrian is a pedestrian who stops for a short, undefined period to rest, use a phone, or communicate with others, and then resumes walking later. Zero-speed pedestrians can be encountered in places such as shopping centers, sidewalks with retail stores, or public gatherings. When integrated into a crowd simulation, zero-speed pedestrians can cause blockages, particularly livelocks, as observed when using the Optimal Reciprocal Collision Avoidance (ORCA) algorithm. A livelock in a simulation is a situation where an agent is blocked but still has room to move. If the livelock is not handled properly, it will invalidate the simulation runs. We provide insight into the causes of livelocks and propose a modification of ORCA that successfully prevents livelocks from occurring in typical situations in which they can be observed using ORCA. Among other applications, this model can be used as a crowd management tool to evaluate the impact of zero-speed pedestrians on crowd flow.
Note: The following files were submitted by the author for peer review, but cannot be converted to PDF. You must view these files (e.g. movies) online.	
ORCA-ZeroSpeed-IEEE.tex bibliography.bib Images.rar	

SCHOLARONE™
Manuscripts

PLEASE KEEP CONFIDENTIAL

Modeling Collision Avoidance Behavior with Zero-speed Pedestrians

Laura C. Echeverri, Jean-Michel Auberlet, and Jean-Paul Hubert

***Abstract*—In this paper, we present an improved collision avoidance algorithm conceived to deal with zero-speed pedestrians in a crowd. A zero-speed pedestrian is a pedestrian who stops for a short, undefined period to rest, use a phone, or communicate with others, and then resumes walking later. Zero-speed pedestrians can be encountered in places such as shopping centers, sidewalks with retail stores, or public gatherings. When integrated into a crowd simulation, zero-speed pedestrians can cause blockages, particularly livelocks, as observed when using the Optimal Reciprocal Collision Avoidance (ORCA) algorithm. A livelock in a simulation is a situation where an agent is blocked but still has room to move. If the livelock is not handled properly, it will invalidate the simulation runs. We provide insight into the causes of livelocks and propose a modification of ORCA that successfully prevents livelocks from occurring in typical situations in which they can be observed using ORCA. Among other applications, this model can be used as a crowd management tool to evaluate the impact of zero-speed pedestrians on crowd flow.**

***Index Terms*—Pedestrian dynamics, Collision Avoidance, Simulation, Virtual Agents.**

I. Introduction

CROWD simulation research involves the development and use of algorithms to understand, reproduce, or predict the behavior of human crowds [1]. These algorithms have many applications, including entertainment (e.g., computer games and movies), safety considerations (e.g., crowd management and evacuation analysis), and urban planning (e.g., infrastructure design and evaluation) [1]–[5].

Agent-based models treat pedestrians as individual agents interacting with each other. This way, human interaction can be modeled through the influence that neighboring agents exert on one another. For this purpose, in each simulation step, a new velocity is computed for each agent depending on the neighboring agents and obstacles, and the rules prescribing its local behavior [1]. By giving each agent autonomy, agent-based models make it possible to reflect the variety of ways in which pedestrians affect each other's behavior.

In the simulation of pedestrian navigation, collision avoidance is the most studied individual-level behavior. According to Cutting et al. [6], humans avoid collisions by answering two successive questions: Will a collision occur? When will the collision occur? The answers to these questions result from the visual perception of the environment and the presence of moving or static obstacles. Accordingly, collision avoidance algorithms mimic the strategies a pedestrian uses to avoid a potential collision with other pedestrians and obstacles by making assumptions about the agents' perceptions.

In the quest to include diverse behaviors within agent-based pedestrian simulations, we consider the possibility for agents to stop for a short time. Pedestrians may stop for a while to rest, use a phone, look at a store window, or communicate with others. They tend to stop where they must do so and it is hard to tell how much the stop will last [7]. People around them have to adapt their behavior to avoid collisions with them. We aim to model collision avoidance in a crowd where people can stop for a moment and continue on their way. To better identify stopping pedestrians in modeling, we use the term *zero-speed agent*.

Second after the Social Force Model, proposed by Helbing and Molnár [8], one of the most widely adopted algorithms for collision avoidance is the Optimal Reciprocal Collision Avoidance (ORCA) algorithm, introduced by van den Berg et al. [9]. Previous studies have successfully employed ORCA to model pedestrian collision avoidance behavior [10], [11]. ORCA anticipates agent collisions by calculating the agent's new velocity and position based on the velocity and position of its closest neighbors. This new position deviates minimally from the agent's intended path in the absence of other agents. However, similar to other collision avoidance algorithms, ORCA is susceptible to scenarios in which agents are blocked by others on their course to their destination [12]. If the blockage is not handled properly, the simulation becomes useless.

One of these situations concerns the presence of zero-speed agents. If a zero-speed agent is in the path of another agent toward its destination, the latter could be blocked, even if there is room to move. We call this type of blockage a *livelock*, as opposed to a *deadlock*, where an agent is completely immobilized with no room to move. ORCA was originally designed to handle agents in perpetual motion, and therefore, the inclusion of zero-speed agents can lead to the occurrence of livelocks that invalidate the simulation runs.

To address this problem, several algorithms have been proposed to mitigate deadlocks, either by integrating ORCA with alternative approaches or by developing novel collision avoidance strategies (see [12]–[16]). Nonetheless, to the best of our knowledge, no existing study has specifically addressed the challenge of livelocks arising from interactions with zero-speed agents.

The main contributions of this paper are as follows:

Manuscript received December 1, 2022; revised x; accepted x. Date of publication x; date of the current version x. This work was supported by French National Research Agency through the "Laboratoire d'Excellence" (LABEX) *Urban Futures*. *(Corresponding author: Laura C. Echeverri.)*

Laura C. Echeverri is with Univ Gustave Eiffel, Ecole des Ponts, LVMT, F-77454 Marne-la-Vallée, France (e-mail: laura.echeverri-guzman@enpc.fr).

Jean-Michel Auberlet is with Univ Gustave Eiffel, COSYS-PICS-L, F-77454 Marne-la-Vallée, France (e-mail: jean-michel.auberlet@univ-eiffel.fr).

Jean-Paul Hubert is with Univ Gustave Eiffel, AME-DEST, F-77454 Marne - la- Vallée, France (e-mail: jean-paul.hubert@univ-eiffel.fr).

PLEASE KEEP CONFIDENTIAL

1) We provide insight into the reasons behind livelocks in ORCA through examples illustrating typical situations in which they can be observed using this algorithm.
2) We propose a modification of ORCA that avoids livelock with zero-speed agents. This modification enriches the model without altering the overall behavior of the algorithm. If a livelock is likely to occur, the model computes a new velocity that is still collision-free but allows the agent to change direction to avoid the livelock. Thus the simulation can continue without interruption. Our modification successfully prevents livelocks in the illustrative examples, demonstrating its usefulness in crowd modelling in the presence of zero-speed pedestrians.

This paper is organized as follows. First, in Section II, we provide an overview of prior research on algorithms for pedestrian collision avoidance, as well as diverse strategies employed to mitigate blockages in ORCA or to outperform it. Next, Section III offers a succinct summary of the ORCA algorithm, facilitating an understanding of the potential emergence of livelocks. Subsequently, in Section IV, we present two examples illustrating the two cases that lead to livelocks when using ORCA and provide insight into their causes. Later on, Section V introduces our proposed modification to the ORCA algorithm, designed to anticipate the occurrence of livelocks involving zero-speed agents. This section also includes the outcomes obtained by applying our proposed solution to the aforementioned examples. Finally, our conclusions and perspectives on future work are presented in Section VI.

II. Related work

In this section, we first present a brief description of the main classes of pedestrian collision avoidance algorithms and position ORCA as a member of one of these classes. We then provide an overview of approaches that aim to mitigate blocking scenarios using ORCA, or to outperform its blocking avoidance capabilities.

A. Algorithms for pedestrian collision avoidance

Collision avoidance algorithms can be classified into four main classes: force-based, velocity-based, vision-based, and data-driven [1]. Force-based models use an analogy with Newtonian physics to consider that an agent is subject to attractive and repulsive forces that act on its acceleration. Attraction forces allow an agent to move toward a specific goal. Repulsive forces cause agents to repel each other and surrounding obstacles. The Social Force Model (SFM), proposed by Helbing and Molnár [8], is the first force-based model and also a model widely used today. Force-based models update an agent's position only based on the current positions of its neighbors, disregarding their velocities. This approach constrains the models' ability to adequately incorporate temporal anticipation. Such an approach differs from the behavior typically observed in pedestrians, who often proactively anticipate the actions of others [17].

In velocity-based models, agents anticipate future collisions by predicting the trajectories of neighboring agents based on their current positions and projected velocities. In this type of model, the navigation space is translated into a velocity space, and the agent must avoid velocities that lead to collisions with other agents, assuming that each agent maintains a constant velocity for at least a short time. Velocity-based models are computationally more expensive than force-based models but are said to better reflect human behavior [1]. A prominent example of these models is the Optimal Reciprocal Collision Avoidance (ORCA) algorithm, proposed by van den Berg et al. [9]. By defining a half-plane containing collision-free velocities with each agent, it selects a velocity from the intersection of these half-planes that minimally deviates from its preferred velocity. This method provides sufficient conditions for collision-free motion by giving each agent half of the responsibility for avoiding pairwise collisions. Moreover, it is computationally more efficient than other velocity-based models [1].

Vision-based algorithms mimic human navigation based on what humans can see. In retina-based approaches, the most developed category, agents and obstacles are abstracted as pixels to interact with. By focusing on human perception, vision-based algorithms do not consider the complete space of feasible velocities, but instead use gradients. This makes these methods more limited than the velocity-based methods in terms of the information available. Additionally, they are more computationally expensive than either velocity-based or force-based algorithms because processing pixels means that each agent needs to process more information to move [1].

Data-driven methods use real human crowd trajectories to replicate their patterns without explicitly stating the rules governing their behavior. They have been criticized for making it difficult to understand the behavior of the produced model. This hinders the possibility of adjusting the model or improving its performance in specific cases [1].

Our research is part of a project that aims to study different sidewalk configurations and their influence on pedestrian behavior. To achieve this, we aim to simulate various behaviors commonly observed on sidewalks, including pedestrians stopping for short periods. Our choice of ORCA is based on its advantages over other collision avoidance techniques. ORCA generates realistic trajectories while ensuring collision-free movement. Nonetheless, it is important to note that ORCA may encounter scenarios where agents are blocked and cannot move, whether they have room to move (livelock) or not (deadlock) [12].

B. Algorithms for preventing blockage

Collision avoidance algorithms are prone to situations where one or more agents are blocked, unable to continue their trajectory to their goal [12]. Such situations can render the current simulation unusable. We distinguish two types of blockage: livelock and deadlock. A *livelock* is when an agent is blocked while still having room to move. A *deadlock* is when an agent is blocked with no room to move, possibly colliding with other agents. Although real pedestrians can face blockage situations they tend to solve them quite quickly. This has not yet been able to be reproduced by many collision avoidance algorithms [12], [13], [18]. That is why one of the

PLEASE KEEP CONFIDENTIAL

core concerns when developing or improving these algorithms is to avoid blockage situations.

In the original formulation of ORCA, van den Berg et al. [9] conceived a strategy to escape from the deadlock in densely packed conditions. In such conditions, velocities can become zero to avoid collisions. Their strategy consists of selecting the safest possible velocity for the blocked agent. This strategy does not, however, prevent blockages from occurring and some blockage avoidance approaches have been proposed to improve or outperform ORCA.

Approaches proposed to outperform ORCA's blockage avoidance include the following. Zhou et al. [13] proposed a Buffered Voronoi Cell (BVC) collision avoidance algorithm in which each agent computes its Voronoi cell and plans a trajectory to the point in the cell that is closest to its goal. A BVC is a set of points that are closer to an agent than to any other agent, retracted by the physical extent of the agent. The authors presented a heuristic solution to deal with livelocks, which arise when certain agents obstruct each other in a manner that prevents at least one agent from reaching its goal through its control algorithm. In such cases, the blocked agent is in a vertex of its BVC that is the closest point to its goal. To resolve the blockage, the agent chooses one of its adjacent edges to detour along. Their algorithm showed comparable performance to ORCA in experiments with multiple dynamic robots moving in arbitrary dimensions. It should be noted, however, that their approach relies solely on current positions for collision avoidance, which limits its ability to effectively anticipate collisions.

Şenbaşlar et al. [14] also used BVC for collision avoidance, taking as input pre-planned trajectories, built using discrete planning. They compared their method to ORCA, and ORCA combined with discrete planning. Their experiments revealed that agents using their approach, as well as those employing ORCA coupled with discrete planning, successfully reached their destination. Conversely, agents following only the ORCA approach could become trapped when attempting to navigate around static obstacles, a situation we can also refer to as a livelock. This problem arises because ORCA focuses primarily on collision avoidance and requires additional strategies for agents to successfully circumvent static obstacles [9]. It is worth noting, however, that their method takes more computation time compared to ORCA.

Lastly, Semnani et al. [15] proposed an adaptation of the flocking algorithm proposed by Reynolds [19] capable of generating paths that are both collision-free and deadlock-free, even within densely packed scenarios. In such scenarios, agents using ORCA might not be able to reach their goal positions because the conventional motion of these agents toward their goals is obstructed by other agents that have already reached their final destinations. Note that these latter agents should not be confused with zero-speed agents, as zero-speed agents retain the possibility of resuming motion.

Approaches that enhance blockage avoidance achieved with ORCA include the following. Dergachev et al. [12] combined three algorithms for multi-agent navigation: Theta* algorithm for individual path planning, ORCA for collision avoidance, and multi-agent path-finding for avoiding deadlocks. When an agent encounters a potential deadlock, such as navigating through a narrow passage, it switches to a coordinated path-planning mode. In this mode, the agent collaborates with its neighbors to construct a joint conflict-free plan, executing it before returning to independent navigation. The authors conducted a comparative study, contrasting their method with the combination of ORCA and Theta*. Their approach notably reduced the occurrence of deadlocks; however, it tends to be time-intensive due to its centralized decision-making nature. Finally, Arul and Manocha [16] coupled BVC and reciprocal velocity obstacles for collision avoidance. Their deadlock resolution strategy consists of identifying if an agent is yet to reach its goal position but with zero speed. If so, the agent identifies the neighboring agent that is closest to the direction of its goal location and switches positions with it to end the blockage. Their method is less conservative than ORCA and has similar runtime performance.

In reviewing the literature, we can make two observations. First, the proposed approaches rely on combining ORCA with other algorithms, such as path planning, or introducing alternative collision avoidance techniques. To the best of our knowledge, no established approach directly modifies ORCA itself to avoid blockage. Second, it becomes apparent that there is a gap in research addressing the problem of livelock arising from agents getting stuck around zero-speed agents. Şenbaşlar et al. [14] proposed a strategy for dealing with livelocks arising from attempts to circumvent static obstacles. However, this approach proves inapplicable to scenarios involving zero-speed agents, since these agents can resume motion at any time. However, livelocks can occur in ORCA when agents interact with zero-speed agents, as we show in Section IV. We address this situation by proposing a modification of ORCA. This modification does not rely on the use of auxiliary algorithms in addition to ORCA. Before presenting this modification, we provide a brief overview of ORCA in the following section.

III. Reciprocal collision avoidance with ORCA

In this section, we provide a brief description of ORCA. This is intended to provide a better understanding of the circumstances that may lead to the potential occurrence of livelocks. For more details on ORCA, see [9].

A. Problem definition

ORCA is a velocity-based algorithm for local navigation in which each agent anticipates collisions by choosing its next velocity according to the position and velocity of its nearest neighbors. In ORCA, each agent assumes that other agents use the same collision avoidance strategy and that the responsibility for avoiding a collision is shared with every other agent. At each simulation step, the model derives for each agent a half-plane of velocities in the velocity space that will not lead to a collision with any other agent over a fixed time horizon into the future. The agent then chooses a velocity from the intersection of all half-planes that is closest to its preferred velocity, i.e., the velocity the agent would have if no agent were in its path. The model uses linear programming to find the optimal velocity. In this way, ORCA guides the

PLEASE KEEP CONFIDENTIAL

agent as directly as possible toward its goal, while ensuring a collision-free motion. In other words, ORCA prioritizes direction over collision avoidance, since the velocity chosen to ensure collision avoidance is intended to minimize deviation from the preferred direction.

ORCA solves the reciprocal n-Body Collision Avoidance problem, which is defined as follows. Let there be a set of n pedestrians modeled as dynamic agents with circular shapes in 2D. Each agent A has a current position $\mathbf{p}_A$, radius r_A, and velocity $\mathbf{v}_A$, that can be observed by other agents. Additionally, it has a maximum speed $s_A^{\max}$ (parameter in m/s) and a preferred velocity $\mathbf{v}_A^{\text{pref}}$ (variable, vector depending on its destination at each step of the simulation).

The objective of the problem is that each agent A selects its new velocity $\mathbf{v}_A^{\text{new}}$ such that all agents are guaranteed to be collision-free for at least a fixed interval of time τ, in which they would continue moving at their new velocity. Moreover, this new velocity should be as close as possible to the agent's preferred velocity, which is in the direction of the agent's destination.

B. Optimal Reciprocal Collision Avoidance

At each time step Δt, each neighboring agent B of agent A induces a set $R^{\tau}_{A|B}$ of relative velocities of A with respect to B that guarantees collision avoidance with B in the time horizon τ, with $\Delta t \leq \tau$. In other words, the set $R^{\tau}_{A|B}$ of permitted velocities for A induced by B contains all relative velocities of A with respect to B that will not lead to a collision between A and B at any time before τ. The set $R^{\tau}_{A|B}$ is a half-plane of velocities in the velocity space. Similarly, A induces a corresponding set of permitted velocities for B, denoted as $R^{\tau}_{B|A}$.

Static obstacles are modeled as collections of line segments. Collision avoidance with these obstacles follows a similar approach to that presented for agents. Let O be an obstacle line segment. The set $R^{\tau}_{A|O}$ denotes the half-plane of permitted velocities for A with respect to O.

The set of velocities permitted for A with respect to all agents and obstacles is the intersection of the half-planes of permitted velocities induced by each other agent and obstacle. This set is denoted as R^{τ}_A and is defined as:

$$R^{\tau}_A = D(\mathbf{0}, s_A^{\max}) \cap \bigcap_{B \neq A} R^{\tau}_{A|B} \cap \bigcap_{O} R^{\tau}_{A|O} \tag{1}$$

Here, $D(\mathbf{0}, s_A^{\max})$ denotes an open disc of radius $s_A^{\max}$ centered at vector $\mathbf{0}$ that represents the maximum speed constraint on agent A.

The agent then selects a new velocity $\mathbf{v}_A^{\text{new}}$ to minimize the distance to its preferred velocity $\mathbf{v}_A^{\text{pref}}$ within the region of permitted velocities, R^{τ}_A, that is,

$$\mathbf{v}_A^{\text{new}} = \arg\min_{\mathbf{v} \in R^{\tau}_A} \|\mathbf{v} - \mathbf{v}_A^{\text{pref}}\|. \tag{2}$$

This selection process can be efficiently performed using linear programming. The region R^{τ}_A is convex, bounded by linear constraints from the half-planes of permitted velocities induced by other agents and obstacles. The optimization problem has a unique local minimum at the intersection point of the lines bounding two of the half-planes that define R^{τ}_A. To solve this, van den Berg et al. [9] utilized the linear programming algorithm introduced by Berg et al. [20]. It is noteworthy that ORCA was initially designed for 2D environments; nevertheless, all definitions and the algorithm can be seamlessly extended to 3D scenarios.

IV. Why Livelock Can Occur in ORCA with Zero-Speed Agents

We define as *zero-speed agent* a pedestrian that remains with zero speed for an indefinite amount of time, with the possibility of resuming its path afterward. We distinguish it from a *regular agent*, a pedestrian that has a speed different from zero. Other agents cannot know how much time a zero-speed agent will remain stopped. This forces them to treat it as a dynamic agent instead of a static object to be circumvented. ORCA works well when considering dynamic agents in constant movement, but can generate livelocks when an agent is moving close to others with zero speed. Depending on its relative position and velocity with respect to the zero-speed agents, an agent can get "stuck" behind them. This would happen if the agent, "guided" by its destination, advances towards a position near the zero-speed agents where it can no longer "get out". While moving in this direction the speed of the agent in the subsequent time steps decreases until it reaches almost zero. In other words, the agent gets "stuck" behind the zero-speed agents. Since there may be other positions available to move to, we say there is a livelock.

We present two scenarios that illustrate the two types of situations that lead to livelock occurrences with ORCA. In both scenarios, an agent navigates from left to right along a 10×5 m sidewalk. The simulations assume the following parameter values: all agents have a radius of $r = 0.3$ m and a maximum speed of $s^{max} = 1$ m/s. The simulations use a time step of $\Delta t = 0.1$ s, and the collision avoidance time horizon is $\tau = 1$ s. These simulation parameters are based on the default parameters of ORCA, since this paper aims to illustrate how the livelocks could occur and how we address them.

Consider the example illustrated in Fig. 1. Agent A is moving from left to right. At this point of the simulation, B and C are zero-speed agents discussing together somewhere in the middle of a sidewalk. They are in the direction of A's goal. The trace of A is depicted while advancing toward its goal and its position is provided at three different times (in seconds) t. From this figure, we can see that A is directed to the middle of B and C, while the agent's speed diminishes when approaching them (notice how the space between the points on the trace diminishes over time). Since ORCA anticipates a collision over a fixed time horizon τ, agent A starts changing orientation and speed, before getting *too close* to the other agents. This, however, is not enough to avoid getting into a livelock. To inspect why this phenomenon occurs, Figures 2(a)-2(c) show the half-planes of permitted velocities for agent A induced by agents B and C at the three different times (in seconds) t. $\mathbf{v}_A^{\text{new}}$, obtained by solving Equation (2), is

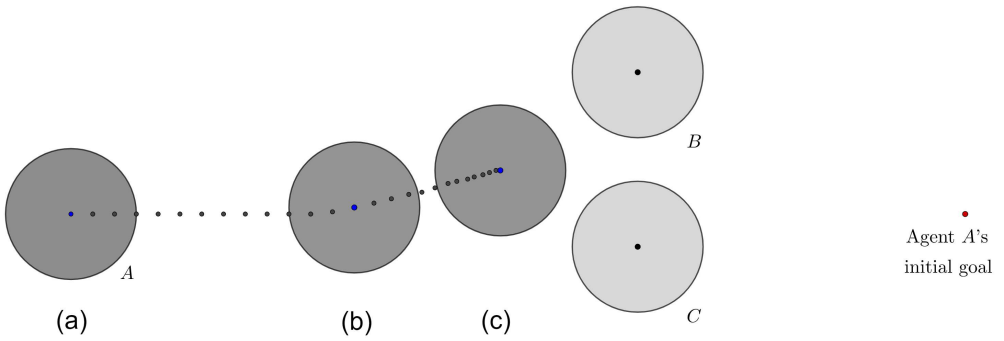

Fig. 1. Trace of agent A while moving from left to right using ORCA. Agents B and C, with zero speeds, are in the middle of a sidewalk in the direction of A's goal. The position of A is shown in times (in seconds) (a) $t = 2$, (b) $t = 3.3$, and (c) $t = 4.5$. All agents have the same size ($r_A = r_B = r_C = 0.3m$), $s_A^{\max} = 1m/s$ and $\tau = 1s$.

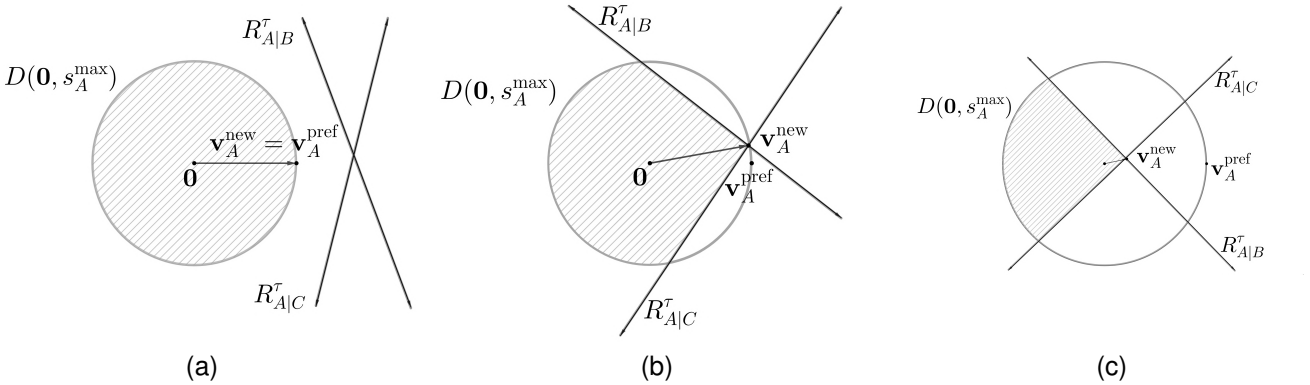

Fig. 2. Half-planes of permitted velocities using ORCA for agent A induced by agents B and C in the example shown in Fig. 1 at times (in seconds) (a) $t = 2$, (b) $t = 3.3$, and (c) $t = 4.5$

.

also depicted for each time. Let $\ell^{\tau}_{A|B}$ and $\ell^{\tau}_{A|C}$ be the lines bounding $R^{\tau}_{A|B}$ and $R^{\tau}_{A|C}$. $\mathbf{v}_A^{\text{new}}$ is the intersection point of $\ell^{\tau}_{A|B}$ and $\ell^{\tau}_{A|C}$. Since neither B nor C start moving, and A continues to head towards the same direction, $\mathbf{v}_A^{\text{new}}$ keeps being the intersection point of $\ell^{\tau}_{A|B}$ and $\ell^{\tau}_{A|C}$, always reducing its magnitude (see the size of the vector to $\mathbf{v}_A^{\text{new}}$ in Figures 2(a)-2(c)). This is due to the fact that $\mathbf{v}_A^{\text{new}}$ is chosen by minimizing the distance between $\mathbf{v}_A^{\text{pref}}$ and $\mathbf{v}_A^{\text{new}}$. The situation exposed is a livelock, since A is blocked behind B and C and will remain so until either B or C restarts, while there are alternative velocities within R^{τ}_A that would not lead to this situation.

Fig. 3 shows the other situation in which this phenomenon can be observed, this time with a regular agent A and a zero-speed agent B near the edge of a sidewalk. Agent A is moving from left to right, facing agent B. Fig. 3 provides the trace of A over time and its position at three different times (in seconds) t. In this figure, we can see that A is directed to a point between the edge of the sidewalk and B owing to its goal. Furthermore, the speed is diminishing while approaching this point (the space between the points on the trace diminishes over time). Let $\ell^{\tau}_{A|B}$ and $\ell^{\tau}_{A|O}$ be the lines bounding $R^{\tau}_{A|B}$ and $R^{\tau}_{A|O}$, with O being the line segment representing the edge of the sidewalk. Figures 4(a)-4(c) show the half-planes of permitted velocities for agent A induced by agent B and the edge of the sidewalk as well as $\mathbf{v}_A^{\text{new}}$ at the three different times (in seconds) t. As can be seen from these figures, in the three times, $\mathbf{v}_A^{\text{new}}$ is the intersection point of $\ell^{\tau}_{A|B}$ and $\ell^{\tau}_{A|O}$. Since A continues to head towards the same direction and B does not move, $\mathbf{v}_A^{\text{new}}$ continues being the intersection point of $\ell^{\tau}_{A|B}$ and $\ell^{\tau}_{A|O}$. The size of the vector to $\mathbf{v}_A^{\text{new}}$ decreases, meaning that the agent slows down until it reaches a speed of almost zero. However, there are other velocities in R^{τ}_A that the agent could take instead.

According to the examples presented above, a livelock in ORCA can be explained as follows. At each time step, ORCA selects a new velocity $\mathbf{v}_A^{\text{new}}$ for agent A from the convex region R^{τ}_A that is closest to A's preferred velocity, $\mathbf{v}_A^{\text{pref}}$. As long as a velocity is feasible and with a minimum distance to the preferred velocity, it will be chosen, regardless of the type of agent or obstacle to which that half-plane intersection belongs. This is due to the choice to prioritize reaching the destination (minimizing the distance to the preferred velocity) over avoiding collisions. In both scenarios, $\mathbf{v}_A^{\text{new}}$ lies at the intersection point of two bounding lines associated with zero-speed agents or static obstacles. Since the zero-speed agents do not restart, A is blocked. Changing the time horizon τ to avoid a collision is not enough to evade the livelock, since the situation depends on the agent positions, the goal of agent A, and the presence of zero-speed agents. The livelock results from ORCA's collision avoidance methodology. Furthermore,

PLEASE KEEP CONFIDENTIAL

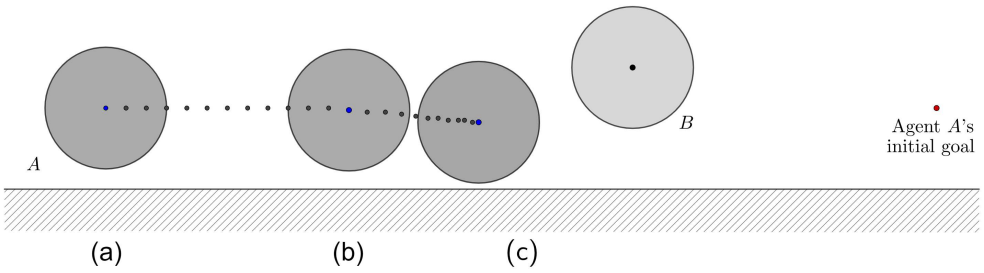

Fig. 3. Trace of agent A while moving from left to right near an edge of a sidewalk using ORCA. Agent B, with zero speed, is in the direction of A's goal. The position of A is shown in times (in seconds) (a)$t = 2$, (b)$t = 3.2$, and (c)$t = 4.3$. All agents have the same size ($r_A = r_B = r_C = 0.3m$), $s_A^{\max} = 1m/s$ and $\tau = 1s$.

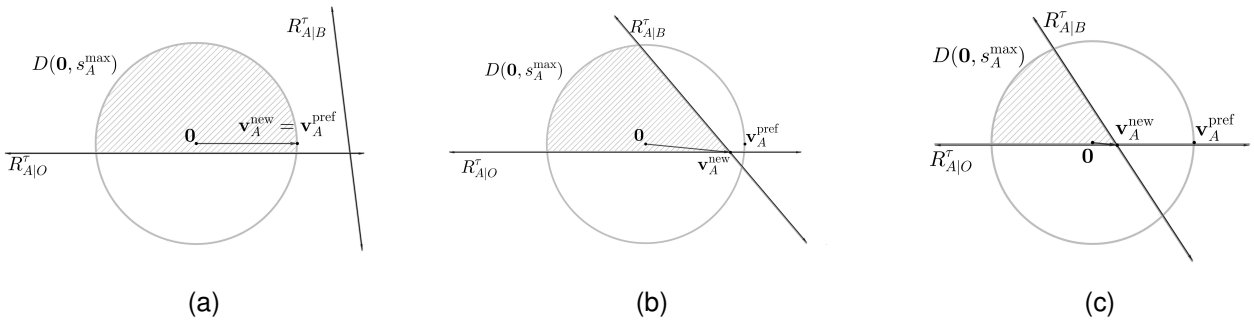

Fig. 4. Half-planes of permitted velocities using ORCA for agent A induced by agent B and an edge of a sidewalk in the example shown in Fig. 3 at times (in seconds) (a) $t = 2$, (b) $t = 3.2$, (c) $t = 4.3$.

agent A cannot know in advance if other agents will stop and the duration of the stop. Therefore it is not possible to plan the change of the agent's preferred velocity (e.g., through global path planning techniques) to circumvent zero-speed agents in the direction of its goal.

To avoid livelocks with zero-speed agents, we propose a modification in the selection of $\mathbf{v}_A^{\text{new}}$, while keeping the same principle of ORCA. In our approach, the new velocity differs from the one obtained by solving Equation (2), but belongs to the region of permitted velocities R_A^τ, thus ensuring collision avoidance. This new velocity is intended to help the agent escape from a livelock before it is produced by anticipating the conditions that would lead it to it.

V. Livelock Avoidance or How to Consider Zero Speeds Pedestrians with ORCA

In this section, we describe the modification we propose to ORCA's new velocity computation to account for zero-speed agents and, consequently, avoid the potential livelocks with them. Afterward, we show the results of applying this modification to the examples presented in Section IV.

A. Livelock avoidance in ORCA

In the two examples presented in Section IV, $\mathbf{v}_A^{\text{new}}$ is the intersection point of lines bounding the half-planes associated with two zero-speed agents (Fig. 2), or a static object and a zero-speed agent (Fig. 4). In essence, the agent, driven by its goal, advances toward a state of livelock due to the presence of zero-speed agents on its path. Our methodology is built upon the anticipation of livelocks arising from interactions with zero-speed pedestrians. We hypothesize that by "banning" certain types of intersection points as potential candidates for $\mathbf{v}_A^{\text{new}}$ and proposing an alternative velocity within R_A^τ, the issue of livelock can be effectively resolved. The alternative velocity will not lead to a collision between A and any other agent or obstacle at any time before time τ, since R_A^τ is defined to guarantee collision avoidance.

The *banned* intersection points correspond to situations involving interactions with either two zero-speed agents, a zero-speed agent and an obstacle, or two obstacles. Even though the last scenario does not involve a zero-speed agent, we included it because of the similarity in agent behavior around static obstacles and zero-speed pedestrians. Our computation of the alternative new velocity involves a modification of the linear programming algorithm developed by Berg et [20].

We label the bounding lines of permitted half-planes to indicate to what type of agent or obstacle they belong to. We distinguish three labels: "regular agent", "zero-speed agent", and "obstacle". The combinations of line types that form a *banned* intersection point are indicated in Table I.

The linear programming algorithm of Berg et al. [20] adds the linear constraints (the half-planes) one after the other, choosing the current $\mathbf{v}_A^{\text{new}}$ at each step. In the original version of the algorithm, the solution to the intermediate linear programs is optimal with respect to the objective function (Eq. (2)). Our version is intended to sacrifice optimality, e.g. to prioritize the destination in the computation of $\mathbf{v}_A^{\text{new}}$, to escape from livelocks with zero-speed agents or static obstacles.

We denote S_A^τ as the set of half-planes that make up the

TABLE I
LINE TYPES COMBINATIONS RELATED TO DIFFERENT TYPES OF AGENTS OR OBSTACLES

	Regular agent	Zero-speed agent	Obstacle
Regular agent	–	–	–
Zero-speed agent	–	X	X
Obstacle	–	X	X

Note: The line types combinations that form a *banned* intersection point have an X.

region of permitted velocities, R^{τ}_{A}. We number the m half-planes belonging to S^{τ}_{A} as $R^{\tau}_{A|B_1}, R^{\tau}_{A|B_2}, \ldots, R^{\tau}_{A|B_m}$. Let $S^{\tau}_{A|B_i}$ be the set of the first i half-planes, together with the maximum speed constraint $D(\mathbf{0}, s^{\max}_{A})$, and let $R^{*\tau}_{A|B_i}$ be the feasible region defined by these constraints:

$$S^{\tau}_{A|B_i} := \{D(\mathbf{0}, s^{\max}_{A}), R^{\tau}_{A|B_1}, R^{\tau}_{A|B_2}, \ldots, R^{\tau}_{A|B_i}\} \quad (3)$$

$$R^{*\tau}_{A|B_i} := D(\mathbf{0}, s^{\max}_{A}) \cap R^{\tau}_{A|B_1} \cap R^{\tau}_{A|B_2} \cap \ldots \cap R^{\tau}_{A|B_i}. \quad (4)$$

We have that

$$R^{*\tau}_{A|B_1} \supseteq R^{*\tau}_{A|B_2} \supseteq \cdots \supseteq R^{*\tau}_{A|B_m} = R^{\tau}_{A} \quad (5)$$

since $R^{*\tau}_{A|B_i} = R^{*\tau}_{A|B_{i-1}} \cap R^{\tau}_{A|B_i}$ for all $1 < i \leq m$.

The solution to each linear program is unique because each feasible region is bounded by $D(\mathbf{0}, s^{\max}_{A})$ (see [20]). It will be one of the vertices of the feasible region. Let $1 \leq i \leq m$ and $\mathbf{v}_i$ be the solution for $R^{*\tau}_{A|B_i}$. When adding linear constraint $R^{\tau}_{A|B_i}$, we verify if the solution for the current feasible region $R^{*\tau}_{A|B_{i-1}}$ belongs to it, that is, if $\mathbf{v}_{i-1} \in R^{\tau}_{A|B_i}$. If so, then $\mathbf{v}_i = \mathbf{v}_{i-1}$. If not, then either $R^{*\tau}_{A|B_i}$ is an empty set or $\mathbf{v}_i$ belongs to the line bounding $R^{\tau}_{A|B_i}$. We call this line $\ell^{\tau}_{A|B_i}$.

If $\mathbf{v}_{i-1} \notin R^{\tau}_{A|B_i}$ and $R^{*\tau}_{A|B_i}$ is not an empty set, the solution will be one of the two intersection points of $\ell^{\tau}_{A|B_i}$ with the lines bounding $R^{*\tau}_{A|B_{i-1}}$. Initially, the chosen solution will be the one minimizing the distance to $\mathbf{v}^{\text{pref}}_{A}$. If this solution corresponds to a *banned* intersection point, we choose the other intersection point, to ensure escaping from a livelock. It is worth noting that our procedure remains independent of simulation parameters other than the types of agents or obstacles present. Furthermore, when an agent encounters neither zero-speed agents nor obstacles, the standard collision avoidance behavior of ORCA applies.

Algorithm 1 provides the general structure of the procedure to obtain $\mathbf{v}^{\text{new}}_{A}$. It requires m, S^{τ}_{A}, $\mathbf{v}^{\text{pref}}_{A}$, and $s^{\max}_{A}$, and ensures $\mathbf{v}^{\text{new}}_{A}$. The procedure starts by initializing $\mathbf{v}^{\text{new}}_{A}$ as the maximum speed in the preferred direction, $\mathbf{v}^{\text{pref}}_{A}$ (lines 1-2). Then the algorithm enters the main for-loop to find $\mathbf{v}^{\text{new}}_{A}$ in R^{τ}_{A} (lines 3-10). When adding each half-plane $R^{\tau}_{A|B_i}$, it checks if $\mathbf{v}^{\text{new}}_{A}$ belongs to it. If so, $\mathbf{v}^{\text{new}}_{A}$ remains the same. If not, the procedure `LinProgMod` computes a point that minimizes the objective function (Eq. (2)) or avoids a potential livelock with zero-speed agents or obstacles, depending on the label of $\ell^{\tau}_{A|B_i}$. If the point does not exist, the linear program is infeasible (R^{τ}_{A} is empty). In such a case, we use the strategy for densely packed conditions as presented in [9].

Algorithm 1 `computeNewVelocity`

Require: $m, S^{\tau}_{A}, \mathbf{v}^{\text{pref}}_{A}, s^{\max}_{A}$
Ensure: $\mathbf{v}^{\text{new}}_{A}$
1: $\mathbf{v}^{\text{pref}}_{A} \leftarrow \mathbf{v}^{\text{pref}}_{A} \cdot s^{\max}_{A}$
2: $\mathbf{v}^{\text{new}}_{A} \leftarrow \mathbf{v}^{\text{pref}}_{A}$
3: **for** $i \leftarrow 1$ **to** n **do**
4: **if** $\mathbf{v}^{\text{new}}_{A} \notin R^{\tau}_{A|B_i}$ **then**
5: $\{\mathbf{v}^{\text{new}}_{A}, feasible\} \leftarrow$ `LinProgMod`$(R^{\tau}_{A|B_i}, R^{*\tau}_{A|B_{i-1}}, \mathbf{v}^{\text{pref}}_{A}, \mathbf{v}^{\text{new}}_{A})$
6: **if** $feasible =$ false **then**
7: Report that the linear program is infeasible and quit
8: **end if**
9: **end if**
10: **end for**

Algorithm 2 presents the function `LinProgMod`. It requires $R^{\tau}_{A|B_i}$, $R^{*\tau}_{A|B_{i-1}}$, $\mathbf{v}^{\text{pref}}_{A}$, and the current value of $\mathbf{v}^{\text{new}}_{A}$, to update it. The function starts by calling `ComputeIntersectionPoints`, that finds the intersection points of $\ell^{\tau}_{A|B_i}$ and the lines bounding $R^{*\tau}_{A|B_{i-1}}$. If it is not possible to find them, the linear program is infeasible and the program quits (lines 1-4). After finding the intersection points, the function retrieves the lines intersecting $\ell^{\tau}_{A|B_i}$ (line 5). Next, it checks whether $\mathbf{v}^{\text{pref}}_{A}$ and $\ell^{\tau}_{A|B_i}$ are on the same direction or not. This is to select which of the intersection points corresponds to the solution of the linear program (lines 6-11). Afterward, it retrieves the labels of $\ell^{\tau}_{A|B_i}$ and the line that intersects it in the corresponding *direction*. This is to check if we have a *banned* intersection point (see Table I). In Algorithm 2, we shorten the labels "zero-speed agent" and "obstacle" as "z-s agent" and "obs". In the case of a *banned* intersection point, we choose the other *direction*, that is, we choose the other intersection point to be $\mathbf{v}^{\text{new}}_{A}$ (lines 12-15).

Algorithm 2 `LinProgMod`

Require: $R^{\tau}_{A|B_i}, R^{*\tau}_{A|B_{i-1}}, \mathbf{v}^{\text{pref}}_{A}, \mathbf{v}^{\text{new}}_{A}$
Ensure: $\mathbf{v}^{\text{new}}_{A}, feasible$
1: $\{[\mathbf{v}^{1}_{A|B_i}, \mathbf{v}^{-1}_{A|B_i}], feasible\} \leftarrow$ `ComputeIntersectionPoints`$(\ell^{\tau}_{A|B_i}, R^{*\tau}_{A|B_{i-1}})$
2: **if** $feasible =$ false **then**
3: Report that the linear program is infeasible and quit
4: **end if**
5: $[\ell^{1}, \ell^{\text{-1}}] \leftarrow$ the lines intersecting $\ell^{\tau}_{A|B_i}$ in $[v^{1}_{A|B_i}, v^{\text{-1}}_{A|B_i}]$
6: $dir \leftarrow 0$
7: **if** $\mathbf{v}^{\text{pref}}_{A}$ and $\ell^{\tau}_{A|B_i}$ are on the same direction **then**
8: $dir \leftarrow$1
9: **else**
10: $dir \leftarrow$ -1
11: **end if**
12: **if** (`label`$(\ell^{\tau}_{A|B_i}) =$"z-s agent" $\|$ `label`$(\ell^{\tau}_{A|B_i}) =$"obs") & (`label`$(\ell^{dir}) =$"z-s agent" $\|$ `label`$(\ell^{dir}) =$"obs") **then**
13: $dir \leftarrow dir \times -1$
14: **end if**
15: $\mathbf{v}^{\text{new}}_{A} \leftarrow \mathbf{v}^{dir}_{A|B_i}$

Fig. 5 provides an example illustrating the approach followed in Algorithms 1 and 2. At the beginning, $D(\mathbf{0}, s^{\max}_{A})$ is computed to bound the feasible region and $\mathbf{v}^{\text{new}}_{A} = \mathbf{v}^{\text{pref}}_{A}$

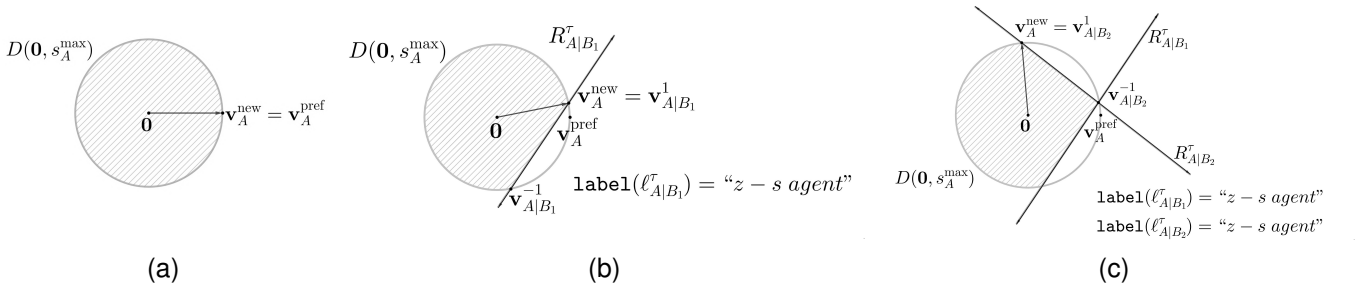

Fig. 5. Example of functioning of Algorithms 1 and 2. (a) $D(\mathbf{0}, s_A^{\max})$ bounds the feasible region. (b) $R^{\tau}_{A|B_1}$ is added, changing $\mathbf{v}_A^{\text{new}}$. (c) $R^{\tau}_{A|B_2}$ is added. Originally, $\mathbf{v}_A^{\text{new}} = \mathbf{v}^{-1}_{A|B_2}$ would be chosen, but $\mathbf{v}^{-1}_{A|B_2}$ is the intersection point of two "zero-speed agent" bounding lines. Thus $\mathbf{v}_A^{\text{new}} = \mathbf{v}^1_{A|B_2}$.

(Fig. 5(a)). Then, we add half-plane $R^{\tau}_{A|B_1}$, but $\mathbf{v}_A^{\text{new}} \notin R^{\tau}_{A|B_1}$, which means that a new velocity must be computed. $\mathbf{v}^1_{A|B_1}$ and $\mathbf{v}^{-1}_{A|B_1}$ are the intersection points of $\ell^{\tau}_{A|B_1}$ with $D(\mathbf{0}, s_A^{\max})$. A *direction* is chosen and, since it only involves one "zero-speed agent" bounding line, there is no need to do anything else (Fig. 5(b)). Next, we add half-plane $R^{\tau}_{A|B_2}$. Again, $\mathbf{v}_A^{\text{new}} \notin R^{\tau}_{A|B_2}$. $\mathbf{v}^1_{A|B_2}$ and $\mathbf{v}^{-1}_{A|B_2}$ are the intersection points of $\ell^{\tau}_{A|B_2}$ with $D(\mathbf{0}, s_A^{\max})$ and $\ell^{\tau}_{A|B_1}$. When choosing a *direction*, it leads to $\mathbf{v}_A^{\text{new}} = \mathbf{v}^{-1}_{A|B_2}$. But $\mathbf{v}^{-1}_{A|B_2}$ is the intersection point of two "zero-speed agent" bounding lines, $\ell^{\tau}_{A|B_1}$ and $\ell^{\tau}_{A|B_2}$. Thus, the other *direction* is chosen and, finally, $\mathbf{v}_A^{\text{new}} = \mathbf{v}^1_{A|B_2}$ (Fig. 5(c)).

B. Results

We compare the results obtained using our methodology with ORCA in the two examples presented in Section IV. Fig. 6 presents the example with regular agent A facing zero-speed agents B and C using our methodology. From the figure, we can see that A changes direction to avoid being trapped behind B and C. Figures 7(a)-7(c) present the half-planes of permitted velocities for agent A induced by B and C as well as $\mathbf{v}_A^{\text{new}}$ at three different times (in seconds) t. At $t = 2$, $\mathbf{v}_A^{\text{new}}$ is the same for both methodologies, since agents B and C are still far from A (see Figures 2(a) and 7(a)). At $t = 3.3$, $\mathbf{v}_A^{\text{new}}$ with ORCA would be a *banned* intersection point. $\mathbf{v}_A^{\text{new}}$ would be the intersection of the lines bounding $R^{\tau}_{A|B}$ and $R^{\tau}_{A|C}$, $\ell^{\tau}_{A|B}$ and $\ell^{\tau}_{A|C}$ (see Fig. 2(b)). In our methodology, since B and C are zero-speed agents, $\ell^{\tau}_{A|B}$ and $\ell^{\tau}_{A|C}$ are labelled as "zero-speed agent" bounding lines. Therefore, our algorithm chooses instead the intersection of $\ell^{\tau}_{A|B}$ with $D(\mathbf{0}, s_A^{\max})$ (see Fig. 7(b)). Finally, at $t = 4.5$, A with ORCA would be blocked in a livelock (see Fig. 2(c)). Instead, with our methodology, it is not blocked and the simulation can continue, as shown in Fig. 7(c).

Similarly, Fig. 8 provides the example with regular agent A moving near the edge of a sidewalk towards zero-speed agent B. The figure shows that agent A changes its trajectory to avoid being blocked behind agent B. Figures 9(a)-9(c) provide the half-planes of permitted velocities for agent A induced by agent B and the edge of the sidewalk together with $\mathbf{v}_A^{\text{new}}$ at three different times. At $t = 3.2$, ORCA would choose a *banned* intersection point, corresponding to zero-speed agent B and the edge of the sidewalk. Instead, our algorithm chooses the intersection of $\ell^{\tau}_{A|B}$ (the line bounding $R^{\tau}_{A|B}$) with $D(\mathbf{0}, s_A^{\max})$ (see Fig. 9(b)). This allows agent A to escape from a livelock. The simulation continues as shown in Fig. 9(c).

VI. Conclusions and Future Work

Agent-based crowd simulation algorithms allow the modeling of the individual behavior of pedestrians in a crowd and how they interact with each other. In the quest to expand the number of behaviors that can be modeled, we considered including pedestrians who can stop for a while and continue walking later, we call them zero-speed agents. Pedestrians make short stops for a variety of reasons, including resting, using their phones, or communicating with others. Since this can be quite spontaneous, the people around them need to know how to react. When integrated into a crowd simulation, zero-speed agents can cause blockages that invalidate the simulation run.

In this paper, we have presented an improved collision avoidance algorithm designed to deal with zero-speed agents and allow the simulation to continue. As a modification of the well-known ORCA algorithm, our model addresses situations where agents may be blocked by zero-speed agents in the original ORCA version. Unlike ORCA, our algorithm prioritizes collision avoidance over steering, preventing potential livelocks—situations where agents cannot move despite available space. Inspired by the natural behavior of real pedestrians navigating around zero-speed counterparts, our algorithm aims to enhance realism in crowd simulations.

Our procedure extends the ORCA algorithm by preventing the agent trajectory from converging in a potential livelock scenario with zero-speed agents. In cases where such a livelock risk is foreseeable, the new velocity deviates from the one that minimizes the distance to the agent's preferred velocity. The new velocity still belongs to the set of velocities not leading to a collision and allows the agent to change direction to avoid a future livelock. This way, our procedure prioritizes collision and livelock avoidance over steering in the presence of zero-speed agents. By not relying on the use of other algorithms on top of ORCA, our method does not increase the difficulty of collision avoidance.

Our approach was successful in avoiding a livelock in the most common cases where a livelock occurs with zero-speed agents in ORCA. Agents change the direction in a way that

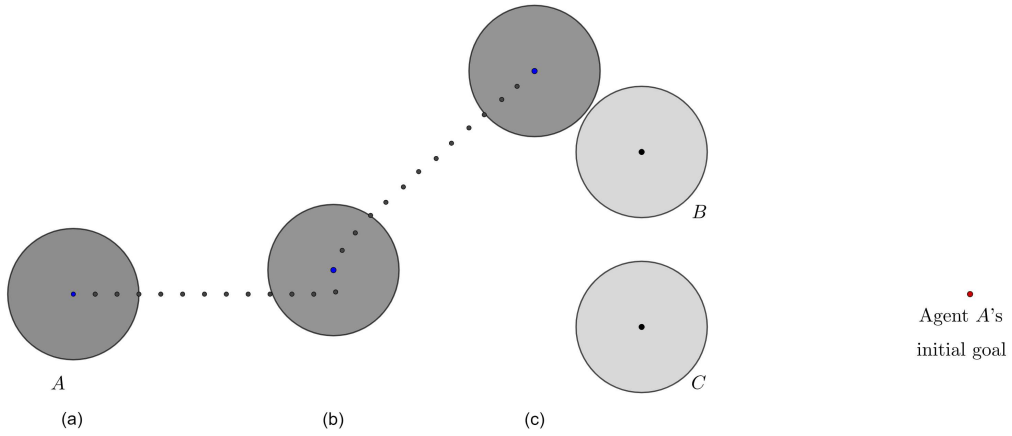

Fig. 6. Trace of agent A while moving from left to right using our modification of ORCA. Agents B and C, with zero speeds, are in the middle of a sidewalk in the direction of A's goal. The position of A is shown in times (in seconds) $t = 2$, $t = 3.3$, and $t = 4.5$. All agents have the same size ($r_A = r_B = r_C = 0.3m$), $s_A^{\max} = 1m/s$ and $\tau = 1s$.

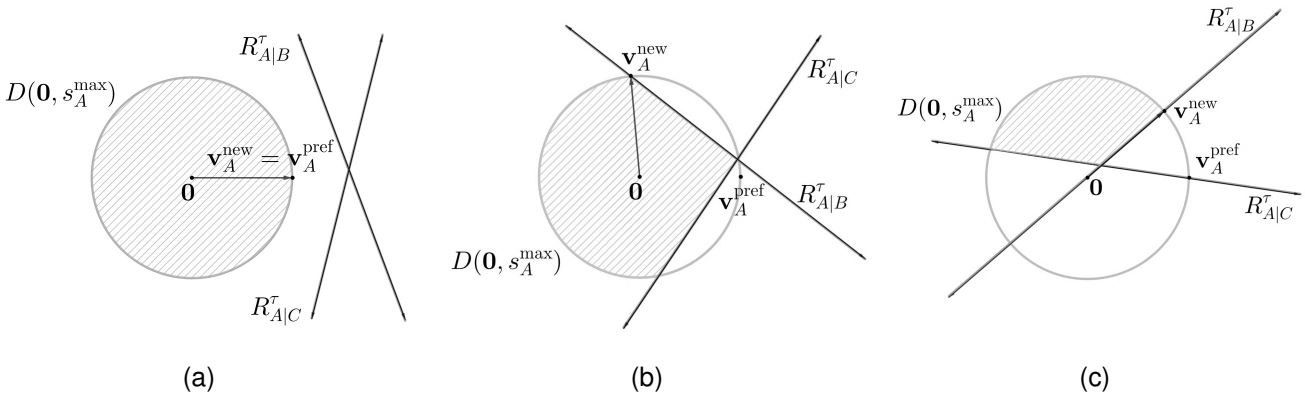

Fig. 7. Half-planes of permitted velocities using our modification of ORCA for agent A induced by agents B and C in the example shown in Fig. 6 at times (in seconds) (a) $t = 2$, (b) $t = 3.3$, and (c) $t = 4.5$

.

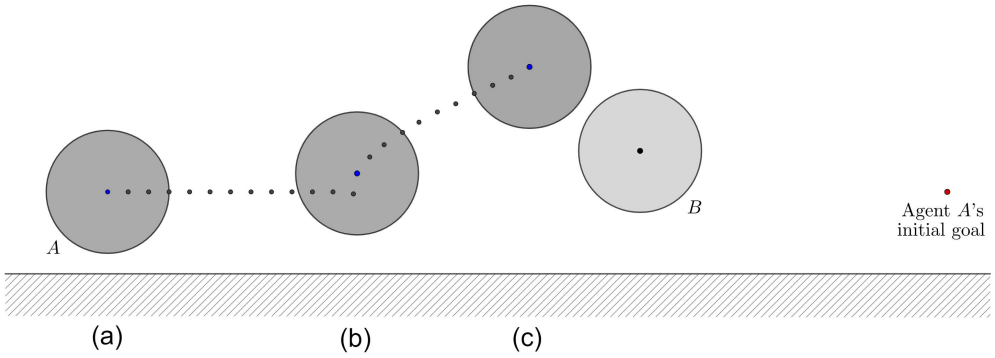

Fig. 8. Trace of agent A while moving from left to right near an edge of a sidewalk using our modification of ORCA. Agent B, with zero speeds, is in the direction of A's goal. The position of A is shown in seconds $t = 2$, $t = 3.2$, and $t = 4.3$. All agents have the same size ($r_A = r_B = r_C = 0.3m$), $s_A^{\max} = 1m/s$ and $\tau = 1s$.

allows them to escape from a livelock while continuing their course to their destination. There are several directions for future work. Further research should be undertaken to explore the selection of other velocities in the region of permitted velocities that are closer to the agent's preferred velocity, but that change the direction moderately. In addition, to prevent pedestrians from stopping suddenly, another perspective is to extend the banned points for very slow pedestrians. Also, it would be interesting to test our model in deadlock situations. Moreover, we would like to compare the output of our model with real data from crowds where pedestrians stop and later resume walking. Finally, our model can be used as a crowd

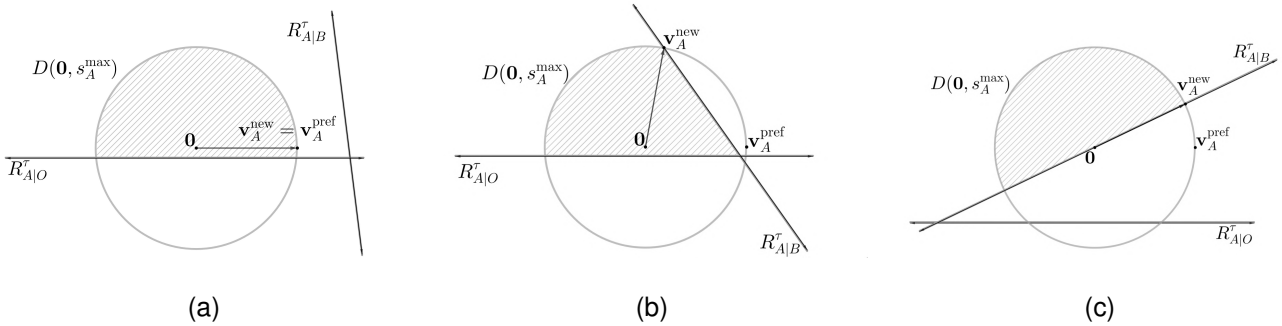

Fig. 9. Half-planes of permitted velocities using our modification of ORCA for agent A induced by agent B and an edge of a sidewalk in the example shown in Fig. 8 at times (in seconds) (a) $t = 2$, (b) $t = 3.2$, (c) $t = 4.3$.

management tool, especially for assessing the impact of zero-speed pedestrians.

REFERENCES

[1] W. van Toll and J. Pettré, "Algorithms for microscopic crowd simulation: Advancements in the 2010s," in *Computer Graphics Forum*, vol. 40, no. 2. Wiley Online Library, 2021, pp. 731–754.

[2] A. Schadschneider, H. Klüpfel, T. Kretz, C. Rogsch, and A. Seyfried, "Fundamentals of pedestrian and evacuation dynamics," in *Multi-Agent Systems for Traffic and Transportation Engineering*. IGI Global, 2009, pp. 124–154.

[3] D. C. Duives, W. Daamen, and S. P. Hoogendoorn, "State-of-the-art crowd motion simulation models," *Transportation Research Part C Emerging Technologies*, vol. 37, pp. 193–209, Dec 2013. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0968090X13000351

[4] C. T. Mathew, P. R. Knob, S. R. Musse, and D. G. Aliaga, "Urban walkability design using virtual population simulation," in *Computer Graphics Forum*, vol. 38, no. 1. Wiley Online Library, 2019, pp. 455–469.

[5] A. Bamaqa, M. Sedky, T. Bosakowski, B. Bakhtiari Bastaki, and N. O. Alshammari, "Simcd: Simulated crowd data for anomaly detection and prediction," *Expert Systems with Applications*, vol. 203, p. 117475, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417422008065

[6] J. E. Cutting, P. M. Vishton, and P. A. Braren, "How we avoid collisions with stationary and moving objects." *Psychological review*, vol. 102, no. 4, p. 627, 1995.

[7] J. Gehl, *Life between buildings*, 6th ed. Island Press, 2011.

[8] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, no. 5, pp. 4282–4286, May 1995. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.51.4282

[9] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha, *Reciprocal n-Body Collision Avoidance*, ser. Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg, 2011, vol. 70, ch. chapter 1, pp. 3–19. [Online]. Available: http://link.springer.com/10.1007/978-3-642-19457-3_1

[10] S. J. Guy, J. Chhugani, S. Curtis, P. Dubey, M. C. Lin, and D. Manocha, "Pledestrians: A least-effort approach to crowd simulation." in *Symposium on computer animation*, 2010, pp. 119–128.

[11] Y. Luo, P. Cai, A. Bera, D. Hsu, W. S. Lee, and D. Manocha, "Porca: Modeling and planning for autonomous driving among many pedestrians," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3418–3425, 2018.

[12] S. Dergachev, K. Yakovlev, and R. Prakapovich, "A combination of Theta*, ORCA and Push and Rotate for Multi-agent Navigation," in *Lecture Notes in Computer Science*. Springer International Publishing, 2020, pp. 55–66.

[13] D. Zhou, Z. Wang, S. Bandyopadhyay, and M. Schwager, "Fast, on-line collision avoidance for dynamic vehicles using buffered voronoi cells," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1047–1054, Apr 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7828016/

[14] B. Şenbaşlar, W. Hönig, and N. Ayanian, "Robust trajectory execution for multi-robot teams using distributed real-time replanning," in *Distributed Autonomous Robotic Systems*, N. Correll, M. Schwager, and M. Otte, Eds. Springer, 2019, pp. 167–181.

[15] S. Hosseini Semnani, A. H. J. de Ruiter, and H. H. T. Liu, "Force-based algorithm for motion planning of large agent," *IEEE Transactions on Cybernetics*, vol. 52, no. 1, pp. 654–665, Jan 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9108576/

[16] S. H. Arul and D. Manocha, "V-RVO: Decentralized Multi-Agent Collision Avoidance Using Voronoi Diagrams and Reciprocal Velocity Obstacles," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 8097–8104.

[17] M. Gérin-Lajoie, C. L. Richards, and B. J. McFadyen, "The negotiation of stationary and moving obstructions during walking: Anticipatory locomotor adaptations and preservation of personal space," *Motor Control*, vol. 9, no. 3, pp. 242–269, Jul 2005. [Online]. Available: http://journals.humankinetics.com/doi/10.1123/mcj.9.3.242

[18] S. Xue, F. Claudio, X. Shi, and T. Li, "Revealing the hidden rules of bidirectional pedestrian flow based on an improved floor field cellular automata model," *Simulation Modelling Practice and Theory*, vol. 100, p. 102044, Apr 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1569190X19301753

[19] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, 1987, pp. 25–34.

[20] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry, Algorithms and Applications*. Springer Berlin Heidelberg, 2008.

Laura C. Echeverri received a Ph.D. degree in Computer Science from the Université de Tours, France, in 2020. She is currently a postdoctoral researcher at the Laboratory "Ville, Mobilité, Transport" (LVMT), Université Gustave Eiffel, ENPC, France. Her research interests include human mobility and transportation network modeling and optimization.

Jean-Michel Auberlet conducts his research at the PICS-L laboratory of the Université Gustave Eiffel. He is mainly focused on both perception and interaction modeling of road users to develop computational models of autonomous agents. He is a member of the Pedestrian Committee of the Transportation Research Board. He is also involved in several international conference committees (PED, TGF, RSS, ICAART).

Jean-Paul Hubert conducts his research at the DEST laboratory of the Université Gustave Eiffel. His research focuses on mobility behavior and the dynamics of urban form, both from a quantitative and statistical point of view and from a qualitative analysis of the emergence of transport modes, including walking, as objects of public policy. He also worked at the French National Statistical Institute (Insee) from 2005 to 2009.

PLEASE KEEP CONFIDENTIAL

Captions

Fig. 1. Trace of agent A while moving from left to right using ORCA. Agents B and C, with zero speeds, are in the middle of a sidewalk in the direction of A's goal. The position of A is shown in times (in seconds) (a) $t = 2$, (b) $t = 3.3$, and (c) $t = 4.5$. All agents have the same size ($r_A = r_B = r_C = 0.3m$), $s_A^{\max} = 1m/s$ and $\tau = 1s$.

Fig. 2. Half-planes of permitted velocities using ORCA for agent A induced by agents B and C in the example shown in Fig. 1 at times (in seconds) (a) $t = 2$, (b) $t = 3.3$, and (c) $t = 4.5$.

Fig. 3. Trace of agent A while moving from left to right near an edge of a sidewalk using ORCA. Agent B, with zero speed, is in the direction of A's goal. The position of A is shown in times (in seconds) (a)$t = 2$, (b)$t = 3.2$, and (c)$t = 4.3$. All agents have the same size ($r_A = r_B = r_C = 0.3m$), $s_A^{\max} = 1m/s$ and $\tau = 1s$.

Fig. 4. Half-planes of permitted velocities using ORCA for agent A induced by agent B and an edge of a sidewalk in the example shown in Fig. 3 at times (in seconds) (a) $t = 2$, (b) $t = 3.2$, (c) $t = 4.3$.

Table I. Line types combinations related to different types of agents or obstacles

Algorithm 1 computeNewVelocity

Algorithm 2 LinProgMod

Fig. 5. Example of functioning of Algorithms 1 and 2. (a) $D(\mathbf{0}, s_A^{\max})$ bounds the feasible region. (b) $R^{\tau}_{A|B_1}$ is added, changing $\mathbf{v}_A^{\text{new}}$. (c) $R^{\tau}_{A|B_2}$ is added. Originally, $\mathbf{v}_A^{\text{new}} = \mathbf{v}^{-1}_{A|B_2}$ would be chosen, but $\mathbf{v}^{-1}_{A|B_2}$ is the intersection point of two "zero-speed agent" bounding lines. Thus $\mathbf{v}_A^{\text{new}} = \mathbf{v}^{1}_{A|B_2}$.

Fig. 6. Trace of agent A while moving from left to right using our modification of ORCA. Agents B and C, with zero speeds, are in the middle of a sidewalk in the direction of A's goal. The position of A is shown in times (in seconds) $t = 2$, $t = 3.3$, and $t = 4.5$. All agents have the same size ($r_A = r_B = r_C = 0.3m$), $s_A^{\max} = 1m/s$ and $\tau = 1s$.

Fig. 7. Half-planes of permitted velocities using our modification of ORCA for agent A induced by agents B and C in the example shown in Fig. 6 at times

PLEASE KEEP CONFIDENTIAL

(in seconds) (a) $t = 2$, (b) $t = 3.3$, and (c) $t = 4.5$.

Fig. 8. Trace of agent A while moving from left to right near an edge of a sidewalk using our modification of ORCA. Agent B, with zero speeds, is in the direction of A's goal. The position of A is shown in seconds $t = 2$, $t = 3.2$, and $t = 4.3$. All agents have the same size ($r_A = r_B = r_C = 0.3m$), $s_A^{\max} = 1m/s$ and $\tau = 1s$.

Fig. 9. Half-planes of permitted velocities using our modification of ORCA for agent A induced by agent B and an edge of a sidewalk in the example shown in Fig. 8 at times (in seconds) (a) $t = 2$, (b) $t = 3.2$, (c) $t = 4.3$.

PLEASE KEEP CONFIDENTIAL

AUTHOR 1:
Laura C. Echeverri
Univ Gustave Eiffel, Ecole des Ponts, LVMT, F-77454 Marne-la-Vallée, France
6-8 Avenue Blaise Pascal 77420 Champs-sur-Marne, France
laura.echeverri-guzman@enpc.fr
(Corresponding author)

AUTHOR 2:
Jean-Michel Auberlet
Univ Gustave Eiffel, COSYS-PICS-L, F-77454 Marne - la- Vallée, France
6-8 Avenue Blaise Pascal 77420 Champs-sur-Marne, France
jean-michel.auberlet@univ-eiffel.fr

AUTHOR 3:
Jean-Paul Hubert
Univ Gustave Eiffel, AME-DEST, F-77454 Marne - la- Vallée, France
6-8 Avenue Blaise Pascal 77420 Champs-sur-Marne, France
jean-paul.hubert@univ-eiffel.fr