# The Emergency Braking Game: a game theoretic approach for maneuvering in a dense crowd of pedestrians <br> Check for updates 

János Szőts ${ }^{1 *}$ © ${ }^{\text {© }}$, Zoltán Gyenes ${ }^{1,2}$, Emese Gincsainé Szádeczky-Kardoss ${ }^{1}$, Ladislau Bölöni ${ }^{2}$ and István Harmati ${ }^{1}$


#### Abstract

We introduce an algorithm that maneuvers a vehicle through an area with randomly moving pedestrians. In non-critical situations, our strategy is to avoid pedestrians by steering, whereas dangerously moving pedestrians are avoided by braking, possibly coming to a complete stop. The distinction between non-critical and dangerous situations, as well as proof of safety, is based on a continuous optimization problem that we define. In this abstract problem, called Emergency Braking Game, one pedestrian is actively trying to collide with a continuously decelerating car. We show how to determine the outcome of the game based on the initial states of the car and the pedestrian. Using this information, our algorithm can initiate deceleration in the real scenario in time to avoid collision. The method's safety is proven theoretically, and its efficiency is shown in simulations with randomly moving pedestrians.


Keywords Collision avoidance, Intelligent control, Game theory, Optimization

## Introduction

The development of self-driving cars, which must operate in the presence of pedestrians, brought an increasing focus to the problem of collision-free autonomous movement [1-3]. To ensure guaranteed safety, it is often convenient to assume that the pedestrians are adversarial, meaning that they seek to collide with the vehicle. While this is, obviously, a worst-case assumption, dense urban intersections like the famous Shibuya Crossing (Fig. 1) or cars leaving parking lots after major sport events create conditions that closely approximate it. In the adversarial scenario, collision avoidance can be formulated as a pur-suit-evasion differential game [4].

[^0]In this paper, we introduce the Emergency Braking Game (EBG) and a novel game-theoretic formulation of a strategy for a vehicle avoiding multiple pedestrians. Inspired by the suicidal pedestrian game [5], EBG proposes that the primary action available to the vehicle to avoid a collision is to decrease its velocity (brake) while retaining its direction of movement. The primary advantage of this strategy is that the actions calculated for the avoidance of different pedestrians can be combined in a convenient way. In contrast, when the vehicle uses swerving as an avoidance strategy [6], the algorithm might recommend incompatible turning directions for the avoidance of different pedestrians.
As in several other techniques that consider an adversarial relationship between the vehicle and pedestrians [7, 8], we divide the scenario into one-on-one games. In contrast to several previous studies, where pedestrians were assumed to adhere to predetermined routes or utilized crosswalks for traversing [9-11], we consider a scenario where the pedestrians can have arbitrary trajectories. The objective of the game is to guarantee safety by
avoiding collision, with the pedestrians' secondary objective being to minimize the miss distances.
The primary contributions of this paper can be summarized as follows:

- We define the Emergency Braking Game (EBG) as a model of a particular solution class for the problem of a vehicle navigating a dense pedestrian area. We provide an analytic solution to the problem under specific assumptions.
- We build a collision avoidance strategy based on EBG.
- We study the behavior generated by the EBG-based collision avoidance in a simulation scenario with randomly moving pedestrians and compare its performance with two recent collision avoidance algorithms.

The remainder of the paper is organized as follows. Section 2 discusses related work in the domain of reactive motion planning algorithms. In Sect. 3, we present the kinematics of the agents present in the game. Later on, in Sect. 4, we define the Emergency Braking Game and describe an analytic solution for the simple case. We describe an evasion algorithm built on this solution in Sect. 5. In Sect. 6 we describe the baseline algorithms, with Sect. 7 containing simulation results and analysis. We provide conclusions and describe future work in Sect. 8.

## Related work

Techniques for collision avoidance by a vehicle in the context of dynamic obstacles have been the subject of several research projects in recent decades [13-16]. The difficulty of this task is largely determined by the amount of knowledge we have about the obstacle movement.


Fig. 1 Shibuya Crossing is one of the busiest pedestrian crossings in the world [12]

The majority of algorithms [17] assume that the obstacles' paths are known, or at least well predicted. This is also the assumption underlying one of the most frequently used dynamic motion planning algorithm families, Velocity Obstacles (VO) [18, 19]; as well as other methods such as the one proposed in [20]. VO has been adapted to various scenarios, such as differential-driven robots [21, 22], ships [23], multi-robot collision avoidance [24-26], and unmanned aerial vehicles [27].

A further assumption that can be made is that the obstacles are holonomic, this being a good approximation for the case of human pedestrians. With this assumption, we can define the reachable set of the obstacles. In practice this can be done by expanding the future positions of the vehicle with the distance the obstacles can cover in the corresponding time interval. This creates a dynamic envelope [28] with the following interpretation: if the dynamic obstacle is initially present inside this envelope, it is possible for it to collide with the vehicle on its predefined path. In contrast to this forward approach, we can also work backward from collision states and define dangerous sets that need to be avoided by the vehicle [29]. The backward approach has the advantage that it is often less computationally demanding.
Another popular method, the Dynamic Window Approach (DWA), can quickly generate solutions in realtime and is capable of taking nonholonomic constraints into account, such as limited turning ability [30, 31]. The DWA algorithm can also be used in dynamic environments to produce collision-free paths. Recent work applied DWA to motion planning for the Forklift Automated Guided Vehicle [32].
In the techniques above, the motion planning algorithm explicitly considers the velocities of the obstacles. It is possible to develop path planning algorithms that only rely on the changing locations of the obstacles, without explicitly requiring the velocity vectors. An example of this class of methods is that of Artificial Potential Fields [33-36] (APF), where obstacles exert a virtual repelling force on the vehicle while the goal exerts an attractive force. The movement is then defined by the resultant of these forces. The method is straightforward to implement, has a low computational demand, and can be effective against slow-moving obstacles. Although APF was originally designed for use in static environments, it has also been adapted to handle dynamic obstacles, such as those encountered in robot soccer [37].

## Pedestrian avoidance dynamics

Let us now consider the problem of obstacle avoidance in the setting of a vehicle aiming to traverse an area where a number of pedestrians are moving unpredictably.

We assume that a pedestrian ( P ) implements a simple holonomic motion, described by the equations

$$
\begin{align*}
& x_{P}(0)=x_{P, 0} \\
& y_{P}(0)=y_{P, 0} \\
& \dot{x}_{P}(t)=w \cos \phi(t)  \tag{1}\\
& \dot{y}_{P}(t)=w \sin \phi(t)
\end{align*}
$$

where $x$ and $y$ are planar coordinates, the 0 index refers to initial values, and $w$ is the (constant) speed of the pedestrian. The control input is the movement direction $\phi$ which the pedestrian can alter instantaneously.
The vehicle ( E ) is described by the state variables of the planar coordinates $x, y$, the heading $\psi$, and velocity $v$. The state variables are subject to the constraints of a minimal turning radius $R$ and a maximal acceleration/ deceleration $a_{\text {max }}$ :

$$
\begin{align*}
x_{E}(0) & =x_{E, 0} \\
y_{E}(0) & =y_{E, 0} \\
\psi(0) & =\psi_{0} \\
v(0) & =v_{0} \\
\dot{x}_{E}(t) & =v(t) \cos \psi(t) \\
\dot{y}_{E}(t) & =v(t) \sin \psi(t) \\
\dot{\psi}(t) & =u_{\psi}(t) \frac{v(t)}{R} \\
\dot{v}(t) & =\left\{\begin{array}{lll}
0 & \text { if } & v(t)=0 \quad \text { and } \quad u_{v}(t)<0 \\
0 & \text { if } & v(t)=v_{\max } \text { and } \quad u_{v}(t)>0 \\
u_{v}(t) a_{\max } & \text { otherwise }
\end{array}\right. \tag{2}
\end{align*}
$$

where the control inputs are $u_{\psi}, u_{v} \in[-1,1]$. The velocity of the vehicle is constrained between 0 and $v_{\text {max }}$.
In case we only need the relative position of a pedestrian and the vehicle, it is useful and customary $[4,5]$ to transform their coordinates to a reduced space. In the reduced space, the vehicle permanently resides in the origin, facing the positive y -axis, as shown in Fig. 2. Then, the full state is described by the relative $x$ and $y$ coordinates-without the $P$ subscript-of the pedestrian and the velocity $v$. The transformation can be given by

$$
\left[\begin{array}{l}
x \\
y
\end{array}\right]=\mathcal{R}(\pi / 2-\psi)\left(\left[\begin{array}{l}
x_{P} \\
y_{P}
\end{array}\right]-\left[\begin{array}{l}
x_{E} \\
y_{E}
\end{array}\right]\right)
$$

with the usual rotation matrix

$$
\mathcal{R}(\bullet)=\left[\begin{array}{cc}
\cos (\bullet) & -\sin (\bullet) \\
\sin (\bullet) & \cos (\bullet)
\end{array}\right]
$$

The initial values in the reduced space are


Fig. 2 Relation of the natural and reduced spaces. The heading of the vehicle coincides with the relative ' $y$ ' axis. The pedestrian's movement direction is explicitly shown

$$
\begin{aligned}
{\left[\begin{array}{l}
x(0) \\
y(0)
\end{array}\right] } & =\left[\begin{array}{l}
x_{0} \\
y_{0}
\end{array}\right]=\mathcal{R}\left(\pi / 2-\psi_{0}\right)\left(\left[\begin{array}{l}
x_{P, 0} \\
y_{P, 0}
\end{array}\right]-\left[\begin{array}{l}
x_{E, 0} \\
y_{E, 0}
\end{array}\right]\right) \\
v(0) & =v_{0}
\end{aligned}
$$

and the relative dynamics-similarly to [5], but with the sign of $u_{\psi}$ oppositely defined-are

$$
\begin{aligned}
& \dot{x}(t)=w \cos \tilde{\phi}(t)+u_{\psi}(t) \frac{v(t)}{R} y(t) \\
& \dot{y}(t)=w \sin \tilde{\phi}(t)-u_{\psi}(t) \frac{v(t)}{R} x(t)-v(t) \\
& \dot{v}(t)=u_{v}(t) a_{\max }
\end{aligned}
$$

where $\tilde{\phi}$ is understood in the rotated frame, hence $\phi+\pi / 2-\psi=\tilde{\phi}$. Since the pedestrian is holonomic, this rotation doesn't affect its capabilities, and thus, in the remainder of this paper, the tilde will be omitted.

## The Emergency Braking Game

The Emergency Braking Game is an abstract formulation of a continuous-time optimization problem with one player, closely related to the Suicidal Pedestrian Problem [5]. Let's consider a pedestrian and a vehicle described in the natural space. The vehicle performs the following movement: it starts from the origin, moves along the $y$-axis, and slows down from its initial speed of $v_{\max }$ to a complete stop with a deceleration of $-a_{\max }$. The coordinates of the vehicle will change as follows:

$$
\begin{aligned}
x_{E}(\tau) & =0 \\
y_{E}(\tau) & =\frac{a_{\max }}{2} T^{2}-\frac{a_{\max }}{2}(T-\tau)^{2}=S-\frac{a_{\max }}{2}(T-\tau)^{2} \\
\tau & \in[0, T]
\end{aligned}
$$

where $T=v_{\max } / a_{\max }$ is the duration of the process. The distance covered is denoted as $S$. The time in this abstract program is represented by $\tau$ to differentiate it from the real-world time $t$.

The player of the game is a "suicidal pedestrian", whose motion is described in Eq. (1). The aim of the pedestrian is to collide with the vehicle as soon as possible. If the pedestrian cannot reach the vehicle before it comes to a stop, it aims to approach the still-moving car as close as possible. If it finds the optimal control for the scenario, $\phi^{*}(\cdot)$, the pedestrian approaches the moving vehicle at a distance $V$, as described by

$$
V=\min _{\phi(\cdot)} \min _{\tau \in[0, T]} \sqrt{x_{P}(\tau)^{2}+\left(y_{P}(\tau)-y_{E}(\tau)\right)^{2}}
$$

If this value is greater than the radius of the circle describing the size of the vehicle, the vehicle will stop before hitting the pedestrian, even when the pedestrian applies optimal minimizing control.

## EBG cases

For every possible initial position of the pedestrian, we want to provide the eventual miss distance $V$ and the optimal controls. First, we display $V$ as a function over the $x, y$ coordinates of the pursuer by plotting contour lines with equal $V$ value.
To do this, we draw a circle with radius $V+w \tau$ around each point along the car's path. In practice, we do this for a sample of these points. Starting from a point on such a circle, the holonomic pedestrian can get to the distance $V$ from the vehicle at time $\tau$. The hull of these circles contains the locations from which the pedestrian can get to the distance $V$ at some point in time. Such a hull is similar to the dynamic envelope described in [28].

Figures 3, 4 and 5, show the hull and its generating circles for $V=0$, as well as only the hulls for other $V$


Fig. 3 Contour lines of the optimal $V$ function of the EBG. Parameters: $v_{\max }=5 \mathrm{~m} / \mathrm{s}, w=2 \mathrm{~m} / \mathrm{s}, a_{\max }=2 \mathrm{~m} / \mathrm{s}^{2}$

$\mathrm{V}=0$
$\mathrm{~V}=0.1$
$\mathrm{~V}=0.2$
$\mathrm{~V}=0.3$
$\mathrm{~V}=0.4$
$\mathrm{~V}=0.5$
$\mathrm{~V}=0.6$
$\mathrm{~V}=0.7$
$\mathrm{~V}=0.8$
$\mathrm{~V}=0.9$
$\mathrm{~V}=1$

Fig. 4 Contour lines of the optimal $V$ function of the EBG. Parameters: $v_{\text {max }}=5 \mathrm{~m} / \mathrm{s}, w=3 \mathrm{~m} / \mathrm{s}, a_{\text {max }}=2 \mathrm{~m} / \mathrm{s}^{2}$
values. The three subfigures were drawn with three different speed parameters. The red line marks the vehicle's path, starting from the triangle, around whose points the circles are drawn.
The figures show that in some cases (Figs. 4 and 5), the hull is a circle, significantly simplifying the solution in the case when the collision is not possible. Optimizing for a collision speed remains complicated for the case with collision. Figure 5 shows a borderline example where the hull is a circle. In Fig. 3, where we have a lower pedestrian speed, the hull has a more complex shape. The necessary and sufficient condition for this simplification is that the circle with radius $w T$ for $V=0$ drawn around the $[0, S]^{T}$ end-position of the vehicle contains the origin:

$$
w T \geq S=\frac{v_{\max } T}{2}
$$

from where


Fig. 5 Contour lines of the optimal $V$ function of the EBG. Parameters: $v_{\text {max }}=5 \mathrm{~m} / \mathrm{s}, \mathrm{w}=2.5 \mathrm{~m} / \mathrm{s}, a_{\text {max }}=2 \mathrm{~m} / \mathrm{s}^{2}$

$$
\begin{equation*}
2 w \geq v_{\max } \tag{3}
\end{equation*}
$$

An equivalent condition is that the distance covered by the vehicle takes less time for the pedestrian to run: $\frac{S}{w} \leq T$. Also, in this case, the optimal control of the pedestrian is to aim for the final position of the vehicle. In the remainder of this discussion, we will assume the velocity of the pedestrian to be at least $v_{\max } / 2$. If the pedestrian is slower, this approximation is conservative and, therefore, does not violate safety regarding collision avoidance.

## The solution to the simple case

If the speed constraint of Eq. (3) is satisfied, the solution for the optimization task involves the pedestrian aiming for the final coordinates of the vehicle $[0, S]^{T}$, with the miss distance being:

$$
V(x, y, v)=\sqrt{x^{2}+(y-S(v))^{2}}-w T(v)
$$

This value depends on the initial relative coordinates of the pedestrian ( $x$ and $y$ ) and the vehicle's initial velocity $v$. This formula remains valid throughout the game, using the remaining time and distance dependent on the current speed: $T(v), S(v)$.
To develop the evasion algorithm, we need a closedform expression of the gradient of the function $V$ in an arbitrary state. Let us introduce the expression

$$
d(x, y, v)=\sqrt{x^{2}+(y-S(v))^{2}}
$$

Then, after elementary operations

$$
\begin{aligned}
& \frac{\partial V}{\partial x}=\frac{x}{d} \\
& \frac{\partial V}{\partial y}=\frac{y-S}{d} \\
& \frac{\partial V}{\partial S}=\frac{S-y}{d} \\
& \frac{\partial V}{\partial T}=-w
\end{aligned}
$$

Using that $T(v)=v / a_{\max }$ and $S(v)=v^{2} / 2 a_{\max }$, we can write

$$
\begin{aligned}
\frac{d T}{d v} & =\frac{1}{a_{\max }} \\
\frac{d S}{d v} & =\frac{v}{a_{\max }}
\end{aligned}
$$

from where

$$
\begin{aligned}
\frac{\partial V(x, y, S(v), T(v))}{\partial v} & =\frac{\partial V}{\partial S} \frac{d S}{d v}+\frac{\partial V}{\partial T} \frac{d T}{d v} \\
& =\frac{S-y}{d} \frac{v}{a_{\max }}-\frac{w}{a_{\max }}
\end{aligned}
$$

## Evasion strategy

In this section, we develop an evasion strategy for the vehicle based on the EBG. The following theorem makes the connection between the abstract optimization problem, the EBG, and the maneuvering task. In particular, the solution and $V$ function of the EBG can be used to guarantee safety in the main scenario.

Theorem 1 If, at a time instant t, for all of the (pointlike) pedestrians' reduced coordinates and the speed of the (pointlike) vehicle, it is true that $V(x, y, v)>0$, then the vehicle is guaranteed to not collide with any pedestrians while coming to a full stop in a straight line decelerating with $a_{\text {max }}$.

## Proof

The origin of the reduced coordinate system is positioned to coincide with the vehicle, and the vehicle momentarily travels on the positive $y$-axis. Fixing the frame now to this point instead of the vehicle, the stopping maneuver is identical to that in the EBG. For any pedestrian, if initially $V(x, y, v)>0$, then the pedestrian cannot reach the vehicle before it stopped. As the movement of the vehicle doesn't depend on the pedestrians, if the condition $V(x, y, v)>0$ holds for all of them, none can reach the vehicle before it has stopped.

## Remark 1

In reality, the agents are not pointlike. Their extent can be incorporated into Theorem 1 by requiring the value to be greater than the sum of the radii of the circumscribed circles: $V(x, y, v)>r_{\text {car }}+r_{\text {pedestrian. Then, the midpoint of a }}$ pedestrian's circle cannot come closer to the midpoint of the vehicle's circle than this value; hence the bodies cannot touch.

## Remark 2

Collision avoidance is guaranteed only while the vehicle is moving. Pedestrians may bump into the stopped vehicle.

The key point of our evasion algorithm is to keep all $V_{i}$ values corresponding to $N$ pedestrians $P_{i}$ above a safety level $V_{\text {safe }}$. Unless there is a critical situation, the lowest $V_{i}$ value is increased through turning while
keeping the maximal velocity of the vehicle. If any of the $V_{i}$ values fall below the safety threshold, the vehicle brakes; this is the critical case. Deceleration is always implemented in a straight line (for the acceleration, this is not always true). From the previous results, it follows that when decelerating with the maximal allowed $-a_{\text {max }}$ value, none of the $V_{i}$ values can decrease. In the worst case, when the critical pedestrian is indeed trying to collide, its corresponding $V_{i}$ is constant.

## Change of the EBG values

Starting from the variables $x, y, v$ of the reduced state of the realistic problem, the change of the $V$ value with respect to the control inputs of the vehicle can be written as follows:

$$
\begin{aligned}
\dot{V}(x, y, v) & =\frac{\partial V}{\partial x} \dot{x}+\frac{\partial V}{\partial y} \dot{y}+\frac{\partial V}{\partial v} \dot{v} \\
& =\frac{x}{d}\left(w \cos \tilde{\phi}+u_{\psi} \frac{v}{R} y\right) \\
& +\frac{y-S}{d}\left(w \sin \tilde{\phi}-u_{\psi} \frac{v}{R} x-v\right) \\
& +\left(\frac{S-y}{d} \frac{v}{a_{\max }}-\frac{w}{a_{\max }}\right) u_{v} a_{\max }
\end{aligned}
$$

which can be divided into parts controlled by the vehicle and parts that are not influenced by it:

$$
\begin{aligned}
\dot{V}(x, y, v) & =\dot{V}\left(x, y, v \mid u_{\psi}\right)+\dot{V}\left(x, y, v \mid u_{v}\right) \\
& +\dot{V}(x, y, v \mid-)
\end{aligned}
$$

Using Isaacs' circular lemma [4], we find that the "optimal" (minimizing) movement direction of the pedestrian is antiparallel to the $w[x / d,(y-S) / d]^{T}$ vector of the terms corresponding to the pedestrian. Given that $[x / d,(y-S) / d]^{T}$ is a unit vector, the pedestrian's effect on $\dot{V}$ is $-w$, which makes the vehicle-independent part

$$
\dot{V}(x, y, v \mid-)=\frac{S-y}{d} v-w
$$

The vehicle's influence is

$$
\begin{aligned}
\dot{V}\left(x, y, v \mid u_{\psi}\right) & =\frac{x y-x y+S x}{d} u_{\psi} \frac{v}{R}=\frac{S x v}{R d} u_{\psi} \\
\dot{V}\left(x, y, v \mid u_{v}\right) & =\left(\frac{S-y}{d} v-w\right) u_{v}
\end{aligned}
$$

It is noteworthy that $\dot{V}=0$ when $u_{\psi}=0$ and $u_{v}=-1$. This aligns with our expectations: when the vehicle is braking in a straight line, and the pedestrian is attempting a collision, the expected miss distance in the EBG remains unchanged, as they are exactly playing the EBG.

## Algorithm

Up to this point, every expression was formulated in continuous time. However, in practice, the algorithm operates in discrete time. We discretize using the Euler method, given knowledge of the sampling time $\Delta t$.

Let us write the actual $V$ as

$$
V_{i}=\sqrt{x_{i}^{2}+\left(y_{i}-S\right)^{2}}-w T
$$

and subsequent $V^{\prime}$ values of the emergency braking games of $P_{i}(i \in\{1, \ldots N\})$, dependent on the control inputs:

$$
\begin{align*}
& V_{i}^{\prime}\left(u_{\psi}, u_{v}\right) \approx V_{i}+\Delta t\left(\frac{S-y_{i}}{d_{i}} v-w\right) \\
& \quad+\Delta t \frac{S x_{i} v}{R d_{i}} u_{\psi}+\Delta t\left(\frac{S-y_{i}}{d_{i}} v-w\right) u_{v} \tag{4}
\end{align*}
$$

The error of the Euler method, due to the changing gradient of the value, is neglected from hereon.
The vehicle tries to travel as fast as possible. The acceleration it can apply is constrained by the maximal acceleration and it may not increase speed above the allowed maximum in the the next sampling interval:

$$
\begin{equation*}
u_{v, \max }=\min \left(1, \frac{v_{\max }-v}{a_{\max } \Delta t}\right) \tag{5}
\end{equation*}
$$

Applying this control is only permissible if retaining the maximal speed or accelerating to it will not cause any of the $N$ game outcomes below the safety level threshold $V_{\text {safe }}$. We apply this condition without considering steering, hence we predict the $V$ values of the next cycle by substituting $u_{\psi}=0$ and $u_{v}=u_{v, \max }$ into Eq. (4), denoting the index of the considered pursuer with $i$ and the acceleration to the maximal speed with the "acc" subscript.

$$
\begin{equation*}
V_{i, a c c}^{\prime}=V_{i}+\Delta t\left(\frac{S-y_{i}}{d_{i}} v-w\right)\left(1+u_{v, \max }\right) \tag{6}
\end{equation*}
$$

There are two possible scenarios: if, for all $i \in\{1, \ldots N\}$, $V_{i, a c c}^{\prime}>V_{\text {safe }}$, the vehicle accelerates to the maximum speed and evades the most critical pedestrian by turning. However, if this condition fails for any of the pedestrians, the vehicle proceeds straight and typically decelerates. As the value $V_{i, a c c}^{\prime}$ is calculated with the maximum possible acceleration, a lesser acceleration may be permitted even in the critical case.

## Critical case

In any situation where an EBG's $V_{i}$ value falls below $V_{\text {safe }}$, the vehicle, adhering to our strategy, will continue in a straight line and either decelerate or accelerate less intensely than the maximum.

From Eq. (6), we can derive the allowable acceleration by substituting the $V$ value from the next cycle with the safety level. The prescribed acceleration coefficient then becomes

$$
\begin{align*}
\tilde{u}_{v} & =\min _{i \in\{1, \ldots . N\}}\left(\frac{V_{\text {safe }}-V_{i}}{\Delta t\left(\frac{S-y_{i}}{d_{i}} v-w\right)}-1\right)  \tag{7}\\
u_{v} & =\max \left(\tilde{u}_{v},-1\right)
\end{align*}
$$

This also includes the lower constraint, in cases when numerical effects cause $V_{\text {safe }}<V_{i}$ and thus $\tilde{u}_{v}<-1$.

If multiple pedestrians are in a critical situation, the above equations select the smallest of the acceleration coefficients, implying the most intensive deceleration. This is safe in all scenarios since, if $\left(\frac{S-y_{i}}{d_{i}} v-w\right)<0$, a stronger deceleration increases $V_{i}^{\prime}$, and if $\left(\frac{S-y_{i}}{d_{i}} v-w\right)>0$, then $V_{i}^{\prime}>V_{i}$ because of Eq. (6). Note that even though the smallest $V_{i}$ would approach $V_{\text {safe }}$ based on the previous equations, in practice, $V_{i}$ will increase when the pedestrian is not attempting to collide.
In the critical case, the vehicle does not turn: $u_{\psi}=0$.

## Non-critical case

If no pedestrians are in a critical situation, the vehicle accelerates to the maximum velocity and increases the smallest $V_{i, a c c}^{\prime}$ by turning. Hence, we need to choose $u_{\psi}$ such that $\min _{i} V_{i}^{\prime}$ is maximal. This can be formulated as a linear program using the auxiliary variable $\xi$ :

$$
\begin{align*}
\max _{u_{\psi}, \xi} & \xi \\
\text { s.t. } \quad \xi & \leq V_{i, a c c}^{\prime}+\Delta t \frac{S x_{i} v}{R d_{i}} u_{\psi} \quad \forall i \in\{1, \ldots N\}  \tag{8}\\
- & \leq u_{\psi} \leq 1
\end{align*}
$$

This linear program always has a solution since the constraints of $u_{\psi}$ are trivial, and $\xi$ is only constrained from above. Hence it always provides the value for $u_{\psi}$. While the linear program does not provide a solution for $u_{\psi}$ if $v=0$, in practice, this is not a problem, because the steering of a standing vehicle does not influence the dynamics.

In the non-critical case, the acceleration coefficient is $u_{v}=u_{v, \text { max }}$ as defined in Eq. (5).

## Incorporating the target

Using the previously defined rules and equations, the vehicle can maintain a safe distance from the pedestrians or stop if that is not possible. However, the vehicle also need to progress towards its target location. We incorporate this goal into the algorithm in the following way. In this subsection, we will work in the natural space coordinates.

The prescribed movement direction is defined as the direction from the vehicle E to the goal G . Any deviation from this direction is denoted by

$$
\Delta \psi=\angle \overrightarrow{E G}-\psi \quad \Delta \psi \in[-\pi, \pi)
$$

This deviation is scaled to represent a distance that can be compared to the miss distances of the EBG. We introduce two new terms that need to be maximized:

$$
\begin{align*}
& V_{N+1}=\bar{V}-\frac{\Delta \psi}{\pi}(\bar{V}-\underline{V}) \\
& V_{N+2}=\bar{V}+\frac{\Delta \psi}{\pi}(\bar{V}-\underline{V}) \tag{9}
\end{align*}
$$

We note that the minimum of these terms is maximal if $\Delta \psi=0$ and $V_{N+1}=V_{N+2}=\bar{V}$. The parameters $\bar{V}>\underline{V}$ define the relation to the $V_{i}$ values of the previous sections. Above a miss distance $\bar{V}$ the importance of the pedestrians is negligible compared to keeping the direction, as $\min \left(V_{N+1}, V_{N+2}\right) \leq \bar{V}$. On the other hand, if, for some pedestrian $P_{i}, V_{i}<\underline{V}$, the movement direction is of negligible interest compared to evasion from $P_{i}$, as $\min \left(V_{N+1}, V_{N+2}\right) \geq \underline{V}$.

While technically the direction could be controlled by a single minimizing term describing the absolute value of the deviation, this could lead to oscillation if the deviation changed in each computational cycle. Maximizing the minimum of the two terms alleviates this oscillation by reaching the prescribed direction exactly in one step. The two linear terms also fit well into the previously defined linear programming scheme.
The vehicle-dependent part of the temporal change of $V_{N+1}$, using Eq. (2) and (9), is

$$
\frac{\partial V_{N+1}}{\partial \psi} \dot{\psi}=\frac{\bar{V}-\underline{V}}{\pi} \frac{v}{R} u_{\psi}
$$

and analogically for $V_{N+2}$, with a negative sign. In summary, for the non-critical case, we extend the linear program of Eq. (8) as follows:

$$
\begin{align*}
\max _{u_{\psi}, \xi} & \xi \\
\text { s.t. } & \xi \leq V_{i, a c c}^{\prime}+\Delta t \frac{S x_{i} v}{R d_{i}} u_{\psi} \quad \forall i \in\{1, \ldots N\} \\
\xi & \leq V_{N+1}+\Delta t \frac{(\bar{V}-\underline{V}) v}{\pi R} u_{\psi}  \tag{10}\\
\xi & \leq V_{N+2}-\Delta t \frac{(\bar{V}-\underline{V}) v}{\pi R} u_{\psi} \\
-1 & \leq u_{\psi} \leq 1
\end{align*}
$$

The extended linear program always has a feasible solution, for the same reason as Eq. (8). Note that the
movement direction is neglected in the critical case. The appropriate choice for the parameters is $\bar{V}>\underline{V} \geq V_{\text {safe }}$.

## Summary of the algorithm

Let us summarize our algorithm in pseudo code. Inputs are the current velocity of the evader agent and the relative positions of the pursuers. Output is the two control variables of the evader.
where $\overrightarrow{E G}$ is the line connecting the vehicle's position with the goal, and $\gamma_{t}$ is the weight (importance) of heading towards the goal.
The repulsive forces exerted by the $N$ pedestrians are

$$
F_{i}=\overrightarrow{P_{i} E} e^{-\left(\frac{\left\|\overrightarrow{P_{i} E}\right\|}{\rho}\right)^{2}} \quad i \in\{1, \ldots N\}
$$

Algorithm 1 EBG-based maneuvering algorithm

```
procedure Compute evader controls
    \(u_{v, \text { max }} \leftarrow\) based on (5)
    for \(i \leftarrow 1, N\) do
        \(V_{i, a c c}^{\prime} \leftarrow\) based on (6)
    end for
    if \(\exists V_{i, a c c}^{\prime}<V_{\text {safe }}\) then \(\triangleright\) Critical case
        \(u_{v} \leftarrow\) based on (7)
        \(u_{\psi} \leftarrow 0\)
    else \(\triangleright\) Non-critical case
        \(u_{v} \leftarrow u_{v, \max }\)
        \(u_{\psi} \leftarrow\) solution of linear program (10)
    end if
end procedure
```


## Two baseline algorithms

As we discussed in the related work section, the research literature contains a large number of different path planning algorithms that consider mobile obstacle avoidance as part of their specification. In order to evaluate the performance characteristics of the EBG algorithm, we briefly describe two recent approaches that will be used as comparative baselines: Gaussian artificial potential fields and safety velocity obstacles.

## Baseline 1: Gaussian artificial potential fields

As our first baseline, we implemented a generic Gaussian artificial potential field method that controls the direction as well as the speed of the vehicle. It applies the common considerations of all APF approaches [33-36].
The APF algorithm has two main steps: first, it sums the attractive force of the goal and the repulsive forces of the pursuers and then calculates the acceleration and turn coefficients from the resultant force. The attractive force exerted by the goal is

$$
F_{\text {target }}=\frac{\overrightarrow{E G}}{\|\overrightarrow{E G}\|} \gamma_{t}
$$

with $\overrightarrow{P_{i} E}$ connecting the positions of the pedestrians and the vehicle, and $\rho$ serving as a tuning parameter.
The resultant force is

$$
F=F_{\text {target }}+\sum_{i=1}^{N} F_{i},
$$

the direction of which is the prescribed movement direction of the vehicle. This direction can either be achieved in one sampling interval, or the vehicle can implement the sharpest possible turn:

$$
\begin{aligned}
& \tilde{u}_{\psi}=\frac{(\angle F-\psi) R}{v \Delta t} \\
& u_{\psi}=\min \left(\max \left(-1, \tilde{u}_{\psi}\right), 1\right)
\end{aligned}
$$

while applying the following acceleration

$$
\begin{aligned}
\tilde{u}_{v} & =\cos (\angle F-\psi)\|F\| \gamma_{a} \\
u_{v} & =\min \left(\max \left(-1, \tilde{u}_{v}\right), 1\right)
\end{aligned}
$$

where $\gamma_{a}$ is the third parameter of the reference strategy, serving as an acceleration coefficient.
As the APF method requires the tuning of the values of three parameters, during our simulation study we experimented with several parameter sets.

## Baseline 2: Safety velocity obstacles

Another reference algorithm, used in robotics and autonomous systems, is the Safety Velocity Obstacles (SVO) technique $[38,39]$ an extension of the Velocity Obstacles algorithm [18]. The goal of SVO is to choose the velocity vector of the vehicle such that it avoids colliding with any obstacle in its environment. The algorithm requires the position and velocity information of these obstacles to be known or to be precisely measurable at the time of decision-making.

Consider an environment with $N$ obstacles $B_{i}$, $i=1 \ldots N$, and a vehicle $A$. The velocity obstacle cone $V O_{i}$ encompasses all possible velocity vectors for $A$ that would lead to a collision with obstacle $B_{i}$ at some point in the future.

$$
\left.V O_{i}=\left\{\mathbf{v}_{\mathrm{A}} \mid \exists t: A\left(\mathbf{p}_{\mathrm{A}}+\mathbf{v}_{A} t\right) \cap B_{i}\left(\mathbf{p}_{\mathrm{Bi}}+\mathbf{v}_{\mathrm{Bi}} t\right) \neq 10\right\}\right)
$$

In this equation, the position and velocity vectors of the agent and the obstacles are represented by $\mathbf{p}_{\mathrm{A}}, \mathbf{v}_{\mathrm{A}}, \mathbf{p}_{\mathrm{Bi}}$, and $\mathbf{v}_{\mathrm{Bi}}$, respectively. The sets of points in the workspace occupied by the vehicle and obstacle $i$ if their positions are $\mathbf{p}$ are represented by $A(\mathbf{p})$ and $B_{i}(\mathbf{p})$. It is assumed that both the vehicle and the obstacles have a disk shape and that their velocities remain constant until time $t$. Additionally, it is assumed that there are no collisions between the obstacles during their motion.
The comprehensive velocity obstacle, $V O$, is formed by unifying all the individual velocity obstacle cones $V O_{i}$.

$$
\begin{equation*}
V O=\bigcup_{i=1}^{N} V O_{i} \tag{12}
\end{equation*}
$$

The set of reachable velocities, $R V$, contains all velocity vectors $\mathbf{v}_{\mathrm{A}}$ that can be achieved by the vehicle within the specified sampling time. The crucial step in the VO method is to determine the reachable avoidance velocities set $R A V$, which can be calculated by subtracting the $V O$ set from $R V$. These are the velocity vectors that the vehicle can select to ensure a collision-free path. A common approach is to discretize the $R A V$ set by dividing it into a grid-like structure.
Figure 6 illustrates the approach in a scenario where a vehicle is navigating through a setting containing both stationary and moving obstacles.
In the SVO, the velocity selection of the vehicle is influenced by two components: speed and safety. These aspects can be simultaneously considered by using an objective function.
The speed component (GO) serves as a metric for evaluating the progression toward the target position (goal):


Fig. 6 An illustration of the VO method for a vehicle at position $\mathbf{p}_{\mathrm{A}}$ navigating an environment consisting of a static obstacle at $\mathbf{p}_{\mathrm{B} 2}$ and a moving obstacle (pedestrian) at $\mathbf{p}_{\mathrm{B} 1}$ with velocity $\mathbf{v}_{\mathrm{B} 1}$

$$
\begin{equation*}
G O\left(\mathbf{v}_{\mathbf{i}}\right)=\frac{r_{i} \cos \Delta \theta_{i}}{v_{\max }} \tag{13}
\end{equation*}
$$

where $v_{\text {max }}$ represents the peak velocity the agent can attain while considering the limitations imposed by its kinematic properties. $\Delta \theta_{i}=\theta_{r g}-\theta_{r v_{i}}$ is the difference between the angle of the goal $\left(\theta_{r g}\right)$ and the angle of the velocity vector $\left(\theta_{r v_{i}}\right)$ while $r_{i}=\left\|\mathbf{v}_{i}\right\|$.
The safety aspect (SA) of the objective function assesses the potential for collision. It is sufficient to consider only the nearest Velocity Obstacle (VO). Additionally, if the robot is unable to reach the closest obstacle within the specified time frame, it is considered to be completely safe, and further distance would not enhance the safety. Based on these considerations, the following expression is obtained:

$$
\begin{equation*}
S A\left(\mathbf{v}_{\mathbf{i}}\right)=\min \left(1, \frac{\min _{\mathbf{v}_{V O} \in V O}\left\|\mathbf{v}_{i}-\mathbf{v}_{V O}\right\|}{v_{\max } \cdot T_{\max }}\right) \tag{14}
\end{equation*}
$$

where $T_{\max }$ is a known parameter representing the maximum time interval considered during the motion of the vehicle.

Finally, the velocity and safety components of the objective function (OF) are combined as a weighted average, where the parameter $\alpha$ represents the significance of safety. The main goal of the SVO algorithm is to select the velocity vector for the vehicle that maximizes the objective function.

$$
O F\left(\mathbf{v}_{i}\right)= \begin{cases}\alpha S A\left(\mathbf{v}_{i}\right)+(1-\alpha) G O\left(\mathbf{v}_{i}\right) & \text { if } \mathbf{v}_{i} \in R A V  \tag{15}\\ 0 & \text { otherwise }\end{cases}
$$

The parameter $\alpha$ plays a critical role in determining the outcome of the velocity selection process. During the testing phase, we evaluated various combinations of parameters to assess their effect on the results. We also
introduced a bang-bang control into the SVO algorithm, where the fastest solution $(\alpha=0)$ can be selected if the obstacles are further from the vehicle than a predetermined distance parameter [38]. Otherwise, the safest solution $(\alpha=1)$ is selected.

## Experimental study

In this section we describe a series of experimental studies, performed in simulation that compare the EBGbased vehicle strategy with the two baselines described in Sect. 6.
The examined scenario assumes a vehicle following the kinematic equations of (2) starting from the origin with zero speed and heading towards the location $[150,0]^{T} m$. The kinematic parameters of the vehicle are $v_{\max }=5 \mathrm{~m} / \mathrm{s}$, $a_{\max }=2 \mathrm{~m} / \mathrm{s}^{2}$ and $R=5 \mathrm{~m}$. The sum of the radii of the vehicle and a pedestrian (the "collision distance") is $d_{c}=2 m$. The goal of the vehicle is to arrive at $d_{c}$ distance from the target point; in an environment without obstacles, it would take 30.85 s .
To create a scenario that tests the ability of the algorithms to maneuver in a dense crowd, we positioned $N=30$ pedestrians between the vehicle and the goal, with the initial coordinates sampled from a uniform distribution with $x \in[10 m, 50 m]$ and $y \in[-20 m, 20 m]$. The pedestrians initial movement directions were random, and their speed was $w=2 m / s$. The pedestrians move along straight stretches, changing direction after random time intervals.
The frequency of direction changes was governed by an approximate discrete Poisson process. In each computation cycle, each pedestrian changes direction with a $p=0.033$ probability independently of other pedestrians and previous computation cycles. The new direction was also a uniformly random independent variable. The average duration of one straight stretch is $\Delta t / p=3 s$, with simulation's sampling time as $\Delta t=0.1 s$.

The simulation run terminates if the vehicle reaches the goal or collides with a pedestrian. In this simulation, a collision is defined as a state where the distance between the vehicle and the pedestrian is less than the collision distance $d_{c}$, while the vehicle is heading towards the pedestrian: $|\psi-\angle \overrightarrow{E P}| \leq 90^{\circ}$.

## Analyzing individual simulation runs

To understand the strategy deployed by the different algorithms, it is useful to analyze individual simulation runs in detail. To facilitate this analysis, the following visual notations will be used in the subsequent discussion. The trajectories of pedestrians will be depicted as lines of various colors, with a black point at the end of each line indicating the final position. The motion of the vehicle will be presented through a series of figures,
each capturing a time interval of $[0,0.2 \mathrm{~T}),[0.2 \mathrm{~T}, 0.4 \mathrm{~T}), \ldots$, [ $0.8 \mathrm{~T}, \mathrm{~T}$ ], where T represents the length of the simulation run. Lastly, a summary figure portrays the entire motion ( 0 to T ) for both the vehicle and the pedestrians.
Figure 7 shows the simulation run for the APF method. This run concludes with a collision at $T=30.9 \mathrm{~s}$. Figure 8 shows the simulation run for the SVO method. In this case as well, the vehicle fails to reach the goal, with the simulation run ending due to a collision at $T=12.3 \mathrm{~s}$. The only approach that had successfully reached the goal in this scenario is the EBG method which reached the goal at $T=85.5 \mathrm{~s}$, as depicted in Fig. 9. Figure 10 shows the complete paths of several specific pedestrians along with the motion of the vehicles controlled using the different algorithms.
In Figs. $7-10$ we see a scenario where the EBG outperforms the baseline agents. However, there is a downside to applying the EBG algorithm. As Fig. 11 depicts, the method is prone to oscillation, causing discomfort to passengers and putting strain on the mechanics and electronics of the robot. This will have to be amended prior to application of the algorithm by introducing smoothing to the control outputs. The effect of such lenient controls on safety may have to be countered by increasing the safety distance.

## Experimental results

To perform a fair comparison between the algorithms, we tested them over 100 scenarios with randomly generated pedestrians. We ran the APF algorithm with 18 different parameter sets and the SVO method with 4 different $\alpha$ parameters. As the EBG parameters can be directly specified from first principles, they were chosen as follows: $V_{\text {safe }}=2 d_{c}=4 m, \underline{V}=4 d_{c}=8 m$, and $\bar{V}=10 d_{c}=20 m$.
The effectiveness of the algorithms was evaluated based on the average running time and the number of accidents that occurred during the simulation runs. Table 1 shows the number of runs that resulted in a collision, and, for the runs that terminated successfully, the average time needed to reach the goal. We found that the EBG algorithm always avoids collisions. For the baseline algorithms, the parameters had a significant impact on the number of collisions. For the APF algorithm, the best parameter choice resulted in 19 collisions, but certain parameter settings had as many as 81 collisions out of 100 trials. The SVO method peformed significantly better, with the number of collisions ranging from 8 to 20 , depending on the value of the parameter $\alpha$. Comparing the SVO and EBG methods, we found that while SVO generated faster solutions, it could not guarantee collision-free target-reaching in all scenarios due to its dependence on the velocity vector of the pedestrians, which could only be calculated between the last two


Fig. 7 The motion of the vehicle using the APF method. In this scenario, the goal cannot be reached because of a collision


Fig. 8 The motion of the vehicle using the SVO method. In this scenario, the goal cannot be reached because of a collision
steps. Hence, sudden changes in the velocity vector of the closest pedestrian could lead to collisions even if the vehicle performed an emergency braking maneuver, as
shown in Fig. 8. The best solution with the SVO method was achieved using the bang-bang control in the parameter set, resulting in eight collisions. However, overall, the


Fig. 9 The motion of the vehicle using the EBG method. In this scenario, the goal was reached, resulting in the only collision-free solution


Fig. 10 The motion of the evader agent using the three different methods. The whole path of some of the obstacles can be seen

EBG algorithm was the most effective in achieving colli-sion-free target-reaching tasks.
In a real-world scenario, we care not only about avoiding accidents but also the safety of the vehicle's movement. To investigate this, for the EBG and the bestperforming APF and SVO parametrizations, we collected all time instants from all collision-free runs where the closest pedestrian is closer than 10 m . In Fig. 12, we display the frequencies of such moments depending on the distance and the speed of the vehicle. In the figure, the color of a speed-distance category corresponds to the logarithm of the number of occurrences.
We find that when using the APF algorithm (left), zero and maximal vehicle speeds are the most frequent,
with the histogram showing an approximately uniform distribution. For the SVO algorithm (center), we find that the vehicle tends to select large velocity vectors to execute evasive maneuvers to avoid collisions, and it never comes to a full stop. However, as previously shown, the SVO method does not always guarantee collision-free motion. In the case of the EBG algorithm (right), we see a curve corresponding to maximal deceleration. This curve is fuzzy because the pedestrians are not following the EBG-optimal direction. It is important to note that the non-zero points in the EBG case are much farther from the high velocity / low distance (upper right) corner of the figure compared to the two


Fig. 11 Acceleration and velocity of the EBG agent in the example simulation run. Heavy oscillation can be seen between cycles 396 and 416 .
baseline algorithms, meaning that even the most critical situations are significantly less dangerous.
All collision avoidance algorithms must strike a balance between safety and the time to reach the obstacle.

We study the tradeoffs made by the different algorithms in Fig. 13.
The comparison includes the EBG algorithm, the APF algorithm with parameters $\gamma_{t}=0.01, \rho=2 d_{c}, \gamma_{a}=2$ and the SVO algorithm following the bang-bang strategy. Only successful (collision-free) runs are examined. The SVO agent reaches the goal significantly faster than both the APF and the EBG, with the time to goal being below 40 s in the majority of situations. On the other hand, the EBG agent never finishes faster than 40 s . The elapsed times of all three agents are consistent, with a few high outliers. The EBG agent's improved safety, comes at the cost of degraded navigation performance.

## Sensitivity to obstacle parameters

We challenged the collision avoidance methods by putting a number of dynamic obstacles between the start and goal positions. We have identified three main factors influencing the performance of the agents: the number of pedestrians, the speed of the pedestrians, and the unpredictability or randomness of their movement. In the following, we analyze the sensitivity to these factors to explore our methods's applicability and find worst case situations.

Table 1 Average run times and number of accidents depending on the APF and SVO parameters

|  | $\gamma_{t}$ | $\rho / d_{c}$ | $\gamma a$ | $\alpha$ | Average time | Accidents |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| APF | 0.1 | 4 | 1 | - | 107.44 | 35 |
|  | 0.1 | 4 | 0.5 | - | 110.22 | 23 |
|  | 0.1 | 4 | 2 | - | 105.07 | 34 |
|  | 0.1 | 2 | 1 | - | 60.81 | 35 |
|  | 0.1 | 2 | 0.5 | - | 66.55 | 35 |
|  | 0.1 | 2 | 2 | - | 59.46 | 42 |
|  | 0.1 | 1 | 1 | - | 42.64 | 81 |
|  | 0.1 | 1 | 2 | - | 37.46 | 78 |
|  | 0.01 | 4 | 1 | - | 111.10 | 31 |
|  | 0.01 | 4 | 0.5 | - | 112.40 | 29 |
|  | 0.01 | 4 | 2 | - | 110.06 | 41 |
|  | 0.01 | 2 | 1 | - | 86.34 | 27 |
|  | 0.01 | 2 | 0.5 | - | 95 | 25 |
|  | 0.01 | 2 | 2 | - | 82.61 | 19 |
|  | 0.01 | 1 | 1 | - | 104.61 | 63 |
|  | 0.01 | 1 | 2 | - | 79.19 | 64 |
|  | 0.01 | 6 | 2 | - | 126.98 | 56 |
|  | 0.01 | 6 | 1 | - | 127.87 | 52 |
| SVO | - | - | - | 0.2 | 31.34 | 20 |
|  | - | - | - | 0.5 | 31.65 | 13 |
|  | - | - | - | 0.8 | 34.04 | 13 |
|  | - | - | - | BB | 38.95 | 8 |
| EBG | - | - | - | - | 81.53 | 0 |

First, let us vary the switching time, the average duration of straight stretches in the pedestrians' movement, while keeping all other parameters unchanged. We expect that more predictable pedestrians are easier to
avoid, leading to better performance. Fig. 14 confirms this expectation: APF and SVO agents have fewer collisions and both the APF and EBG algorithms reach the target in a shorter time. Note that the average time in


Fig. 12 Heatmap of near-collision states for the different algorithms: (left) APF, (center) SVO and (right) EBG. Brighter colors depict higher frequencies. Dark blue corresponds to zero occurrences


Fig. 13 Histograms of the time to target over 100 simulation runs, with only the successful runs plotted. The optimal time, with no pedestrians present is 30.85 s . (left) APF, (center) SVO, (right) EBG


Fig. 14 Results dependent on the average duration of straight stretches in the pedestrians' paths. (left) Number of collisions, (right) Average and median time to reach the goal
case of an EBG agent increases for a switching time of 10 s , but the median time decreases. This is due to outlier scenarios where a pedestrian approaches the target area and thus prevents the agent from reaching it,
With regard to speed, slower pedestrians are less dangerous. This is confirmed in Fig. 15: fewer collisions happen with slower pedestrians. Interestingly, for the APF and EBG agents, the time taken to reach the goal increases greatly when the obstacle speed is reduced. This effect can be explained by pointing out that these two agets often stop and wait for a pedestrian to pass, whereas the SVO leans more on swerve maneuvers. A slow pedestrian can be bypassed easily, but one has to wait longer for it to pass.
The number of pedestrians has an obvious effect on performance. Fig. 16 shows that increasing the number of obstacles leads to more collisions, and longer elapsed times. It is still worth noting, that the influence on time
is much greater in the case of the EBG than the other two agents. Once more this can be attributed to the EBG's conservativeness.
Let us note two important observations. First, the EBG agent moves collision-free, regardless of the simulation parameters. Second, the worst case for our algorithm is when obstacles are moving slowly; then, the handicap regarding efficiency is greatest.

## Conclusions

In this paper, we introduced the Emergency Braking Game and, based on it, an algorithm that allows a vehicle to avoid collisions when navigating through a dense crowd of pedestrians, even accounting for worst-case pedestrian behavior. Analyzing the technique under some simplifying assumptions, we provide a formal proof of its safety. In an experimental study comparing it to two baseline algorithms, we validated the safety proof, and


Fig. 15 Results dependent on the speed of the pedestrians. (left) Number of collisions, (right) Average and median time to reach the goal



Fig. 16 Results dependent on the number of pedestrians. (left) Number of collisions, (right) Average and median time to reach the goal

## found that the baseline techniques had a significant number of collisions.

The next step in the development shall be the elimination of oscillating controls. Future research will additionally focus on omitting the simplifying assumptions we have made in this work. In particular, it will involve handling observation uncertainties and partial observation, actuator noise, as well as incorporating static obstacles and the boundaries of the usable space.

## Acknowledgements

Supported by the ÚNKP-23-4-I-BME-360 New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund.

## Author contributions

All Authors read and approved the final manuscript.

## Funding

Open access funding provided by Budapest University of Technology and Economics. The research reported in this paper is part of project no. BME-NVA-02, implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the TKP2021 funding scheme.

## Availability of data and materials

Not applicable.

## Declarations

## Ethics approval and consent to participate

Not applicable.

## Competing interests

The authors declare that they have no competing interests.

Received: 17 June 2023 Accepted: 13 November 2023
Published online: 23 February 2024

## References

1. Gelbal SY, Aksun-Guvenc B, Guvenc L (2020) Collision avoidance of Iow speed autonomous shuttles with pedestrians. Int J Automot Technol 21:903-917
2. Yang W, Zhang Z, Jiang K, Lei Q, Zong K (2022) Optimization of emergency braking pedestrian collision avoidance for autonomous vehicle fusing the fuzzy neural network with the genetic algorithm. Int J Control Automat Syst 20(7):2379-2390
3. Rafiei A, Fasakhodi AO, Hajati F (2022) Pedestrian collision avoidance using deep reinforcement learning. Int J Automot Technol 23(3):613-622
4. Isaacs R (1965) Differential games. Wiley, New York
5. Exarchos I, Tsiotras P, Pachter M (2015) On the suicidal pedestrian differential game. Dyn Games Appl 5(3):297-317. https://doi.org/10.1007/ s13235-014-0130-2
6. Dang T, Desens J, Franke U, Gavrila D, Schäfers L, Ziegler W (2012) Steering and evasion assist. In: Eskandarian A (ed) Handb Intell Veh. Springer, London, pp 759-782
7. Exarchos I, Tsiotras P, Pachter M (2016) UAV collision avoidance based on the solution of the suicidal pedestrian differential game. Proc AIAA Guid Navig Control Conf. https://doi.org/10.2514/6.2016-2100
8. Malone N, Chiang H-T, Lesser K, Oishi M, Tapia L (2017) Hybrid dynamic moving obstacle avoidance using a stochastic reachable set-based potential field. IEEE Transact Robot 33(5):1124-1138. https://doi.org/10. 1109/TRO.2017.2705034
9. Bonnin S, Weisswange TH, Kummert F, Schmüdderich J (2014) Pedestrian crossing prediction using multiple context-based models. In: Bonnin S (ed) 17th International IEEE conference on intelligent transportation systems (ITSC). Qingdao, IEEE, pp 378-385
10. Ahmed T, Moeinaddini M, Almoshaogeh M, Jamal A, Nawaz I, Alharbi F (2021) A new pedestrian crossing level of service (PCLOS) method for promoting safe pedestrian crossing in urban areas. Int J Environ Res Public Health 18(16):8813
11. Bhatia TS, Solmaz G, Turgut D, Bölöni L (2015) Two algorithms for the movements of robotic bodyguard teams. In: Proceeding of workshop on knowledge, skill, and behavior transfer in autonomous robots, pp 2-8
12. Roseau Pensant Inc.: Shibuya crossing. https://thegate12.com/spot/471. Accessed on 11 April 2023
13. Lefèvre S, Vasquez D, Laugier C (2014) A survey on motion prediction and risk assessment for intelligent vehicles. Robomech J 1(1):1-14
14. Guinot L, Ando K, Takahashi S, Iwata H (2023) Analysis of implicit robot control methods for joint task execution. ROBOMECH J 10(1):1-21
15. Fujii N, Ogata T, Hara T, Ota J (2014) Realization method for a rearrangement task by multiple mobile robots in consideration of map errors. ROBOMECH J 1:1-11
16. Wang J, Takahashi Y (2018) Indoor mobile robot self-localization based on a low-cost light system with a novel emitter arrangement. ROBOMECH J 5(1):1-17
17. Mohanan MG, Salgoankar A (2018) A survey of robotic motion planning in dynamic environments. Robot Auton Syst 100:171-185. https://doi. org/10.1016/j.robot.2017.10.011
18. Fiorini P, Shiller Z (1998) Motion planning in dynamic environments using velocity obstacles. Int J Robot Res 17(7):760-772
19. Alonso-Mora J, Beardsley P, Siegwart R (2018) Cooperative collision avoidance for nonholonomic robots. IEEE Transact Robot 34(2):404-420. https://doi.org/10.1109/TRO.2018.2793890
20. Vannoy J, Xiao J (2008) Real-time adaptive motion planning (RAMP) of mobile manipulators in dynamic environments with unforeseen changes. IEEE Transact Robot 24(5):1199-1212. https://doi.org/10.1109/TRO.2008. 2003277
21. Wilkie D, Van Den Berg J, Manocha D (2009) Generalized velocity obstacles. In: International conference on intelligent robots and systems (IROS-09). pp 5573-5578
22. Snape J, Van Den Berg J, Guy SJ, Manocha D (2010) Smooth and collisionfree navigation for multiple robots under differential-drive constraints. In: International conference on intelligent robots and systems (IROS-10). pp. 4584-4589
23. Huang Y, Chen L, Van Gelder PHAJM (2019) Generalized velocity obstacle algorithm for preventing ship collisions at sea. Ocean Eng 173:142-156
24. Van Den Berg J, Snape J, Guy SJ, Manocha D (2011) Reciprocal collision avoidance with acceleration-velocity obstacles. In: Proceedings of IEEE International conference on robotics and automation (ICRA-11). pp 3475-3482
25. Snape J, Van Den Berg J, Guy SJ, Manocha D (2011) The hybrid reciprocal velocity obstacle. IEEE Transact Robot 27(4):696-706
26. Douthwaite JA, Zhao S, Mihaylova LS (2019) Velocity obstacle approaches for multi-agent collision avoidance. Unmanned Syst 7(01):55-64
27. Battisti T, Muradore R (2020) A velocity obstacles approach for autonomous landing and teleoperated robots. Autonom Robots 44(2):217-232
28. Vatcha R, Xiao J (2014) Detection of robustly collision-free trajectories in unpredictable environments in real-time. Autonom Robots 37(1):81-96. https://doi.org/10.1007/s10514-013-9377-5
29. Mitchell IM, Bayen AM, Tomlin CJ (2005) A time-dependent HamiltonJacobi formulation of reachable sets for continuous dynamic games. IEEE Transact Autom Control 50(7):947-957. https://doi.org/10.1109/TAC. 2005. 851439
30. Fox D, Burgard W, Thrun S (1997) The Dynamic Window approach to collision avoidance. IEEE Robot Autom Mag 4(1):23-33
31. Brock O, Khatib O (1999) High-speed navigation using the global dynamic window approach. In: 1999 IEEE International conference on robotics and automation, volume 1, pp 341-346
32. Wu B, Chi X, Zhao C, Zhang W, Lu Y, Jiang D (2022) Dynamic path planning for forklift agv based on smoothing $A^{*}$ and improved dwa hybrid algorithm. Sensors 22(18):7079
33. Reif JH, Wang H (1999) Social potential fields: a distributed behavioral control for autonomous robots. Robot Autonom Syst 27(3):171-194
34. Kovács B, Szayer G, Tajti F, Burdelis M, Korondi P (2016) A novel potential field method for path planning of mobile robots by adapting animal motion attributes. Robot Autonom Syst 82:24-34. https://doi.org/10. 1016/j.robot.2016.04.007
35. Chiang H-T, Malone N, Lesser K, Oishi M, Tapia L (2015) Path-guided artificial potential fields with stochastic reachable sets for motion planning in highly dynamic environments. In: Proceeding IEEE international conference on Robotics and Automation (ICRA-15). pp 2347-2354
36. Li G, Tamura Y, Yamashita A, Asama H (2013) Effective improved artificial potential field-based regression search method for autonomous mobile robot path planning. Int J Mechatron Autom 3(3):141-170
37. Qixin C, Yanwen H, Jingliang Z (2006) An evolutionary artificial potential field algorithm for dynamic path planning of mobile robot. In: Proceeding of IEEE International conference on intelligent robots and systems (IROS-06), pp 3331-3336. https://doi.org/10.1109/IROS.2006.282508
38. Gyenes Z, Szadeczky-Kardoss EG (2018) Motion planning for mobile robots using the safety velocity obstacles method. In: Proceeding of 19th International Carpathian Control Conference (ICCC-18). IEEE. pp 389-394
39. Yuan Y, Shi Y, Yue S, Xue S, Yi C, Chen B (2022) A dynamic obstacle avoidance method for AGV based on improved speed barriers. Electronics 11(24):4175. https://doi.org/10.3390/electronics11244175

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.


[^0]:    *Correspondence:
    János Szőts
    szotsjanos@iit.bme.hu
    ${ }^{1}$ Department of Control Engineering and Information Technology, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Múegyetem rkp. 3., Budapest H-1111, Hungary
    ${ }^{2}$ Department of Computer Science, University of Central Florida, 4328 Scorpius St, Orlando, FL 32816, USA

