Multi-agent motion planning using differential games with lexicographic preferences

Kristina Miller*, Sayan Mitra*
*Electrical and Computer Engineering Department
University of Illinois Urbana-Champaign

***Abstract*—Multi-player games with lexicographic cost functions can capture a variety of driving and racing scenarios and under certain conditions are known to have pure-strategy Nash Equilibria. The standard Iterated Best Response (IBR) procedure for finding such equilibria can be slow because, in general, computing the best response for each agent involves solving a non-convex optimization problem. In this paper, we introduce a type of game which uses a lexicographic cost function. We show that for this class of games, the best responses can be effectively computed through piece-wise linear approximations. This in turn enables us to approximate the Nash Equilibria using a linearized version of IBR. We show that the gap between the linear approximations returned by our linearized IBR and the true best response drops asymptotically. We have implemented the algorithm and our experiments show that it can find approximate Nash Equilibria for handful of agents driving in realistic scenarios in less than 10 seconds.**

I. Introduction

Motion planning in multi-agent environments is a challenging problem with many applications in autonomous driving, human-robot interactions, and urban air mobility. Standard single agent motion planning algorithms can be ported to multi-agent problems, under restrictive assumptions such as no interaction among agents or accurate predictions for agents. By treating other agents as dynamic but known obstacles, a variety of motion strategies have been used to tackle the multi-agent problem, such as sampling-based planners [29, 22, 7], temporal logic-based planers [21, 5, 1], reach-avoid synthesis [18, 6, 10, 2, 23, 16], and model predictive control [17, 3, 24, 27]. These approaches can provide safe trajectories for navigation, however, typically they do not account for dynamic interaction between agents.

Differential games [12] are a natural framework to study strategic interactions between rational agents moving in some workspace. Using differential games to solve the multi-agent motion planning problem is a growing area of interest [14, 8, 15, 28, 13, 25, 12]. Recently, a special class of games, called an *urban driving game*, was introduced in [26] for studying decision making in autonomous vehicles around other vehicles and pedestrians. In urban driving games, agents have a shared goal of non-collision and independent individual goals. Specifically, the cost function for each agent has two parts: (i) a *collision cost* which forces the agents to avoid collisions, and (ii) a *personal cost* which only dependents on the agent's own actions. The personal cost could, for example, encode minimization of distance traveled, fuel expended, or time.

The standard way for evaluating steady-state behavior in games is to study its Nash equilibria [19]. At a Nash equilibrium, none of the agents can improve their cost by unilaterally changing their actions, and thus, each agent is playing its optimal response to the action profile of the other agents. Thus, the Nash equilibria can be seen as prediction of rational outcomes. In [26], it is shown that the Nash equilibrium exists for urban driving games.

One limitation of the urban driving game formulation of [26] is that the personal cost function only depends on the individual agent's actions. This formulation cannot be applied in scenarios where one agent's personal costs dependents on its opponent's cost, such as in a race. In this work, we expand upon the existing notion urban driving game, and add a zero sum component to the agents personal cost function. In this way, the urban driving game can be used in a wider variety of scenarios, such as a racing scenario. We show that this revised urban driving game is still a potential game, and therefore, has a pure strategy Nash equilibrium.

In order to find the Nash equilibrium of the revised urban driving game, we propose a linearized version of an iterative best response algorithm. In this algorithm, we use linear constraints to minimize the collision cost function, and search over these actions for the best response. Thus, we approximate the agents' trajectories as piecewise linear curves. The algorithm returns the piecewise linear approximation that minimizes the agent's personal cost function, giving rise to the notion of linear Nash equilibrium. Finally, we show that for a large enough number of segments, we can find the asymptotic bound on the gap between the true Nash equilibrium and the linear Nash equilibrium.

Finally, we implement the linearized version of iterative best response. This is done using a mixed integer linear program, similar to [6] and [18]. We show that this prototype tool works in a variety of scenarios, including traffic scenarios. Additionally, we show that the time to find the approximate Nash equilibrium for most scenarios is within seconds.

II. Lexicographic games and Nash Equilibria

We use sets such as $\mathcal{N} = \{0, \ldots, n-1\}$ to index or name the players in the game. For an $i \in \mathcal{N}$, $\{-i\}$ is used as a shorthand for $\mathcal{N} \backslash \{i\}$. The set of real and positive real numbers is denoted as $\mathbb{R}, \mathbb{R}_{\geq 0}$. The *lexicographic ordering* on $\mathbb{R}^2$ is defined as $(a_1, b_1) \preceq (a_2, b_2)$ iff (1) $a_1 \leq a_2$ or (2) $a_1 = a_2$

and $b_1 \leq b_2$. If $(a_1, b_1) \preceq (a_2, b_2)$ and $(a_1, b_1) \neq (a_2, b_2)$ then $(a_1, b_1) \prec (a_2, b_2)$.

A. Lexicographic general sum games

In this work, we discuss a class of games called lexicographic general sum game which is shown in Definition 1. This lexicographic general sum game is a generalization of the urban driving game introduced in [26]. Here, we modify the personal cost component of the lexicographic cost function so that each agent's personal cost is also dependent on its opponent's actions.

Definition 1. An *n-player lexicographic general sum game (LG)* $\mathcal{G}$ is defined by three components $\mathcal{G} = \langle \mathcal{N}, \{\mathcal{Z}_i\}, \{J_i\} \rangle$, where:

(i) $\mathcal{N} = \{0, \dots, n-1\}$ is a set of n agents,

(ii) $\mathcal{Z}_i$ is a compact set of uniformly continuous curves $z_i : [0,1] \to \mathbb{R}^d$, which specifies a trajectory for agent i. The *joint action space* for $\mathcal{G}$ is $\mathcal{Z} = \prod_{i \in \mathcal{N}} \mathcal{Z}_i$, and a particular joint action $z \in \mathcal{Z}$ can be written as $z = [z_0, \dots, z_{n-1}]$ where each $z_i \in \mathcal{Z}_i$. Finally,

(iii) $J_i : \mathcal{Z} \to \mathbb{R}^2$ is a *lexicographic cost function* for agent i with $J_i(z) = (J_i^{\text{col}}(z), J_i^{\text{per}}(z))$, with two parts, (a) the *collision cost*

$$J_i^{\text{col}}(z) = \sum_{j \in \{-i\}} f_{i,j}(z_i, z_j), \tag{1}$$

is defined in terms of a collection of bounded real-valued symmetric functions, $f_{i,j}(z_i, z_j) = f_{j,i}(z_j, z_i)$, $i \neq j$. And (b) the *personal cost*

$$J_i^{\text{per}}(z) = g_i(z_i) - \sum_{j \in \{-i\}} g_j(z_j), \tag{2}$$

is defined in terms of individual costs g_i which are continuous and bounded over $\mathcal{Z}_i$.

This 2-part definition of the cost function J provides some analytical advantages as we shall see in Section III, and it separates concerns: (a) the collision cost depends on action pairs and it incentivizes certain social outcomes over others; (b) the personal cost depends only on individual choices. We note here that the uniform continuity of the actions z_i implies that the actionspace $\mathcal{Z}_i$ is compact by Arzela-Ascoli Theorem [4].

Example: Some examples of an LG at a traffic intersection can be seen in Figure 1. Each agent lives in $\mathbb{R}^2$ is given a starting position (rectangles) and a goal position (circles). The action spaces are the trajectories that the agent's take to reach the goal position. The collision cost function is given in (3)

$$f_{i,j}(z_i, z_j) = \begin{cases} 0 & \text{if } z_i \cap \mathcal{B}_c(z_j) = \varnothing \\ 1 & \text{else} \end{cases} \tag{3}$$

There are a variety of individual cost functions that can be used in these scenarios. A simple example of the individual cost function is one in which the agents wish to minimize their traveled distance, seen in (4).

$$g_i(z_i) = \int_0^1 \sqrt{1 + (\frac{d}{ds} z_i(s))^2} ds \tag{4}$$

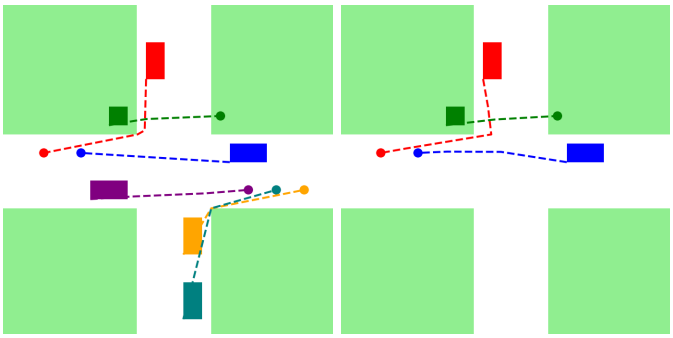

Fig. 1. Examples of general sum driving games at an intersection. Left: the vehicles must drive through the intersection while avoiding collisions with each other, and a pedestrian attempts to cross the intersection. Right: two vehicles attempt to drive through the intersection while the pedestrian attempts to cross the intersection. The rectangles are the starting positions, and the circles denote the goal positions. The dotted trajectories are examples of what the Nash equilibrium may look like.

B. Best responses and Nash equilibria

The predictions from game theory can help us understand social behavior by predicting how agents will act in a game defined by the costs. The predictions about player behavior are called equilibria, and one particular type of equilibria is called Nash equilibrium, which is defined in terms of agent's best responses to other agents's choice. In 1951, John Nash proved that all n-player games with *finite action sets* have a mixed-strategy Nash equilibrium [19][1]

Definition 2 (Best response)**.** Given an action profile $z_{-i} \in \mathcal{Z}_{-i}$ of agents in $\{-i\}$, the *best response* set of agent i is the set of actions that minimizes the cost $J_i(z_i, z_{-i})$. That is,

$$\mathcal{R}^{\text{BR}}(z_{-i}) = \{z_i \in \mathcal{Z}_i \mid J_i(z_i, z_{-i}) \preceq J_i(z_i', z_{-i}) \ \forall z_i' \in \mathcal{Z}_i\}$$

The notion of a Nash equilibrium [19] of a game captures the idea that for some joint actions, no agent can unilaterally improve their cost. In this work, a Nash equilibrium refers to a *pure strategy* Nash equilibrium, where agents select one action from $\mathcal{R}_i^{\text{BR}}(z_{-i})$, as compared to a *mixed strategy* Nash equilibrium, where agents select actions from $\mathcal{R}_i^{\text{BR}}(z_{-i})$ according to a probability distribution.

Definition 3 (Nash equilibrium)**.** A joint action $z^* \in \mathcal{Z}$ of an LG is a pure strategy *Nash equilibrium(NE)* if $\forall i \in \mathcal{N}$, $\forall z_i \in \mathcal{Z}_i$,

$$J_i(z^*) \preceq J_i(z_i, z_{-i}^*).$$

[1]A *mixed strategy* is a probabilistic distribution on the set of actions of each player such that this distribution is the best response to the other agent's distributions. In this paper, we focus on pure Nash equilibria, and mixed-strategies would be considered in a future work.

It follows from these definitions that z^* is an NE if and only if for every $i \in \mathcal{N}$, $z_i^* \in \mathcal{R}_i^{\text{BR}}(z_{-i}^*)$.

The existence of a Nash equilibrium for general continuous game with continuous utility functions can be proven using a generalization of Kakutani's fixed point theorem [9]. While this existence result has been around for a long time, there has been limited work on practical algorithms for computing Nash equilibria in general continuous games. Several special classes of games have been identified, for example, separable games [20], where the problem becomes tractable. For later use, we introduce the notion of an ε-Nash equilibrium of a lexicographic game $\mathcal{G}$, for any $\varepsilon > 0$, which is a joint action $z^\varepsilon \in \mathcal{Z}$ such that the neither the collision cost nor the personal cost can be improved by more than ε. That is, z^ε is a ε-NE if for all $i \in \mathcal{N}$ for all $z_i \in \mathcal{Z}_i$ the inequalities in (II-B) hold true.

$$\begin{aligned} J_i(z^\varepsilon) &\preceq \langle J_i^{\mathsf{col}}(z_i, z_{-i}^\varepsilon) + \varepsilon, J_i^{\mathsf{per}}(z_i, z_{-i}^\varepsilon)\rangle \\ J_i(z_i^\varepsilon) &\preceq \langle J_i^{\mathsf{col}}(z_i, z_{-i}^\varepsilon), J_i^{\mathsf{per}}(z_i, z_{-i}^\varepsilon) + \varepsilon\rangle. \end{aligned} \tag{5}$$

An example of potential Nash equilibria can be seen in Figure 1.

Problem Statement: We would like to develop an algorithm that given a lexiographic game $\mathcal{G}$ and $\varepsilon > 0$, computes an ε-Nash equilibrium of $\mathcal{G}$.

In driving games, computing the NE can be very difficult, especially in scenarios with a large (or infinite) action space, or in games that have a large number of agents [26]. In this paper we aim to simplify the computation of NE through a linearized iterative best response. In the next section, we will discuss the iterative best response framework for computing NE, and then we will present the linearized version. Then we will analyze the algorithm, and show that we can obtain a reasonable approximation of the NE.

C. Iterated Best Response for Computing NE

Iterated best response (IBR) is a standard method for finding pure strategy NE. The procedure starts with an initial guess of a joint action $z \in \mathcal{Z}$ to be a candidate NE. Then each agent's action z_i in z is updated to be the best response $\mathcal{R}_i^{\text{BR}}$ to the other agent's current choice of z_{-i}. This continues iteratively until and unless there is no better response to the joint action for any of the agents. The standard IBR procedure is shown Algorithm 1. The computation of the best response for each agent involves solving an optimization problem, which can be challenging for general infinite games. Further, the procedure is not guaranteed to converge, however, if the game is a *potential game* (defined below), then convergence is guaranteed. We proceed by showing that lexicographic general sum games are a special type of potential game (Proposition 1).

D. Lexicographic and potential games

We first show that our game is an ordinal potential game[2], which are guaranteed to have pure strategy NE.

[2]A *potential games* is one where and an exact potential function P exists such that $J_i(z_i', z_{-i}) - J_i(z_i, z_{-i}) = P(z_i', z_{-i}) - P(z_i, z_{-i})$/ We will see that this condition is too strong and not necessary for analysis of LG.

Algorithm 1: Iterative best response (IBR)

Input: $\mathcal{G}$, initial guess $z \in \mathcal{Z}$
Output: z^*

```
do
    z* ← z
    for i ∈ N do
        z_i ∈ R_i^BR(z_{-i})
    end
while z* ≠ z;
return z*
```

Definition 4 (Ordinal potential game)**.** *An LG $\mathcal{G}$ is an* ordinal potential game (OPG) *if there exists a function $P : \mathcal{Z} \to \mathcal{X}$ such that $\forall i \in \mathcal{N}$, $\forall z_{-i} \in \mathcal{Z}_{-i}$, $\forall z_i, z_i' \in \mathcal{Z}_i$*

$$J_i(z_i', z_{-i}) \preceq J_i(z_i, z_{-i}) \iff P(z_i', z_{-i}) \preceq P(z_i, z_{-i}).$$

Here $(\mathcal{X}, \preceq)$ is some totally ordered set.

Such a function P is called an *ordinal potential function (OPF)* of the game. It can be shown that any lexicographic game has an OPF. The proof for Proposition 1 is a modification of Theorem 1 from [26], and is included in the full version of this paper.

Proposition 1. *Any lexicographic game $\mathcal{G}$ is an ordinal potential game with potential function*

$$P(z) = \langle \frac{1}{2} \sum_{j \in \mathcal{N}} J_j^{\mathsf{col}}(z), \sum_{j \in \mathcal{N}} g_j(z_j)\rangle. \tag{6}$$

The fact that $\mathcal{G}$ is an ordinal potential game leads to some nice properties: First, a global minimum of P exists [26], and this global minimum corresponds to a pure strategy Nash equilibrium [11]. Secondly, an ε-approximation of this NE can be computed using the iterated best response approach. These two results are stated here as Propositions 2 and 3.

Proposition 2. *Lexicographic general sum games have a pure strategy Nash equilibrium.*

For lexicographic games, the action spaces $\mathcal{Z}_i$ are indeed compact, which means that for every $z_i \in \mathcal{Z}_i$, for every $\varepsilon > 0$, there exist $\delta > 0$ such that $\|z_i(x) - z(y)\| \leq \varepsilon$, and $x, y \in [0, 1]$ and $\|x - y\| \leq \delta$ (generalized Arzela-Ascoli Thm from [4]). Then, using similar reasoning to [26], for continuous cost functions, a pure strategy NE exists.

Proposition 3. *For any lexicographic game $\mathcal{G}$, and any $\varepsilon > 0$, the IBR procedure converges to an ε-NE in a finite number of iterations.*

The proof of this proposition comes from the fact that on each iteration, each agent attempts to improve their cost by ε. If there is no updated action that each agent can take to improve their cost by ε, then the joint action is an ε-NE, and the algorithm terminates.

III. Linear IBR Algorithm

We propose an algorithm called L-IBR that searches for linear best responses that are increasingly closer to the actual best response of the agents. Instead of searching for best response actions, which are continuous functions satisfying certain cost constraints imposed by the actions of other agents, the algorithm searches for best responses that are piecewise linear (PWL) functions. PWL functions can be described by a sequence of waypoints in $\mathbb{R}^d$, and under additional approximation of the collision cost, the search for optimal waypoints can be formulated as a mixed integer linear program (MILP). We show that as the number of waypoints describing a PWL function increases, the solution found by L-IBR approaches the linear best response, and the gap between the cost of this linear best response and the actual NE, can be bounded. The pseudocode for L-IBR is shown in Algorithm 2.

L-IBR takes as input the lexicographic game $\mathcal{G}$, and an initial guess for the joint action $z \in \mathcal{Z}$ that described by a collection of sequences of waypoints $\{P_i\}_{\mathcal{N}}$. Then, IBR is performed for the current joint action described by the waypoint sequences, until they converge to an equilibrium. The algorithm returns the PWL joint action as the computed approximate linear equilibrium z^{lin}.

Algorithm 2: L-IBR

Input: game $\mathcal{G} = \langle \mathcal{N}, \{\mathcal{Z}_i\}_{\mathcal{N}}, \{J_i\}_{\mathcal{N}} \rangle$, initial guess $\{P_i^{[0]}\}_{\mathcal{N}}$, collision bound c
Output: PWL equilibrium z^{lin}

$k \leftarrow 1$
while $k = 1$ *or* $P^{[k]} \neq P^{[k-1]}$ **do**
  $\{P_i^{[k]}\}_{\mathcal{N}} \leftarrow \{P_i^{[k-1]}\}_{\mathcal{N}}$
  for $j \in \mathcal{N}$ **do**
    $Q \leftarrow \texttt{linUpdate}(\{P_i\}_{-j}, J_j^{\mathsf{per}}(\cdot), c)$
    $P_j^{[k]} \leftarrow$ $\text{argmin}_{\Gamma \in \{Q, P_j^{[k]}\}}(J_j^{\mathsf{per}}(\texttt{Wp2C}(\Gamma), \{\texttt{Wp2C}(P_i^{[k]})\}_{-j}))$
  end
end
for $i \in \mathcal{N}$ **do**
  $z_i^{\text{lin}} \leftarrow \texttt{Wp2C}(P_i^{[k]})$
end
return z^{lin}

In order to find a ε-NE, $J_j(\texttt{Wp2C}(P^{[k]}))$ must never increase for any $j \in \mathcal{N}$. Then, we get Invariant 1.

Invariant 1. *For every $k > 0$ and $\forall j \in \mathcal{N}$,*

$$J_j(\texttt{Wp2C}(P^{[k]})) \preceq J_j(\texttt{Wp2C}(P^{[k-1]}))$$

Since the cost of each agent never increases in each iteration, then when the agents can non longer update their actions improve their cost, then the algorithm terminates and returns the approximate linear equilibrium.

Outline of Section III: In Section III-A, we first discuss the construction of PWL curves given a sequence of waypoints. In Section III-B, we discuss the construction of the bounding boxes that ensure that there are no collisions between agents. In Section III-C, we propose a linear formulation for computing the best response of agent i to z_{-i} by encoding constraints that guarantee $J_i^{\mathsf{col}}(z_i, z_{-i})$ is minimized, and choosing z_i that minimizes $J_i^{\mathsf{per}}(z_i, z_j)$ while satisfying our linear constraints. However, the set of actions that satisfies the linear constraints is an *under-approximation* of the true best response set. Thus, we will show that L-IBR can find the ε-NE, and we find the asymptotic lower bound of ε.

Geometric preliminaries: Given two points $p, p' \in \mathbb{R}^d$, a line segment is denoted by $\overline{pp'} \subset \mathbb{R}^d$. A ball of radius $c \geq 0$ centered at the point $p \in \mathbb{R}^d$ is denoted by $\mathcal{B}_c(p)$. The set of all points within c distance of a curve $z : [0, s] \to \mathbb{R}^d$ is given by the set $\mathcal{B}_c(z) := \bigcup_{s \in [0,1]} \mathcal{B}_c(z(s))$. Similarly, the set of all points within c distance of a line segment $\overline{pp'}$ is given by the set $\mathcal{B}_c(\overline{pp'}) := \bigcup_{q \in \overline{pp'}} \mathcal{B}_c(q)$

A. Constructing piecewise linear actions from waypoints

Recall from Definition 1 that the action space $\mathcal{Z}_i$ consists of uniformly continuous curves $z_i : [0, 1] \to \mathbb{R}^d$. In L-IBR, we use piecewise linear (PWL) curves constructed from a sequence of $m+1$ waypoints in $\mathbb{R}^d$ as follows: Given sequence of points $P_i = \{p_0, \ldots, p_m\}$, where each $p_j \in \mathbb{R}^d$, the PWL function z_i is constructed by mapping the domain $[0, 1]$ to the m line segments $\overline{p_0 p_1}, \ldots, \overline{p_{m-1} p_m}$. The PWL function is given by $\texttt{Wp2C}(P_i) : [0, 1] \to \mathbb{R}^d$. The domain is split into m sections $[t_{j-1}, t_j]$, $j \in [1, m]$, and each $\texttt{Wp2C}(P_j)(t_j) = p_j$. Over some duration $[t_{j-1}, t_j]$, $\texttt{Wp2C}(P_i)(s) \in \overline{p_{j-1} p_j}$, and for any $s, \sigma \in [t_{j-1}, t_j]$, $\sigma > s$, $\|\texttt{Wp2C}(P_i)(s) - p_{j-1}\| < \|\texttt{Wp2C}(P_i)(\sigma) - p_{j-1}\|$. Many different PWL functions could be constructed with different $\frac{\partial z_i}{\partial t}$ values ($t \in [0, 1]$), meaning that the linearity of the functions comes from their construction in $\mathbb{R}^d$. We fix any arbitrary mapping Wp2C throughout this paper and the resulting curve is written as $z_i = \texttt{Wp2C}(P_i)$. An example is shown in Figure 2. Note that the resulting PWL curves are uniformly continuous, and thus satisfy action space assumption in Definition 1. This can be seen in Figure 2. Note that the resulting curve is uniformly continuous, but possible non-differentiable at each disjunction point p_j, $j \in \{1, \ldots, m\}$.

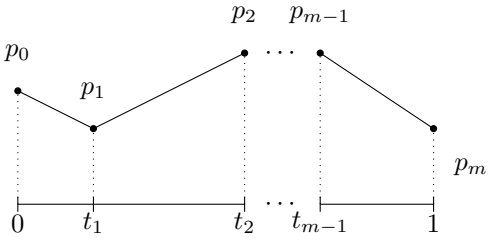

Fig. 2. Example of the Wp2C function converting a sequence of waypoints to a piecewise linear function. Each point $p_j \in \mathbb{R}^d$, and the associated time stamp for each point is shown in the timeline below.

Proposition 4. *For any sequence of waypoints $P_i \in \prod_{j=0}^{m} \mathbb{R}^d$, $z_i = \texttt{Wp2C}(P_i)$ is uniformly continuous.*

Now that we have defined the PWL curves, we will use them to approximate the nonlinear curves in $\mathcal{Z}_i$. Let $z_i : [0,1] \to \mathbb{R}^d$ be the uniformly continuous curve we are trying to approximate, and P_i be a sequence of $m+1$ waypoints in $\mathbb{R}^d$. The approximation error between $\texttt{Wp2C}(P_i)$ and z_i is given by $\delta = \max_{s\in[0,1]} \|\texttt{Wp2C}(P_i)(s) - z_i(s)\|$. Ideally, the P_i that we use to approximate z_i minimizes δ.

Lemma 1. *For any z_i, and $\delta > 0$, there exists $m \in \mathbb{N}$ and $P_i = \{p_0, \dots, p_m\}$ such that $\|\texttt{Wp2C}(P_i)(s) - z_i(s)\| \leq \delta$.*

We omit the proof for brevity, but it follows from the fact that for any uniformly continuous function z_i, and any $s, t \in [0,1]$, $s \neq t$, there exists $\zeta > 0$ and $\xi > 0$ such that $\|s - t\| < \zeta \implies \|z_i(s) - z_i(t)\| < \xi$.

Now that we have bound the error between $\texttt{Wp2C}(P_i)$ and any action z_i, we can begin to find a bound on ε. Here, we use the assumption that $J_i^{\mathsf{per}}(z_i, \{z_j\}_{-i})$ is Lipschitz continuous with constant K in the first argument. Then, $\varepsilon = K\delta$.

Lemma 2. *For personal cost functions $J_i^{\mathsf{per}}(z_i, \{z_j\}_{-i})$, which are Lipschitz continuous with respect to the first argument with Lipschitz constant K, $z_i \in \mathcal{Z}_i$, $\{z_j\}_{-i}$, and $\delta > 0$, there exists $m \in \mathbb{N}$ and $P_i = \{p_0, \dots, p_m\}$ such that $\|J_i^{\mathsf{per}}(z_i, \{z_j\}_{-i}) - J_i^{\mathsf{per}}(\texttt{Wp2C}(P_i), \{z_j\}_{-i})\| \leq K\delta$.*

Proof: Fix an arbitrary $\delta > 0$ and $\{z_{-i}\}$. By Lemma 1, we know that there exists $m > 0$ and $P_i = \{p_0, \dots, p_m\}$ such that $\|\texttt{Wp2C}(P_i)(s) - z_i(s)\| \leq \delta$ for all $s \in [0,1]$. Thus, by the Lipschitz continuity of J_i^{per}, $\|J_i^{\mathsf{per}}(\texttt{Wp2C}(P_i), \{z_{-i}\}) - J_i^{\mathsf{per}}(z_i, \{z_{-i}\})\| \leq K\delta$. ∎

B. Constructing bounding boxes to avoid collisions

A common type of collision cost used in autonomous driving and platooning scenarios is one in which the agents must remain at least c away from each other. Thus, we use the following definition of a collision. Two agents i and j collide if their trajectories z_i and z_j come within c of each other. The collision cost is minimized when z_j is at least c away from z_i. Thus, given an agent $j \in \mathcal{N}$, $J_j^{\mathsf{col}}(z_j, z_{-j})$ is minimized when,

$$\forall i \in \{-j\},\ \mathcal{B}_c(z_i) \cap z_j = \varnothing \tag{7}$$

Note that this checks for collisions over the entire trajectory, not just collisions in time. However, by splitting each trajectory into shorter trajectories in time, then this would check for collisions in time. In our collision heuristic, we consider the collision cost to be minimized when the agents remain at least c away from each other. A common way of checking for collisions is checking for intersections between bounding boxes. In this section we will discuss how to construct the bounding boxes used.

Each segment in a PWL trajectory has its own bounding box. Consider two points $p, p' \in \mathbb{R}^d$. The minimum bounding box for this segment is given by $\mathcal{O}_c(p,p') = \{y \in \mathbb{R}^d | \mathbf{A}y < \mathbf{b}\}$, where $\mathbf{A} \in \mathbb{R}^{d\times 2d}$, $\mathbf{b} \in \mathbb{R}^{2d}$. Additionally, $\mathcal{O}_c(p,p') \supset \mathcal{B}_c(\overline{pp'})$ and there is no other $\mathcal{O}'_c(p,p') \supset \mathcal{B}_c(\overline{pp'})$ such that $\mathcal{O}_c(p,p') \supset \mathcal{O}'_c(p,p')$. In this section, we will show how we construct such $\mathcal{O}_c(p,p')$. For convenience in this section, we will make the dependence on p, p' implicit.

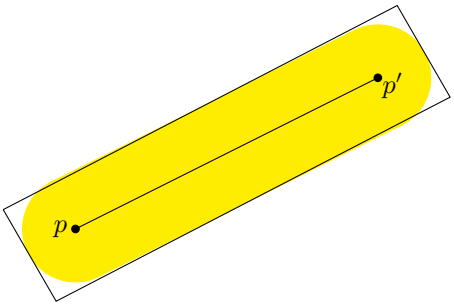

Fig. 3. Example of a bounding box in $\mathbb{R}^2$. The yellow tube is $\mathcal{B}_c(p,p')$, and the rectangle drawn around it is the bounding box.

The line segment that connects p and p' is given by $\overline{pp'}$. The vector that is tangent to this line is given by $\overrightarrow{n}_0 = \frac{p'-p}{\|p'-p\|}$. Then, in order to construct a bounding box, we choose any basis that spans $\mathcal{W}$ that contains $\overrightarrow{n}_0$. This basis is given by $\{\overrightarrow{n}_0, \dots, \overrightarrow{n}_{d-1}\}$ where each $\overrightarrow{n}_i$ is a unit vector.

The bounding boxes are defined by half-spaces. Each of these half-spaces is indexed by $\{0, 0', \dots, (d-1), (d-1)'\}$, meaning

$$\mathbf{A} = \begin{bmatrix} A_0 \\ A_{0'} \\ \vdots \\ A_{(d-1)} \\ A_{(d-1)'} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_0 \\ b_{0'} \\ \vdots \\ b_{(d-1)} \\ b_{(d-1)'} \end{bmatrix}$$

Here, each $A_i \in \mathbb{R}^d$, and $b \in \mathbb{R}$. Let us construct the i^{th} half-space. Let $\mathcal{A}_i y = \beta_i$ be the hyperplane spanned by $\{\overrightarrow{n}_{-i}\}$ centered on $p' + c\overrightarrow{n}_i$. Then, the half-space is either given by $\mathcal{A}_i y < \beta_i$ or $-\mathcal{A}_i y < -\beta_i$, whichever contains $\overline{pp'}$. We will call this half space $\{A_i y < b\}$, where $A_i = \pm\mathcal{A}_i$ and $b_i = \pm\beta_i$.

Similarly, $\mathcal{A}_{i'} y = \beta_{i'}$ is the hyperplane spanned by $\{\overrightarrow{n}_{-i}\}$ centered on $p - c\overrightarrow{n}_i$, and the half-space is either given by $\mathcal{A}_{i'} y \leq \beta_{i'}$ or $-\mathcal{A}_{i'} y < -\beta_{i'}$, whichever contains $\overline{pp'}$. Note that if the i^{th} half-space is given by $\pm\mathcal{A}_i y \leq \pm\beta_i$, then the i'^{th} half-space is given by $\mp\mathcal{A}_{i'} y < \mp\beta_{i'}$. We call this half-space $\{A_i y < b_i\}$.

Then, the bounding boxes are defined as $\mathcal{O}_c(p,p') = \{y \in \mathbb{R}^d | \mathbf{A}y < \mathbf{b}\}$. An example of a bounding box can be seen in Figure 3. Note that each $A_i y = b_i$, $A_{i'} y = b_{i'}$ are parallel, and any $A_i y = b_i$ and $A_j y = b_j$, $j \in \{-i\}$, $j \neq i'$ are orthogonal to each other.

Proposition 5. *For any $c > 0$, $p, p' \in \mathbb{R}^d$, $q, q' \in \mathbb{R}^d$ we note the following:*

1) *if $A_i q \geq b_i$ for some $i \in [0, d-1]$, then $A_{i'} q < b_{i'}$*
2) *$\exists q \in \mathbb{R}^d$ such that for some $i, j \in [0, d-1]$ and $j \neq i'$, $A_i q \geq b_i$ and $A_j q \geq b_j$.*

3) *if* $A_i q \geq b_i$ *and* $A_i q' \geq b_i$ *for some* $i \in [0, d-1]$, *then* $\overline{qq'} \cap \mathcal{O}_c = \varnothing$
4) *there exists* $\overline{qq'}$ *such that* $\overline{qq'} \cap \mathcal{O}_c = \varnothing$ *that does not satisfy (2).*
5) *for any* $\overline{qq'}$ *described in (3), there exists* $q'' \in \mathbb{R}^d$ *such that* $\overline{qq''}$ *and* $\overline{q''q'}$ *satisfy (2).*

Note that these propositions still hold true if i and i' are switched.

Proof:

Proposition 5.1: Consider the half-spaces defined by $A_i y < b_i$ and $A_{i'} y < b_{i'}$. We know from the construction of the half-spaces that $\overline{pp'} \subset \{A_i y < b_i\} \cap \{A_{i'} y < b_{i'}\}$. This means that $\{A_i y \geq b_i\} \subset \{A_{i'} y < b_{i'}\}$. Therefore, if $A_i q \geq b_i$, then $A_{i'} q < b_{i'}$.

Proposition 5.2: Consider two half-spaces defined by $A_i y < b_i$ and $A_j y < b_j$. Since $\{A_i y = b_i\}$ and $\{A_j y = b_j\}$ are orthogonal to each other, $\{A_i y \geq b_i\} \cap \{A_j y \geq b_j\} \neq \varnothing$. Therefore, there exists $q \in \{A_i y \geq b_i\} \cap \{A_j y \geq b_j\}$, and the proposition holds.

Proposition 5.3: Consider some q, q' such that $A_i q \geq b_i$ and $A_i q' \geq b_i$. Then, it is easy to see that $\overline{qq'} \cap \mathcal{O}_c = \varnothing$.

Propositions 5.4 and 5.5: Consider some $q \in \mathbb{R}^d$ and $i \in [0, d-1]$ such that $A_i q \geq b_i$, $q'' \in \mathbb{R}^d$ and $j \neq i$ such that $A_i q'' \geq b_i$ and $A_j q'' \geq b_j$, and $q' \in \mathbb{R}^d$ such that $A_j q'' \geq b_j$. Then, by Proposition 5.2, we can see that $\overline{qq''} \cap \mathcal{O}_c = \varnothing$ and $\overline{q''q'} \cap \mathcal{O}_c = \varnothing$. If $q'' \in \overline{qq'}$, then we can see that $\overline{qq'} \cap bbox_c = \varnothing$, and thus, Propositions 5.4 and 5.5 hold. ∎

Since $\mathcal{O}_c$ is an over-approximation of $\mathcal{B}_c(pp')$, we note there is a modification on the smallest δ that can be achieved in Lemma 1. This modification is given in Lemma 3.

Lemma 3. *For any* $p, p' \in \mathbb{R}^d$, *bounding box* $\mathcal{O}_c(p, p') \subset \mathbb{R}^d$, *and action* $z_i \in \mathcal{Z}_i$ *such that* $\mathcal{B}_c(\overline{pp'}) \cap z_i(s) = \varnothing$ *for all* $s \in [0, 1]$, *and* $\delta > c(\sqrt{d} - 1)$, *there exists* $m \in \mathbb{N}$ *and* $Q = \{q_0, \ldots, q_m\}$ *such that* $\|\texttt{Wp2C}(Q)(s) - z_i(s)\| \leq \delta$.

The proof follows from the fact that for any point $q \in \mathbb{R}^d$ that lies on the boundary of $\mathcal{B}_c(\overline{pp'})$, there exists a point q' on the corresponding bounding box $\mathcal{O}_c$ such that $\|q' - q\| \leq c(\sqrt{d} - 1)$.

C. Linear formulation for approximating $\mathcal{R}^{BR}$

In Line 5 of Algorithm 2, given the current guess $\{P_i\}_{\mathcal{N}}$, a candidate sequence of waypoints Q for agent j is computed. In this section, we present a mixed integer linear program (MILP), that minimizes $J_j^{\mathsf{col}}(\texttt{Wp2C}(P_j), \{\texttt{Wp2C}(P_i\}_{-j})$ via the encoded constraints.

Recall the collision constraint in (7). In the case of our PWL curves, this can be rewritten as in (8) for any waypoint sequences $\{P_j\}_{\mathcal{N}}$. Given an agent $j \in \mathcal{N}$, $J_j^{\mathsf{col}}(\texttt{Wp2C}(P_j), \{\texttt{Wp2C}(P_i)\}_{-j})$

$$\forall i \in \{-j\},\ \mathcal{B}_c(\texttt{Wp2C}(P_i)) \cap \texttt{Wp2C}(P_j) = \varnothing \tag{8}$$

Now we will walk through the steps of computing z_j for $\texttt{linUpdate}(\{P_i\}_{-j}, J_j^{\mathsf{per}}(\cdot), c)$ in Line 5 of Algorithm 2. First, we check that individual segments

Consider the waypoints $p, p', q, q' \in \mathbb{R}^d$. Given a constant $c > 0$, the segments $\overline{pp'}$ and $\overline{qq'}$ are at least c away from each other if $\mathcal{B}_c(\overline{pp'}) \cap \overline{qq'} = \varnothing$. However, the formulation to check this is not linear. Thus, we use the minimum bounding boxes $\mathcal{O}_c(p, p') = \{y \in \mathbb{R}^d | \mathbf{A}y < \mathbf{b}\}$ introduced in Section III-B to check for collisions.

The variable α is an array of binary variables that determine if q, q' lie outside the same face of $\mathcal{O}_c(p, p')$. The $\Lambda >> 0$ is used to encode disjunctions.

$$\begin{aligned}\texttt{SafeSegment}(p, p', c) = \\ \bigwedge_{r \in \{0, 0', \ldots, d-1, d-1'\}} \Big(& b_r - A_r q < \Lambda(1 - \alpha_r) \\ & \wedge\ b_r - A_r q' < \Lambda(1 - \alpha_r) \Big) \\ \wedge \sum_{r \in \{0, 0', \ldots, d-1, d-1'\}} & \alpha_r \geq 1\end{aligned} \tag{9}$$

By writing the collision constraints in this way, we can see that the resulting formulation is linear and contains no disjunctions.

Lemma 4. *Given* $c > 0$ *and* $p, p' \in \mathbb{R}^d$, *for any* $q, q' \models \texttt{SafeSegment}(p, p', c)$

$$\mathcal{B}_c(\overline{pp'}) \cap \overline{qq'} = \varnothing$$

Proof: From Proposition 5.1, we know that if q, q' lie outside the same half-space of $\mathcal{O}_c(p, p')$, then $\overline{qq'} \cap \mathcal{O}_c(p, p') = \varnothing$, and since $\mathcal{B}_c(\overline{pp'}) \subset \mathcal{O}_c$, this also means $\mathcal{B}_c(\overline{pp'}) \cap \overline{qq'} = \varnothing$. Thus, in this proof, we will show that q, q' lie outside the same half-space of $\mathcal{O}_c(p, p')$.

Choose some $p \in \mathbb{R}^d s$ such that $A_r q \geq b_r$ for some $r \in \{0, 0', \ldots, (d-1), (d-1)'\}$. Then, $b_r - A_r q \leq 0$, and $\alpha_r \in \{0, 1\}$. Now choose $q' \in \mathbb{R}^d$. If $A_r q' \geq b_r$, then $b_r - A_r q' \leq 0$, and $\alpha_r = 1$. However, if $A_r q' < b_r$, then $b_r - A_r q' > 0$, and $\alpha_r = 0$. By the third constraint, $\sum_{r \in \{0, 0', \ldots, (d-1), (d-1)'\}} \alpha_r \geq 1$, meaning if $q, q' \models \texttt{SafeSegment}(p, p', c)$, then at least one $\alpha_r = 1$, and therefore $A_r q' \geq b_r$ and $A_r q \geq b_r$. Thus, $\mathcal{B}_c(\overline{pp'}) \cap \overline{qq'} = \varnothing$. ∎

Now we check that a sequence of waypoints Q is at least c away from another sequence of waypoints P. For any sequence of $m+1$ waypoints $P = \{p_0, \ldots, p_m\}$ and any sequence of $n+1$ waypoints $Q = \{q_0, \ldots, q_n\}$, we check that $\texttt{Wp2C}(Q)$ is at least c away from $\texttt{Wp2C}(P)$. Here, we use the fact that if $q_{i-1}, q_i \models \bigwedge_{j=1}^{m} \texttt{SafeSegment}(p_{j-1}, p_j, c)$ for every $i = 1, \ldots, n$, then we say that $Q \models \texttt{SafeSequence}(P, c)$, which is shown in (10).

$$\texttt{SafeSequence}(P, c) = \bigwedge_{i=1}^{n} (q_{i-1}, q_i) \models \bigwedge_{j=1}^{m} \texttt{SafeSegment}(p_{j-1}, p_j, c) \tag{10}$$

Corollary 1. *For any $c > 0$, any sequence of $m+1$ waypoints P, and any sequence of $n+1$ waypoints Q, if for all $i = 1, \dots, n$*

$$Q \models \texttt{SafeSequence}(P, c)$$

then

$$\mathcal{B}_c(\texttt{Wp2C}(P)) \cap \texttt{Wp2C}(Q) = \varnothing$$

Proof: By Lemma 4, we know that if $(q_{i-1}, q_i) \models \texttt{SafeSegment}(p_{j-1}, p_j)$, then $\mathcal{B}_c(\overline{p_{j-1}p_j}) \cap \overline{q_{i-1}q_i} = \varnothing$. Thus, if $(q_{i-1}, q_i) \models \bigwedge_{j=1}^{m} \texttt{SafeSegment}(p_{j-1}, p_j)$, then

$$\bigcup_{i \in \{0,\dots,m-1\}} \mathcal{B}_c(\overline{p_i p_{i+1}}) \cap \overline{q_j q_{j+1}} = \varnothing$$

Furthermore, if this is true for every $i = 1, \dots, m$, then

$$\bigcup_{i \in \{0,\dots,m-1\}} \mathcal{B}_c(\overline{p_i p_{i+1}}) \cap \bigcup_{j \in \{0,\dots,n-1\}} \overline{q_j q_{j+1}} = \varnothing$$

Due to the construction of $\texttt{Wp2C}(\cdot)$, we can see this is equivalent to

$$\mathcal{B}_c(\texttt{Wp2C}(P)) \cap \texttt{Wp2C}(Q) = \varnothing$$

■

The MILP formulation to find the sequence of waypoints $\{Q\}$ that minimizes some cost function $J_i^{\text{col}}(\texttt{Wp2C}(Q)), \{\texttt{Wp2C}(P_i)\}_{-i})$ is given in (11).

$$\begin{aligned} &\texttt{linUpdate}(\{P_j\}_{-i}, J_i^{\text{per}}(\cdot), c) = \\ &\qquad \arg\min_{Q} J_i^{\text{per}}(\texttt{Wp2C}(Q), \{\texttt{Wp2C}(P_j)\}_{-i}) \\ &\qquad Q \models \bigwedge_{j \in \{-i\}} \texttt{SafeSequence}(P_j, c) \end{aligned} \tag{11}$$

Corollary 2. *For any $c > 0$, any $\{P_j\}_{-j}$, and any $Q \models \bigwedge_{j \in \{-i\}} \texttt{SafeSequence}(P_j, c)$,*

$$\mathcal{B}_c(\texttt{Wp2C}(P_j)) \cap \texttt{Wp2C}(Q) = \varnothing$$

for every $j \in \{-i\}$.

Proof: Consider some sequence of $m+1$ waypoints Q that satisfies our constraints in $\texttt{linUpdate}(\{P_j\}_{-i}, J_i^{\text{per}}(\cdot), c)$. By Corollary 1, if $\texttt{SafeSequence}(P_j, c)$ is satisfied for every $j = \{-i\}$, then it is easy to see that

$$\forall j \in \{-i\},\ \mathcal{B}_c(\texttt{Wp2C}(P_j)) \cap \texttt{Wp2C}(Q) = \varnothing$$

■

Thus, we can see that any Q returned by $\texttt{linUpdate}(\cdot)$ minimizes $J_i^{\text{col}}(\cdot)$. Note that the Q found using $\texttt{linUpdate}$ is not necessarily the Q that minimizes $J_i(\cdot)$ while satisfying (8) (due to our linear approximation). However, combining all parts above, we can find a bound on the ε error in L-IBR. This is shown in Theorem 1.

Theorem 1. *For some agent $i \in \mathcal{N}$ with personal cost function $J_i^{\text{per}}(z_i, \{z_j\}_{-i})$, which is Lipschitz continuous with respect to the first argument with Lipschitz constant K, some $\{P_j\}_{-i}$, and som constant $c > 0$ such that for any $\varepsilon > Kc(\sqrt{d} - 1)$ there exists $m \in \mathbb{N}$ and $Q = \{q_0, \dots, Q_{m_i}\}$ found using $\texttt{linUpdate}(\{P_j\}_{-i}, J_i^{\text{per}}(\cdot), c)$ such that*

$$\begin{aligned} J_i^{\text{per}}(\texttt{Wp2C}(Q), \{\texttt{Wp2C}(P_j\}_{-i}) \leq \\ J_i^{\text{per}}(z_i^*, \{\texttt{Wp2C}(P_j)\}_{-i}) + \varepsilon \end{aligned}$$

where z_i^* is the best response of agent i to $\{\texttt{Wp2C}(P_j)\}_{-i}$.

Proof: Fix some $c > 0$ and $\{P_j\}_{-i}$. Then, from Lemma 3, we know that in our MILP formulation, the smallest error between any P_i and z_i is $c(\sqrt{d} - 1)$. From Lemma 2, we can then see that for this Q, $\|J_i^{\text{per}}(\texttt{Wp2C}(Q), \{\texttt{Wp2C}(P_j\}_{-i}) - J_i^{\text{per}}(\texttt{Wp2C}(Q^*), \{\texttt{Wp2C}(P_j\}_{-i})\| \leq Kc(\sqrt{d} - 1)$ ■

Thus, we can see that as the number of segments to approximate the agent trajectories increases, the algorithm returns a joint action closer to the NE. In Section IV, we will see this tradeoff.

IV. Experimental results

Implementation details: We implement L-IBR using Python 3 and Gurobi. We run the intersection examples seen in Figure 1. The collision cost function remains the same as in (1). In these experiments, we vary the number of agents and the number of segments in the PWL trajectories. Each individual agents cost function is the distance traveled. The results of the experiments are seen in Table I. Each agent's individual cost function is the distance traveled. Some example results can be seen in Figure 4

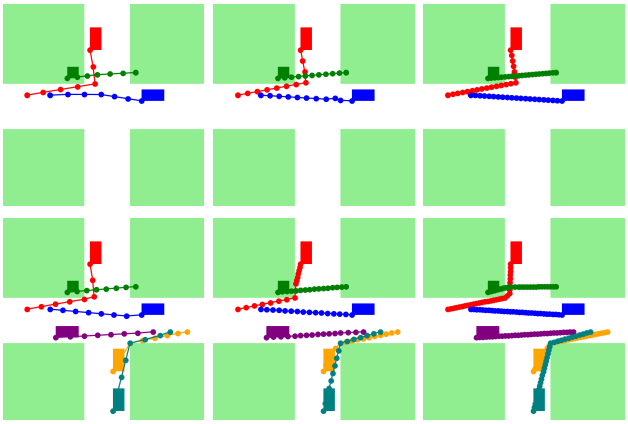

Fig. 4. Example results from the examples. Note that across the varying number of segments used to approximate the PWL paths, the actual paths taken by the agents remains the same.

In Table I, we compare the computational time, and the sum of all the agents' individual costs. Here, we can see that the time to compute the linear NE increases with the number of agents in the scenario and the number of segments used to approximate the trajectories. Additionally, the total cost of the agents individual cost functions decreases as the number of

Scenario	Num agents	Num segments	Total cost	Comp time (s)
Ped, 2 Car	3	20	2.268	1.277
Ped, 2 Car	3	10	4.584	0.578
Ped, 2 Car	3	6	7.684	0.427
Ped, 5 car	6	30	3.395	65.16
Ped, 5 car	6	15	6.730	3.567
Ped, 5 car	6	7	14.981	7.168

TABLE I
RESULTS OF L-IBR FOR AN INTERSECTION SCENARIO.

segments to approximate the trajectories increases, which is in line with our main result from Theorem 1. Thus, we can see that there is a tradeoff between the computation time, and how closely we can achieve the true NE. Note that in the ped, 5 car scenario with 15 segments, the computation time is lower than the scenario with 7 segments. This is due to the number of iterations run to find the linear NE.

V. CONCLUSION

In this work, we presented a formulation for a lexicographic general sum game. This game is shown to be a potential game, and thus possesses a pure strategy Nash equilibrium. We then presented an iterative best response algorithm that linearized the constraints to find actions that minimized the first part of the lexicographic cost function, thus making it simpler to find actions that would also minimize the second part of the lexicographic cost function. Finally, we presented some experiments and showed that our prototype tool could find solutions to our game.

In the future, we can study how to relax the type of collision costs we are minimizing for to explore a larger variety of games. Additionally, we can use this problem set up to study the types equilibria we get by solving a long single shot game versus solving a sequence of shorter multi-stage games. Finally, we can explore equilibria selection to help with the design of cost functions.

REFERENCES

[1] A. M. Ayala, S. B. Andersson, and C. Belta. Temporal logic motion planning in unknown environments. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5279–5284. IEEE, 2013.

[2] A. Bajcsy, S. L. Herbert, D. Fridovich-Keil, J. F. Fisac, S. Deglurkar, A. D. Dragan, and C. J. Tomlin. A scalable framework for real-time multi-robot, multi-human collision avoidance. In *2019 international conference on robotics and automation (ICRA)*, pages 936–943. IEEE, 2019.

[3] A. Bemporad, F. Borrelli, and M. Morari. Model predictive control based on linear programming - The explicit solution. *IEEE TAC*, 47(12):1974–1985, 2002.

[4] B. Berckmoes. An arzel\a-ascoli theorem for the hausdorff measure of noncompactness. *arXiv preprint arXiv:1303.2368*, 2013.

[5] G. E. Fainekos, A. Girard, H. Kress-Gazit, and G. J. Pappas. Temporal logic motion planning for dynamic robots. *Automatica*, 45(2):343–352, 2009.

[6] C. Fan, K. Miller, and S. Mitra. Fast and guaranteed safe controller synthesis for nonlinear vehicle models. In *CAV*, pages 629–652. Springer, 2020.

[7] E. Frazzoli, M. A. Dahleh, and E. Feron. Real-time motion planning for agile autonomous vehicles. *Journal of guidance, control, and dynamics*, 25(1):116–129, 2002.

[8] D. Fridovich-Keil, E. Ratner, L. Peters, A. D. Dragan, and C. J. Tomlin. Efficient iterative linear-quadratic approximations for nonlinear multi-player general-sum differential games. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 1475–1481. IEEE, 2020.

[9] D. Fudenberg and J. Tirole. *Game theory*. MIT press, 1991.

[10] S. L. Herbert, M. Chen, S. Han, S. Bansal, J. F. Fisac, and C. J. Tomlin. Fastrack: A modular framework for fast and guaranteed safe motion planning. In *CDC*, pages 1517–1522. IEEE, 2017.

[11] J. P. Hespanha. *Noncooperative Game Theory*. Princeton University Press, 2017.

[12] R. Isaacs. *Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization*. Courier Corporation, 1999.

[13] D. K. Jha, M. Zhu, and A. Ray. Game theoretic controller synthesis for multi-robot motion planning-part ii: Policy-based algorithms. *IFAC-PapersOnLine*, 48(22):168–173, 2015.

[14] F. Laine, D. Fridovich-Keil, C.-Y. Chiu, and C. Tomlin. The computation of approximate generalized feedback nash equilibria. *arXiv preprint arXiv:2101.02900*, 2021.

[15] S. M. LaValle. *A game-theoretic framework for robot motion planning*. PhD thesis, University of Illinois at Urbana-Champaign, 1995.

[16] A. Majumdar and R. Tedrake. Funnel libraries for real-time robust feedback motion planning. *The International Journal of Robotics Research*, 36(8):947–982, 2017.

[17] D. Q. Mayne. Model predictive control: Recent developments and future promise. In *Automatica*, volume 50, pages 2967–2986. Pergamon, 2014.

[18] K. Miller, C. Fan, and S. Mitra. Planning in dynamic and partially unknown environments. *IFAC-PapersOnLine*, 54(5):169–174, 2021.

[19] J. Nash. Non-cooperative games. *Annals of mathematics*, pages 286–295, 1951.

[20] N. D. Stein, A. Ozdaglar, and P. A. Parrilo. Separable and low-rank continuous games. *International Journal of Game Theory*, 37(4):475–504, 2008.

[21] D. Sun, J. Chen, S. Mitra, and C. Fan. Multi-agent motion planning from signal temporal logic specifications. *arXiv preprint arXiv:2201.05247*, 2022.

[22] J. Van Den Berg, D. Ferguson, and J. Kuffner. Anytime path planning and replanning in dynamic environments. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2366–2371. IEEE, 2006.

[23] S. Vaskov, S. Kousik, H. Larson, F. Bu, J. Ward, S. Worrall, M. Johnson-Roberson, and R. Vasudevan. Towards provably not-at-fault control of autonomous robots in arbitrary dynamic environments. *arXiv preprint arXiv:1902.02851*, 2019.

[24] M. Vitus, V. Pradeep, G. Hoffmann, S. Waslander, and C. Tomlin. Tunnel-milp: Path planning with sequential convex polytopes. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, page 7132, 2008.

[25] A. Zanardi, S. Bolognani, A. Censi, and E. Frazzoli. Game theoretical motion planning: Tutorial icra 2021. 2021.

[26] A. Zanardi, E. Mion, M. Bruschetta, S. Bolognani, A. Censi, and E. Frazzoli. Urban driving games with lexicographic preferences and socially efficient nash equilibria. *IEEE Robotics and Automation Letters*, 6(3):4978–4985, 2021.

[27] M. N. Zeilinger, C. N. Jones, and M. Morari. Real-time suboptimal model predictive control using a combination of explicit MPC and online optimization. *IEEE TAC*, 56(7):1524–1534, 2011.

[28] M. Zhu, M. Otte, P. Chaudhari, and E. Frazzoli. Game theoretic controller synthesis for multi-robot motion planning part i: Trajectory based algorithms. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1646–1651. IEEE, 2014.

[29] M. Zucker, J. Kuffner, and M. Branicky. Multipartite rrts for rapid replanning in dynamic environments. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 1603–1609. IEEE, 2007.