2022 IEEE 61st Conference on Decision and Control (CDC)
December 6-9, 2022. Cancún, Mexico

Multi-agent motion planning using differential games with lexicographic preferences

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

2022 IEEE 61st Conference on Decision and Control (CDC) | 978-1-6654-6761-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/CDC51059.2022.9992957

***Abstract*—Multi-player games with lexicographic cost functions can capture a variety of driving and racing scenarios and are known to have pure-strategy Nash Equilibria (NE) under certain conditions. The standard Iterated Best Response (IBR) procedure for finding such equilibria can be slow because computing the best response for each agent generally 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 enables us to approximate the NE using a linearized version of IBR. We show the gap between the linear approximations returned by our linearized IBR and the true best response drops asymptotically. We implement the algorithm and show that it can find approximate NE for a handful of agents driving in realistic scenarios in under 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. Single agent planning algorithms can be ported to multi-agent problems under restrictive assumptions such as accurate predictions for agents or no interactions. Treating other agents as dynamic but known obstacles, a variety of motion strategies have been used such as sampling-based planners [23, 17, 6], temporal logic-based planers [16, 5, 1], reach-avoid synthesis [13, 2, 18], and model predictive control [12, 3, 21]. These can provide safe trajectories for navigation, but typically do not account for dynamic interaction between agents.

Differential games (DGs) [10] study strategic interactions between rational agents moving in some workspace. Using DGs to solve the multi-agent motion planning problem is a growing area of interest [7, 11, 22, 19]. A special class of games, called *Urban driving games (UDG)*, were introduced in [20] to study decision making in autonomous vehicles around other vehicles and pedestrians. This is different from typical DGs, as a two-part cost function is used where: (i) a *collision cost* is a shared cost which enforces no collision, and (ii) a *personal cost* dependent on an individual agent's actions, and can encode minimization of distance traveled, fuel expended, or time. Nash equilibria (NE) [14] is a standard way for evaluating steady-state behavior in games. At NE, no agent can improve their cost by unilaterally changing their action. Thus, each agent is playing its optimal response to the action profile of the other agents. In [20], it is shown that NE exists for UDG.

The authors acknowledge support from the research grants FMITF NSF CCF 19-18531. The views, opinions, and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the U.S. Government.

One limitation of a UDG is that the personal cost function only depends on each individual agent's actions. It 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 of a UDG, and add a zero sum component to the personal cost function and thus, can be used in a wider variety of scenarios. We show this revised UDG is still a potential game, and thus has a pure strategy NE. To find this NE, we propose a linearized version of an iterative best response algorithm using linear constraints to minimize the collision cost function, and search over these actions for the best response. The algorithm returns the piecewise linear approximation of the agents' trajectories that minimize the agent's personal cost function, giving rise to the notion of linear NE. We show that for a large enough number of segments, we can find the asymptotic bound on the gap between the true NE and returned approximation. To do this, we use a mixed integer linear program, similar to [13]. The prototype tool works in a variety of scenarios and can typically find the approximate NE within seconds.

II. Lexicographic games and Nash Equilibria

Sets such as $\mathcal{N} = \{0, \ldots, n-1\}$ index the players in the game. For $i \in \mathcal{N}$, $\{-i\}$ 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) $a_1 < a_2$ 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 a lexicographic general sum game, shown in Definition 1. This is a generalization of the urban driving game introduced in [20]. 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 *lexicographic general sum game (LG)* is given by $\mathcal{G} = \langle \mathcal{N}, \{\mathcal{Z}_i\}, \{J_i\} \rangle$, where: (i) $\mathcal{N} = \{0, \ldots, n-1\}$ is a set of n agents, (ii) $\mathcal{Z}_i$ is a compact set of uniformly continuous curves $z_i : [0,1] \rightarrow \mathbb{R}^d$, which specifies a trajectory (action) for agent i. The *joint action space* for $\mathcal{G}$

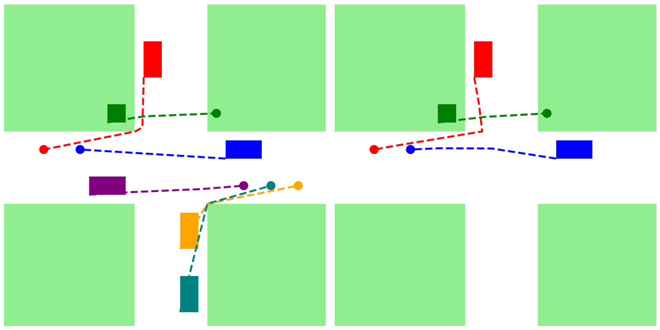

Fig. 1. Examples of general sum driving games at an intersection. Vehicles must navigate an intersection while avoiding collisions with each other and a pedestrian. The dotted trajectories are potential NE.

is $\mathcal{Z} = \prod_{i \in \mathcal{N}} \mathcal{Z}_i$, and a particular joint action $z \in \mathcal{Z}$ can be written as $z = [z_0, \ldots, z_{n-1}]$, 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* 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* 2 is defined in terms of individual costs g_i which are continuous and bounded over $\mathcal{Z}_i$.

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

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

The uniform continuity of each z_i implies that $\mathcal{Z}_i$ is compact by Arzela-Ascoli theorem [4]. The two part definition of the cost function provides some analytical advantages seen in Section III, and incentivizes certain outcomes using a collision cost dependent on action pairs, whereas each personal cost is dependent on an agent's action with respect to every other agents' actions.

Example: Examples of an LG at a traffic intersection are shown in Figure 1. Each agent lives in $\mathbb{R}^2$ is given a starting position (rectangles) and goal position (circles). The action spaces are the reference trajectories that the agents follow to reach the goal position. The collision cost function is given in (3), and the definition of a collision is later specified in (6). Individual costs may minimize the traveled distance or fuel consumption.

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

B. Best responses and Nash equilibria

Game theory can help us understand social behavior by predicting how agents will act in a game defined by the costs. Such predictions are called equilibria. One particular type is called a Nash equilibrium (NE) which is defined in terms of an agent's best response (Definition 2) to other's choices.

Definition 2 (Best response)**.** Given an agent $i \in \mathcal{N}$ and an action profile $z_{-i} \in \prod_{j \in \{-i\}} \mathcal{Z}_j$, the *best response* set of agent i is the set of actions that minimizes $J_i(z_i, z_{-i})$. 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 NE captures the idea that for some joint action, no agent can unilaterally improve their cost. John Nash proved in 1951 that all n-player games with *finite action sets* have a mixed-strategy NE [14][1] In this work, NE refers to a *pure strategy* NE, where agents select one action from $\mathcal{R}_i^{\text{BR}}(z_{-i})$, whereas for *mixed strategy* NE, 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 $\forall i \in \mathcal{N}$, $\forall z_i \in \mathcal{Z}_i$, $J_i(z^*) \preceq J_i(z_i, z_{-i}^*)$.

It follows 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 NE for general continuous games with continuous utility functions can be proven using a generalization of Kakutani's fixed point theorem [8]. While this existence result has been known, there is limited work on practical algorithms for computing NE. Several special classes of games have been identified, such as separable games [15], where the problem becomes tractable. For later use, we introduce the notion of an ε-NE of an LG 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.

$$\begin{aligned} J_i(z^\varepsilon) &\preceq \langle J_i^{\text{col}}(z_i, z_{-i}^\varepsilon) + \varepsilon, J_i^{\text{per}}(z_i, z_{-i}^\varepsilon) \rangle \\ J_i(z_i^\varepsilon) &\preceq \langle J_i^{\text{col}}(z_i, z_{-i}^\varepsilon), J_i^{\text{per}}(z_i, z_{-i}^\varepsilon) + \varepsilon \rangle. \end{aligned} \tag{4}$$

Problem Statement: In this paper, we would like to develop an algorithm that given an LG $\mathcal{G}$ and $\varepsilon > 0$, computes an ε-Nash equilibrium of $\mathcal{G}$. Computing the NE can be very difficult, especially in scenarios with a large (or infinite) action space, or when there are a large number of agents [20]. We aim to simplify the computation of NE through a linearized iterative best response. In Section II-C, we discuss the iterative best response framework for computing NE, and in Section III we present the linearized version. We then 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 NE. The procedure starts with an initial guess of a joint action $z \in \mathcal{Z}$ to be a candidate NE. Each agent's updates its action $z_i \in \mathcal{Z}_i$ to be in the best response set $\mathcal{R}_i^{\text{BR}}(z_{-i})$ until and unless there is no better response to the joint action for any of the agents. The standard procedure is shown Algorithm 1. The best response computation for each

[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.

agent involves solving an optimization problem, which can be challenging for general infinite games. Convergence is not guaranteed, however, if $\mathcal{G}$ is a *potential game* (Definition 4), convergence is guaranteed.

Algorithm 1: Iterative best response (IBR)

Input: $\mathcal{G}$, initial guess $z \in \mathcal{Z}$
Output: z^*
do
    $z^* \leftarrow z$
    for $i \in \mathcal{N}$ **do**
        $z_i \in \mathcal{R}_i^{\mathrm{BR}}(z_{-i})$
    end
while $z^* \neq z$;
return z^*

D. Lexicographic and potential games

We proceed by showing that an LG is an ordinal potential game[2], which are guaranteed to have pure strategy NE.

Definition 4 (Ordinal potential game)**.** An LG $\mathcal{G}$ is an *ordinal potential game (OPG)* if there exists a function $P : \mathcal{Z} \rightarrow \mathcal{X}$ 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. In Proposition 1, it is shown that any LG has an OPF. The proof is a modification of Theorem 1 from [20], and is included in the full version.

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

$$P(z) = \langle \frac{1}{2} \sum_{j \in \mathcal{N}} J_j^{\mathit{col}}(z), \sum_{j \in \mathcal{N}} g_j(z_j) \rangle. \tag{5}$$

The fact that an LG is an ordinal potential game leads to some nice properties. First, a global minimum of P exists [20], and corresponds to a pure strategy Nash equilibrium [9]. Secondly, an ε-NE can be computed using IBR. These results are stated as Propositions 2 and 3.

Proposition 2. *LGs have a pure strategy NE.*

For LGs, the action spaces $\mathcal{Z}_i$ are indeed compact, which means that for every $z_i \in \mathcal{Z}_i$ and every $\varepsilon > 0$, there exists $\delta > 0$ 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 [20], for continuous cost functions, a pure strategy NE exists.

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

[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.

The proof follows 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 Linear Iterated Best Response (L-IBR) that linearly searches for best responses. To do this, the best responses are approximated piecewise linear (PWL) functions described by sequences of waypoints in $\mathbb{R}^d$. Under additional approximations 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 true best response, and the gap between the cost of the returned action and the actual NE can be bounded.

The pseudocode for L-IBR is shown in Algorithm 2. It takes as input the LG $\mathcal{G}$ and an initial guess for the joint action described by a collection of sequences of waypoints $\{P_i^{[0]}\}_{\mathcal{N}}$, each $P_i^{[0]} = \{p_0, \cdots, p_m\}$, and a collision bound $c > 0$. In each iteration, each P_i is updated using linear constraints. This continues until the each P_i converges to an equilibrium, and the computed approximate linear equilibrium z^{lin} is returned.

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}
1 $k \leftarrow 1$
2 **while** $k = 1$ *or* $P^{[k]} \neq P^{[k-1]}$ **do**
3     $\{P_i^{[k]}\}_{\mathcal{N}} \leftarrow \{P_i^{[k-1]}\}_{\mathcal{N}}$
4   **for** $j \in \mathcal{N}$ **do**
5         $Q \leftarrow \mathtt{linUpdate}(\{P_i\}_{-j}, J_j^{\mathsf{per}}(\cdot), c)$
6         $P_j^{[k]} \leftarrow \mathrm{argmin}_{\Gamma \in \{Q, P_j^{[k]}\}} (J_j^{\mathsf{per}}(\mathtt{Wp2C}(\Gamma), \{\mathtt{Wp2C}(P_i^{[k]})\}_{-j}))$
7   **end**
8 **end**
9 **for** $i \in \mathcal{N}$ **do**
10     $z_i^{\mathrm{lin}} \leftarrow \mathtt{Wp2C}(P_i^{[k]})$
11 **end**
12 **return** z^{lin}

To find a ε-NE, $J_j(\mathtt{Wp2C}(P^{[k]}))$ must never increase for any $j \in \mathcal{N}$. This is enforced in Line 6, leading to Invariant 1. Thus, L-IBR terminates when the no actions can be updated to improve an agent's personal cost.

Invariant 1. *For every $k > 0$ and $\forall j \in \mathcal{N}$, $J_j(\mathtt{Wp2C}(P^{[k]})) \preceq J_j(\mathtt{Wp2C}(P^{[k-1]}))$*

First, we discuss the construction of PWL curves and the bounding boxes to check for collisions. We propose a linear formulation for computing the best response of agent i to z_{-i} by encoding constraints that minimize $J_i^{\mathsf{col}}(z_i, z_{-i})$, then choosing z_i that minimizes $J_i^{\mathsf{per}}(z_i, z_j)$ while satisfying our linear constraints. The set of actions that satisfies the linear constraints is an *under-approximation* of the true best response set, so we show that L-IBR can find the ε-NE, and 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 $\mathcal{B}_c(z) := \bigcup_{s \in [0,1]} \mathcal{B}_c(z(s))$. The set of all points within c distance of a line segment $\overline{pp'}$ is given by $\mathcal{B}_c(\overline{pp'}) := \bigcup_{q \in \overline{pp'}} \mathcal{B}_c(q)$.

A. Constructing piecewise linear actions from waypoints

Recall from Definition 1 that $\mathcal{Z}_i$ 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 $\mathtt{Wp2C}(P_i) : [0, 1] \to \mathbb{R}^d$. The domain is split into m sections $[t_{j-1}, t_j]$, $j \in [1, m]$, and each $\mathtt{Wp2C}(P_j)(t_j) = p_j$. For any $s \in [t_{j-1}, t_j]$, $\mathtt{Wp2C}(P_i)(s) \in \overline{p_{j-1} p_j}$, and for any $s, \sigma \in [t_{j-1}, t_j]$, $\sigma > s$, $\|\mathtt{Wp2C}(P_i)(s) - p_{j-1}\| < \|\mathtt{Wp2C}(P_i)(\sigma) - p_{j-1}\|$. Many different PWL functions can 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 $\mathbb{R}^d$. In this paper, we fix any arbitrary mapping $\mathtt{Wp2C}$. The resulting curve is written as $z_i = \mathtt{Wp2C}(P_i)$. The resulting PWL curves are uniformly continuous as stated in Proposition 4, satisfying the action space assumption in Definition 1. The curve may possibly be non-differentiable at each disjunction point.

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

The PWL curves can approximate uniformly continuous $z_i \in \mathcal{Z}_i$. Given $z_i : [0, 1] \to \mathbb{R}^d$ and a sequence P_i of $m+1$ waypoints in $\mathbb{R}^d$, the approximation error between $\mathtt{Wp2C}(P_i)$ and z_i is given by $\delta = \max_{s \in [0,1]} \|\mathtt{Wp2C}(P_i)(s) - z_i(s)\|$. For fixed m, the chosen P_i chosen should ideally minimize δ. Furthermore, given some $\delta > 0$, we can choose m such that the approximation error is less than δ.

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

The proof follows from the fact that for any uniformly continuous z_i, and any $s, t \in [0, 1]$, $s \neq t$, there exists $\zeta > 0$ and $\xi > 0$ such that $\|s - t\| < \zeta \implies \|z_i(s) - z_i(t)\| < \xi$. Assuming $J_i^{\mathsf{per}}(z_i, \{z_j\}_{-i})$ is Lipschitz continuous with constant K in the first argument, we can find a seqence P_i such that $\|J_i^{\mathsf{per}}(z_i, \{z_j\}_{-i}) - J_i^{\mathsf{per}}(\mathtt{Wp2C}(P_i), \{z_j\}_{-i})\| \leq K\delta$ as stated in Lemma 2.

Lemma 2. *For $J_i^{\mathsf{per}}(z_i, \{z_j\}_{-i})$, 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 $P_i = \{p_0, \ldots, p_m\}$ such that $\|J_i^{\mathsf{per}}(z_i, \{z_j\}_{-i}) - J_i^{\mathsf{per}}(\mathtt{Wp2C}(P_i), \{z_j\}_{-i})\| \leq K\delta$.*

The proof follows from Lemma 2 and the Lipschitz continuity of J_i^{per}.

B. Constructing bounding boxes to avoid collisions

To prevent collisions in autonomous driving and platooning scenarios, agents typically must remain at least c apart. Two agents i and j collide if their trajectories z_i and z_j come within c of each other. For an agent $j \in \mathcal{N}$, $J_j^{\mathsf{col}}(z_j, z_{-j})$ is minimized when (6) holds.

$$\forall i \in \{-j\},\ \mathcal{B}_c(z_i) \cap z_j = \varnothing \tag{6}$$

This checks for collisions over the entire trajectory, not just collisions in time, which can be done by splitting each trajectory into shorter trajectories in time. Collisions are checked through intersections between boxes that bound trajectories. We now construct these boxes.

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 $\mathcal{O}_c(p, p') \supset \mathcal{O}'_c(p, p')$. For convenience in this section, we will make the dependence on p, p' implicit. To construct $\mathcal{O}_c$, we compute the vector tangent to $\overline{pp'}$, given by $\overrightarrow{n}_0 = \frac{p' - p}{\|p' - p\|}$, and choose any basis that spans $\mathbb{R}^d$ and contains $\overrightarrow{n}_0$. The basis given by $\{\overrightarrow{n}_0, \ldots, \overrightarrow{n}_{d-1}\}$, each $\|\overrightarrow{n}_i\| = 1$.

Bounding boxes are defined by half-spaces. Each half-space is indexed by $\mathsf{dP}(A) = \{0, 0', \ldots, (d-1), (d-1)'\}$, meaning $\mathbf{A}^\top = [A_0^\top\ A_{0'}\ \cdots\ A_{(d-1)}^\top\ A_{(d-1)'}^\top]^\top$ and $\mathbf{b}^\top = [b_0\ b_{0'}\ \cdots\ b_{(d-1)}\ b_{(d-1)'}]^\top$. Each $A_i \in \mathbb{R}^d$ and $b_i \in \mathbb{R}$. We now construct the i^{th} half-space. Let $\mathcal{A}_i y = \beta_i$ be the hyperplane spanned by $\{\overrightarrow{n}_{-i}\}$ centered on $p' + c\overrightarrow{n}_i$. 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 call this half space $\{A_i y < b_i\}$, 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 $p - c\overrightarrow{n}_i$. This 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 $\mp\mathcal{A}_{i'} y < \mp\beta_{i'}$, and called $\{A_{i'} y < b_{i'}\}$. The bounding boxes are defined as $\mathcal{O}_c = \{y \in \mathbb{R}^d | \mathbf{A}y < \mathbf{b}\}$. 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.

Proposition 5. *For any $c > 0$, $p, p' \in \mathbb{R}^d$, $q, q' \in \mathbb{R}^d$ we note the following: (i) if $A_i q \geq b_i$ for some $i \in [0, d-1]$, then $A_{i'} q < b_{i'}$. (ii) $\exists q \in \mathbb{R}^d$ such that for some $i, j \in [0, d-1]$ and $j \neq i'$, $A_i q \geq b_i$ and $A_j q \geq b_j$. (iii) if $A_i q \geq b_i$ and $A_i q' \geq b_i$ for some $i \in [0, d-1]$, then $\overline{qq'} \cap \mathcal{O}_c = \varnothing$. (iv) there exists $\overline{qq'}$ such that $\overline{qq'} \cap \mathcal{O}_c = \varnothing$ that does not satisfy (2). (v) for any $\overline{qq'}$ described in (3), there exists $q'' \in \mathbb{R}^d$ such that $\overline{qq''}$ and $\overline{q''q'}$ satisfy (2).*

We omit the proofs here for brevity, but they will be included in the full version of this paper. Note that Proposition 5 still holds true if i and i' are switched. Since $\mathcal{O}_c$ is an over-approximation of $\mathcal{B}_c(pp')$, a slight modification on Lemma 1 is given in Lemma 3.

Lemma 3. *For any $p, p' \in \mathbb{R}^d$, bounding box $\mathcal{O}_c \subset \mathbb{R}^d$, 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 $q \in \mathbb{R}^d$ that lies on the boundary of $\mathcal{B}_c(\overline{pp'})$, there exists $q' \in \mathbb{R}^d$ on the corresponding $\mathcal{O}_c$ such that $\|q' - q\| \leq c(\sqrt{d}-1)$.

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

In Line 5 of Algorithm 2, a candidate sequence of waypoints Q for agent j is computed given the current guess of $\{P_i\}_{-j}$. We present a MILP formulation that minimizes $J_j^{\text{col}}(\texttt{Wp2C}(P_j), \{\texttt{Wp2C}(P_i\}_{-j})$ via encoded constraints. For PWL curves, the collision condition in (6) can be rewritten as in (7), and if true, minimizes $J_j^{\text{col}}(\texttt{Wp2C}(P_j), \{\texttt{Wp2C}(P_i)\}_{-j})$.

$$\forall i \in \{-j\},\ \mathcal{B}_c(\texttt{Wp2C}(P_i)) \cap \texttt{Wp2C}(P_j) = \varnothing \qquad (7)$$

We now check that individual segments do not collide. Consider some $p, p', q, q' \in \mathbb{R}^d$, and constant $c > 0$. Then, $\overline{pp'}$ and $\overline{qq'}$ are at least c away from each other if $\mathcal{B}_c(\overline{pp'}) \cap \overline{qq'} = \varnothing$. The naive formulation to check for this is nonlinear, so we use the minimum bounding boxes from Section III-B. The linear, disjunction free formulation is called $\texttt{SafeSegment}(p, p', c)$ and shown in (8). Here, α is an array of binary variables that determine if q, q' lie outside the same face of $\mathcal{O}_c(p, p')$, and $\Lambda >> 0$ is used to encode disjunctions.

$$\bigwedge_{r \in \mathsf{dP}(A)} \Big(b_r - A_r q < \Lambda(1-\alpha_r) \wedge b_r - A_r q' < \Lambda(1-\alpha_r) \Big) \wedge \sum_{r \in \mathsf{dP}(A)} \alpha_r \geq 1 \qquad (8)$$

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: If q, q' lie outside the same half-space of $\mathcal{O}_c(p, p')$, then $\overline{qq'} \cap \mathcal{O}_c(p, p') = \varnothing$ by Proposition 5. By construction, $\mathcal{B}_c(\overline{pp'}) \subset \mathcal{O}_c(\overline{pp'})$, meaning $\mathcal{B}_c(\overline{pp'}) \cap \overline{qq'} = \varnothing$. We now show that q, q' lie outside the same half-space of $\mathcal{O}_c(p, p')$. Choose some $p \in \mathbb{R}^d$ such that $A_r q \geq b_r$ for some $r \in \mathsf{dP}(A)$. 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 $\sum_{r \in \mathsf{dP}(A)} \alpha_r \geq 1$, at least one $\alpha_r = 1$, and thus $A_r q' \geq b_r$ and $A_r q \geq b_r$. Therefore, $\mathcal{B}_c(\overline{pp'}) \cap \overline{qq'} = \varnothing$. ■

Given $P = \{p_0, \ldots, p_m\}$ and $Q = \{q_0, \ldots, q_n\}$, we check that $\texttt{Wp2C}(Q)$ is at least c away from $\texttt{Wp2C}(P)$. 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 $i = 1, \cdots, n$, then $Q \models \texttt{SafeSequence}(P, c)$, shown in (9).

$$\bigwedge_{i=1}^{n} (q_{i-1}, q_i) \models \bigwedge_{j=1}^{m} \texttt{SafeSegment}(p_{j-1}, p_j, c) \qquad (9)$$

Corollary 1. *For any $c > 0$, any sequence of $m+1$ waypoints P, and any sequence of $n+1$ waypoints Q, if for all $i = 1, \ldots, n$ $Q \models \texttt{SafeSequence}(P, c)$ then $\mathcal{B}_c(\texttt{Wp2C}(P)) \cap \texttt{Wp2C}(Q) = \varnothing$*

The proof follows from Lemma 4 and the construction of $\texttt{Wp2C}(\cdot)$. If $(q_{i-1}, q_i) \models \bigwedge_{j=1}^m \texttt{SafeSegment}(p_{j-1}, p_j)$ is true for every $i = 1, \ldots, m$, then $\bigcup_{i \in \{0, \ldots, m-1\}} \mathcal{B}_c(\overline{p_i p_{i+1}}) \cap \overline{q_j q_{j+1}} = \varnothing$. If this is true for every $i = 1, \ldots, m$, $\bigcup_{i \in \{0, \ldots, m-1\}} \mathcal{B}_c(\overline{p_i p_{i+1}}) \cap \bigcup_{j \in \{0, \ldots, n-1\}} \overline{q_j q_{j+1}} = \varnothing$, leading to Corollary 1. The MILP forumlation for $\texttt{linUpdate}(\{P_j\}_{-i}, J_i^{\text{per}}(\cdot), c)$ is given in (10). In Corollary 2. we show that the resulting trajectories do not collide, minimizing $J_i^{\text{col}}(\cdot)$.

$$\begin{aligned} &\arg\min_{Q} J_i^{\text{per}}(\texttt{Wp2C}(Q), \{\texttt{Wp2C}(P_j)\}_{-i}) \\ &Q \models \bigwedge_{j \in \{-i\}} \texttt{SafeSequence}(P_j, c) \end{aligned} \qquad (10)$$

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 $j \in \{-i\}$.*

The proof follows from Corollary 1 and the fact that $\texttt{SafeSequence}(P_j, c)$ is satisfied for every $j = \{-i\}$. Thus J_j^{col} is minimized. The Q found using $\texttt{linUpdate}$ is not necessarily the Q that minimizes $J_i^{\text{per}}(\cdot)$ while satisfying (7) due to the approximation. We can find a bound on the ε error in L-IBR, stated 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, $\{P_j\}_{-i}$, and constant $c > 0$ such that for any $\varepsilon > Kc(\sqrt{d}-1)$ there exists $m \in \mathbb{N}$ and $Q = \{q_0, \ldots, Q_{m_i}\}$ found using $\texttt{linUpdate}(\{P_j\}_{-i}, J_i^{\text{per}}(\cdot), c)$ such that*

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

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

The proof follows from Lemmas 2 and 3, where the smallest error between any $\texttt{Wp2C}(P_i)$ and z_i is $c(\sqrt{d}-1)$, and thus $\|J_i^{\text{per}}(\texttt{Wp2C}(Q), \{\texttt{Wp2C}(P_j\}_{-i}) - J_i^{\text{per}}(\texttt{Wp2C}(Q^*), \{\texttt{Wp2C}(P_j\}_{-i})\| \leq Kc(\sqrt{d}-1)$. As the number of segments to approximate the agent trajectories increases, L-IBR returns a joint action closer to the NE.

IV. Experimental results

L-IBR is implemented using Python 3 and Gurobi, and run in the examples in Figure 2. The collision cost function is

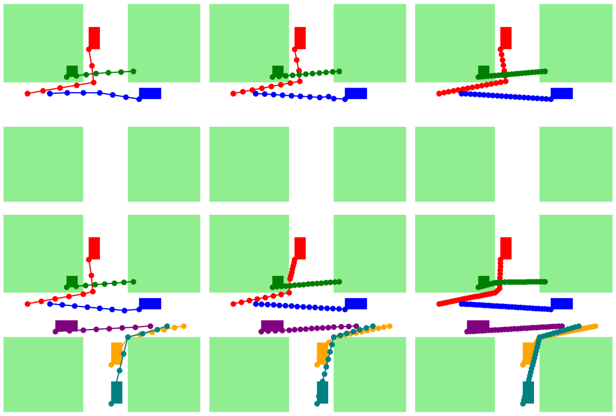

Fig. 2. Example results. Across varying numbers of segments used to approximate PWL paths, the actual paths taken remains the same.

Scenario	Num segments	Total cost	Comp time
2 car, 1 ped	20	2.268	1.277
2 car, 1 ped	10	4.584	0.578
2 car, 1 ped	6	7.684	0.427
5 car, 1 ped	30	3.395	65.16
5 car, 1 ped	15	6.730	3.567
5 car, 1 ped	7	14.981	7.168

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

that in (1) and the indivicual cost function is the distance traveled.

In Table I, we compare the computational time in seconds and total cost the agents' individual costs. Computational time increases with the number of agents and the number of segments used to approximate the trajectories. The total cost decreases as the number of segments increases, which is in line with our main result from Theorem 1. We see a tradeoff between computation time and how closely we can achieve the true NE. The computation time is lower in the 5 car, 1 ped scenario with 15 segments than the one with 7 segments due to the number of iterations run to find the NE.

V. CONCLUSION

We presented a formulation for a lexicographic general sum game, and show that it is a potential game possesing a pure strategy Nash equilibrium. We proposed an iterative best response algorithm that searches for actions that minimize the second part of the lexicographic cost function using linear constraints to minimize the first part We ran some experiments and showed we can find solutions to the game.

In the future, we can relax the type of collision costs to explore a larger variety of games. We can also use this problem set up to study the equilibria found in a long single shot game versus a sequence of shorter multi-stage games. We can also 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 *IROS-RSJ*, 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 *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] 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.

[7] 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 *ICRA*, pages 1475–1481. IEEE, 2020.

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

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

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

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

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

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

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

[15] 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.

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

[17] J. Van Den Berg, D. Ferguson, and J. Kuffner. Anytime path planning and replanning in dynamic environments. In *ICRA*, pages 2366–2371. IEEE, 2006.

[18] 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.

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

[20] 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 RA-L*, 6(3):4978–4985, 2021.

[21] 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.

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

[23] M. Zucker, J. Kuffner, and M. Branicky. Multipartite rrts for rapid replanning in dynamic environments. In *ICRA*, pages 1603–1609. IEEE, 2007.