You Can't Always Get What You Want: Games of Ordered Preference

Dong Ho Lee, Lasse Peters, and David Fridovich-Keil

***Abstract*—We study noncooperative games, in which each player's objective is composed of a sequence of ordered—and potentially conflicting—preferences. Problems of this type naturally model a wide variety of scenarios: for example, drivers at a busy intersection must balance the desire to make forward progress with the risk of collision. Mathematically, these problems possess a nested structure, and to behave properly players must prioritize their most important preference, and only consider less important preferences to the extent that they do not compromise performance on more important ones. We consider multi-agent, noncooperative variants of these problems, and seek generalized Nash equilibria in which each player's decision reflects both its hierarchy of preferences *and* other players' actions. We make two key contributions. First, we develop a recursive approach for deriving the first-order optimality conditions of each player's nested problem. Second, we propose a sequence of increasingly tight relaxations, each of which can be transcribed as a mixed complementarity problem and solved via existing methods. Experimental results demonstrate that our approach reliably converges to equilibrium solutions that strictly reflect players' individual ordered preferences.**

***Index Terms*—Autonomous agents, optimization and optimal control, constrained motion planning.**

I. Introduction

In optimal decision-making, a user's preferences often reflect competing goals such as safety and efficiency. For example, consider the intersection scenario in Fig. 1 where each vehicle has a different order of preferences regarding reaching the goal, driving under the speed limit, driving within the lane, and minimizing fuel usage. In such cases, treating all preferences as equally important can be problematic, especially when some preferences encode hard constraints, such as respecting lane boundaries. When formulated as an optimization problem, conflicting preferences can lead to infeasibility and ultimately cause solver failure.

In many cases—such as the autonomous driving example above—there is a clear hierarchy among the conflicting preferences. A naïve approach to encode this concept of ordered preference is to construct a single objective function with weighted contributions from each preference, which can be adjusted manually or learned from data [1], [2]. However, such formulations can easily become ill-conditioned, and it is not always straightforward to design weights which yield the desired behavior.

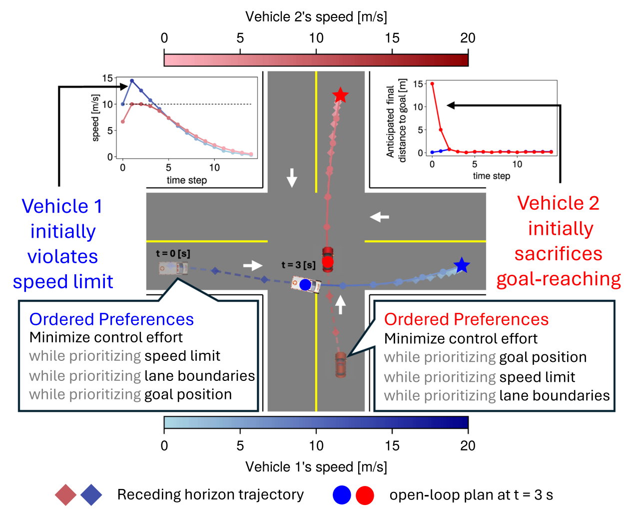

Fig. 1. A two-vehicle intersection scenario involving four levels of preferences. The star indicates the goal position for each vehicle. Our game of ordered preference (GOOP) framework identifies equilibrium trajectories by selectively relaxing less important preferences only when they compromise the performance of more important ones. In this figure, the dashed lines with diamond markers depict the complete closed-loop trajectories computed via receding horizon planning—where an open-loop plan is computed at each time step and only the first control action is implemented—while the solid lines with circle markers show a representative open-loop plan generated at $t = 3$ s. Additional subplots illustrate vehicle speeds and anticipated final distance to the goal in the closed-loop trajectories. Vehicle 1 (blue) initially violates the speed limit in order to satisfy goal-reaching at the final time step. On the other hand, Vehicle 2 (red) initially maintains their speed under the limit and sacrifices goal-reaching instead.

Hierarchical optimization problems have been well studied in the operations research literature [3], [4]. These problems are naturally characterized as a sequence of nested mathematical programs, where the decision variable at each level is constrained to be a minimizer of the problem at the level below. Several studies such as [5], [6], [7], [8] have explored theoretical properties such as optimality conditions and constraint qualifications in bilevel settings. Nested problems of this kind can also be solved via "lexicographic minimization," in which each subproblem is addressed in order—from the lowest level to the highest level—while preserving the optimality of higher-priority preferences (at lower levels) by incorporating additional constraints [9], [10], [11].

While the lexicographic approaches produce solutions with desirable properties in single-agent settings, their computational

Received 21 January 2025; accepted 17 May 2025. Date of publication 30 May 2025; date of current version 6 June 2025. This article was recommended for publication by Associate Editor S. Tonneau and Editor O. Stasse upon evaluation of the reviewers' comments. This work was supported by the National Science Foundation CAREER Award under Grant 2336840. *(Corresponding author: Dong Ho Lee.)*

Dong Ho Lee and David Fridovich-Keil are with the Department of Aerospace Engineering and Engineering Mechanics, the University of Texas at Austin, Austin, TX 78712 USA (e-mail: leedh0124@utexas.edu; dfk@utexas.edu).

Lasse Peters is with the Department of Cognitive Robotics, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: l.peters@tudelft.nl).

Digital Object Identifier 10.1109/LRA.2025.3575324

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

methods do not readily extend to multi-agent domains. In single-agent contexts, hierarchical least-squares quadratic problems have been studied [12], particularly in the realm of real-time robot control. More generally, connectivity structures—often referred to as mathematical program networks—have also been characterized [13].

While various methods to cope with hierarchical preferences have been developed—such as the aforementioned strategy of weighting agents' preferences according to their priority as in [14]—most focus on single-agent scenarios, and there are very limited results for multi-agent, noncooperative settings. For example, recent work [15] applies lexicographic minimization to an urban driving game via an *iterated best response* (IBR) scheme. However, this approach is limited to a certain class of games where IBR is guaranteed to converge. Follow-on work [16] considers preferences which are only partially ordered, necessitating a substantially different solution approach. Recent works [17], [18] introduce social or game-theoretic models integrating high-level intent and compliance, while others [19], [20] study game-theoretic planning in competitive scenarios, but all assume scalarized or single-level objectives. In contrast, our work focuses on settings where agents' decisions are governed by ordered preferences, allowing some to be selectively relaxed to achieve higher priority goals.

In this paper, we study multi-agent, game-theoretic variants of problems of (totally) ordered preference, which we refer to as game of ordered preferences (GOOPs). Our contributions are twofold: *(i)* We reformulate each agent's problem of ordered preference by sequentially replacing inner-level optimization problems with their corresponding Karush-Kuhn-Tucker conditions. This yields a mathematical program with complementarity constraints (MPCC) for each agent. *(ii)* We develop a relaxation technique that smoothens the boundary of the feasible set in these problems in order to facilitate numerical computation. From this set of relaxed MPCCs, we derive a single mixed complementarity problem whose solution is a (local) generalized Nash equilibrium solution of the original GOOP. We present experimental results which demonstrate that the proposed algorithm reliably converges to approximate generalized Nash solutions which reflect individual player's hierarchy of preferences, and compare the results with a family of penalty-based approximation baselines.

II. Preliminaries and Related Work

In this section, we introduce two important concepts underpinning our work and discuss the related literature in each area. In Section II-A, we discuss how we formulate the problem of ordered preferences as a hierarchical optimization problem, and transcribe it as an MPCC. Next, in Section II-B, we introduce generalized Nash equilibrium problems (GNEPs) and discuss their relationship to mixed complementarity problems (MiCPs), for which an off-the-shelf solver is available.

A. From Hierarchical Preferences to MPCCs

We begin by discussing a *single*-agent problem with *two* levels; future sections will generalize to the N-agent, K-level setting. We use subscripts to denote the preference level and assume that a higher preference index indicates higher priority. In other words, the *innermost* problem carries the highest level of preference. This yields a problem of the following form:

$$\min_{\mathbf{z}_1} \quad J_1(\mathbf{z}_1) \tag{1a}$$

$$\text{s.t.} \quad \mathbf{z}_1 \in \operatorname*{argmin}_{\mathbf{z}_2} J_2(\mathbf{z}_2) \tag{1b}$$

$$\text{s.t.} \quad \mathbf{z}_2 \in \mathcal{Z}, \tag{1c}$$

where $\forall i \in \{1,2\}, \mathbf{z}_i \in \mathbb{R}^n, J_i(\cdot) : \mathbb{R}^n \to \mathbb{R}$, and the inner feasible set $\mathcal{Z}$ is defined in terms of continuously differentiable functions $g : \mathbb{R}^n \to \mathbb{R}^m$ and $h : \mathbb{R}^n \to \mathbb{R}^p$ as $\mathcal{Z} := \{\mathbf{z} \in \mathbb{R}^n | g(\mathbf{z}) = 0, h(\mathbf{z}) \geq 0\}$. This formulation captures the fact that any outer-level variables are constrained to be in the set of minimizers of the lower-level problem. By inspection, we can readily see that the inner problem is a constrained nonlinear program. In general, the Karush-Kuhn-Tucker (KKT) conditions are only necessary for optimality, provided that some constraint qualifications are satisfied [21]. If $\mathcal{Z}$ is convex, then the KKT conditions are also sufficient. The necessary conditions for optimality correspond to a mixed complementarity problem (MiCP), which is the KKT system comprised of primal ($\mathbf{z}_2$) and dual (λ_2, μ_2) variables of the inner problem. It is convenient to express the result in terms of the Lagrangian of the inner problem, defined as $\mathcal{L}_2(\mathbf{z}_2, \lambda_2, \mu_2) := J(\mathbf{z}_2) - \lambda_2^\top h(\mathbf{z}_2) - \mu_2^\top g(\mathbf{z}_2)$:

$$\min_{\mathbf{z}_1, \lambda_2, \mu_2} \quad J_1(\mathbf{z}_1) \tag{2a}$$

$$\text{s.t.} \quad \nabla_{\mathbf{z}_2} \mathcal{L}_2(\mathbf{z}_1, \lambda_2, \mu_2) = 0, \tag{2b}$$

$$0 \leq h(\mathbf{z}_1) \perp \lambda_2 \geq 0, \tag{2c}$$

$$g(\mathbf{z}_1) = 0. \tag{2d}$$

The optimization problem in (2) is a single-level program that involves the Lagrange dual variables of the lower level problem in (1b) and (1c) as primal variables. To be specific, the dual variables $(\lambda_2, \mu_2) \in \mathbb{R}^{p+m}$, which are introduced at the inner problem, become primal variables (λ_1, μ_1) for the outer problem (in addition to $\mathbf{z}_1 \in \mathbb{R}^n$). We call these additional primal variables as the *induced* primal variables since they are introduced in the process of building a single-level program. In particular, constraint (2b) refers to the stationarity condition of the Lagrangian function with respect to the primal variable ($\mathbf{z}_1$) of the inner level problem. Constraint (2c) encodes the complementarity relationship between the inequality constraints in (1c) and the associated dual variables. This constraint indicates that for each coordinate $i \in [p]$, at least one of $h_i(\mathbf{z})$ and λ_i is zero, while the other is nonnegative. Lastly, (2d) is the equality constraint from (1c).

The reformulated problem in (2) is known as a mathematical program with complementarity constraints (MPCC). In general, MPCCs are ill-posed as the complementarity constraints in (2c) violate constraint qualifications (CQs) such as the Mangasarian-Fromowitz constraint qualification (MFCQ) and linear independence constraint qualification (LICQ) at every feasible point [22]. This inherent lack of regularity in the structure of MPCCs makes it difficult to use standard nonlinear programming (NLP) solvers directly. In particular, the absence of a CQ implies that the KKT conditions of the reformulation in (2) may no longer hold at a locally optimal solution. These theoretical and numerical difficulties led to the development of tailored theory and methods for solving MPCCs [23], [24], [25], [26], [27], [28]. In this context, we develop a relaxation-based approach for solving GOOPs, which we explore in detail in Section III.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

B. Generalized Nash Equilibrium Problems

In this section, we formally introduce generalized Nash equilibrium problems (GNEPs) and provide a brief overview of how local solutions may be identified [29]. A GNEP involves N players, whose variables are denoted as $\mathbf{z}^i \in \mathbb{R}^{n_i}$. The dimension of the game is $n := \sum_{i=1}^N n_i$. We denote by $\mathbf{z}^{\neg i} \in \mathbb{R}^{n-n_i}$ the state variables of all players except player Ri. Each player Ri has an objective function denoted by $J^i(\mathbf{z}^i, \mathbf{z}^{\neg i})$ and a feasible set $\mathcal{Z}^i(\mathbf{z}^{\neg i})$ on which their decisions depend. Each feasible set is defined algebraically via (nonlinear) equality and/or inequality constraints : $\mathcal{Z}^i(\mathbf{z}^{\neg i}) := \{\mathbf{z}^i | g^i(\mathbf{z}^i, \mathbf{z}^{\neg i}) = 0, h^i(\mathbf{z}^i, \mathbf{z}^{\neg i}) \geq 0\}$. We call these constraints *private* since they are "owned" by each player Ri. Furthermore, we also consider constraints that are *shared* among N players, which we denote as $g^s(\mathbf{z}) = 0, h^s(\mathbf{z}) \geq 0$ where $\mathbf{z} := [\mathbf{z}^1, \mathbf{z}^2, \ldots, \mathbf{z}^N]^\top$. For simplicity, we assume that these constraints are shared by *all* players so that everyone is equally responsible for satisfying them.

Definition II.1 (Generalized Nash Equilibrium): Mathematically, a generalized Nash equilibrium problem (GNEP) is expressed via coupled optimization problems:

$$\forall i \in [N] \quad \begin{cases} \min\limits_{\mathbf{z}^i} & J^i(\mathbf{z}^i, \mathbf{z}^{\neg i}) \\ \text{s.t.} & \mathbf{z}^i \in \mathcal{Z}^i(\mathbf{z}^{\neg i}) \end{cases} \tag{3a}$$

$$\text{s.t.} \quad g^s(\mathbf{z}) = 0, \quad h^s(\mathbf{z}) \geq 0. \tag{3b}$$

The generalized Nash equilibrium (GNE) solution of (3), $\mathbf{z}^* := [\mathbf{z}^{1*}, \ldots, \mathbf{z}^{N^*}]^\top$, satisfies the inequality $J^i(\mathbf{z}^i, \mathbf{z}^{\neg i*}) \geq J^i(\mathbf{z}^*)$ for all feasible choices $\mathbf{z}^i \in \mathcal{Z}^i(\mathbf{z}^{\neg i*})$, for all players $i \in [N]$. This means that at equilibrium, no player has an incentive to unilaterally deviate from their equilibrium $\mathbf{z}^{i*}$.

In practice, it is intractable to solve for a (global) GNE solution. Instead, it is common to transcribe the formulation in (3) as a mixed complementarity problem (MiCP) and use off-the-shelf solvers to find a *local* GNE solution. In essence, solving this MiCP is equivalent to finding a point that satisfies the system of first-order (KKT) conditions of each player's optimization problem. In this paper, we use the PATH solver [30], which constructs an equivalent nonsmooth system of equations and solves them via a generalized Newton method. We note that solving for GNE solutions via solving the corresponding MiCP has been widely used in [31], [32].

III. Games of Ordered Preferences

In this section, we formalize a variant of the hierarchical problems described in Section II-A which extends to the multi-agent, noncooperative games of Section II-B. We term this multi-agent variant a *game of ordered preference*.

A. Mathematical Formulation of GOOPs

We begin by introducing the mathematical formulation of GOOP which we shall contextualize with a running example.

1) General Formulation: Unlike the GNEP in Definition 2.1, where each player's optimization problem is a standard NLP, a GOOP consists of N optimization problems for each player, but each player's problem is hierarchical, of the type discussed in Section II-A. Each player's hierarchical problem may involve a different number of levels. To this end, we use $k^i \in [K^i]$ to denote the kth level of preference for player Ri, where K^i refers to the number of preferences for Ri. Mathematically, we express a GOOP as follows:

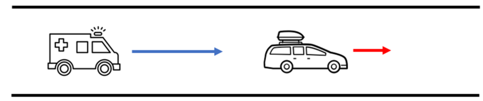

Fig. 2. A highway driving scenario with 2 vehicles.

$$\min_{\mathbf{z}_1^i} \quad J_1^i(\mathbf{z}_1^i, \mathbf{z}_1^{\neg i}) \tag{4a}$$

$$\text{s.t.} \quad \mathbf{z}_1^i \in \operatorname*{argmin}_{\mathbf{z}_2^i} \; J_2^i(\mathbf{z}_2^i, \mathbf{z}_1^{\neg i}) \tag{4b}$$

$$\ddots$$

$$\text{s.t.} \quad \mathbf{z}_{K^i-1}^i \in \operatorname*{argmin}_{\mathbf{z}_{K^i}^i} \; J_{K^i}^i(\mathbf{z}_{K^i}^i, \mathbf{z}_1^{\neg i}) \tag{4c}$$

$$\text{s.t.} \quad \mathbf{z}_{K^i}^i \in \mathcal{Z}_{K^i}^i(\mathbf{z}_1^{\neg i}) \tag{4d}$$

$$g^s(\mathbf{z}_1) = 0, \quad h^s(\mathbf{z}_1) \geq 0. \tag{4e}$$

Here, $\mathcal{Z}_{K^i}^i(\mathbf{z}^{\neg i}) := \{\mathbf{z}^i \in \mathbb{R}^{n_i} | g^i(\mathbf{z}^i, \mathbf{z}^{\neg i}) = 0, h^i(\mathbf{z}^i, \mathbf{z}^{\neg i}) \geq 0\}$, and (4e) represents the shared constraints between Ri and the rest of the players.

Running example: We will use the following 2-player running example to illustrate the GOOP formalism. We will study more complex interactions in Section IV.

Consider the highway driving scenario of Fig. 2, in which $N = 2$ vehicles must plan their future actions over the next T time steps. In this example, vehicle 1 is an ambulance and its highest priority preference is to reach a desired goal position. Its secondary preference is to drive below the speed limit. In contrast, vehicle 2 is a passenger car whose highest priority preference is to respect the speed limit, and whose secondary preference is to reach a goal location. Both vehicles' lowest priority objective is to minimize their individual control effort, and no vehicle wants to collide. These conflicting preferences make it natural to describe the interaction as a GOOP. [1]

We model each vehicle as a player in the game and denote the ith vehicle's trajectory as $\mathbf{z}^i := [\mathbf{x}^i, \mathbf{u}^i]^\top, \forall i \in [N]$. Here, $\mathbf{x}^i := [x_1^i, \ldots, x_T^i]^\top \in \mathbb{R}^{4T}$ with $x_t^i = [p_{x,t}^i, p_{y,t}^i, v_{x,t}^i, v_{y,t}^i]^\top \in \mathbb{R}^4$ encoding the state of ith vehicle, comprised of position and velocity in the horizontal and vertical directions. Further, we denote a sequence of control inputs by $\mathbf{u}^i := [u_1^i, \ldots, u_T^i]^\top \in \mathbb{R}^{2T}$ where the ith vehicle's control input at time t, $u_t^i = [a_{x,t}^i, a_{y,t}^i]^\top \in \mathbb{R}^2$, is the acceleration in the horizontal and vertical directions, respectively. Each vehicle follows double-integrator dynamics, discretized at a resolution Δt, i.e.

$$x_{t+1}^i = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} p_{x,t}^i \\ p_{y,t}^i \\ v_{x,t}^i \\ v_{y,t}^i \end{bmatrix}}_{x_t^i} + \begin{bmatrix} \frac{1}{2}\Delta t^2 & 0 \\ 0 & \frac{1}{2}\Delta t^2 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \underbrace{\begin{bmatrix} a_{x,t}^i \\ a_{y,t}^i \end{bmatrix}}_{u_t^i}. \tag{5}$$

Note that (5) should be interpreted as equality constraints that partially define $\mathcal{Z}_{K^i}^i(\mathbf{z}_1^{\neg i})$ in (4d). Both vehicles must also drive

[1] Although our running example considers only two levels of preferences, in practice one could also introduce another level which encodes an absolute maximum safety speed limit. This additional limit at a higher priority level would prevent the ambulance from exceeding the speed limit *indefinitely*.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

within the highway lane in the horizontal direction. We encode this requirement as inequality constraints:

$$\underline{p}_y \leq p^i_{y,t} \leq \overline{p}_y \tag{6}$$

The equality constraints (5) and inequality constraints (6) together specify the *private* feasible set $\mathcal{Z}^i_{K^i}(\mathbf{z}^{\neg i}_1)$ in (4d). Both vehicles also share a collision-avoidance constraint:

$$h^s(\mathbf{x}^1, \mathbf{x}^2) = \left[\left(p^1_{x,t} - p^2_{x,t}\right)^2 + \left(p^1_{y,t} - p^2_{y,t}\right)^2 - d^2_{col}\right]^T_{t=1} \in \mathbb{R}^T \tag{7}$$

where d_{col} is the minimum distance between the two vehicles to avoid collision.

To encode each player's individual ordered preferences, we define the following cost components:

$$J^i_{\text{ctrl}}(\mathbf{u}^i) = \sum_{t=1}^{T} \sum_{j\in\{x,y\}} \left(a^i_{j,t}\right)^2 \tag{8a}$$

$$J^i_{\text{goal}}(\mathbf{x}^i) = \sum_{j\in\{x,y\}} [\hat{p}^i_j - p^i_{j,T}]_+ \tag{8b}$$

$$J^i_{\text{obey}}(\mathbf{x}^i) = \sum_{\substack{t=1\\ j\in\{x,y\}}}^{T} [\underline{v}^i_j - v^i_{j,t}]_+ + [v^i_{j,t} - \overline{v}^i_j]_+, \tag{8c}$$

where $[\cdot]_+ := \max(0, \cdot)$, $(p^i_{x,T}, p^i_{y,T})$ represents the terminal position of the vehicle, $(\hat{p}^i_x, \hat{p}^i_y)$ refers to the desired goal position, and $(\underline{v}^i_x, \overline{v}^i_x)$ and $(\underline{v}^i_y, \overline{v}^i_y)$ denote the lower and upper limits of the velocity in the horizontal and vertical directions.

Using these cost components, we define vehicle 1's ordered preferences to prioritize goal reaching (8b) over obeying the speed limit (8c), i.e., $J^1_2(\mathbf{x}^i) = J^1_{\text{obey}}(\mathbf{x}^1)$ and $J^1_3(\mathbf{x}^1) = J^1_{\text{goal}}(\mathbf{x}^1)$. In contrast, vehicle 2 prioritizes obeying the speed limit (8c) over goal reaching (8b); i.e. $J^2_2(\mathbf{x}^2) = J^2_{\text{goal}}(\mathbf{x}^2)$ and $J^2_3(\mathbf{x}^2) = J^2_{\text{obey}}(\mathbf{x}^2)$.

Intuitively, the ambulance may violate the speed limit to reach the goal more quickly. Similarly, the passenger car in front may pull to the side, de-prioritizing goal-reaching to yield to the ambulance, or temporarily violate the speed limit to avoid a collision. Once the ambulance has passed, however, the car must strictly adhere to the speed limit. GOOP solutions naturally give rise to appropriate negotiation of preferences, relaxing less important preferences first when not all preferences can be perfectly satisfied.

To support this intuition, we provide a sample solution for the highway running example. Fig. 3 shows the interaction between two vehicles with different priorities for this scenario where we consider horizontal dynamics only. Vehicle 1 (blue) prioritizes minimizing the distance to the goal at the final time step. However, it slows down in order to avoid collision with vehicle 2. Vehicle 2 (red) prioritizes driving within the maximum speed limit, but to avoid collision with the fast-approaching Vehicle 1 (blue), it temporarily exceeds this limit. Here, GOOP allows optimal violations of preferences to satisfy hard constraints like collision avoidance.

2) From Hierarchical to Single-Level: Next, we discuss how to derive first-order necessary conditions for GOOP. We shall use these conditions to identify equilibrium solutions in Section III-B.

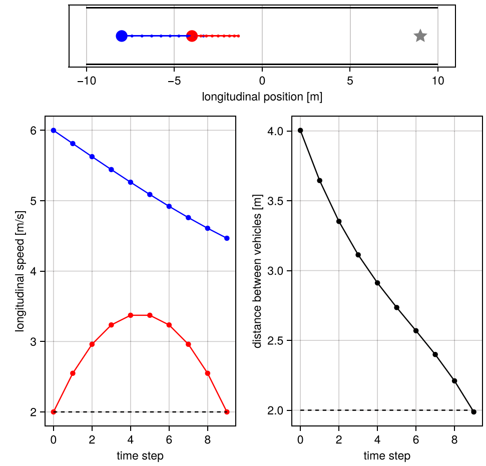

Fig. 3. A GOOP solution of the running example. The grey marker is the goal position for both vehicles. The dashed line in the left plot (speed) indicates the maximum speed limit. The dashed line in the right plot (distance between the vehicles) represents the minimum safe distance for collision avoidance.

Following the procedure in Section II-A, we may transcribe Ri's hierarchical problem (4) into a single level. To do this, we will successively replace each nested problem within (4b), (4c), (4d) with its corresponding KKT conditions, starting from the inner-most problem, which encodes the highest priority preference. As a result of this operation, we obtain a mathematical program with complementarity constraints (MPCC) of the following form:

$$\tilde{\mathbf{z}}^{i*}_1 \in \arg\min_{\tilde{\mathbf{z}}^i_1} \; J^i_1(\tilde{\mathbf{z}}^i_1, \tilde{\mathbf{z}}^{\neg i*}_1) \tag{9a}$$

$$\text{s.t. } g^i(\tilde{\mathbf{z}}^i_1, \tilde{\mathbf{z}}^{\neg i*}_1) = 0, h^i(\tilde{\mathbf{z}}^i_1, \tilde{\mathbf{z}}^{\neg i*}_1) \geq 0, \tag{9b}$$

$$G^i(\tilde{\mathbf{z}}^i_1, \tilde{\mathbf{z}}^{\neg i*}_1) \geq 0, H^i(\tilde{\mathbf{z}}^i_1, \tilde{\mathbf{z}}^{\neg i*}_1) \geq 0, \tag{9c}$$

$$G^i(\tilde{\mathbf{z}}^i_1, \tilde{\mathbf{z}}^{\neg i*}_1)^\top H^i(\tilde{\mathbf{z}}^i_1, \tilde{\mathbf{z}}^{\neg i*}_1) = 0, \tag{9d}$$

$$g^s(\tilde{\mathbf{z}}^i_1, \tilde{\mathbf{z}}^{\neg i*}_1) = 0, \; h^s(\tilde{\mathbf{z}}^i_1, \tilde{\mathbf{z}}^{\neg i*}_1) \geq 0. \tag{9e}$$

and we interpret the constraints in (9b) to (9d) as a specification of Ri's private constraint set $\mathcal{Z}^i(\tilde{\mathbf{z}}^{\neg i}_1)$ for the GNEP in (3a), and (9e) encode the shared constraints in (3b).

Note that problem (9) involves new variables $\tilde{\mathbf{z}}^i_1, \forall i \in [N]$. These include the original primal variables $\mathbf{z}^i_1$ along with additional variables—the dual variables from lower-level problems in (4)—induced by the aforementioned recursive procedure. In particular, $\tilde{\mathbf{z}}^i_1 := [\mathbf{z}^i_1, \lambda^i_2, \mu^i_2, \ldots, \lambda^i_{K^i}, \mu^i_{K^i}]^\top, \forall i \in [N]$, and the variables $(\lambda^i_2, \mu^i_2, \ldots, \lambda^i_{K^i}, \mu^i_{K^i})$ are Lagrange multipliers from the KKT conditions of lower-level problems. The functions $g^i(\tilde{\mathbf{z}}^i_1, \tilde{\mathbf{z}}^{\neg i}_1), h^i(\tilde{\mathbf{z}}^i_1, \tilde{\mathbf{z}}^{\neg i}_1), G^i(\tilde{\mathbf{z}}^i_1, \tilde{\mathbf{z}}^{\neg i}_1)$ and $H^i(\tilde{\mathbf{z}}^i_1, \tilde{\mathbf{z}}^{\neg i}_1)$ collect equality and inequality constraints that arise throughout. For clarity, we present an explicit formulation of (9) in the following running example.

Running example: For our running example, we have three priority levels for each vehicle, i.e., $K^i = 3, \forall i \in [2]$. For this simple case, we can see how the dual variables become *induced* primal variables for the outermost problem. Beginning with the innermost level ($k^i = 3$), the intermediate level ($k^i = 2$) problem becomes:

$$\min_{\mathbf{z}^i_2, \lambda^i_3, \mu^i_3} \quad J^i_2(\mathbf{z}^i_2, \tilde{\mathbf{z}}^{\neg i}_1) \tag{10a}$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

$$\text{s.t.}\quad \nabla_{\mathbf{z}_3^i}\mathcal{L}_3^i(\mathbf{z}_2^i,\tilde{\mathbf{z}}_1^{\neg i},\lambda_3^i,\mu_3^i)=0, \tag{10b}$$

$$0\le h^i(\mathbf{z}_2^i,\tilde{\mathbf{z}}_1^{\neg i})\perp\lambda_3^i\ge 0, \tag{10c}$$

$$g^i(\mathbf{z}_2^i,\tilde{\mathbf{z}}_1^{\neg i})=0. \tag{10d}$$

The KKT conditions for (10) define the feasible set of the outermost ($k^i=1$) problem:

$$\min_{\mathbf{z}_1^i,\lambda_3^i,\mu_3^i,\lambda_2^i,\mu_2^i}\; J_1^i(\mathbf{z}_1^i,\tilde{\mathbf{z}}_1^{\neg i}) \tag{11a}$$

$$\text{s.t.}\quad \nabla_{\mathbf{z}_2^i,\lambda_3^i,\mu_3^i}\mathcal{L}_2^i(\mathbf{z}_1^i,\tilde{\mathbf{z}}_1^{\neg i},\lambda_3^i,\mu_3^i,\lambda_2^i,\mu_2^i)=0, \tag{11b}$$

$$\nabla_{\mathbf{z}_3^i}\mathcal{L}_3^i(\mathbf{z}_1^i,\tilde{\mathbf{z}}_1^{\neg i},\lambda_3^i,\mu_3^i)=0, \tag{11c}$$

$$0\le h^i(\mathbf{z}_1^i,\tilde{\mathbf{z}}_1^{\neg i})\perp\lambda_2^{i,1}\ge 0, \tag{11d}$$

$$0\le\lambda_3^i\perp\lambda_2^{i,2}\ge 0, \tag{11e}$$

$$h^i(\mathbf{z}_1^i,\tilde{\mathbf{z}}_1^{\neg i})^\top\lambda_3^i=0, g^i(\mathbf{z}_1^i,\tilde{\mathbf{z}}_1^{\neg i})=0, \tag{11f}$$

$$g^s(\tilde{\mathbf{z}}_1)=0,\quad h^s(\tilde{\mathbf{z}}_1)\ge 0. \tag{11g}$$

where $\lambda_2^i=[\lambda_2^{i,1},\lambda_2^{i,2}]^\top$, $\mu_2^i=[\mu_2^{i,1},\mu_2^{i,2},\mu_2^{i,3}]^\top$ denote dual variables for the inequality and equality constraints (respectively) of the intermediate level problem. Note that the Lagrangian for the outermost problem is $\mathcal{L}_2^i(\mathbf{z}_2^i,\tilde{\mathbf{z}}_1^{\neg i},\lambda_3^i,\mu_3^i,\lambda_2^i,\mu_2^i)=J_2^i(\mathbf{z}_2^i,\tilde{\mathbf{z}}_1^{\neg i})-h^i(\mathbf{z}_2^i,\tilde{\mathbf{z}}_1^{\neg i})^\top\lambda_2^{i,1}-{\lambda_3^i}^\top\lambda_2^{i,2}-\nabla_{\mathbf{z}_3^i}\mathcal{L}_3^i(\mathbf{z}_2^i,\tilde{\mathbf{z}}_1^{\neg i},\lambda_3^i,\mu_3^i)^\top\mu_2^{i,1}-g^i(\mathbf{z}_2^i,\tilde{\mathbf{z}}_1^{\neg i})^\top\mu_2^{i,2}-h^i(\mathbf{z}_2^i,\tilde{\mathbf{z}}_1^{\neg i})^\top\lambda_3^i\cdot\mu_2^{i,3}$. Observe that the formulation in (11) is in the form of an MPCC as given in (9). To be specific, we have that $g^i(\tilde{\mathbf{z}}_1^i,\tilde{\mathbf{z}}_1^{\neg i})$ consists of the equality constraints (11b), (11c), (11f). The shared constraints in (9e) are identical to (11g) and the complementarity constraints in (9) correspond to:

$$G^i(\tilde{\mathbf{z}}_1^i,\tilde{\mathbf{z}}_1^{\neg i})=\begin{bmatrix}h^i(\mathbf{z}_1^i,\tilde{\mathbf{z}}_1^{\neg i})\\ \lambda_3^i\end{bmatrix},\; H^i(\tilde{\mathbf{z}}_1^i,\tilde{\mathbf{z}}_1^{\neg i})=\begin{bmatrix}\lambda_2^{i,1}\\ \lambda_2^{i,2}\end{bmatrix}. \tag{12}$$

Next, we discuss how to solve problems of the form (9) numerically.

B. Numerical Solution of GOOP

1) MPCC Relaxation: As noted earlier in Section II-A, the MPCC in (9) can be numerically challenging to solve due to irregularities in the geometry of the feasible set. Therefore, we propose a relaxation scheme that mitigates the aforementioned issue by solving a sequence of GOOPs which are regularized by altering the complementarity constraints in (9d). To this end, we replace the equality constraint in (9d) with an inequality as follows:

$$G^i(\tilde{\mathbf{z}}_1^i,\tilde{\mathbf{z}}_1^{\neg i})^\top H^i(\tilde{\mathbf{z}}_1^i,\tilde{\mathbf{z}}_1^{\neg i})\le\sigma. \tag{13}$$

When $\sigma=0$, this constraint encodes the original complementarity condition. For $\sigma>0$, this reformulation enlarges the feasible set and ensures that it has a nonempty interior. Using this relaxation scheme, the MPCC in (9) becomes

$$\tilde{\mathbf{z}}_1^{i*}\in\arg\min_{\tilde{\mathbf{z}}_1^i}\; J_1^i(\tilde{\mathbf{z}}_1^i,\tilde{\mathbf{z}}_1^{\neg i*}) \tag{14a}$$

$$\text{s.t. } g^i(\tilde{\mathbf{z}}_1^i,\tilde{\mathbf{z}}_1^{\neg i*})=0, h^i(\tilde{\mathbf{z}}_1^i,\tilde{\mathbf{z}}_1^{\neg i*})\ge 0, \tag{14b}$$

$$G^i(\tilde{\mathbf{z}}_1^i,\tilde{\mathbf{z}}_1^{\neg i*})\ge 0, H^i(\tilde{\mathbf{z}}_1^i,\tilde{\mathbf{z}}_1^{\neg i*})\ge 0, \tag{14c}$$

$$G^i(\tilde{\mathbf{z}}_1^i,\tilde{\mathbf{z}}_1^{\neg i*})^\top H^i(\tilde{\mathbf{z}}_1^i,\tilde{\mathbf{z}}_1^{\neg i*})\le\sigma, \tag{14d}$$

$$g^s(\tilde{\mathbf{z}}_1,\tilde{\mathbf{z}}_1^{\neg i*})=0,\; h^s(\tilde{\mathbf{z}}_1,\tilde{\mathbf{z}}_1^{\neg i*})\ge 0. \tag{14e}$$

2) From Relaxed MPCC to MiCP: To solve this transcribed game, we formulate the KKT conditions of the coupled optimization problem, i.e.,

$$\begin{bmatrix}\nabla_{\tilde{\mathbf{z}}_1^1}\tilde{\mathcal{L}}^1\\ g^1\\ \vdots\\ \nabla_{\tilde{\mathbf{z}}_1^N}\tilde{\mathcal{L}}^N\\ g^N\\ g^s\end{bmatrix}=0 \text{ and } 0\le\begin{bmatrix}c^1\\ \vdots\\ c^N\\ h^s\end{bmatrix}\perp\boldsymbol{\lambda}\ge 0, \tag{15}$$

where

$$\tilde{\mathcal{L}}^i(\tilde{\mathbf{z}}_1^i,\tilde{\lambda}_1^i,\tilde{\mu}_1^i,\lambda^s,\mu^s)=J_1^i(\tilde{\mathbf{z}}_1^i,\tilde{\mathbf{z}}_1^{\neg i})-c^i(\tilde{\mathbf{z}}_1^i,\tilde{\mathbf{z}}_1^{\neg i})^\top\tilde{\lambda}_1^i -g^i(\tilde{\mathbf{z}}_1^i,\tilde{\mathbf{z}}_1^{\neg i})^\top\tilde{\mu}_1^i-h^s(\tilde{\mathbf{z}}_1)^\top\tilde{\lambda}^s-g^s(\tilde{\mathbf{z}}_1)^\top\tilde{\mu}^s, \tag{16}$$

is the Lagrangian of the ith player's problem with Lagrange multipliers $(\tilde{\lambda}_1^i,\tilde{\mu}_1^i,\tilde{\mu}^s,\tilde{\lambda}^s)$,

$$c^i(\tilde{\mathbf{z}}_1^i,\tilde{\mathbf{z}}_1^{\neg i}):=\begin{bmatrix}h^i(\tilde{\mathbf{z}}_1^i,\tilde{\mathbf{z}}_1^{\neg i})\\ G^i(\tilde{\mathbf{z}}_1^i,\tilde{\mathbf{z}}_1^{\neg i})\\ H^i(\tilde{\mathbf{z}}_1^i,\tilde{\mathbf{z}}_1^{\neg i})\\ \sigma-G^i(\tilde{\mathbf{z}}_1^i,\tilde{\mathbf{z}}_1^{\neg i})^\top H^i(\tilde{\mathbf{z}}_1^i,\tilde{\mathbf{z}}_1^{\neg i})\end{bmatrix} \tag{17}$$

denotes the aggregated vector of player ith (private) inequality constraints in (14b) to (14d), and $\boldsymbol{\lambda}$ denotes the aggregation of all players Lagrange multiplies associated with inequality constraints. The resulting KKT conditions in (15) take the form of a standard MiCP [21, Definition 1.1.6] for which off-the-shelf solvers exist.

3) Proposed Algorithm: With the above relaxation scheme at hand, we numerically solve the original (unrelaxed) GOOP via a sequence of successively tightened relaxations (15); i.e. with σ successively approaching zero.

Our proposed procedure is summarized in Algorithm 1 in which GOOP(σ) denotes the relaxed MiCP at tightness σ. Specifically, we start by initializing $\tilde{\mathbf{z}}$ as a vector of zeros of the appropriate dimension and setting σ as a small positive number. We then solve the resulting MiCP using the PATH solver [30], and repeat for successively smaller σ using each solution $\tilde{\mathbf{z}}$ as an initial guess for the next round. In this way, we gradually drive σ to zero and find a local GNE solution such that the maximum violation of complementarity, $\max_j\{G_j^i(\tilde{\mathbf{z}}_1^i,\tilde{\mathbf{z}}_1^{\neg i})H_j^i(\tilde{\mathbf{z}}_1^i,\tilde{\mathbf{z}}_1^{\neg i})\}_{i=1}^N$ is below a certain tolerance, $\gamma>0$. The convergence of such annealing procedures has been widely studied in the context of general mathematical program with equilibrium constraintss (MPECs) and MPCCs. Under tailored constraint qualifications outlined in [22], [25], the stationary points of the relaxed problems converge to a weak stationary point of the underlying MPEC. For more details on convergence results, we refer readers to [22], [25]. If, at any iteration, the current solution does not change significantly from the previous one, i.e., by more than a fixed tolerance $\epsilon>0$, we consider the solution has converged. We refer to [22] for guidelines on selecting $(\sigma,\kappa,\gamma,\epsilon)$ in Algorithm III-B3.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Algorithm 1: Relaxed Game of Ordered Preferences.

$\mathbf{z}_0, \sigma_0, \kappa, \gamma, \epsilon \leftarrow$ initial guess, relaxation factor, update factor, complementarity tolerance, converged tolerance
Set $k \leftarrow 1$
while $\max_j \{G_j^i(\mathbf{z}_k) H_j^i(\mathbf{z}_k)\}_{i=1}^N \geq \gamma$ *or* $k = 1$ **do**
  $\mathbf{z}_k \leftarrow$ solution of GOOP(σ_{k-1}) initialized at $\mathbf{z}_{k-1}$
  if $\max_j \{G_j^i(\mathbf{z}_k) H_j^i(\mathbf{z}_k)\}_{i=1}^N \leq \gamma$ **then**
    break ; `// Solution is found`
  else if $||\mathbf{z}_k - \mathbf{z}_{k-1}||_2 < \epsilon$ **then**
    break ; `// Low precision solution`
  else
    $\sigma_k \leftarrow \kappa \sigma_{k-1}$; `// Reduce` $\sigma \downarrow 0$
    $k \leftarrow k + 1$
return: $\mathbf{z}_k, \sigma_k, \max_j \{G_j^i(\mathbf{z}_k) H_j^i(\mathbf{z}_k)\}_{i=1}^N$

IV. Experiments

This section evaluates the performance of the proposed GOOP approach in a Monte Carlo study and compares it with a baseline that encodes the ordered preferences via penalty-based scalarization in a non-hierarchical game formulation. These experiments are designed to support the claims that (i) GOOP reliably reflect agents' individual ordered preferences and that (ii) penalty-based approximate scalarization schemes fail to capture such solutions. Finally, we also present a scenario with more complex dynamics and preferences to illustrate the practical generality of the GOOP framework.

A. Experiment Setup

Evaluation Scenario: Our experiment extends the previous running example of highway driving scenario, where we consider $N = 3$ vehicles: vehicle 1 (blue) is an ambulance that wishes to travel at high speed, and vehicles 2 (red) and 3 (green) are passenger cars just ahead of the ambulance. Each vehicle adheres to a specific hierarchy of preferences, as outlined in (8). Note that the road length is set to $56\,\mathrm{m}$, lane width to $13\,\mathrm{m}$ and the speed limit to $5.6\,\mathrm{m/s}$. Vehicles are modeled as point-masses and are required to drive within the lane. To avoid collision, a minimum required safety distance of $5.6\,\mathrm{m}$ between vehicles is enforced.

Initial State Distribution: In order to evaluate the performance of each method, we consider a wide variety of initial conditions. To focus on more challenging scenarios, i.e. those with conflicting objectives, we construct the set of initial conditions as follows. First, we generate 10 base scenarios at which at least one vehicle cannot achieve all of their preferences perfectly. We then sample 10 additional initial states from a uniform distribution centered around each base scenario. We thus obtain a total 100 challenging scenarios.

Evaluation Metrics: For each of these test problems, (i) we evaluate methods based on the preferences *at each priority level* for each player and (ii) we measure the L_1 distance between the trajectories found by each method. To account for the existence of multiple equilibria, we solve the GOOP 20 times, each time using a different initial guess. We report the distance between the baseline trajectory and the *closest* GOOP trajectory.

B. Baseline: Explicitly Weighting Preferences

When one does not have access to a solver capable of encoding preference hierarchies explicitly—the key feature of our proposed approach—one may instead attempt to encode the concept of ordered preferences via scalarized objective. A natural scalarization scheme is a weighted sum of objectives per player—a technique that as been previously explored by [14] in non-game-theoretic motion planning. We use a game-theoretic variant of this approach as a baseline. Thus, for the baseline, each player solves a problem of the following form:

$$\min_{\mathbf{z}^i} \quad \alpha_1 J_1^i(\mathbf{z}^i, \mathbf{z}^{\neg i}) + \alpha_2 J_2^i(\mathbf{z}^i, \mathbf{z}^{\neg i}) + \alpha_3 J_3^i(\mathbf{z}^i, \mathbf{z}^{\neg i}) \tag{18a}$$

$$\text{s.t.} \quad g^i(\mathbf{z}^i, \mathbf{z}^{\neg i}) = 0, h^i(\mathbf{z}^i, \mathbf{z}^{\neg i}) \geq 0, \tag{18b}$$

$$g^s(\mathbf{z}) = 0, h^s(\mathbf{z}) \geq 0. \tag{18c}$$

Here, $[\alpha_1, \alpha_2, \alpha_3] = [1, \alpha, \alpha^2]^\top$ in (18a) is the vector of penalty weights assigned to each prioritized preference. To encode relative importance analogously to the hierarchical formulation in (4), we choose $\alpha > 1$.

Baseline variants: Observe that, for large penaty weights, the scalarized objective (18a) ensures a large separation of preferences at different hierarchy levels. Hence, one may be tempted to choose $\alpha \gg 1$. However, large penalty weights negatively affect the conditioning of the problem (18). Since it is not straightforward to determine the lowest value of α that enforces the preference hierarchy, we instead consider several variants of the baseline with $\alpha \in \{1, 10, 20, 30, 40, 50\}$.

C. Implementation Details

We implement Algorithm 1 and the aforementioned baseline in the Julia programming language.[2] To ensure a fair comparison, we implement all methods using the same MiCP solver, namely PATH [30].

Non-smooth objectives: Note that some of the objectives are not smooth, cf. (8b), and (8c), posing a challenge for numerical optimization. However, since these objectives take the form $J_k^i(\mathbf{z}_k^i, \mathbf{z}_1^{\neg i}) := \max(0, -f_k^i(\mathbf{z}_k^i, \mathbf{z}_1^{\neg i}))$, we can introduce a slack variable transformation to obtain a smooth problem, i.e., we can reformulate $\min_{\mathbf{z}_k^i} \max(0, -f_k^i(\mathbf{z}_k^i, \mathbf{z}_1^{\neg i}))$ as:

$$\min_{\mathbf{z}_k^i, s_i^k} \quad s_k^i \tag{19a}$$

$$\text{s.t.} \quad s_k^i \geq -f_k^i(\mathbf{z}_k^i, \mathbf{z}_1^{\neg i}), \tag{19b}$$

$$s_k^i \geq 0. \tag{19c}$$

D. Large-Scale Quantitative Results

Table I shows the performance gap between our method (game of ordered preference (GOOP) (4) as implemented by Algorithm 1) and the baseline variants (game (18)) with different penalty parameters. Here, $\tilde{J}_k$ and J_k denote the performance at preference level k for the *baseline* and *our* method, respectively.

Out of 100 test cases, Algorithm 1 did not converge for six of the initial conditions at which the three vehicles were approximately collinear; we hypothesize that these instances correspond to boundaries between homotopy classes. Therefore,

[2]Source code is available at https://github.com/CLeARoboticsLab/ordered-preferences.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE I
DIFFERENCE OF PREFERENCES VALUES AT EACH PRIORITY LEVEL ACROSS DIFFERENT α

Robot	α	$\tilde{J}_3 - J_3$	$\tilde{J}_2 - J_2$
R1 (Ambulance)	1	0.168 ± 0.004	-1.76 ± 0.54
	10	0.179 ± 0.003	-1.87 ± 0.54
	30	0.043 ± 0.044	-0.22 ± 0.70
	50	0.046 ± 0.055	-0.01 ± 1.03
R2 (Passenger Car)	1	0.000 ± 0.001	0.00 ± 0.01
	10	0.000 ± 0.001	0.00 ± 0.01
	30	0.000 ± 0.001	0.00 ± 0.01
	50	0.002 ± 0.024	0.00 ± 0.01
R3 (Passenger Car)	1	0.00 ± 0.00	0.00 ± 0.00
	10	0.00 ± 0.00	0.00 ± 0.00
	30	0.00 ± 0.00	0.00 ± 0.00
	50	0.00 ± 0.00	0.00 ± 0.00

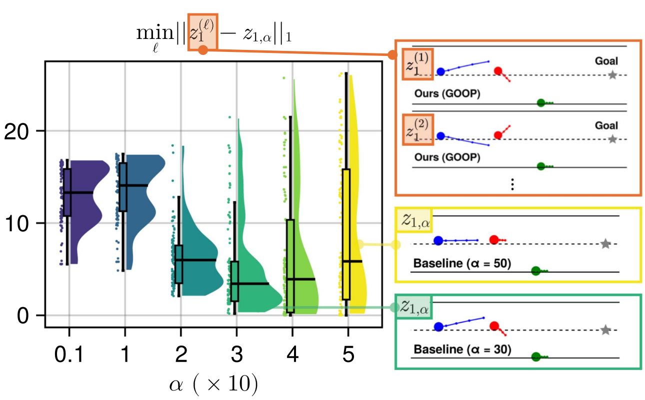

Fig. 4. L_1-trajectory distance between GOOP solutions and baseline approximations at penalty strength α ($z_{1,\alpha}$) for the 3-vehicle ambulance scenario. To facilitate a fair comparison when multiple solutions exist, we compute the distance as the *minimum* difference between the baseline trajectory and *all* available GOOP equilibria computed for the same initial condition. Note $z_1^{(\ell)}$ refers to the ℓth GOOP trajectory for the given initial condition, where $\ell = 1, 2, \ldots, 20$.

the results below reflect only the remaining 94 test cases. For reference, the baselines converged for all cases.

Main Result 1: Preference Prioritization in GOOP: Table I shows the performance gap with respect to the multiple preference levels. We see that the performance gap at the highest preference level (level 3) is always positive (up to solver precision), indicating that our method finds solutions that perform better with respect to the highest priority preference. Furthermore, Table I indicates that our method achieves this performance by "backing down" on lower priority preferences as indicated by the largely negative gap with respect to this metric. In sum, these results support the claim that our method respects the order of preferences: GOOP solutions relax less important preferences in favor of more important ones.

Baseline performance: The baseline attenuates the performance gap as the penalty parameter α increases. However, even with the largest penalty weight, i.e., $\alpha = 50$, the baseline fails to consistently match our method's performance and exhibits a high variance. This effect can be attributed to poor numerical conditioning of the problem for large weights.

Main Result 2: Distance between baseline and GOOP solutions: Fig. 4 measures the L_1 distance between the baseline and GOOP equilibrium trajectories for each test case. Although

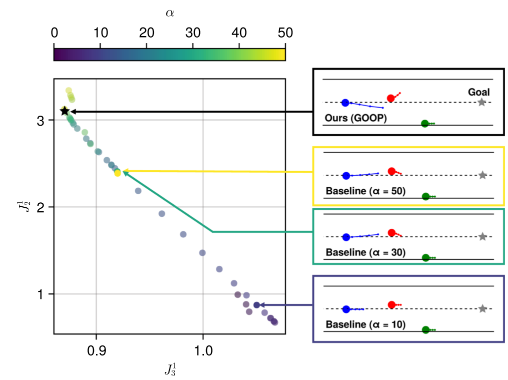

Fig. 5. Comparison of vehicle 1's highest (get to goal) and second highest (obey speed limit) preference values for GOOP and baseline for different values of α. Increasing α initially improve the trajectory for R1. However, the performance improvement is not monotonic since at $\alpha = 30$ and $\alpha = 50$, the baseline yields a degraded trajectory for R1, i.e. farther away from the goal position.

higher α values occasionally improve the baseline performance (as the lower end of the distributions approaches zero), for sufficiently high values of α the baseline exhibits poor numerical conditioning, resulting in a large variance in the solution quality, i.e., the scalarized approximations do not always recover the GOOP equilibria. This result shows the limitations of approximating GOOP solutions via scalarization.

E. Detailed Analysis for a Fixed Scenario

To provide additional intuition beyond the large-scale evaluation in Section IV-D, next, we assess a *fixed* scenario in greater detail. Fig. 5 visualizes the solutions identified by both Algorithm 1 and the baseline for a single initial state and a dense sweep over the penalty weight, i.e., $\alpha \in \{1, 2, \ldots, 49, 50\}$. In Fig. 5, we plot player 1's preferences at level 3 (J_3^1) over their preferences at level 2 (J_2^1). To illustrate how the solutions in Fig. 5 correspond to open-loop trajectories, we link selected points to their respective trajectories on the right side.

Quantitative Results: Our GOOP solution, marked by a star at the top left, outperforms all baselines, achieving the lowest value of preference at the most important level. All baseline solutions are located to the right of the GOOP solution, indicating that the baselines do not consistently match GOOP in optimizing the highest priority preference. In line with the large-scale evaluation in Section IV-D, we observe that larger weights do not consistently improve performance. In fact, R1's trajectory worsens for $\alpha = 30$ and $\alpha = 50$.

Qualitative Results: Recall that for R1, reaching its goal has the highest priority. By accurately encoding this prioritization, our method finds a solution that brings R1 closer to the goal at the final time step than all baseline variants. For R2 and R3, all methods achieve comparable performance with respect to all prioritized preferences. In summary, these results further support the claim that the equilibrium solutions computed by Algorithm 1 reflect players' hierarchical preferences.

F. An Intersection Scenario

We present an intersection scenario involving two vehicles, each with *four* levels of preferences. In accordance with the order of preferences outlined in Fig. 1, vehicle 1 accelerates beyond the speed limit (of $10\,\mathrm{m/s}$). In contrast, vehicle 2 maintains its speed but sacrifices reaching the goal. Both vehicles achieve the top preference by sacrificing their less important preferences, showing that our GOOP framework accurately captures the hierarchy of preferences in settings with deeply nested objectives.

V. Conclusion and Future Work

In this paper, we proposed the game of ordered preference (GOOP), a multi-agent, noncooperative game framework where each player optimizes over their own hierarchy of preferences. We recursively derived first-order optimality conditions for each player's optimization problem, which introduces complementarity constraints. We proposed a relaxation-based algorithm for solving the N-player KKT system for approximate (local) GNE solutions. Our experiments show that our algorithm outperforms penalty-based baselines while accurately reflecting each individual's order of preferences by relaxing lower-priority preferences when needed. Future work may focus on tailored numerical solvers that avoid the need for iteratively solving relaxed MiCP subproblems. Our formulation's problem size grows exponentially in the number of hierarchy levels. While real-time performance is not the immediate goal of this work, future work should address this limitation, in which case our work can serve as a reference solution technique. Future work may also explore amortized optimization via neural network policies that approximate equilibrium solutions, potentially enabling larger-scale deployments with many players and deeper preference hierarchies. Finally, extending GOOP to incorporate feedback mechanisms for dynamic information presents an exciting opportunity for applications such as autonomous mobile agents.

References

[1] S. Levine and V. Koltun, "Continuous inverse optimal control with locally optimal examples," in *Proc. 29th Int. Cof. Int. Conf. Mach. Learn.*, 2012, pp. 475–482.

[2] A. Y. Ng and S. J. Russell, "Algorithms for inverse reinforcement learning," in *Proc. 17th Int. Conf. Mach. Learn.*, 2000, pp. 663–670.

[3] G. Anandalingam and T. Friesz, "Hierarchical optimization: An introduction," *Ann. Operations Res.*, vol. 34, pp. 1–11, 1992.

[4] Y.-J. Lai, "Hierarchical optimization: A satisfactory solution," *Fuzzy Sets Syst.*, vol. 77, pp. 321–335, 1996.

[5] G. B. Allende and G. Still, "Solving bilevel programs with the KKT-approach," *Math. Program.*, vol. 138, pp. 309–332, 2013.

[6] S. Dempe, V. V. Kalashnikov, and N. Kalashnykova, "Optimality conditions for bilevel programming problems," in *Optimization with Multivalued Mappings: Theory, Applications, and Algorithms*. 2006, pp. 3–28.

[7] S. Dempe and A. B. Zemkoho, "The bilevel programming problem: Reformulations, constraint qualifications and optimality conditions," *Math. Program.*, vol. 138, pp. 447–473, 2013.

[8] S. Dempe, V. Kalashnikov, G. A. Pérez-Valdés, and N. Kalashnykova, "Bilevel programming problems," in *Energy Systems*. New York, NY, USA: Springer, 2015.

[9] M. Kochenderfer, "*Algorithms for Optimization*. Cambridge, MA, U.K.: The MIT Press Cambridge, 2019.

[10] M. Anilkumar et al., "Lexicographic optimization based MPC: Simulation and experimental study," *Comput. Chem. Eng.*, vol. 88, pp. 135–144, 2016.

[11] S. Khosravani, M. Jalali, A. Khajepour, A. Kasaiezadeh, S. Chen, and B. Litkouhi, "Application of lexicographic optimization method to integrated vehicle control systems," *IEEE Trans. Ind. Electron.*, vol. 65, no. 12, pp. 9677–9686, Dec. 2018.

[12] A. Escande et al., "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *Int. J. Robot. Res.*, vol. 33, pp. 1006–1028, 2014.

[13] F. Laine, "Mathematical program networks," 2024, *arXiv:2404.03767*.

[14] S. Veer, K. Leung, R. K. Cosner, Y. Chen, P. Karkus, and M. Pavone, "Receding horizon planning with rule hierarchies for autonomous vehicles," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 1507–1513.

[15] 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 Robot. Automat. Lett.*, vol. 6, no. 3, pp. 4978–4985, Jul. 2021.

[16] A. Zanardi et al., "Posetal games: Efficiency, existence, and refinement of equilibria in games with prioritized metrics," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 1292–1299, Apr. 2022.

[17] W. Schwarting et al., "Social behavior for autonomous vehicles," in *Proc. Nat. Acad. Sci.*, 2019, pp. 24972–24978.

[18] J. F. Fisac, E. Bronstein, E. Stefansson, D. Sadigh, S. S. Sastry, and A. D. Dragan, "Hierarchical game-theoretic planning for autonomous vehicles," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 9590–9596.

[19] M. Wang, Z. Wang, J. Talbot, J. C. Gerdes, and M. Schwager, "Game-theoretic planning for self-driving cars in multivehicle competitive scenarios," *IEEE Trans. Robot.*, vol. 37, no. 4, pp. 1313–1325, Aug. 2021.

[20] J. Geary, S. Ramamoorthy, and H. Gouk, "Resolving conflict in decision-making for autonomous driving," in *Proc. Robot.: Sci. Syst.*, 2021.

[21] J.-S. P. F. Facchinei, *Finite-Dimensional Variational Inequalities and Complementarity Problems*. New York, NY, USA: Springer, 2007.

[22] S. Scholtes, "Convergence properties of a regularization scheme for mathematical programs with complementarity constraints," *SIAM J. Optim.*, vol. 11, pp. 918–936, 2001.

[23] S. Leyffer et al., "Interior methods for mathematical programs with complementarity constraints," *SIAM J. Optim.*, vol. 17, pp. 52–77, 2006.

[24] A. Schwartz, "Mathematical programs with complementarity constraints: Theory, methods and applications," Ph.D. dissertation, Universität Würzburg, Würzburg, Germany, 2011.

[25] T. Hoheisel et al., "Theoretical and numerical comparison of relaxation methods for mathematical programs with complementarity constraints," *Math. Program.*, vol. 137, pp. 257–288, 2013.

[26] M. Anitescu, "On using the elastic mode in nonlinear programming approaches to mathematical programs with complementarity constraints," *SIAM J. Optim.*, vol. 15, pp. 1203–1236, 2005.

[27] S. Dempe and A. Zemkohoo, "Bilevel optimization: Advances and next challenges," in *Springer Optimization and Its Applications*, vol. 161. Springer, 2020.

[28] A. Nurkanović, A. Pozharskiy, and M. Diehl, "Solving mathematical programs with complementarity constraints arising in nonsmooth optimal control," *Vietnam J. Math.*, pp. 1–39, 2024.

[29] F. Facchinei and C. Kanzow, "Generalized Nash equilibrium problems," *Ann. Operations Res.*, vol. 175, pp. 177–211, 2010.

[30] S. P. Dirkse and M. C. Ferris, "The path solver: A nommonotone stabilization scheme for mixed complementarity problems," *Optim. Methods Softw.*, vol. 5, pp. 123–156, 1995.

[31] T. F. Rutherford, "Mixed complementarity programming with gams," *Lecture Notes for Econ*, vol. 6433, pp. 1299–1324, 2002.

[32] R. W. Cottle, J.-S. Pang, and R. E. Stone, *The Linear Complementarity Problem*. Philadelphia, PA, USA: SIAM, 2009.