A Convex Formulation of Game-Theoretic Hierarchical Routing

Dong Ho Lee, *Graduate Student Member, IEEE*, Kaitlyn Donnel, Max Z. Li, *Member, IEEE*, and David Fridovich-Keil, *Member, IEEE*

***Abstract*—Hierarchical decision-making is a natural paradigm for coordinating multi-agent systems in complex environments such as air traffic management. In this letter, we present a bilevel framework for game-theoretic hierarchical routing, where a high-level router assigns discrete routes to multiple vehicles who seek to optimize potentially noncooperative objectives that depend upon the assigned routes. To address computational challenges, we propose a reformulation that preserves the convexity of each agent's feasible set. This convex reformulation enables a solution to be identified efficiently via a branch-and-bound algorithm. Our approach ensures global optimality while capturing strategic interactions between agents at the lower level. We demonstrate the solution concept of our framework in two-vehicle and three-vehicle routing scenarios.**

***Index Terms*—Game theory, autonomous systems, optimization.**

I. Introduction

Coordinating multiple vehicles in a controlled fashion is a crucial challenge for decision-making in applications such as autonomous transportation systems and air traffic management. Such problems are inherently hierarchical: routing decisions assign paths to multiple vehicles, and individual vehicles determine optimal trajectories in response to their assigned routes. Current approaches treat these problems in isolation, i.e., higher-level routing decisions operate on discrete graphs which abstract away the continuous lower-level trajectory optimization process. In this letter, we present a technique for solving these coupled problems *simultaneously*, and showcase its performance in an example inspired by air traffic management. Ultimately, our results suggest that integrating the discrete (multi-)vehicle routing problem with low-level trajectory design can reduce vehicles' control effort and yield more efficient system-wide performance.

A natural approach to capturing this hierarchical structure is through bilevel optimization, where the upper level handles discrete decision-making in the form of route assignments, and the lower level encodes continuous optimization of individual vehicles' trajectories. Information flows between these levels bidirectionally. The upper-level routing problem accounts for the physical state and operational requirements of individual vehicles, while, at the lower-level, the vehicles generate trajectories that respect the prescribed routing plan.

On their own, routing problems are well-studied in the context of (mixed) integer programs [1], [2]. Similarly, the lower-level trajectory optimization is well-explored in both single- and multi-agent contexts. We consider the more general multi-agent, potentially noncooperative, variants of these problems.

Concretely, we make the following contributions. *(i)* We formulate hierarchical vehicle routing and trajectory design problems as a mixed-integer bilevel program, where the upper level models a discrete routing problem and the lower level encodes a (potentially noncooperative) trajectory design game played among M vehicles. *(ii)* We propose a convex reformulation that integrates the two levels, and solve the resulting problem via a branch-and-bound algorithm that guarantees global optimality. Experimental results demonstrate how solving these coupled problems in tandem can yield more efficient solutions and showcase the proposed approach in a scenario inspired by air traffic management.

II. Related Work

Current optimization models for air traffic management typically focus on strategic decisions, such as optimizing flight delays [3], [4] or airspace sector route assignments [5]. Recent work examines hierarchical congestion pricing and route planning via a bilevel optimization approach, but ignores vehicle dynamics entirely [6]. On the other hand, aircraft and air vehicle trajectory optimization studies (e.g., [7]) typically do not jointly optimize higher-level strategic routing or traffic flow management decisions. This separation between strategic decision-making and trajectory optimization can lead to inefficiencies, as strategic plans may not fully account for vehicle dynamics and interactions at the trajectory level. This letter contributes an approach which begins to close this gap.

At the trajectory level, multi-agent interactions play a critical role in ensuring efficient airspace management. Game-theoretic approaches [8], [9], [10], [11], [12], [13], [14] have

Received 17 March 2025; revised 4 May 2025; accepted 12 May 2025. Date of publication 19 May 2025; date of current version 3 June 2025. This work was supported by the National Science Foundation CAREER Award under Grant 2336840. Recommended by Senior Editor C. Briat. *(Corresponding author: Dong Ho Lee.)*

Dong Ho Lee, Kaitlyn Donnel, 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; kaitlyndonnel@utexas.edu; dfk@utexas.edu).

Max Z. Li is with the Department of Aerospace Engineering, the Department of Civil and Environmental Engineering, and the Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: maxzli@umich.edu).

Digital Object Identifier 10.1109/LCSYS.2025.3571655

been widely explored, and efficient methods exist which can compute approximate and/or local Nash equilibrium strategies in multi-agent, noncooperative settings. However, these approaches generally assume a fixed environment *without* a high-level planner that can influence individual agents' decisions.

Hierarchical game-theoretic frameworks have also been proposed for trajectory optimization, primarily in the context of human-autonomous vehicle interactions [15], [16]. These works typically model motion planning as a dynamic Stackelberg game, where an autonomous vehicle (follower) anticipates and reacts to a human driver (leader).

Existing works on bilevel optimization for routing problems [17], [18], [19] consider hierarchical decision-making between multiple stakeholders. However, they often assume cooperative settings and do not model interactions between individual entities at the lower level. Moreover, their solution methods often rely on metaheuristic algorithms such as genetic algorithms, or involve linearizing nonconvex constraints to obtain tractable relaxed formulations.

The control literature has similarly addressed multi-vehicle routing and trajectory optimization problems via decentralized frameworks [20], receding-horizon controllers [21], [22] or heuristics-based approximations [23]. For example, recent work [24] studies optimal assignment of a fleet of electric aircraft and their charge scheduling aimed at reducing energy consumption. However, these prior works generally either consider discrete route assignment *only* or treat discrete routing separately from continuous vehicle dynamics.

Existing work on hierarchical routing problems in control contexts typically approaches discrete pathfinding and continuous trajectory optimization *sequentially* [25], [26]. Our core contribution is to bridge this gap by presenting a unified mixed-integer bilevel framework that *simultaneously* integrates high-level routing with trajectory optimization.

Our proposed framework facilitates more efficient management of multiple vehicles and is readily applicable for future air traffic management automation systems (c.f. Section VI).

III. Problem Formulation

In this section, we formulate a hierarchical routing game problem. This formulation can easily accommodate additional application-specific constraints, cf. Section IV.

A. Preliminaries

We represent the environment as a graph with N nodes, each corresponding to a point of interest to be visited (e.g., airspace navigational aids), and M vehicles, each generating their own continuous state/action trajectories. The nodes are defined in a 2D Cartesian plane as $\mathcal{P} := \{\hat{\mathbf{p}}_i\}_{i=1}^N = \{(\hat{x}_i, \hat{y}_i)\}_{i=1}^N$. Additionally, each vehicle's trajectory starts at their own *start* node S^j, whose position is $\hat{\mathbf{p}}_S^j = (\hat{x}_S^j, \hat{y}_S^j)$, and ends at *terminal* node T^j, which is located at $\hat{\mathbf{p}}_T^j = (\hat{x}_T^j, \hat{y}_T^j)$. A *waypoint* is any node included in the route; a route has K waypoints, and cannot have more than N waypoints, i.e., $K \leq N$. Between each pair of consecutive waypoints, T intermediate *subwaypoints* define a trajectory segment.

B. Proposed Bilevel Framework

The proposed bilevel program is of the form:

$$\min_{\substack{\mathbf{z}_u \in \mathcal{Z}_u \\ \mathbf{z}_l \in \mathcal{Z}_l}} \quad J_u(\mathbf{z}_u, \mathbf{z}_l) \tag{1a}$$

$$\text{s.t.} \quad \forall j \in [M] \quad \left\{ \mathbf{z}_l^j \in \operatorname*{argmin}_{\tilde{\mathbf{z}}_l^j \in \mathcal{Z}_l^j\left(\mathbf{z}_u, \tilde{\mathbf{z}}_l^{-j}\right)} J_l^j\left(\mathbf{z}_u, \tilde{\mathbf{z}}_l\right), \right. \tag{1b}$$

where $\mathbf{z}_u \in \mathbb{R}^{n_u}$ and $\mathbf{z}_l \in \mathbb{R}^{n_l}$ represent the upper-level (discrete) and the lower-level (continuous) decision variables, respectively. Similarly, $J_u(\cdot) : \mathbb{R}^{n_u} \times \mathbb{R}^{n_l} \to \mathbb{R}$ and $J_l(\cdot) : \mathbb{R}^{n_u} \times \mathbb{R}^{n_l} \to \mathbb{R}$ denote the objective functions at the upper and lower level, respectively. We formulate the upper level as a discrete problem that consists of binary decision variables $\mathbf{z}_u := [\mathbf{z}^1, \ldots, \mathbf{z}^M]$ where $\mathbf{z}^j := [z^j_{S^j,1}, \ldots, z^j_{S^j,K}, z^j_{1,1}, \ldots z^j_{i,k}, \ldots z^j_{T^j,K}] \in \{0,1\}^{NK}, \ \forall j \in [M]$. The binary variable $z^j_{i,k}$ is defined such that $z^j_{i,k} = 1$ if node i is the k^{th} waypoint for vehicle j, and $z^j_{i,k} = 0$ otherwise. The lower-level decision variables $\mathbf{z}_l := [\mathbf{z}_l^1, \ldots, \mathbf{z}_l^M]$ are continuous, and for each vehicle j, $\mathbf{z}_l^j := (\mathbf{x}^j, \mathbf{u}^j) = (x^j_{1:K,1:T}, u^j_{1:K,1:T})$ where $x^j_{k,t} \in \mathbb{R}^{n_x}$ and $u^j_{k,t} \in \mathbb{R}^{n_u}$ represent the state and the control input variables of the j^{th} vehicle. The superscript on $\mathbf{z}_l^j \in \mathbb{R}^{n_l^j}$ in (1b) refers to the j^{th} vehicle's variables. We use $\mathbf{z}_l^{-j} \in \mathbb{R}^{n_l - n_l^j}$ to denote the variables of all vehicles except j. $[M]$ in (1b) denotes the set $\{1, \ldots, M\}$. In subsequent sections, we characterize the structure of the constraints and cost functions in (1).

C. Upper-Level Routing Problem

The upper level routing problem in (1) models a *router* that determines the sequence of waypoints for all vehicles' routes, given fixed *initial* and *terminal* nodes for each vehicle. $\mathcal{Z}_u$ in (1a) denotes the feasible set for $\mathbf{z}_u$ and is defined in terms of constraints that must be satisfied for a feasible route. By definition, the following constraints naturally encode the beginning and the end of a trajectory:

$$z^j_{S^j,1} = 1, z^j_{T^j,K} = 1, \ \forall j \in [M]. \tag{2}$$

No vehicle can be assigned two waypoints at the same time, and at most one vehicle can be assigned any waypoint at the same time. These constraints are encoded by:

$$\sum_{i=1}^N z^j_{i,k} = 1, \ \forall k \in [K], j \in [M], \tag{3}$$

$$\sum_{j=1}^M z^j_{i,k} \leq 1, \ \forall i \in [N], k \in [K]. \tag{4}$$

Furthermore, a node may not be visited more than once by any vehicle, i.e.,

$$\sum_{k=1}^K z^j_{i,k} \leq 1, \ \forall i \in [N], j \in [M]. \tag{5}$$

Equations (2) to (5) together define $\mathcal{Z}_u$ in (1a).

D. Lower-Level Trajectory Game

The lower-level trajectory planning problem in (1b) is mathematically formulated as a multi-agent trajectory game, whose solution is a Nash equilibrium (NE) [27]. We consider a variant of linear quadratic (LQ) games [28] for this level, i.e., (1b) is in the form of:

$$\min_{\mathbf{x}^j,\mathbf{u}^j} \overbrace{\frac{1}{2}\sum_{\substack{k=1\\t=1}}\Big(Q\Big(x^j_{k,t}, z^j_{1:N,k}\Big) + \|u^j_{k,t}\|_2^2\Big) + I\Big(\mathbf{x}^j, \mathbf{x}^{\neg j}\Big)}^{J^j_l(\mathbf{z}_u,\mathbf{x},\mathbf{u}^j)} \tag{6a}$$

$$\text{s.t. } x^j_{k,t+1} = Ax^j_{k,t} + Bu^j_{k,t},\ \forall k < K,\ \forall t < T, \tag{6b}$$

$$x^j_{k+1,1} = Ax^j_{k,T} + Bu^j_{k,T},\ \forall k < K, \tag{6c}$$

$$\underline{x} \le x^j_{k,t} \le \bar{x},\ \forall k \in [K],\ \forall t \in [T], \tag{6d}$$

$$\underline{u} \le u^j_{k,t} \le \bar{u},\ \forall k \in [K],\ \forall t \in [T]. \tag{6e}$$

Equations (6b) to (6e) together characterize $\mathcal{Z}_l$ in (1). The subscripts (k, t) in (6) correspond to time step t, after passing the k^{th} waypoint. Specifically, (6b) describes state transition within the trajectory segment k, i.e., from the k^{th} to the $(k+1)^{\text{th}}$ waypoint. Similarly, (6c) defines the changes from the final time step T of segment k to the initial time step of segment $k+1$. Equations (6d) to (6e) encode the bounds on the state and control variables for the j^{th} vehicle. Note that—despite their apparent restriction to linear dynamical systems and quadratic agent objectives—LQ games can also apply in more general settings, e.g., those in which the game dynamics are differentially flat and agent objectives are quadratic in the flat outputs. We discuss the cost structure (6a) of the lower-level problem in the next section.

E. Structure of Cost Functions

We now characterize the structure of objective functions at the upper (1a) and lower (1b) levels. The upper-level routing problem minimizes the total control effort, as a proxy for fuel consumption, i.e., $J_u(\mathbf{z}_u, \mathbf{z}_l) := \sum_{j=1}^{M}\sum_{k=1}^{K}\sum_{t=1}^{T}\|u^j_{k,t}\|_2^2$. Note that the variable $u^j_{k,t}$ (6) is determined at the lower level *in response to* the routing choice $\mathbf{z}_u$ (1) at the upper level.

The cost function at the lower level, $J^j_l(\mathbf{z}_u, \mathbf{x}, \mathbf{u}^j)$, in (6a) is comprised of two parts. The first term captures individual vehicle costs such as state penalties and control effort, while the second term accounts for interaction effects among vehicles. The function $Q(x^j_{k,t}, z^j_{1:N,k})$ is convex and quadratic with respect to both $x^j_{k,t}$ and $z^j_{1:N,k}$. In Section V, we describe how $Q(x^j_{k,t}, z^j_{1:N,k})$ can penalize trajectory deviations from the upper-level routing plan. The inter-agent interaction term $I(\mathbf{x}^j, \mathbf{x}^{\neg j})$ is also a convex quadratic function, and encodes the influence of other vehicles' trajectories on vehicle j's cost. In Section V, we use this term to model aircraft formation flight.

The lower-level game consists of linear constraints and quadratic objective functions for all vehicles, i.e., the lower-level game is a convex game parameterized by $\mathbf{z}_u$. Thus, the Karush-Kuhn-Tucker (KKT) conditions for each agent's problem are both necessary and sufficient for optimality [27].

IV. Methodology

In this section, we present a KKT reformulation [29], [30] of the hierarchical routing game in (1). We introduce auxiliary variables to transform the problem into a mixed integer quadratic problem (MIQP), and solve the resulting problem via a branch-and-bound algorithm [31].

A. KKT Reformulation

For the ease of discussion, we express the lower-level LQ game (6) in the following form:

$$\textit{(Vehicle j's problem)}\quad \min_{\mathbf{x}^j,\mathbf{u}^j}\ J^j_l\Big(\mathbf{z}_u, \mathbf{x}, \mathbf{u}^j\Big) \tag{7a}$$

$$\text{s.t. } g\Big(\mathbf{x}^j, \mathbf{u}^j\Big) = 0, \tag{7b}$$

$$h\Big(\mathbf{x}^j, \mathbf{u}^j\Big) \ge 0, \tag{7c}$$

where $\mathbf{x} := [\mathbf{x}^j, \mathbf{x}^{\neg j}]$. Equation (7b) encodes the equality constraints from Equations (6b) and (6c). These govern the dynamics of the j^{th} vehicle's trajectory. Similarly, (7c) refers to the bounds in Equations (6b) and (6e), which encode the state and control bounds.

Assumption 1 (Slater's Condition for the Lower Level): For a given feasible $\mathbf{z}_u \in \mathcal{Z}_u$ of (1), Slater's constraint qualification holds for the lower-level problem, i.e., there exists $\tilde{\mathbf{x}}^j, \tilde{\mathbf{u}}^j$ s.t. $g(\tilde{\mathbf{x}}^j, \tilde{\mathbf{u}}^j) = 0$ and $h(\tilde{\mathbf{x}}^j, \tilde{\mathbf{u}}^j) > 0$.

From Assumption 1, it follows that the original bilevel problem (1) and its single-level reformulation using KKT conditions of (1b) are equivalent. In other words, we can rewrite (7) using its KKT conditions and obtain the following single-level reformulation:

$$\min_{\mathbf{z}_u,\mathbf{z}_l,\lambda,\mu}\ J_u(\mathbf{z}_u, \mathbf{z}_l) \tag{8a}$$

$$\text{s.t. } \mathbf{z}_u \in \mathcal{Z}_u, \tag{8b}$$

$$\nabla_{\mathbf{x}^j,\mathbf{u}^j}\mathcal{L}\Big(\mathbf{z}_u, \mathbf{x}, \mathbf{u}^j, \lambda^j, \mu^j\Big) = 0, \tag{8c}$$

$$0 \le h\Big(\mathbf{x}^j, \mathbf{u}^j\Big) \perp \lambda^j \ge 0, \tag{8d}$$

$$g\Big(\mathbf{x}^j, \mathbf{u}^j\Big) = 0, \quad \forall j \in [M], \tag{8e}$$

where the j^{th} agent's *Lagrangian* function is defined as $\mathcal{L}(\mathbf{z}_u, \mathbf{x}, \mathbf{u}^j, \lambda^j, \mu^j) := J^j_l(\mathbf{z}_u, \mathbf{x}, \mathbf{u}^j) - \lambda^{j\top}h(\mathbf{x}^j, \mathbf{u}^j) - \mu^{j\top}g(\mathbf{x}^j, \mathbf{u}^j)$. Note that μ^j and λ^j refer to the dual variables corresponding to the inequality and equality constraints of the j^{th} vehicle in (7), respectively.

We observe that the KKT reformulation introduces complementarity constraints in (8d), i.e., $h(\mathbf{x}^j, \mathbf{u}^j) \perp \lambda_j \equiv h(\mathbf{x}^j, \mathbf{u}^j)^\top \lambda_j = 0$, which involve bilinear terms that break the convexity of the feasible set in (8). For general nonlinear programming (NLP) problems at the upper level, these complementarity constraints (8) can be handled using relaxation-based approaches [32], [33]. However, since the upper level variables z_u are binary, this formulation results in a nonconvex mixed integer nonlinear problem (MINLP), which is intractable to solve. A standard strategy in the MINLP literature is to obtain convex relaxations of the underlying nonconvex problems [34], [35]. However, rather than developing tighter convex relaxations of an inherently nonconvex problem, we propose a more direct approach: reformulating the problem to preserve convexity from the outset.

B. Introducing Auxiliary Variables to Preserve Convexity

The nonconvexity of (8) arises from the inequality constraints in (7c) of the LQ game. To bypass this issue, we introduce auxiliary variables at the upper level to *reconstruct* the equilibrium trajectories from the lower-level problems.

To this end, we define auxiliary variables $\breve{\mathbf{z}}_u := (\breve{\mathbf{x}}^j, \breve{\mathbf{u}}^j) = (\breve{x}^j_{1:K,1:T}, \breve{u}^j_{1:K,1:T})$, and obtain the following problem:

$$\min_{\mathbf{z}_u, \breve{\mathbf{z}}_u, \mathbf{z}_l} \; J_u\big(\breve{\mathbf{z}}_u, \mathbf{z}_l\big) + \alpha \sum_{j=1}^{M} \left\| \begin{bmatrix} \breve{\mathbf{x}}^j \\ \breve{\mathbf{u}}^j \end{bmatrix} - \begin{bmatrix} \mathbf{x}^j \\ \mathbf{u}^j \end{bmatrix} \right\|_2^2 \tag{9a}$$

$$\text{s.t.} \; \mathbf{z}_u \in \mathcal{Z}_u, \tag{9b}$$

$$g\big(\breve{\mathbf{x}}^j, \breve{\mathbf{u}}^j\big) = 0, \; h\big(\breve{\mathbf{x}}^j, \breve{\mathbf{u}}^j\big) \geq 0, \forall j \in [M], \tag{9c}$$

$$\mathbf{z}^j_l \in \operatorname*{argmin}_{\tilde{\mathbf{z}}^j_l \in \tilde{\mathcal{Z}}^j_l(\mathbf{z}_u, \tilde{\mathbf{z}}^{\neg j}_l)} J^j_l\big(\mathbf{z}_u, \tilde{\mathbf{z}}_l\big), \forall j \in [M], \tag{9d}$$

where $J_u(\breve{\mathbf{z}}_u, \mathbf{z}_l) := \sum_{j=1}^{M}\sum_{k=1}^{K}\sum_{t=1}^{T} \|\breve{u}^j_{k,t}\|_2^2$ and $\alpha > 0$ is a weight parameter that penalizes large deviations from $(\mathbf{x}^j, \mathbf{u}^j)$. We can interpret $(\mathbf{x}^j, \mathbf{u}^j)$ as a *reference* trajectory that arises from the noncooperative game played at the lower level (i.e., accounting for individual and inter-agent interaction costs, cf. Section III). Note that, in (9d), $\tilde{\mathcal{Z}}^j_l := \{(\mathbf{x}^j, \mathbf{u}^j) \,|\, g(\mathbf{x}^j, \mathbf{u}^j) = 0\}$, which consists of equality constraints (7b) of vehicle j's problem. As a result, in this reformulation, the lower-level game (9d) does not have inequality constraints anymore. Instead, the inequality constraints are enforced at the top-level problem on the auxiliary variables, i.e., $(\breve{\mathbf{x}}^j, \breve{\mathbf{u}}^j)$. These auxiliary variables refer to the *adjusted* trajectory (for the j^{th} vehicle) that must still satisfy the dynamics $g(\breve{\mathbf{x}}^j, \breve{\mathbf{u}}^j) = 0$ and state-control bounds $h(\breve{\mathbf{x}}^j, \breve{\mathbf{u}}^j) \geq 0$.

With the auxiliary variables defined in (9), we can write its KKT reformulation as follows:

$$\min_{\substack{\mathbf{z}_u, \breve{\mathbf{z}}_u, \\ \mathbf{z}_l, \mu}} \; J_u\big(\breve{\mathbf{z}}_u, \mathbf{z}_l\big) + \alpha \sum_{j=1}^{M} \left\| \begin{bmatrix} \breve{\mathbf{x}}^j \\ \breve{\mathbf{u}}^j \end{bmatrix} - \begin{bmatrix} \mathbf{x}^j \\ \mathbf{u}^j \end{bmatrix} \right\|_2^2 \tag{10a}$$

$$\text{s.t.} \; \mathbf{z}_u \in \mathcal{Z}_u, \tag{10b}$$

$$g\big(\breve{\mathbf{x}}^j, \breve{\mathbf{u}}^j\big) = 0, \; h\big(\breve{\mathbf{x}}^j, \breve{\mathbf{u}}^j\big) \geq 0, \quad \forall j \in [M], \tag{10c}$$

$$\nabla_{\mathbf{x}^j, \mathbf{u}^j} \tilde{\mathcal{L}}\big(\mathbf{z}_u, \mathbf{x}, \mathbf{u}^j, \mu^j\big) = 0, \quad \forall j \in [M], \tag{10d}$$

$$g\big(\mathbf{x}^j, \mathbf{u}^j\big) = 0, \quad \forall j \in [M], \tag{10e}$$

where $\tilde{\mathcal{L}}(\mathbf{z}_u, \mathbf{x}, \mathbf{u}^j, \mu^j) = J^j_l(\mathbf{z}_u, \mathbf{x}, \mathbf{u}^j) - \mu^{j\top} g(\mathbf{x}^j, \mathbf{u}^j)$.

Proposition 1 establishes the connection between (8) and the reformulation via auxiliary variables in (10).

Proposition 1: Let $(\mathbf{z}^*_u, \mathbf{z}^*_l, \lambda^*, \mu^*)$ be an optimal solution of (8) such that $h(\mathbf{x}^{j*}, \mathbf{u}^{j*}) > 0 \; \forall j \in [M]$. Then, $\lambda^* = 0$ and the tuple $(\mathbf{z}^*_u, \breve{\mathbf{z}}^*_u, \mathbf{z}^*_l, \mu^*)$ with $\breve{\mathbf{z}}^*_u = \mathbf{z}^*_l$ is optimal for (10) with the same objective value as (8).

Proof: Because each $h(\mathbf{x}^{j*}, \mathbf{u}^{j*}) > 0$, (8d) forces $\lambda^{j*} = 0 \; \forall j$. This implies $\mathcal{L}(\mathbf{z}^*_u, \mathbf{x}^*, \mathbf{u}^{j*}, \lambda^{j*}, \mu^{j*}) = \tilde{\mathcal{L}}(\mathbf{z}^*_u, \mathbf{x}^*, \mathbf{u}^{j*}, \mu^{j*})$. Setting $\breve{\mathbf{x}}^{j*} = \mathbf{x}^{j*}$, $\breve{\mathbf{u}}^{j*} = \mathbf{x}^{j*} \; \forall j$ in (10a) recovers the same objective value as (8a) and $(\mathbf{z}^*_u, \breve{\mathbf{z}}^*_u, \mathbf{z}^*_l, \mu^*)$ satisfies the KKT conditions of (10). ■

We now state a key lemma that formalizes the convexity of the continuous relaxation of the KKT reformulation in (10).

Lemma 1: The continuous relaxation of (10) (i.e., allowing z_u to be fractional) is a convex quadratic program (QP).

Proof: Under the assumptions of Section III—that each vehicle j's lower-level cost $J^j_l(\mathbf{z}_u, \mathbf{x}, \mathbf{u}^j)$ and the upper-level cost $J_u(\breve{\mathbf{z}}_u, \mathbf{z}_l)$ are convex and quadratic and all constraints $h(\cdot)$ and $g(\cdot)$ are affine—the reformulation in the (10) does not entail complementarity constraints. As a result, the feasible set in (10) remains convex (apart from the binary nature of the variables $\mathbf{z}_u$). This, in turn, implies that the problem (10) is a mixed integer quadratic problem (MIQP). Thus, any continuous relaxation of the upper-level binary variables $\mathbf{z}_u$ yields a convex quadratic programming (QP) problem. ■

Next, we present a proposition that characterizes the global optimality of the solution to (10), which is obtained via a branch-and-bound algorithm [31].

Proposition 2: Under the same conditions as Lemma 1, one can find a global optimal solution to (10) via a branch-and-bound algorithm.

Proof: Branch-and-bound algorithms operate by solving tree-structured subproblems, each of which imposes a stricter relaxation of the integer constraints on $\mathbf{z}_u$ than its parent problem. From Lemma 1, it follows the solution to each subproblem provides a valid lower bound on the objective (10a) for every child subproblem (with smaller feasible set). At any subproblem, if a solution is found that improves upon the current best, then the current best solution is updated, which ensures that the sequence of candidate optimal values (J^{k*}_u) is decreasing monotonically. Since there are a finite number of subproblems to explore—2^{MNK}—the algorithm terminates in a finite number of steps with a globally optimal solution. ■

V. Experimental Results

In this section, we (i) compare our proposed bilevel approach with a decomposition-based baseline to illustrate how solving the two levels of planning simultaneously improves system-wide efficiency, and (ii) showcase the performance of our proposed hierarchical routing framework in a more complex example (in which agents' low-level trajectory costs are coupled), leveraging the reformulation in (10).

Comparison with baseline. To illustrate the improved efficiency provided by our method, we consider a three-vehicle routing scenario, where the conventional approach is to first solve high-level discrete routing independently of vehicle dynamics (i.e., state and control input) and subsequently generate dynamically feasible continuous trajectories based on the prescribed routes.

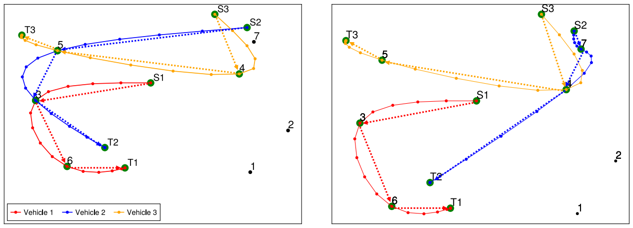
Modeling formation flight via interaction term $I(\mathbf{x}^j, \mathbf{x}^{\neg j})$. To demonstrate how our proposed framework can account for inter-agent interaction effects (at the lower-level), we consider a formation flight scenario involving two vehicles and three vehicles to demonstrate how the vehicles' interactions influence their individual trajectories in accordance with the high-level routing decisions.

A. Experiment Setup

We first formalize vehicle dynamics at the lower level, which are the equality constraints in Equations (6b) and (6c). Concretely, we model each vehicle as a double integrator with $x^j_{k,t} = [p^{j,x}_{k,t}, p^{j,y}_{k,t}, v^{j,x}_{k,t}, v^{j,y}_{k,t}]^\top \in \mathbb{R}^4$ encoding the state of the vehicle, and $u^j_{k,t} = [a^{j,x}_{k,t}, a^{j,y}_{k,t}]^\top \in \mathbb{R}^2$ representing the control input at time step $t \in [T]$. The state vector x^j_t consists of position and velocity in the horizontal and vertical directions, and the control vector u^j_t consists of acceleration in the horizontal and vertical directions, respectively. We discretize the double-integrator dynamics at resolution Δt, i.e.,

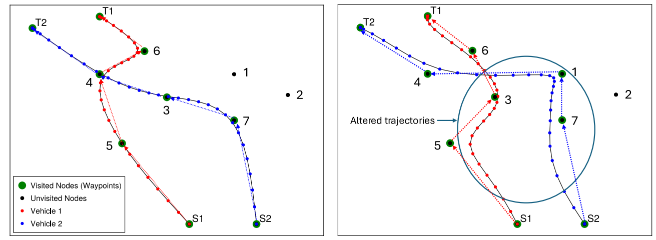
$$x^j_{k,t+1} = \underbrace{\begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{A} x^j_{k,t} + \underbrace{\begin{bmatrix} \frac{1}{2}\Delta t^2 & 0 \\ 0 & \frac{1}{2}\Delta t^2 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix}}_{B} u^j_{k,t}. \tag{11}$$

Note that (11) encodes the constraint (6b). A similar variant of (11) can be formed for (6c).



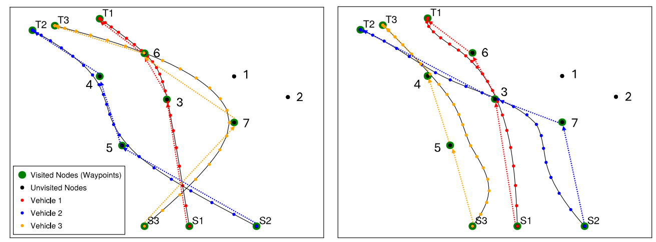
(a) Our method (b) Baseline: Brute-force

Fig. 1. Comparison of our bilevel method and a baseline, decoupled routing and trajectory design approach on a sample three-vehicle scenario.



(a) No interaction cost (b) With interaction cost

Fig. 2. Comparison of optimal vehicle routes and trajectories in a two-vehicle formation flight routing game. Dashed straight lines indicate each vehicle's visited nodes.



(a) No interaction cost (b) With interaction cost

Fig. 3. Comparison of optimal vehicle routes and trajectories in a three-vehicle formation flight routing game. Dashed straight lines indicate each vehicle's visited nodes.

Next, we define the individual vehicles' objective functions (6a). The component of each vehicle's objective which pertains only to their upper- and lower-level variables is encoded by the function $Q(x^j_{k,t}, z^j_{S^j:T^j,k}) := \frac{1}{2}\sum_{k=1}^{K} \|\mathbf{p}^j_{k,1} - \sum_{i=S^j}^{T^j} z^j_{i,k}\hat{\mathbf{p}}_i\|_2^2$. This cost term penalizes any deviations of the vehicle's state at the start of each trajectory segment k, $\mathbf{p}^j_{k,1} = (x^{j,x}_{k,1}, x^{j,y}_{k,1})$, from the *chosen* waypoint, whose location is determined by the product $z^j_{i,k}\hat{\mathbf{p}}_i$.

Comparison with baseline. The baseline method searches for the global optimal solution by brute-force enumeration, identifying the set of routes with the minimum total route cost, which is evaluated based on Euclidean distances between nodes. In the brute-force search, we perform conflict resolution among all candidate routes to ensure that no two vehicles visit the same waypoint at the same time. We then generate continuous state-action trajectories following the prescribed route plan.

Modeling formation flight via interaction term $I(\mathbf{x}^j, \mathbf{x}^{-j})$. We introduce $I(\mathbf{x}^j, \mathbf{x}^{-j})$ in (6a) to account for inter-agent interaction for the lower-level game. We model this interaction as $I(\mathbf{x}^j, \mathbf{x}^{-j}) = \sum_{j'\neq j}^{M}\sum_{k=1}^{K}\sum_{t=1}^{T}\|C(\mathbf{x}^j_{k,t} - \mathbf{x}^{j'}_{k,t} + \mathbf{r}^{jj'})\|_2^2$. In this section, we consider a one-dimensional variant of $I(\mathbf{x}^j, \mathbf{x}^{-j})$, i.e., $C = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}$. This means that $C\mathbf{x}^j_{k,t} = p^{j,x}_{k,t}$. In the two-vehicle setting (Figure 2), we consider $C\mathbf{r}^{12} = r$, and $C\mathbf{r}^{21} = -r$. Here, $r > 0$ models the situation where Vehicle 1 wishes to stay on the left of Vehicle 2 *and* Vehicle 2 wants to be on the right of Vehicle 1, with a separation of r along the horizontal axis. In the three-vehicle setting (Figure 3), each of the $\mathbf{r}^{jj'}$ are distinct.

B. Detailed Analysis of Results

Comparison with baseline. In this scenario, we have $N = 9$, $K = 4$, $T = 5$ and $M = 3$. Based on Euclidean distances, Vehicle 1 (red) vists the nodes $S_1 \to 3 \to 6 \to T_1$, Vehicle 2 (blue) visits $S_2 \to 7 \to 4 \to T_2$, and Vehicle 3 (orange) follows $S_3 \to 4 \to 5 \to T_3$, incurring a total control input cost (across all vehicles) of 188.4 (c.f. Figure 1b). In contrast, our bilevel method directs Vehicle 2 to visit $S_2 \to 5 \to 3 \to T_2$ (keeping the same routes for the others), resulting in a total control input cost of 147.3 (c.f. Figure 1a). This substantial improvement of total control cost is because our method accounts for vehicle dynamics and avoids visiting nearby nodes that require sharp turns, as in the baseline's trajectory for Vehicle 2.

Modeling formation flight via interaction term $I(\mathbf{x}^j, \mathbf{x}^{-j})$. In the following experiment with two vehicles, we consider a scenario with $N = 9$ nodes, $K = 5$ and $T = 7$. For the three-vehicle case, we set the number of waypoints to $K = 4$, while keeping all other parameters unchanged.

Figure 2 and Figure 3 present the impact of interaction costs in (6a) in two-vehicle and three-vehicle settings, respectively. In the absence of $I(\mathbf{x}^j, \mathbf{x}^{-j})$, each vehicle generates their trajectory to *only* minimize their own cost while visiting required waypoints. This implies that the game in (9) is essentially a set of decoupled optimization problems, for any fixed choice of routing variables $\mathbf{z}_u$. In contrast, we observe distinct alterations in the trajectories due to the interaction term. The circled region in Figure 2b highlights how the trajectories have changed substantially so that the vehicles can maintain formation. We also observe that the routing selections change as well. In Figure 2a, Vehicle 1 (red) chooses to visit nodes $\text{S1} \to 5 \to 4 \to 6 \to \text{T1}$ and Vehicle 2 (blue)'s route is $\text{S2} \to 7 \to 3 \to 4 \to \text{T2}$. However, accounting for the interaction term, their trajectories change in Figure 2b, i.e., Vehicle 1 selects nodes $\text{S1} \to 5 \to 3 \to 6 \to \text{T1}$ and Vehicle 2 visits nodes $\text{S2} \to 7 \to 1 \to 4 \to \text{T2}$. This behavior stems from inter-vehicle interactions: each vehicle adjusts its optimal trajectory based on the paths chosen by the others.

We observe similar trajectory adjustments in the three-vehicle formation flight scenario. Without interaction costs (Figure 3a), Vehicle 1 (red) follows the route $S_1 \to 3 \to 6 \to T_1$, Vehicle 2 (blue) follows $S_2 \to 5 \to 4 \to T_2$, and Vehicle 3 (orange) follows $S_3 \to 7 \to 6 \to T_3$. However, when accounting for inter-agent proximity (in the horizontal direction), the vehicles adjust their trajectories to maintain formation. In Figure 3b, Vehicle 1 remains in the middle and makes minimal modifications to its trajectory. On the other hand, Vehicles 2 and 3 alter their paths to ensure proper spacing from Vehicle 1. To this end, Vehicle 2 visits the nodes $S_2 \to 7 \to 3 \to T_2$ and Vehicle 3 follows the path $S_3 \to 5 \to 4 \to T_3$.

These routing changes occur because our formulation explicitly couples the choice of vehicles' waypoints at the high level with their low-level planning process. The examples above illustrate that, when vehicles' trajectory planning preferences change (e.g., to include a preference for flying in formation), the hierarchical router can *proactively* generate a route which accounts for that change. Furthermore, due to the convexity of the proposed formulation and by Proposition 2, the new route is globally optimal.

VI. Conclusion

In this letter, we present a bilevel game-theoretic framework for multi-agent hierarchical routing that integrates discrete route assignments with continuous trajectory optimization. We propose a convex reformulation of the resulting mixed-integer program, and demonstrate our results in two- and three-vehicle formation flight scenarios. Our proposed method is generalizable and can accommodate domain-specific constraints and objectives, such as routing and scheduling constraints in an emergency response scenario and time window constraints in last-mile delivery using heterogeneous fleets of ground robots and drones. Future work will focus on extending the proposed framework to real-world applications, particularly in next-generation air traffic management decision support tools. For example, the U.S. Federal Aviation Administration seeks to adopt a new automation system, *Flow Management Data and Services* (FMDS), which is a potentially multi-hundreds of millions USD investment [36]. FMDS seeks to integrate *data streams* such as real- and near-real time aircraft trajectory information with *services* such as air traffic demand-capacity balancing. The modeling results herein provide a foundation for future services that can run on FMDS, potentially impacting thousands of commercial flights serviced by FMDS.

References

[1] B. Zeng and Y. An. "Solving bilevel mixed integer program by reformulations and decomposition." 2014. [Online]. Available: https://optimization-online.org/wp-content/uploads/2014/07/4455.pdf

[2] T. Kleinert, M. Labbé, I. Ljubić, and M. Schmidt, "A survey on mixed-integer programming techniques in bilevel optimization," *EURO J. Comput. Optim.*, vol. 9, Jun. 2021, Art. no. 100007.

[3] O. Richetta et al., "Dynamic solution to the ground-holding problem in air traffic control," *Transp. Res. A Policy Pract.*, vol. 28, no. 3, pp. 167–185, 1994.

[4] P. B. Vranas et al., "The multi-airport ground-holding problem in air traffic control," *Oper. Res.*, vol. 42, no. 2, pp. 249–261, 1994.

[5] D. Bertsimas and S. S. Patterson, "The air traffic flow management problem with enroute capacities," *Oper. Res.*, vol. 46, no. 3, pp. 406–422, 1998.

[6] H. Wu, M. Z. Li, J. Henderson, E. M. Bongo, and L. A. Weitz, "Managing congestion in advanced air mobility operations using a bi-level optimization approach," in *Proc. AIAA SCITECH Forum*, 2025, pp. 11–15.

[7] A. G. Taye, R. Valenti, A. Rajhans, A. Mavrommati, P. J. Mosterman, and P. Wei, "Safe and scalable real-time trajectory planning framework for urban air mobility," *J. Aerosp. Inf. Syst.*, to be published.

[8] M. Zhu et al., "Game theoretic controller synthesis for multi-robot motion planning—Part I: Trajectory based algorithms," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2014, pp. 1645–1651.

[9] Z. Williams et al., "Distributed potential ILQR: Scalable game-theoretic trajectory planning for multi-agent interactions," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2023, pp. 1–7.

[10] 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 *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2020, pp. 1475–1481.

[11] B. Di and A. Lamperski, "Newton's method and differential dynamic programming for unconstrained nonlinear dynamic games," in *Proc. IEEE 58th Conf. Decis. Control (CDC)*, 2019, pp. 4073–4078.

[12] F. Laine et al., "The computation of approximate generalized feedback Nash equilibria," *SIAM J. Optim.*, vol. 33, no. 1, pp. 294–318, 2023.

[13] N. Mehr et al., "Maximum-entropy multi-agent dynamic games: Forward and inverse solutions," *IEEE Trans. Robot.*, vol. 39, no. 3,pp. 1801–1815, May 2023.

[14] E. L. Zhu and F. Borrelli, "A sequential quadratic programming approach to the solution of open-loop generalized Nash equilibria," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2023, pp. 3211–3217.

[15] J. F. Fisac et al., "Hierarchical game-theoretic planning for autonomous vehicles," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, 2019, pp. 9590–9596.

[16] K. Ji et al., "Hierarchical and game-theoretic decision-making for connected and automated vehicles in overtaking scenarios," *Transp. Res. C Emerg. Technol.*, vol. 150, May 2023, Art. no. 104109.

[17] Y. Marinakis, A. Migdalas, and P. M. Pardalos, "A new bilevel formulation for the vehicle routing problem and a solution method using a genetic algorithm," *J. Global Optim.*, vol. 38, pp. 555–580, Aug. 2007.

[18] W. Fan and R. B. Machemehl, "Bi-level optimization model for public transportation network redesign problem: Accounting for equity issues," *Transp. Res. Rec.*, vol. 2263, no. 1, pp. 1–11, 2011.

[19] L. Bianco et al., "A bilevel flow model for HAZMAT transportation network design," *Transp. Res. C Emerg. Technol.*, vol. 17, no. 2, pp. 175–196, 2009.

[20] M. Pavone et al., "Adaptive and distributed algorithms for vehicle routing in a stochastic and dynamic environment," *IEEE Trans. Autom. Control*, vol. 56, no. 6, pp. 1259–1274, Jun. 2011.

[21] W. Li et al., "A cooperative receding horizon controller for multivehicle uncertain environments," *IEEE Trans. Autom. Control*, vol. 51, no. 2, pp. 242–257, Feb. 2006.

[22] C. Yao et al., "Cooperative receding horizon control for multi-agent rendezvous problems in uncertain environments," in *Proc. 49th IEEE Conf. Decis. Control (CDC)*, 2010, pp. 4511–4516.

[23] A. B. Asghar et al., "Multi-robot routing for persistent monitoring with latency constraints," in *Proc. Amer. Control Conf. (ACC)*, 2019, pp. 2620–2625.

[24] F. Vehlhaber and M. Salazar, "Electric aircraft assignment, routing, and charge scheduling considering the availability of renewable energy," *IEEE Control Syst. Lett.*, vol. 7, pp. 3669–3674, 2023.

[25] F. Altché and A. de La Fortelle, "Partitioning of the free space-time for on-road navigation of autonomous ground vehicles," in *Proc. IEEE 56th Annu. Conf. Decis. Control (CDC)*, 2017, pp. 2126–2133.

[26] L. Fagiano, M. Tanaskovic, L. C. Mallitasig, A. Cataldo, and R. Scattolini, "Hierarchical routing control in discrete manufacturing plants via model predictive path allocation and greedy path following," in *Proc. 59th IEEE Conf. Decis. Control (CDC)*, 2020, p. 12.

[27] T. Başar and G. J. Olsder, *Dynamic Noncooperative Game Theory*, 2nd ed. London, U.K.: Academic, 1998.

[28] S. P. Dirkse and M. C. Ferris, "The path solver: A nommonotone stabilization scheme for mixed complementarity problems," *Optim. Methods Softw.*, vol. 5, no. 2, pp. 123–156, 1995.

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

[30] S. Dempe, V. Kalashnikov, G. A. Pérez-Valdés, and N. Kalashnykova, "Bilevel programming problems," in *Energy Systems*. Heidelberg, Germany: Springer, 2015.

[31] L. A. Wolsey, *Integer Programming*. Hoboken, NJ, USA: Wiley, 2020.

[32] S. Scholtes, "Convergence properties of a regularization scheme for mathematical programs with complementarity constraints," *SIAM J. Optim.*, vol. 11, no. 4, pp. 918–936, 2001.

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

[34] P. Belotti et al., "Mixed-integer nonlinear optimization," *Acta Numerica*, vol. 20, pp. 301–306, Apr. 2013.

[35] J. Kronqvist et al., "A review and comparison of solvers for convex MINLP," *Optim. Eng.*, vol. 20, pp. 397–455, Jun. 2019.

[36] *Flow Management Data and Services (FMDS)*, Fed. Aviation Admin., Washington, DC, USA, 2025.