A Game Theoretic Method for Two-Team Multi-Player Autonomous Racing

Zhenghao Hu, Xiuxian Li, *Senior Member, IEEE,*, Min Meng, and Shiyu Zhao

***Abstract*—This paper explores an autonomous driving competition between two teams, where the number of members in one team is greater than or equal to the other team's, but their maximum speed is lower. The paper proposes a hierarchical decision-making approach to address how the slower team can compete against their opponents through collaboration. Initially, in the first layer, the teams are paired using weighted bipartite graph matching, followed by the task reassignment to address opponents who pose a significant threat. In the second layer, each player computes its optimal path through the matching-based iterated best response, taking into account the opponents determined by the first-layer matching. Through this hierarchical decision-making module, each player assumes specific roles and tasks, enabling cooperative blocking, aiding lagging teammates to catch up or contributing to the team's leading member to amplify their advantage. This method aims to increase the team's chances of winning competitions at a higher rate.**

***Index Terms*—Autonomous racing, team competition, game theory, best response, weighted bipartite graph.**

I. Introduction

In recent years, there has been rapid advancement in autonomous driving. After the DARPA Urban Challenge [1], research has primarily focused on urban driving [2]. Urban driving scenarios necessitate vehicles to adhere to traffic rules, interact with each other, and make real-time decisions based on surrounding vehicles' actions [3]. Additionally, autonomous racing has garnered increased attention, leading to the emergence of official self-driving competitions such as Roborace [4], Indy Autonomous [5], Formula Student Driverless (The Formula Student cars are widely employed in research concerning autonomous driving planning and control [6]–[8]), and the VDI Autonomous Driving Challenge. Compared to urban driving, the regulations in autonomous racing are less complex, devoid of traffic lights or lane-changing requirements. Competitions are generally classified into two categories: 1) Single player races, where the winner is determined based on the time taken to complete the track. 2) Head-to-head competitions, where the player reaching the finishing line first is declared the winner.

Manuscript received: February, 3, 2024; Revised: May, 12, 2024; Accepted: July, 1, 2024.

This paper was recommended for publication by Editor T. Asfour and Editor M. A. Hsieh upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the National Key Research and Development Program of China No. 2022ZD0119702, the National Natural Science Foundation of China under Grant 62103305, 62088101, U23B2059, and the Shanghai Municipal Science and Technology Major Project, No. 2021SHZDZX0100.

[1]The authors are with the Department of Control Science and Engineering, College of Electronics and Information Engineering, Tongji University, Shanghai 201800, China (e-mail: zhenghao_hu@tongji.edu.cn, xxli@ieee.org, mengmin@tongji.edu.cn).

[2]X. Li and M. Meng are also with Shanghai Research Institute for Intelligent Autonomous Systems, and Shanghai Institute of Intelligent Science and Technology, Tongji University, Shanghai, China.

[3]S. Zhao is with the School of Engineering at Westlake University and the Institute of Advanced Technology at Westlake Institute for Advanced Study, Hangzhou, China. (e-mail: zhaoshiyu@westlake.edu.cn).

Digital Object Identifier (DOI): see top of this page.

A. Related Work

For the first type, vehicles are expected to fully utilize their capabilities to complete the racetrack. Many researchers regard this as an optimal control problem to determine the global optimal trajectory that meets both dynamic properties and track boundary constraints. The trajectory can be optimized for minimum lap time, as demonstrated in [9], which showcases a non-linear model-predictive framework, crafting an optimal control problem with time as the principal goal, and [10] by computing the minimum lap time trajectory with endpoint constraints offline and tracking it through NMPC. Alternatively, Heilmeier et al. [11] solve a quadratic optimization problem involving constraints on vehicle dynamic limits. This approach aims to determine a path with minimal curvature around the racetrack, reducing the lateral acceleration and tire force. Additionally, energy consumption is considered, particularly in electric car or drone racing. Hermann et al. [12] take lap time and energy consumption into consideration to solve an optimal control problem. Vehicles can plan a local trajectory within a fixed planning horizon based on the reference line, avoiding collisions with obstacles or adversaries in the environment. Liniger et al. [13] develop the MPCC approach, which simplifies the calculation of projection while integrating track maintenance and collision avoidance constraints via dynamic programming. Their work demonstrates the viability of these approaches using a 1:43-scale car. With the development of deep reinforcement learning, some end-to-end methods such as DNNs [14], Deep Deterministic Policy Gradient [15], Soft Actor-Critic [16] and Model-based or Model-free RL [17] are applied to autonomous driving.

In head-to-head racing, merely driving at maximum speed might not be optimal due to variations in vehicle performance. Considering interactions with other agents creates a more complex scenario where each player needs to respond optimally to the strategies employed by others. Hence, applying game theory becomes reasonable to deal with the multi-agent environment. A sensitivity enhanced iterative best response approach (SE-IBR) designed to approximate the Nash equilibrium between the ego player and its opponents is developed in [18]–[21] both in multi-player drone racing and vehicle racing. To address the challenge of poor performance resulting from inaccuracies in opponent models or states affecting the SE-IBR algorithm, Notomista et al. [22] combine the algorithm with a control barrier function to adapt situations with incomplete information. In [23], Liniger et al. introduce an alternative method that utilizes the Stackelberg pattern to show blocking maneuvers in a receding horizon fashion. But the bi-matrices method applied is unsuitable as the number of players increases, because the time complexity of searching for optimal trajectories grows exponentially. Zheng et al. [24] adopt population-based optimization combined with counterfactual regret minimization to compute the optimal behaviors of vehicles across various scenarios. Jung et al. [25] integrate the Stackelberg game with the MPCC method. This approach allows the ego car to determine whether to maintain its velocity or overtake opponents but neglects blocking actions. In [26], a Gaussian process is applied to understand the leading vehicle's behavior. Subsequently, the car behind utilizes a stochastic MPC to plan optimistic trajectories for potential overtaking maneuvers, guided by the Gaussian process outputs. He et al. [27] propose a method that enables the ego car to switch between time-optimization and overtaking modes under safety considerations, based on the specific scenario.

In the team-based scenario, [28] reviews methods for team members to collaborate in completing certain tasks. For racing scenario, each team member has designated roles and responsibilities, while simultaneously executing specific tasks corresponding to opponents. This setting not only involves competition but also cooperation. Di et al. [29] utilize SE-IBR algorithm with another wingman preventing the leader from being overtaken by the opponent. Cui et al. [30] consider a many-to-one competition with three combination strategies for the team to collaboratively block the faster opponent. Li et al. [31] apply the SE-IBR algorithm to a scenario with obstacles, taking into account both teammates and opponents. Thakkar et al. [32] study a team competition with limited lane-changing maneuvers, discretize the racetrack and propose a hierarchical control approach to obtain the optimal waypoints for players.

This paper investigates a multi-player competition between two teams, with one team having a lower speed than the other while having an equal or greater number of members. Each member considers predetermined opponents and undertakes specific tasks, aiming to achieve teamwork in the competition.

B. Our Contributions

1) **Racing scale:** Compared to the many-to-one racing in [29], [30], the two-team multi-player competition is more complex and lack of literature currently. We design a matching-based hierarchical framework to obtain the optimal trajectories for each player, allowing the slower team to achieve a high winning rate.

2) **Task assignment:** We design a task assignment algorithm based on a weight function, especially considering the leading position, and pay more attention to the opponents with greater threats. It can alleviate the burden on players before planning trajectories, as it is challenging for a slower player to compete with multiple opponents simultaneously shown in [18]–[20]. The matching-based algorithm is a novel introduction in autonomous racing, as previous research solely incorporates the influences of teammates into the objective function, as demonstrated in [31], [32].

3) **Role transition and teamwork:** Based on the task assignment algorithm, players can switch roles and perform different tasks in the dynamic environment. The planning module considers the best responses of teammates and opponents, thus players can engage in actions such as assisting lagging teammates and cooperating to block opponents.

II. Preliminaries

A. Autonomous Racing Scenario

The research studies two-team multi-vehicle racing, defining team 1 as $\mathcal{T}_1 = \{A_1, A_2, ..., A_m\}$ and team 2 as $\mathcal{T}_2 = \{B_1, B_2, ..., B_n\}$, where A_i and B_j represent the indices of team members. In the competition, it is assumed that members in the same team possess identical maximum speeds. However, the maximum speed of members in team 1 is lower than that of team 2. The purpose of the study is to enable team 1 to increase the probability of any of its members being the first to cross the finishing line through implementing task allocation, cooperative blocking and other strategies.

As shown in Fig. 1, player i located at $\boldsymbol{p}_i = [x, y]^T \in \mathbb{R}^2$ on the track has a projection point $\boldsymbol{\tau}_i$ on the reference line, from which the tangent vector $\boldsymbol{t}_i$ and normal vector $\boldsymbol{n}_i$ of the reference line at that point can be obtained. The reference line which is a curve parameterized by the arc-length is known to the player as

$$\boldsymbol{\tau} : [0, l] \mapsto \mathbb{R}^2,$$

where l represents the total length of the center line. This means that the projection point $\boldsymbol{\tau}_i$ on the reference line can be obtained by the longitudinal position of player i (arc-length from the origin to the projection point).

To ensure that vehicles remain within the boundaries of the track, they need to satisfy the track boundary constraints, which means their lateral distances are less than the half-width of the track,

$$\left|\boldsymbol{n}(s_i)^T \left[\boldsymbol{p}_i - \boldsymbol{\tau}(s_i)\right]\right| \leq w_\tau, \tag{1}$$

where $s_i \in [0, l]$ is the longitudinal position of the player, and the arc-length of the point on the reference line closest to $\boldsymbol{p}_i$,

$$s_i(\boldsymbol{p}_i) = \arg\min_s \frac{1}{2}\left\|\boldsymbol{\tau}(s) - \boldsymbol{p}_i\right\|^2, \tag{2}$$

and s_i also represents the progress in the competition. It is a crucial metric for the player to optimize its trajectories along the race track.

Meanwhile, during the competition, players should pay attention to the collision avoidance to ensure safety, so all pairs of players should obey the distance constraint:

$$\left\|\boldsymbol{p}_i - \boldsymbol{p}_j\right\| \geq \overline{d}, \quad \forall i, j \in T_1 \cup T_2, \tag{3}$$

where $\overline{d}$ is the minimum distance between two cars.

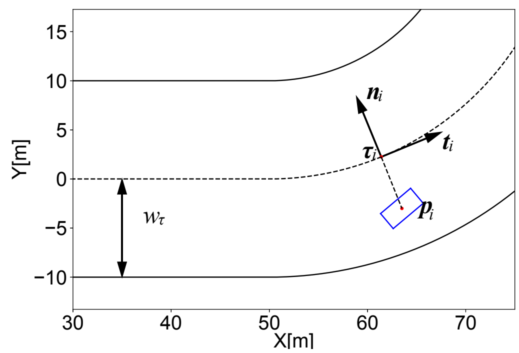

Fig. 1. The track with half-width w_τ for the competition. The reference line is the center line (dashed line). The blue rectangle is player i, and $\boldsymbol{p}_i$ represents the position of the car. $\boldsymbol{t}_i$ and $\boldsymbol{n}_i$ respectively represent the local tangent and normal vectors to the track in the projection $\boldsymbol{\tau}_i$ of $\boldsymbol{p}_i$.

B. Polynomial Trajectory Generation

We use the bicycle model and piecewise time polynomial trajectories applied in [18] to calculate the trajectories of players in the planning horizon. Let N be the number of waypoints whose positions should be planned by players, with a constant time interval of Δt between any two adjacent waypoints. The $(k+1)$-th polynomial between two waypoints at time $t_k = k\Delta t$ can be described as

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \boldsymbol{p}^k + \begin{bmatrix} \alpha_{k,0} \\ \beta_{k,0} \end{bmatrix} t + \begin{bmatrix} \alpha_{k,1} \\ \beta_{k,1} \end{bmatrix} t^2, \quad t \in [0, \Delta t], \tag{4}$$

where $k = 0, 1, ..., N-1$. (4) assumes that within Δt, the vehicle is uniformly accelerating in both the x and y directions with respect to time t. The coefficients $\alpha_{k,0}$ and $\beta_{k,0}$ are related to velocities, while $\alpha_{k,1}$ and $\beta_{k,1}$ are associated with accelerations. Including the initial states $\boldsymbol{p}^0$ and $\boldsymbol{u}^0$, the trajectory totally has $N+1$ waypoints

$$\begin{aligned} \boldsymbol{p}^k &= \boldsymbol{p}^{k-1} + \begin{bmatrix} \alpha_{k,0} \\ \beta_{k,0} \end{bmatrix} \Delta t + \begin{bmatrix} \alpha_{k,1} \\ \beta_{k,1} \end{bmatrix} \Delta t^2, \\ \boldsymbol{u}^k &= \dot{\boldsymbol{p}}^k, \end{aligned} \tag{5}$$

where $k = 1, 2, ..., N$. In addition to the continuity constraints (5) above, these discrete waypoints should also satisfy several other constraints at each step k (For simplicity, the subscript k in $\alpha_{k,0/1}$ and $\beta_{k,0/1}$ is omitted below).

1) Maximum speed constraints: The speed of the car is

$$v^2(t) = \|\dot{\boldsymbol{p}}\|^2 = \dot{x}^2(t) + \dot{y}^2(t). \tag{6}$$

Since the vehicles accelerate or decelerate on each trajectory segment, the maximum speed of the vehicle is obtained at the end points, which means that

$$\left\|\dot{\boldsymbol{p}}^k\right\| = \left\|\boldsymbol{u}^k\right\| \le \bar{v}, \tag{7}$$

where $\bar{v}$ is the maximum speed of the player.

2) Acceleration constraints: From the bicycle model, the acceleration is

$$\begin{aligned} a^2(t) &= \ddot{x}^2(t) + \ddot{y}^2(t) - v^2(t)\dot{\theta}^2(t) \\ &\le \ddot{x}^2(t) + \ddot{y}^2(t) = 4(\alpha_1^2 + \beta_1^2) \le \bar{a}^2, \end{aligned} \tag{8}$$

where θ is the heading of the car, α_1 and β_1 are the coefficients in (5), $\bar{a}$ is the maximum acceleration of the vehicle, and the constraints are relaxed.

3) Curvature constraints: The curvature of the planned trajectory should satisfy the vehicle's steering angle with

$$\begin{aligned} \kappa(t) &= \frac{\dot{x}(t)\ddot{y}(t) - \dot{y}(t)\ddot{x}(t)}{(\dot{x}^2(t) + \dot{y}^2(t))^{3/2}} \\ &= \frac{2(\alpha_0\beta_1 - \alpha_1\beta_0)}{[\alpha_0^2 + \beta_0^2 + 4(\alpha_0\alpha_1 + \beta_0\beta_1)t + 4(\alpha_1^2 + \beta_1^2)t^2]^{3/2}}. \end{aligned} \tag{9}$$

Noting that in (9), the denominator contains a quadratic polynomial $h(t) = \alpha_0^2+\beta_0^2+4(\alpha_0\alpha_1+\beta_0\beta_1)t+4(\alpha_1^2+\beta_1^2)t^2$, this constraint can be simplified to

$$\kappa_{\max} = \frac{2(\alpha_0\beta_1 - \alpha_1\beta_0)}{\min_{t\in[0,\Delta t]}[h(t)]^{3/2}} \le \bar{\kappa}, \tag{10}$$

where κ_{max} is the maximum curvature of the trajectory, and $\bar{\kappa}$ is the maximum feasible curvature.

C. Multi-Player Autonomous Racing

By discretizing the planning horizon, a sequence $\boldsymbol{\zeta}_i = (\boldsymbol{p}_i^1, ..., \boldsymbol{p}_i^N, \boldsymbol{u}_i^1, ..., \boldsymbol{u}_i^N)$ including positions and control inputs can be defined. In the multi-player game without teammates, assuming there are M players, each vehicle strives to maximize its lead over the opponents in the following form:

$$\max_{\zeta_i} \; s_i(\zeta_i) - \frac{1}{M-1} \sum_{j=1, j\neq i}^{M} s_j(\zeta_j) \tag{11}$$

$$s.t. \quad \boldsymbol{f}_i(\zeta_i) = 0 \tag{11a}$$

$$\boldsymbol{g}_i(\zeta_i) \le 0 \tag{11b}$$

$$\chi_i(\zeta_i, \zeta_j) \le 0, \tag{11c}$$

where

1) $\boldsymbol{f}_i$ represents the continuity constraints (5) for player i;
2) $\boldsymbol{g}_i$ represents the constraints (1), (7), (8), and (10) for player i;
3) χ_i represents the collision avoidance constraints (3).

From the perspective of game theory, considering the collision avoidance constraints between players, the strategies of both sides affect each other's actions (such as steering to avoid collisions). Taking Nash equilibrium into account that the opponents will respond optimally to the strategy of the ego player, according to the sensitivity analysis in [19], the objective function can therefore be replaced by

$$\max_{\zeta_i} s_i(\zeta_i) - \sum_{j=1, j\neq i}^{M} \alpha_{ij} s_j^*(\zeta_i), \tag{12}$$

where $s_j^*(\zeta_i)$ defines player j's best response to player i's strategy ζ_i. α_{ij} is a sensitivity parameter designed by player i with respect to player j. To obtain a closed-form expression for it, a linear approximation, derived through sensitivity analysis, is based on the current estimation of the Nash equilibrium strategy profile. Specifically, assume that in the l-th iteration, a fixed guess $(\zeta_1^{l-1}, \ldots, \zeta_M^{l-1})$ for all players has already been updated. Afterward, each player follows (12) to sequentially predict the strategies of all players. During the update of player i's strategy, the strategies of other players are considered fixed, taking on values ζ_j^l, $j < i$, as player j has updated its strategy before player i, or ζ_j^{l-1}, $j > i$. Each player updates its strategy in this manner, predicting the strategies of opponents. Since ζ_i^{l-1} is available for player j when player i solves an optimization problem with respect to player j, its optimal payoff $s_j^*(\zeta_i)$ in the vicinity of ζ_i^{l-1} can be obtained using a first-order Taylor approximation

$$s_j^*(\zeta_i) \approx s_j^*(\zeta_i^{l-1}) + \frac{ds_j^*}{d\zeta_i}\bigg|_{\zeta_i=\zeta_i^{l-1}} (\zeta_i - \zeta_i^{l-1}). \tag{13}$$

From Lemma 1 in [18], the derivation can be substituted as

$$\frac{ds_j^*}{d\zeta_i}\bigg|_{\zeta_i=\zeta_i^{l-1}} = -\boldsymbol{\mu}_{ji}^l \frac{\partial \chi_j}{\partial \zeta_i}\bigg|_{(\zeta_i^{l-1},\zeta_j^l)}, \tag{14}$$

where $\boldsymbol{\mu}_{ji}^l$ is the row vector of Lagrange multipliers associated to constraints (11c) at the l-th iteration. By disregarding the constants unrelated to ζ_i, (12) leads to

$$\max_{\zeta_i} s_i(\zeta_i) + \sum_{j=1, j\neq i}^{M} \sum_{k=1}^{N} \alpha_{ij} \mu_{ji}^{k,l} (\boldsymbol{\sigma}_{ij}^{k,l})^T \boldsymbol{p}_i^k, \tag{15}$$

where $\mu_{ji}^{k,l}$ is the k-th element of $\boldsymbol{\mu}_{ji}^l$, and

$$\boldsymbol{\sigma}_{ij}^{k,l} = \frac{\boldsymbol{p}_j^{k,l} - \boldsymbol{p}_i^{k,l-1}}{\left\| \boldsymbol{p}_j^{k,l} - \boldsymbol{p}_i^{k,l-1} \right\|}.$$

Players can calculate their optimal strategies by (15). Based on the aforementioned theory, the following sections will give the details to solve two-team multi-player problems.

III. Design of the Planning Module

This research focuses on the competition among multiple players from two teams. Given that team 1 with slower speeds has a disadvantage, it is assumed that the number of players satisfy $|\mathcal{T}_1| \geq |\mathcal{T}_2|$. Each player A_i in team 1 needs to be assigned a specific role, categorized into two types:

1) Advancers, whose strategy involves advancing as fast as possible along the track;
2) Defenders, who are allocated their respective opponents and obstructing the opponents they are assigned to.

By cooperatively blocking opponents, some lagging players within the team can seize the opportunity to catch up, or others who are leading can further expand their advantage. The framework is demonstrated in Fig. 2.

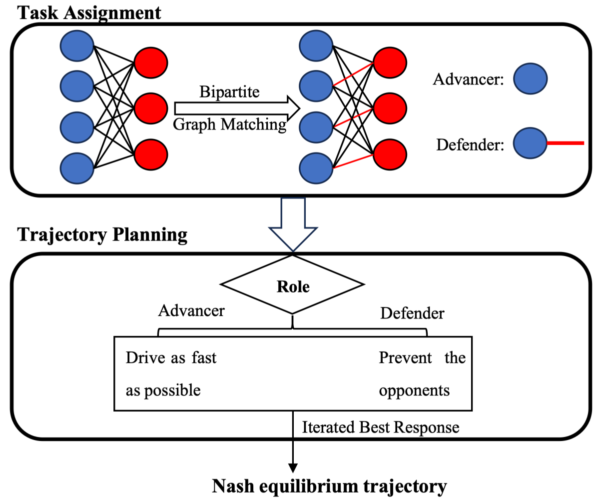

Fig. 2. The decision-planning framework for the slower team.

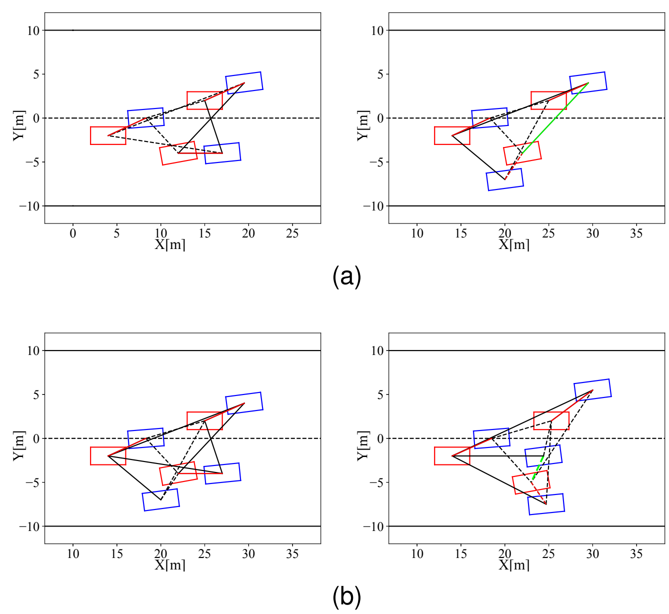

Fig. 3. (a) $m = n$ scenarios. (b) $m > n$ scenarios. Blue and red nodes represent team 1 and team 2 members, respectively. Solid lines between A_i and B_j denote opponents considered by A_i, while dashed lines signify those beyond A_i's range. Red lines display matched pairs from bipartite graph matching, while green lines show added pairs post-reassignment. Players move from left to right.

A. Minimum Weight Bipartite Graph Matching

To assign roles and tasks to each player, a matching-based approach is introduced. Each vehicle is considered as a node, and its neighbors consist of the opponents (Fig. 3). The weight of each edge is determined by the lateral and longitudinal distances between two players. Therefore, the weighted bipartite graph matching method, also known as the Kuhn-Munkres (KM) algorithm [33], is applicable in this scenario.

The weighted bipartite graph is defined as $\mathcal{G} = (\mathcal{T}_1 \bigcup \mathcal{T}_2, \mathcal{E})$, the weight of the edge $e_{A_iB_j}$ is defined as

$$e_{A_iB_j} = \begin{cases} w_d \left|d_{A_i} - d_{B_j}\right| + w_s(s_{A_i} - s_{B_j}) & \text{if } \frac{l_c}{2} \leq s_{A_i} - s_{B_j} \leq \bar{l} \\ & \text{and } \left|d_{A_i} - d_{B_j}\right| \leq w_\tau, \\ \varepsilon + w_d \left|d_{A_i} - d_{B_j}\right| - w_s(s_{A_i} - s_{B_j}) & \text{else,} \end{cases} \tag{16}$$

where w_d and w_s are weights, $d.$ represents the lateral distance, l_c is the length of the car and $\bar{l} = N\Delta t(\bar{v}_{A_i} + \bar{v}_{B_i})$ is the maximum allowed longitudinal distance to the opponent. Player A_i considers opponents behind itself in a limited distance and with a lateral distance not exceeding the track's radius. Otherwise, the edge weight will receive $\varepsilon \gg 0$. Then the task allocation objective is

$$\begin{aligned}\min \sum_{i=1}^{m}\sum_{j=1}^{n} & e_{A_iB_j}\cdot x_{A_iB_j}\\ s.t.\quad & x_{A_iB_j}\in\{0,1\},\\ & \sum_{j=1}^{n} x_{A_iB_j}\le 1,\\ & \sum_{i=1}^{m} x_{A_iB_j}=1.\end{aligned}\tag{17}$$

From (17), team 1 will be paired based on the lateral and longitudinal distances between members and opponents, where $x_{A_iB_j} = 1$ indicates that A_i is matched with B_j. All the opponents will match a distinct member A_i after (17). The opponent assigned to vehicle A_i is defined as $\mathcal{P}_{A_i}$. Situations similar to the right of Fig. 3a and Fig. 3b might exist, where a member, due to lagging or being overtaken, poses slight threat to the faster opponent. In such cases, redistribution of these opponents is necessary. The opponents in need of reallocation form the set $\mathcal{U}$,

$$\mathcal{U}=\left\{B_j|B_j\in\mathcal{P}_{A_i}\cap\left(s_{A_i}-s_{B_j}<\frac{l_c}{2}\cup\left|d_{A_i}-d_{B_j}\right|>w_\tau\right)\right\},$$

which includes those about to overtake or have already overtaken, as well as opponents with significant lateral differences. The task reassignment method is designed for $B_j \in \mathcal{U}$ who poses a considerable threat. The whole task assignment algorithm is demonstrated in Algorithm 1. A_i who has no matched opponent is regarded as an advancer. In theory, due to the speed disadvantage, it's more ideal for one or multiple members to obstruct a single opponent (Fig. 3a), whereas a single member trying to block multiple opponents often leads to failure (Fig. 3b).

Algorithm 1: Task Assignment Algorithm

1 **Initialization:**
2 Obtain initial matching result: $\mathcal{P}_{A_i} = \{B_j\}$ from (17);
3 Obtain unmatched set $\mathcal{U}$;
4 Sort $\mathcal{U}$ and $\mathcal{T}_1$ by longitudinal progress descendingly;
5 **for** *any* $B_j \in \mathcal{U}$ **do**
6 **for** $A_i \in \mathcal{T}_1$ **do**
7 **if** $s_{A_i} > s_{B_j}$ *and* $\mathcal{P}_{A_i} = \emptyset$ **then**
8 add B_j to $\mathcal{P}_{A_i}$;
9 **end**
10 **end**
11 **if** *no A_i satisfies Line 7* **then**
12 add B_j to $\mathcal{P}_{A_i^*}$ where $A_i^* = \arg\min_{A_i} \frac{1}{|\mathcal{P}_{A_i}|}\sum_{B_k\in\mathcal{P}_{A_i}}\left|s_{B_j}-s_{B_k}\right|+\left|d_{B_j}-d_{B_k}\right|$ $s.t.\ s_{A_i} > s_{B_j}$
13 **end**
14 **end**

In Algorithm 1, Lines 7-9 lead a many-to-one result and Line 12 resembles a clustering method to identify a member whose opponents have the closest Manhattan distance to the opponent being reassigned. This enables the member to have capability to block multiple opponents simultaneously.

Remark 1: Algorithm 1 determines the roles of the players before they calculate their trajectories.

1) Algorithm 1 uses the minimum weight matching (MWM) combined with (16) to minimize the overall cost of the team when the game is divided into one-to-one blocking.
2) The algorithm reallocates the members to the opponents with greater threats, using a greedy matching approach in Lines 6-10 and a clustering method in Lines 11-13.

Remark 2: In contrast to previous related works, Algorithm 1 exhibits several advantages:

1) In many-to-many scenarios, based on the weight function, Algorithm 1 demonstrates a more reasonable matching approach compared to the simple distance-based allocation method in [30].
2) In the dynamic environment, Algorithm 1 can switch players' roles, unlike the method in [34], where opponents are predetermined at the beginning of the game.
3) Compared to the maximum cardinality matching (MCM) in [35]–[37] for 3-D reach-avoid games, Algorithm 1 considers the global prevention cost along with the leading position, a crucial metric in autonomous racing.

The time complexity of MCM based on maximum network flow [38] and applied in [35] is $O(|\mathcal{T}_1|\,(|\mathcal{E}|+2\,|\mathcal{T}_1|))$, where $|\mathcal{E}|$ is the number of edges in $\mathcal{G}$. In contrast, Algorithm 1 here has the time complexity $O(|\mathcal{T}_1|^3)$, which is more favorable for large-scale scenarios, i.e., $|\mathcal{E}| > |\mathcal{T}_1|^2 - 2|\mathcal{T}_1|$.

B. Strategies for Two-Team Multi-Player Competition

If a competition is considered between two teams, the objective of one team can be described as follows:

$$\max_{\{\zeta_{A_i}\}_{A_i\in\mathcal{T}_1}} \frac{1}{m}\sum_{i=1}^{m} s_{A_i}(\boldsymbol{p}_{A_i}^N)-\frac{1}{n}\sum_{j=1}^{n} s_{B_j}(\boldsymbol{p}_{B_j}^N),\tag{18}$$

where m is the number of members in $\mathcal{T}_1$, and n is the number of members in $\mathcal{T}_2$. The team aims to maximize the difference between the average progress of all members and that of the opponents. After the opponents are matched by Algorithm 1, the competition will be split into multiple one-to-one races, with a small number involving one-to-many or many-to-one scenarios. Based on the multi-player competition algorithm (Sec. II-C), the objective function of team 1 in the l-th iteration is modified as follows, considering the opponents each member matches.

$$\max_{\{\zeta_{A_i}\}_{A_i}\in\mathcal{T}_1}\sum_{i=1}^{m}\left[s_{A_i}(\boldsymbol{p}_{A_i}^N)+\sum_{B_j\in\mathcal{P}_{A_i}}\sum_{k=1}^{N}\alpha_{A_iB_j}\mu_{B_jA_i}^{k,l}(\boldsymbol{\sigma}_{A_iB_j}^{k,l})^T\boldsymbol{p}_{A_i}^k\right],\tag{19}$$

where s_{A_i} is given in (2). The objective function means that the team aims to maximize the progress difference between each of its members and the opponents they consider. Players can set $\boldsymbol{\mu}_{B_jA_i} > \mathbf{0}$ to activate collision avoidance constraints, allowing them to make more aggressive decisions. And if B_j is far away from A_i, then one can set $\boldsymbol{\mu}_{B_jA_i} = \mathbf{0}$ to neglect the opponent. The selection of $\boldsymbol{\mu}_{B_jA_i}$ can be achieved by setting distance threshold and reselecting adjacent opponents as

$$\mathcal{N}_{A_i}=\left\{B_j\in\mathcal{P}_{A_i}\left|s_{A_i}-s_{B_j}\le\bar{l}\right.\right\}.\tag{20}$$

The matching-based iterated best response algorithm is summarized in Algorithm 2. The Line 10 means that the faster team just considers the progress and drive as fast as possible along the racetrack to do overtaking actions. Because once they successfully achieve overtaking, it becomes difficult for slower vehicles to re-overtake. Therefore, there is no need for blocking behaviors. The algorithm can be implemented using parallel programming [20] to speed up. In this manner, team 1 and team 2 update their $(l+1)$-th strategies simultaneously with only l-th strategies.

Algorithm 2: Matching-Based Iterated Best Response

1 Set the maximum number of iterations L;
2 Initialize all vehicles' strategies $\boldsymbol{\zeta}^0$;
3 Use Algorithm 1 to perform matching;
4 Use (20) to reselect opponents;
5 **for** $l = 0, 1, ..., L-1$ **do**
6   **for** A_i *in* $\mathcal{T}_1$ **do**
7   Solve the objective (19) for A_i with $\{\boldsymbol{\zeta}_{A_1}^{l+1}, ..., \boldsymbol{\zeta}_{A_i}^{l}, ..., \boldsymbol{\zeta}_{A_m}^{l}, \boldsymbol{\zeta}_{B_1}^{l}, ..., \boldsymbol{\zeta}_{B_n}^{l}\}$, obtain the optimal strategy $\boldsymbol{\zeta}_{A_i}^{l+1}$;
8   **end**
9   **for** B_j *in* $\mathcal{T}_2$ **do**
10   Solve the objective (19) with $\boldsymbol{\mu}_{A_iB_j} = \mathbf{0}$ for B_j with $\{\boldsymbol{\zeta}_{A_1}^{l+1}, ..., \boldsymbol{\zeta}_{A_m}^{l+1}, \boldsymbol{\zeta}_{B_1}^{l+1}, ..., \boldsymbol{\zeta}_{B_j}^{l}, ..., \boldsymbol{\zeta}_{B_n}^{l}\}$, obtain the optimal strategy $\boldsymbol{\zeta}_{B_j}^{l+1}$;
11   **end**
12 **end**
13 $\mathcal{T}_1$ updates strategies once again;
14 Obtain the optimal strategies $\boldsymbol{\zeta}_{A_i}$ and $\boldsymbol{\zeta}_{B_j}$.

Remark 3: Compared to previous autonomous racing research, Algorithm 2 has the following advantages:

1) Compared to [18]–[20], [31], [32], due to Lines 3 and 4, players only focus on their matched opponents within a certain range, reducing ineffective blocking actions.
2) In [30], a GNG framework is designed with an internal game among team members, doubling the iteration time compared to Algorithm 2. And Algorithm 2, in contrast, directly incorporates teammates' strategies in (19), avoiding suppression among teammates.
3) Algorithm 2, in contrast to [13] and [39] which obtain Stackelberg equilibria from payoff matrices, addresses more fair scenarios in multi-player racing, because, from a game perspective, all players are considered identical in this paper.
4) Compared to the approach in [25], where blocking actions are not considered, Algorithm 2 enables overtaking and blocking, with the team considering all opponents.

IV. Simulation Results

A. Simulation Setup

Consider a two-team multi-vehicle racing, where the maximum speed of one teams is set to be less than the other, and the team with the slower speed gains an initial longitudinal distance lead at the beginning. The time step is chosen $\Delta t = 0.5$s and the total planning horizon is 4s. Rather than employing vision-based tracking and estimation methods in the simulation, the interaction behaviors between the agents are tested only under the assumption that the agents can acquire the positions of their opponents and teammates. After obtaining the optimal trajectory of the vehicle from (19), the vehicle proceeds to track the first path point.

To ensure initial interaction between the vehicles of both teams, the simulation guarantees that the starting position of the faster vehicle is positioned behind the slower vehicle. The initial position of the faster vehicles are sampled from a uniform distribution within the rectangle $[-1, 1] \times [-8, 8]$m. Similarly, the starting positions of the slower vehicles are uniformly sampled from a rectangle bounded by $[4, 6] \times [-8, 8]$m. The vehicles are modeled as rectangles with a length of 4m and a width of 2m. Collision detection is implemented during the initial position setting, and any positions violating collision avoidance constraints are resampled. The winning criterion is that any member of any team reaching the finish line first. It is considered a success if a member A_i is the first to reach the finish line. Due to the existence of collision avoidance constraints, the results show that the probability of collision during the race is relatively low in the simulation. However, the optimization problem may encounter the situation where no solution can be found, according to the basic work [21], the hard constraints are converted into soft constraints and the problem is re-solved. If the objective function still has no solution, vehicles will continue to follow the last planned trajectories.

B. Simulation Results

Algorithm 2 is evaluated on the circular track with a total length of approximately 388m. The race scenarios considered involve two-versus-two, three-versus-two, and three-versus-three competitions. First of all, the algorithm is tested without the matching Algorithm 1, and the results show that the slower team loses all the competitions because they lacked a proper strategy for collaboration. The members continue to compete individually rather than adopting a cooperative approach, rendering their blocking efforts futile when considering the situations involving all opponents. Fig. 4 depicts the positions of vehicles at different moments during a competition without using the task assignment algorithm. It can be observed that, in this scenario, the strategy choices of slower vehicles appear chaotic. The slower vehicles may tend to block farther opponents, leading to blocking failures and loss of their original leading position. The results indicate that blocking multiple opponents simultaneously is not a viable strategy for a slower vehicle.

The results successfully employing Algorithm 2 based on Algorithm 1 are shown in Table I. As the speed difference narrows, the winning rate of the slower team increases, especially when the teams have an equal number of members. Additionally, the slower team can enhance its winning rate by leveraging the advantage of member quantity and employing various cooperative strategies. The velocity curves of players in both x and y directions (v_x and v_y) are shown in Fig. 5. It

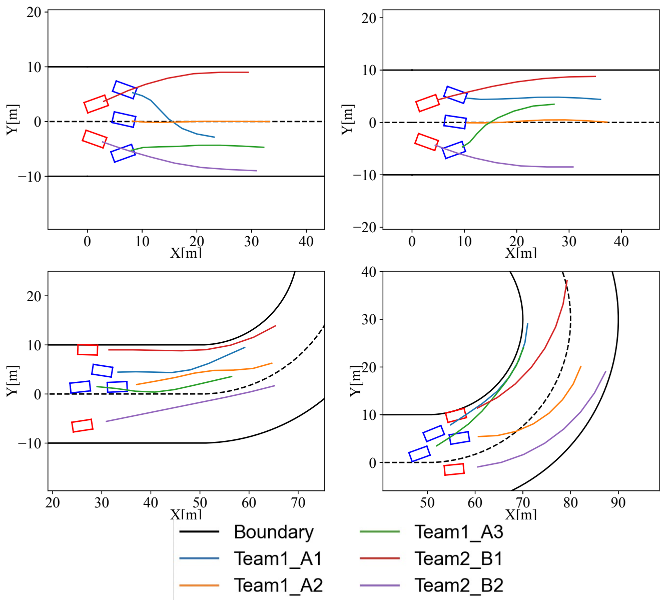

Fig. 4. Snapshots in the 8m/s vs. 10m/s scenario without Algorithm 1.

can be observed that faster vehicles attempt to overtake with higher $|v_y|$ on the straight, which means the defenders can force the opponents to change direction to overtake, thereby preventing them from reaching their maximum speed.

TABLE I
WINNING RATE IN DIFFERENT COMPETITIONS

Racing scenarios	7m/s vs. 10m/s winning times	8m/s vs. 10m/s winning times	9m/s vs. 10m/s winning times
2 vs. 2	21/129	113/37	147/3
3 vs. 2	65/85	126/24	150/0
3 vs. 3	2/148	97/53	110/40

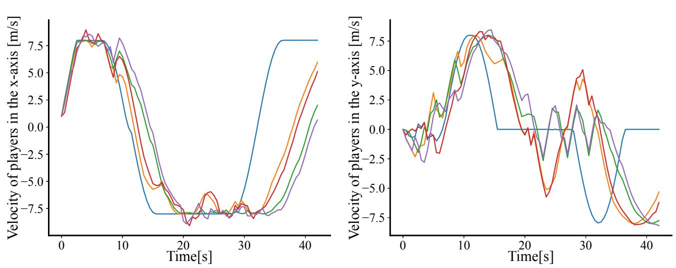

Fig. 5. Plots of velocity in x-axis and y-axis versus time for each vehicle in three-versus-two scenario (8m/s vs. 10m/s). The legends are same with Fig. 4.

In situations where the disadvantaged team has a greater number of members, the slower players in team 1 exhibit diverse driving behaviors. As shown in Fig. 6a, it depicts the scenario where team members obstruct opponents to assist another teammate behind. The teammate seizes the opportunity to catch up successfully, and its role transitions from an advancer to a defender. And Fig. 6b shows the situation where an opponent is overtaking. Two blue cars pay more attention to the opponent with a significant threat and collectively block it. Members switch back to one-to-one blocking strategies when the threat from opponents diminishes. Fig. 6c showcases the outcomes of one-to-one blocking achieved through task allocation in an equal-number scenario. Representative simulation examples are shown in the video (available online at https://youtu.be/OdTuzl_Ocag).

From the simulations, it is verified that the proposed method outperforms the GTP method [18] (Fig. 4), which considers more opponents in the competition, and does not act effectively on blocking opponents in two-team scenarios. With Algorithm 1, the slower team operates more cohesively, effectively assigning tasks among its members and focusing on the opponents around them.

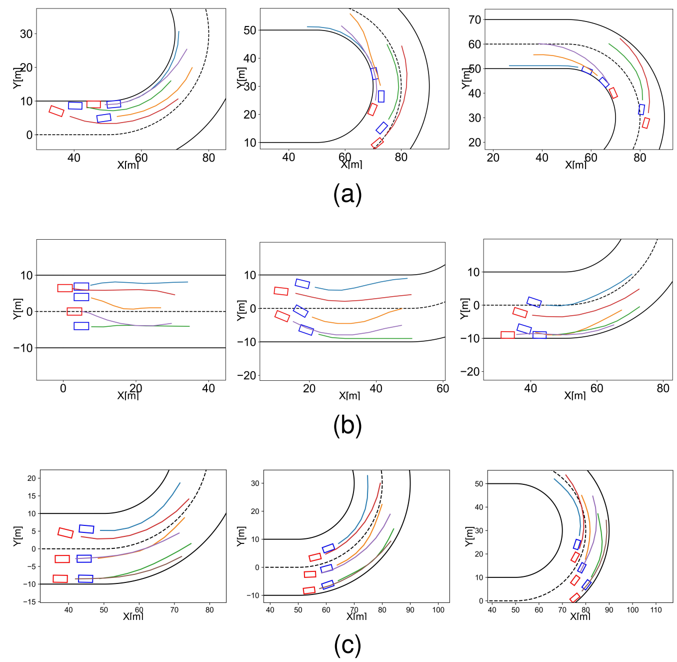

Fig. 6. Snapshots in different 8m/s vs. 10m/s scenarios. (a) shows the scenario where the teammates block the opponents to help a teammate. (b) shows the scenario where players cooperatively block an opponent. (c) shows one-to-one blocking scenarios.

V. CONCLUSION

This paper explores the autonomous racing involving two teams of multi vehicles. The objective is to find an approach for the slower team to win the game. This is achieved through an iterated best response approach combined with a designed task assignment algorithm.

We design a weight function to employ minimum weight bipartite graph matching method, and based on this, a task assignment algorithm is proposed. Furthermore, approaches for redistribution to address edge cases are also considered in Algorithm 1. The vehicles dynamically determine their roles in real-time based on the task assignment algorithm, enabling them to fulfill tasks such as advancing or blocking in dynamic scenarios. We demonstrate the effectiveness of the task assignment algorithm through simulations, comparing the results with and without Algorithm 1.

Then, a matching-based algorithm is designed for vehicles to iteratively update their strategies based on the task assignment algorithm, and the algorithm converges. We exhibit the results in different scenarios and showcase the varying driving behaviors among team members, thereby confirming the effectiveness of Algorithm 2.

References

[1] M. Buehler, K. Iagnemma, and S. Singh, *The DARPA urban challenge: autonomous vehicles in city traffic*. Springer, 2009, vol. 56.

[2] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[3] J. Betz, H. Zheng, A. Liniger, U. Rosolia, P. Karle, M. Behl, V. Krovi, and R. Mangharam, "Autonomous vehicles on the edge: A survey on autonomous vehicle racing," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 3, pp. 458–488, 2022.

[4] F. Christ, A. Wischnewski, A. Heilmeier, and B. Lohmann, "Time-optimal trajectory planning for a race car considering variable tyre-road friction coefficients," *Vehicle System Dynamics*, vol. 59, no. 4, pp. 588–612, 2021.

[5] A. Wischnewski, M. Geisslinger, J. Betz, T. Betz, F. Fent, A. Heilmeier, L. Hermansdorfer, T. Herrmann, S. Huch, P. Karle *et al.*, "Indy autonomous challenge-autonomous race cars at the handling limits," in *12th International Munich Chassis Symposium*, 2022, pp. 163–182.

[6] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger, "Learning-based model predictive control for autonomous racing," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3363–3370, 2019.

[7] J. Ni and J. Hu, "Path following control for autonomous formula racecar: Autonomous formula student competition," in *IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1835–1840.

[8] J. Ni, J. Hu, and C. Xiang, "Robust path following control at driving/handling limits of an autonomous electric racecar," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5518–5526, 2019.

[9] E. Pagot, M. Piccinini, and F. Biral, "Real-time optimal control of an autonomous rc car with minimum-time maneuvers and a novel kineto-dynamical model," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 2390–2396.

[10] J. L. Vázquez, M. Brühlmeier, A. Liniger, A. Rupenyan, and J. Lygeros, "Optimization-based hierarchical motion planning for autonomous racing," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 2397–2403.

[11] A. Heilmeier, A. Wischnewski, L. Hermansdorfer, J. Betz, M. Lienkamp, and B. Lohmann, "Minimum curvature trajectory planning and control for an autonomous race car," *Vehicle System Dynamics*, 2019.

[12] T. Herrmann, F. Christ, J. Betz, and M. Lienkamp, "Energy management strategy for an autonomous electric racecar using optimal control," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 720–725.

[13] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 scale RC cars," *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 628–647, 2015.

[14] P. Drews, G. Williams, B. Goldfain, E. A. Theodorou, and J. M. Rehg, "Vision-based high-speed driving with a deep dynamic observer," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1564–1571, 2019.

[15] A. Remonda, S. Krebs, E. Veas, G. Luzhnica, and R. Kern, "Formula RL: Deep reinforcement learning for autonomous racing using telemetry data," *arXiv preprint arXiv:2104.11106*, 2021.

[16] Y. Song, H. Lin, E. Kaufmann, P. Dürr, and D. Scaramuzza, "Autonomous overtaking in gran turismo sport using curriculum reinforcement learning," in *International Conference on Robotics and Automation (ICRA)*, 2021, pp. 9403–9409.

[17] A. Brunnbauer, L. Berducci, A. Brandstätter, M. Lechner, R. Hasani, D. Rus, and R. Grosu, "Model-based versus model-free deep reinforcement learning for autonomous racing cars," *arXiv preprint arXiv:2103.04909*, 2021.

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

[19] Z. Wang, T. Taubner, and M. Schwager, "Multi-agent sensitivity enhanced iterative best response: A real-time game theoretic planner for drone racing in 3d environments," *Robotics and Autonomous Systems*, vol. 125, p. 103410, 2020.

[20] Z. Wang, R. Spica, and M. Schwager, "Game theoretic motion planning for multi-robot racing," in *Distributed Autonomous Robotic Systems: The 14th International Symposium*. Springer, 2019, pp. 225–238.

[21] R. Spica, E. Cristofalo, Z. Wang, E. Montijano, and M. Schwager, "A real-time game theoretic planner for autonomous two-player drone racing," *IEEE Transactions on Robotics*, vol. 36, no. 5, pp. 1389–1403, 2020.

[22] G. Notomista, M. Wang, M. Schwager, and M. Egerstedt, "Enhancing game-theoretic autonomous car racing using control barrier functions," in *International Conference on Robotics and Automation (ICRA)*, 2020, pp. 5393–5399.

[23] A. Liniger and J. Lygeros, "A noncooperative game approach to autonomous racing," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 3, pp. 884–897, 2019.

[24] H. Zheng, Z. Zhuang, J. Betz, and R. Mangharam, "Game-theoretic objective space planning," *arXiv preprint arXiv:2209.07758*, 2022.

[25] C. Jung, S. Lee, H. Seong, A. Finazzi, and D. H. Shim, "Game-theoretic model predictive control with data-driven identification of vehicle model for head-to-head autonomous racing," *arXiv preprint arXiv:2106.04094*, 2021.

[26] T. Brüdigam, A. Capone, S. Hirche, D. Wollherr, and M. Leibold, "Gaussian process-based stochastic model predictive control for overtaking in autonomous racing," *arXiv preprint arXiv:2105.12236*, 2021.

[27] S. He, J. Zeng, and K. Sreenath, "Autonomous racing with multiple vehicles using a parallelized optimization with safety guarantee using control barrier functions," in *International Conference on Robotics and Automation (ICRA)*, 2022, pp. 3444–3451.

[28] S. W. Loke, "Cooperative automated vehicles: A review of opportunities and challenges in socially intelligent vehicles beyond networking," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 4, pp. 509–518, 2019.

[29] J. Di, S. Chen, P. Li, X. Wang, H. Ji, and Y. Kang, "A cooperative-competitive strategy for autonomous multidrone racing," *IEEE Transactions on Industrial Electronics*, 2023.

[30] Y. Cui, J. Qiu, W. Sun, T. Huang, and X. Gong, "How to beat speed with quantity: A many-to-one multi-vehicle racing framework," *IEEE Transactions on Vehicular Technology*, pp. 1–13, 2023.

[31] L. Li, Q. Yuan, S. Li, C. Wang, and G. Xie, "Cooperative-competitive game-based method for local path planning with obstacle avoidance in racing scenarios," in *2022 41st Chinese Control Conference (CCC)*, 2022, pp. 1803–1808.

[32] R. S. Thakkar, A. S. Samyal, D. Fridovich-Keil, Z. Xu, and U. Topcu, "Hierarchical control for cooperative teams in competitive autonomous racing," *arXiv preprint arXiv:2204.13070*, 2022.

[33] D. Bruff, "The assignment problem and the hungarian method," *Notes for Math*, vol. 20, no. 29-47, p. 5, 2005.

[34] M. Chen, Z. Zhou, and C. J. Tomlin, "Multiplayer reach-avoid games via pairwise outcomes," *IEEE Transactions on Automatic Control*, vol. 62, no. 3, pp. 1451–1457, 2017.

[35] R. Yan, X. Duan, Z. Shi, Y. Zhong, and F. Bullo, "Matching-based capture strategies for 3d heterogeneous multiplayer reach-avoid differential games," *Automatica*, vol. 140, p. 110207, 2022.

[36] H. Hu, M. Bui, and M. Chen, "Multi-agent reach-avoid games: Two attackers versus one defender and mixed integer programming," in *IEEE Conference on Decision and Control (CDC)*, 2023, pp. 7227–7233.

[37] R. Yan, Z. Shi, and Y. Zhong, "Task assignment for multiplayer reach–avoid games in convex domains via analytical barriers," *IEEE Transactions on Robotics*, vol. 36, no. 1, pp. 107–124, 2020.

[38] L. Ford Jr and D. Fulkerson, "Flows in networks," 1962.

[39] Z. Hu, X. Li, and M. Meng, "A stackelberg game approach for three-player autonomous racing," in *2023 42nd Chinese Control Conference (CCC)*, 2023, pp. 4886–4891.