

Markov Potential Game with Final-Time Reach-Avoid Objectives

Sarah. H.Q. Li, Abraham P. Vinod

Abstract—We formulate a Markov potential game with final-time reach-avoid objectives by integrating potential game theory with stochastic reach-avoid control. Our focus is on multi-player trajectory planning where players maximize the same multi-player reach-avoid objective: the probability of all participants reaching their designated target states by a specified time, while avoiding collisions with one another. Existing approaches require centralized computation of actions via a global policy, which may have prohibitively expensive communication costs. Instead, we focus on approximations of the global policy via local state feedback policies. First, we adapt the recursive single player reach-avoid value iteration to the multi-player framework with local policies, and show that the same recursion holds on the joint state space. To find each player’s optimal local policy, the multi-player reach-avoid value function is projected from the joint state to the local state using the other players’ occupancy measures. Then, we propose an iterative best response scheme for the multi-player value iteration to converge to a pure Nash equilibrium. We demonstrate the utility of our approach in finding collision-free policies for multi-player motion planning in simulation.

I. INTRODUCTION

As advanced air mobility systems become a reality, the expectation for air mobility’s operation scale and vehicle sizes will change significantly, creating potentially unprecedented traffic flows within more confined air spaces [1], [2]. Aerial collisions can have catastrophic impacts [3]. As a result, existing air traffic management prioritizes individual vehicle safety during operation, and often rely on human operators to certify and minimize vehicle encounters in dense air spaces such as take-off and landing zones [4]. Although trained human operators are irreplaceable for making time-critical decisions for air space encounters, their mental limitations makes them a bottleneck in increasing the traffic throughput of future mobility systems.

To overcome these limitations posed by human operators, we propose a safety-prioritized, multi-player routing model using a Markov Decision Process (MDP) that may eventually support tactical time routing in advanced air mobility systems. We aim to overcome a key challenge with routing game-type models for ground-based mobility systems: the lack of accountability for individual vehicle’s safety probability over a finite time horizon. For ground mobility systems, individual vehicles rely on their on-board pilots (humans or autonomous) to ensure individual safety during operation. While this makes sense for ground-based transportation

systems where individual vehicles have sufficient maneuverability and local observability, aerial and spatial vehicles have more constrained observability and maneuverability. Therefore, we seek a mathematical model in which individual safety is directly maximized and guaranteed.

Contributions. We propose a reach-avoid Markov potential game in which players directly maximize their finite horizon probability of collectively reaching their respective destination while avoiding other players. We show that 1) the proposed multi-player reach-avoid objective is multilinear in each player’s local policy space, 2) the Nash equilibrium condition is a relaxation of the globally optimal condition, and 3) despite using local policies, the value function for the multi-player reach-avoid objective must be recursively defined over the joint state space. We extend the existing multiplicative dynamic programming framework to an iterative best response scheme that utilizes the occupancy measure of opponent players to project the joint state value function to a local state value function, and use it to find the optimal local policy. Finally, we prove and validate in simulation that our best response scheme can effectively find the Nash equilibrium policy. We conclude with a simulation study that evaluates the dependence of the computation complexity on state size and the number of players.

II. RELATED LITERATURE

Our proposed routing model’s objective differs from that of established routing models for traffic management [5], [6]. In established routing models, a time-instantaneous safety metric is used to formulate a sum-separable objective. When this objective is minimized, the congestion levels of individual routes are minimized with respect to the origin-destination demands of the vehicle population [7]. Alternatively, [8] solves the multi-player routing problem in air traffic management via a collection of single-player routing problems, which can be subsequently solved via existing techniques in reachability [9]–[11]. Additionally, [9], [10] also derived dynamic programming-based solutions for maximizing the reach-avoid objective.

Subsequent works solve the reach-avoid optimization problem with time-varying and joint chance constraints [11], [12]. Game-theoretical extensions of reach-avoid objectives have been investigated primarily under zero sum-types of interaction, where two teams of players maximize/minimize the same objective [13]–[16]. In [17], the authors formulated a hierarchical framework for simultaneously incorporating high fidelity interaction and high fidelity vehicle dynamics. In contrast, our primary focus is on a collaborative game

Sarah H. Q. Li is with the Automatic Control Laboratory, ETH Zürich, Physikstrasse 3, Zürich, 8092, Switzerland (email: huili@ethz.ch).

Abraham P. Vinod is with the Mitsubishi Electric Research Laboratories, Cambridge, MA, USA(email:abraham.p.vinod@ieee.org)

setting where players jointly minimize a common potential function via unilateral policy changes. Potential game theory has been an integral part of traffic research since the formulation of the Wardrop equilibrium in [18]. Game-theoretic modeling of mobility systems is exemplified by the routing game model from [19] and its stochastic variants from [7], [20]. However, reach-avoid objectives have not been explored within a potential game framework, to the best of our knowledge.

Notation: A set with N elements is denoted as $[N] = \{0, \dots, N-1\}$. The set of matrices of i rows and j columns with real (non-negative) valued entries is given by $\mathbb{R}^{i \times j}(\mathbb{R}_+^{i \times j})$. The ones column vector is denoted by $\mathbb{1}_N = [1, \dots, 1]^T \in \mathbb{R}^{N \times 1}$. The simplex in \mathbb{R}^N , $\{x \in \mathbb{R}_+^N | \mathbb{1}^\top x = 1\}$, is denoted by Δ_N . The set of random variables that take on values from sample space Ω is denoted as \mathcal{X}_Ω .

III. MULTI-PLAYER REACH-AVOID MDP

Our problem concerns designing trajectories for N players under interaction constraints within a shared state space and a common finite time horizon. Each player aims to reach a target set at the end of the horizon while avoiding the other players over the entire time horizon. Our formulation is motivated by existing work on stochastic reach-avoid problems [9]–[11],

A. Multi-player reach-avoid MDP

We model each player as a finite horizon, finite state-action MDP, where i^{th} player's MDP is given by $(\mathcal{S}, \mathcal{A}_i, T, P_i, p_i, \mathcal{T}_i)$. Each player has the same state space \mathcal{S} and the same time horizon $T \in \mathbb{N}$. The action set \mathcal{A}_i is player-specific with $A_i \in \mathbb{N}$ elements. Each action is admissible from each state. Each player i has initial state $s_i(0)$, which is stochastically distributed over \mathcal{S} according to the probability distribution $p_i \in \Delta_{\mathcal{S}}$, and aims to reach its target state set at time T , given by $\mathcal{T}_i \subseteq \mathcal{S}_i$. At each time step $t \in [T]$, player i 's state $s_i(t)$ is a time-varying Markov process. Each player has independent transition dynamics.

Assumption 1 (INDEPENDENT MARKOV TRANSITIONS). *For every player $i \in [N]$ and time step $t \in [0, T)$, its next state $s_i(t+1)$ is a Markov random variable that depends on its current state $s_i(t)$ and action $a_i(t)$, denoted by $P_i(s_i(t), a_i(t)) \in \mathcal{X}_{\mathcal{S}}$ such that*

$$s_i(t+1) \sim P_i(s_i(t), a_i(t)), \forall s_i(t) \in \mathcal{S}, t \in [T], a_i(t) \in \mathcal{A}_i.$$

We denote the probability distribution of $P_i(s_i(t), a_i(t))$ as $\mathbb{P}_i[\cdot | s_i, a_i] \in \Delta_{\mathcal{S}}$, for all $s_i \in \mathcal{S}$ and $a_i \in \mathcal{A}_i$.

Player i selects each action $a_i(t)$ via a state-dependent and time-varying policy π_i . We consider player policies that depends strictly on each player's local state.

Definition 1 (LOCAL FEEDBACK). *For every player $i \in [N]$, a policy $\pi_i(\cdot, t)$ is a function of player i 's state $s_i(t)$,*

$$\pi_i : \mathcal{S} \times [T] \mapsto \mathcal{X}_{\mathcal{A}_i}, a_i(t) \sim \pi_i(s_i(t), t), \forall t \in [T], s_i \in \mathcal{S}. \quad (1)$$

We use Π_i to denote the set of all policies π_i satisfying (1).

Under policy π_i , we use $y_i^{\pi_i}(s_i, \hat{s}_i, t)$ to denote player i 's probability of transitioning to \hat{s}_i at time $t+1$ if player i was at s_i at time t . Intuitively, $y_i(s_i, \hat{s}_i, t)$ is the (\hat{s}_i, s_i) element of the Markov transition matrix under policy $\pi_i(\cdot, t)$. We observe that each $y_i^{\pi_i}(\cdot, s_i(t), t)$ is linear in $\pi_i(s_i(t), t)$,

$$y_i^{\pi_i}(s_i, \hat{s}_i, t) = \sum_{a_i \in \mathcal{A}_i} \mathbb{P}_i[\hat{s}_i | s_i, a_i] \mathbb{P}[a_i | \pi_i(s_i, t)]. \quad (2)$$

When the context is clear, we drop the superscript π_i for y_i (2) to simplify the notation. We denote player i 's state trajectory over $T+1$ time steps as $\tau_i \in \mathcal{S}^{T+1}$. Each trajectory $\tau_i = (s_i(0), \dots, s_i(T))$ is a realization of sequential random variables determined by a Markov process $h_i(\pi_i) \in \mathcal{X}_{\mathcal{S}^{T+1}}$, such that the probability of trajectory τ_i occurring is

$$\mathbb{P}[\tau_i | h_i(\pi_i)] = p_i(s_i(0)) \left(\prod_{t=0}^{T-1} y_i^{\pi_i}(s_i(t), s_i(t+1), t) \right). \quad (3)$$

Multi-player reach-avoid objective. All players share similar objective: a) avoid other players at all time steps and b) reach their respective target set \mathcal{T}_i at time T . If either of these conditions are violated for any player, all players receive zero reward. To model this objective, we introduce the following indicator functions,

$$X_i(s) = \mathbb{1}(s \in \mathcal{T}_i), \quad (4)$$

$$Y_{ij}(s_i, s_j) = \begin{cases} \mathbb{1}(s_i \neq s_j) & j \neq i \\ 1 & j = i \end{cases}, \forall i, j \in [N]. \quad (5)$$

For each joint player trajectory $\{\tau_i\}_{i \in [N]}$ with $\tau_i \in \mathcal{S}^{T+1}$, we use the indicator function $R(\tau_1, \dots, \tau_N)$ to indicate whether the joint trajectory achieves the multi-player reach-avoid objective, defined as

$$R(\tau_1, \dots, \tau_N) = \prod_{i \in [N]} X_i(T) \prod_{t=0}^T \prod_{j \in [N]} Y_{ij}(t). \quad (6)$$

We define the expected value of R (6) under the joint policy (π_1, \dots, π_N) as

$$F(\pi_1, \dots, \pi_N) = \mathbb{E}[R(\tau_1, \dots, \tau_N) | \tau_i \sim h_i(\pi_i), \forall i \in [N]]. \quad (7)$$

Each player uses their policy π_i (1) to maximize F (7) with respect to the other players' policies π_{-i} , such that player i 's reach-avoid MDP is given by

$$\max_{\pi_i \in \Pi_i} F(\pi_i, \pi_{-i}), \quad \forall i \in [N]. \quad (8)$$

Player i 's individual reach-avoid problem (8) can become trivially defined depending on the opponent policy π_{-i} . Under π_{-i} , there must be trajectories $\tau_i \in \mathcal{S}^{T+1}$ such that $R(\tau_1, \dots, \tau_N) = 1$ and $\tau_i \sim h(\pi_i)$ is realizable with positive probability. Otherwise, $F(\pi_i, \pi_{-i}) = 0$ for all $\pi_i \in \Pi_i$.

The objective F (7) is the same for every player's reach-avoid MDP. The result is that each player *equally* prioritizes the reachability of their own targets and the reachability of the other players' targets. If a joint trajectory achieves target reachability and opponent avoidance for player i , but causes another player j to fail target reachability or creates collision for players $j, k \neq i$, then player i does not prefer such a trajectory when solving (8).

B. Markov potential game and Nash equilibrium policies

We seek a joint policy (π_1, \dots, π_N) that is jointly optimal for the N -coupled MDPs posed in (8).

Definition 2 (NASH EQUILIBRIUM). *The joint policy $(\pi_1^*, \dots, \pi_N^*)$ is a Nash equilibrium if and only if for all $i \in [N]$,*

$$F(\pi_i^*, \pi_{-i}^*) \geq F(\pi_i, \pi_{-i}^*), \quad \forall \pi_i \in \Pi_i. \quad (9)$$

Players reach a Nash equilibrium when no one can further improve the multi-player reach-avoid objective in (8) via unilateral policy changes. The Nash equilibrium has the following interpretation: in the event that all players except i fix their policies to π_{-i}^* , player i 's policy π_i^* maximizes the likelihood of player i 's trajectory τ_i reaching target \mathcal{T}_i while avoiding other players over the entire time horizon T .

Connections to policies with global feedback. A natural alternative to (8) is to directly optimize the multi-player reach-avoid objective over the policies that have global state feedback. For all players $i \in [N]$, consider the policy $\hat{\pi}_i : \mathcal{S}^N \times [T] \mapsto \mathcal{X}_{\mathcal{A}_i}$ that takes global state feedback as

$$a_i(t) \sim \hat{\pi}_i(s_1(t), \dots, s_N(t), t), \quad \forall t \in [T], \quad (10)$$

and the global optimization problem given by

$$\max_{\hat{\pi}_1, \dots, \hat{\pi}_N} \mathbb{E}[R(\tau_1, \dots, \tau_N) | \tau_j \sim h_j(\hat{\pi}_j), \quad \forall j \in [N]]. \quad (11)$$

Problem (11) may be viewed as a multi-player extension of the single-player reach-avoid problem from [9]–[11] under Assumption 1. It differs from (8) in two aspects: first, global policy solution space vs local policy solution space, and second, global optimality vs unilateral optimality. Since the global policy space (10) subsumes the local policy space (1) and unilateral optimality is a necessary condition for global optimality, the joint policy that maximizes (11) will achieve a better multi-player reach-avoid objective than the Nash equilibrium policy.

However, the joint policy $(\hat{\pi}_1, \dots, \hat{\pi}_N)$ has impractical memory and operation requirements: each $\hat{\pi}_i$'s memory requirement grows exponentially with additional players, and at every time step, players must share their states in an all-to-all communication network to gather the necessary inputs for each policy. To avoid these impractical requirements while leveraging the single player multiplicative dynamic programming solution, we relax the global optimality conditions from (11) using the Nash equilibrium conditions (9) and the policy memory requirements via Assumption 1. As seen in Definition 2, a Nash equilibrium implies that the joint policy is coordinate-wise optimal: the joint policy is optimal if collaborations between players are ignored. From this perspective, the Nash equilibrium conditions are necessary towards the optimality of (11), and Nash equilibrium is an approximation and a lower bound to (11).

Connections to Markov potential games. The game defined in (8) is a potential game [21]—i.e., there exists an ordinal potential function $F : \Pi_1 \times \dots \times \Pi_N \mapsto \mathbb{R}$ that satisfies $\forall \pi_i, \hat{\pi}_i \in \Pi_i, \quad \forall i \in [N]$,

$$F_i(\pi_i, \pi_{-i}) > F_i(\hat{\pi}_i, \pi_{-i}) \Leftrightarrow F(\pi_i, \pi_{-i}) > F(\hat{\pi}_i, \pi_{-i}). \quad (12)$$

Given that each player's objective F_i are identical, $F_i = F$ (7) is the obvious choice of the potential function.

As a Markov potential game, (8) has a set of Nash equilibria that possess well-behaved computational and theoretical properties, some of which we list below.

Solution structure. A Markov game has at least one pure Nash equilibrium $(\pi_1^*, \dots, \pi_N^*)$ where each π_i^* is deterministic: at every state, a unique action is always chosen. [21]

Connections to single-player dynamic programming. The reach-avoid MDP in both single and multi-player settings is a non-convex optimization problem on which first-order gradient methods do not have good guarantees. In the single player setting, existing work show that multiplicative dynamic programming provably converges to the optimal policy [9], [10] and one may use convex optimization for grid-free computation when the dynamics are linear [11]. Via potential game theory, we can extend the single-player multiplicative dynamic programming to multi-player policy update schemes by leveraging existing work on multi-player learning dynamics. In the next section, we show that iterative best response is one such learning dynamic that will find an equilibrium in the joint policy space under NE assumptions. Additionally, other gradient-based methods such as Frank-Wolfe [22] and gradient play [23] can also be used in conjunction with Algorithm 2 to compute the Nash equilibrium.

IV. MULTI-PLAYER MULTIPLICATIVE DYNAMIC PROGRAMMING

In this section, we evaluate the multi-player reach-avoid objective (7) using the conditional transition distributions $\{y_i\}_{i \in \mathbb{N}}$ (2) and show that the objective can be recursively computed via a value function on the joint state space. We project the value function using each player's occupancy measures to construct an iterative best response scheme that extends the single-player multiplicative dynamic programming solution.

A. Multilinear program formulation

We show that the objective (7) is multilinear in each player's policy π_i by showing that the objective is multilinear in the conditional transition distributions $\{y_i\}_{i \in \mathbb{N}}$ (2).

Lemma 1. *Any real-valued function $G : \mathcal{S}^{NK} \mapsto \mathbb{R}$ that takes in a joint trajectory $\{\tau_i\}_{i \in \mathbb{N}}$, where $\tau_i \in \mathcal{S}^K$ and is a Markov process as described by $h(\pi_i)$ (3), then the expectation of G with respect to $\{\pi_i\}_{i \in \mathbb{N}}$ is multilinear in $\{y_i\}_{i \in \mathbb{N}}$ (2), i.e.,*

$$\begin{aligned} & \mathbb{E}[G(\tau_1, \dots, \tau_N) | \tau_j \sim h_j(\pi_j), \forall j \in [N]] = \\ & \sum_{\substack{\tau_1, \dots, \tau_N \\ \in \mathcal{S}^{NK}}} G(\tau_1, \dots, \tau_N) \prod_{t=0}^T \prod_{j \in [N]} p_j(s_j(0)) y_j(s_j(t), s_j(t+1), t), \end{aligned} \quad (13)$$

where $\tau_i = (s_i(0), \dots, s_i(K-1))$ for all $i \in [N]$.

Proof. Let τ_1, \dots, τ_N represent a joint trajectory where at each time step t , player i is at state $s_i(t)$ for all $i \in [N]$, $t \in [K-1]$, and let Γ denote the set of all realizable joint

trajectories, then $\mathbb{E}[G(\tau_i, \tau_{-i}) | \tau_j \sim h_j(\pi_j), \forall j \in [N]]$ can be evaluated as

$$F_i(\pi_i, \pi_{-i}) = \sum_{\tau \in \Gamma} \prod_i \mathbb{P}[\tau_i] G_i(\tau_i, \tau_{-i}), \quad (14)$$

where $\mathbb{P}[\tau_i]$ denotes the joint probability of player i being at state $s_i(t)$ for all time steps $t = 0, \dots, K-1$. We can directly evaluate $\mathbb{P}[\tau_i]$ as

$$\mathbb{P}[\tau_i] = \mathbb{P}[s_i(0)] \prod_{t=0}^{K-1} \mathbb{P}[s_i(t+1) | s_i(t), a_i(t) \sim \pi_i(s_i(t), t)].$$

where $\mathbb{P}[s_i(t+1) | s_i(t), a_i(t) \sim \pi_i(s_i(t), t)] = y(s_i(t), s_i(t+1), t)$ as defined in (2) and $\mathbb{P}[s_i(0)] = y(0, \tau_i, \pi_0)$ is the initial state distribution. \square

If we apply Lemma 1 to (7), we observe that (8) is in fact a multilinear optimization problem over a compact policy domain. Its global optimal solution can therefore be difficult to compute and certify. Interestingly, when $N = 1$, the single player reach-avoid MDP (11), despite being multi-linear in π_i still, has a globally optimal solution that multiplicative dynamic programming is guaranteed to find [9]–[11].

Since the multi-player reach-avoid objective F (7) is an multilinear function of individual player policies π_1, \dots, π_N , finding even a locally optimal solution to (8) via optimization algorithms can be challenging; most gradient-based algorithms tend to converge to KKT solutions that are not sufficient for guaranteeing optimality in the nonconvex setting.

Instead, we leverage game-theoretic learning dynamics to synthesize distributed algorithms for finding Nash equilibrium policies. By relaxing the global collaborative reach-avoid problem (11) as a potential game, we uncover iterative best response as a possible multi-player learning scheme for finding joint policy equilibria. As a potential game, we know that iterative best response over the deterministic policy domain converges to the Nash equilibrium policies. However, it remains to be seen how each player can compute the best response to the opponent policies.

B. Multi-player value function

A single player reach-avoid MDP with deterministic obstacles can be solved via multiplicative dynamic programming [10]. When adapting the single player reach-avoid MDP to the multi-player reach-avoid MDP, two critical gaps to address are 1) how does having having multiple players with individual states and policies change the value functions' structure, and 2) how does having *stochastic obstacles* that correspond to players change multiplicative dynamic programming?

Similar to the single-player value function definition from [9], [10], we formulate the multi-player value function below and show that it recursively computes the multi-player reach-avoid objective.

$$\begin{aligned} V_T^\pi(s_1, \dots, s_N) &= \prod_j X_j(s_j) \prod_{i,j} Y_{ij}(s_i, s_j), \\ V_t^\pi(s_1, \dots, s_N) &= \prod_{i,j} Y_{ij}(s_i, s_j) \sum_{\hat{s} \in \mathcal{S}^N} \prod_j y_j(s_j, \hat{s}_j, t) V_{t+1}^\pi(\hat{s}). \end{aligned} \quad (15)$$

We note that each $s, \hat{s} \in \mathcal{S}^N$ from (15) corresponds to a joint state of all players: $s = (s_1, \dots, s_N)$ and $\hat{s} = (\hat{s}_1, \dots, \hat{s}_N)$.

Proposition 1. *Under the joint policy $\pi = (\pi_1, \dots, \pi_N)$, V_0^π, \dots, V_T^π as defined in (15) are the expected value of the random variable*

$$R_t^T(\tau_1, \dots, \tau_N) = \prod_i X_i(s_i(T)) \prod_{\hat{t}=t}^T \prod_{i,j} Y_{ij}(s_i(\hat{t}), s_j(\hat{t})). \quad (16)$$

with respect to π —i.e., $V_t^\pi(s_1, \dots, s_N)$ (15) is equivalent to

$$\begin{aligned} V_t^\pi(s_1, \dots, s_N) &= \mathbb{E}^\pi \left[R_t^T(\tau_1, \dots, \tau_N) \right] \\ &\quad \tau_i \sim h_i(\pi_i), \tau_i(0) = s_i, \forall i \in [N]. \end{aligned} \quad (17)$$

The proof is provided in App. A. Proposition 1 specifies the more general results from [9], [10] to the finite state-action MDP under decoupled player dynamics in Assumption 1.

C. Computing player i 's best response

A key difference between the single player reach-avoid MDP and the multi-player reach-avoid MDP is the state availability for policy feedback. In the single-player value function, the global state is available, while in the multi-player reach-avoid MDP, only the local state is available for each player. We first observe that when all players except i take on policies $\pi_{-i}(t), \dots, \pi_{-i}(T-1)$, the multi-player reach-avoid value $V_t^{\pi_i, \pi_{-i}}(s)$ as function of π_i is given by (15) with y_{-i} explicitly evaluated via the other players policies. In addition to this, the expected multi-player reach-avoid value at state s_i depends on the occupancy measures of all other players at time y , which depends on the policies $\pi_{-i}(0), \dots, \pi_{-i}(t-1)$. Collectively, this implies that the expected multi-player reach-avoid value can be directly computed as

$$\begin{aligned} \mathbb{E} \left[V_t^{\pi_i, \pi_{-i}}(s_i, s_{-i}) | \pi_{-i} \right] &= \sum_{s_{-i}} \rho_{-i}(s_{-i}, t) \prod_{j,\ell} Y_{ij}(s_i, s_j) \\ &\quad \sum_{\hat{s}_i} \mathbb{P}[\hat{s}_i | s_i(t), \pi_i] \sum_{\hat{s}_{-i}} \prod_{j \neq i} y_j(s_j, \hat{s}_j, t) V_{t+1}(\hat{s}_i, \hat{s}_{-i}), \end{aligned} \quad (18)$$

where $\rho_{-i}(s_{-i}, t) = \prod_{j \neq i} \rho_j(s_j, t)$ correspond to the occupancy measures of players $[N] \setminus \{i\}$ and can be found via the forward propagation of policies $\pi_{-i}(0), \dots, \pi_{-i}(t-1)$ through player i 's Markov dynamics (Algorithm 1). Together, $\rho_j(s_j, t) y_j(s_j, \hat{s}_j, t)$ denote the joint probability that player i was in state s_j at time t and state \hat{s}_j at time $t+1$.

Obstacles with Markov dynamics. In (18), we observe that $\sum_{\hat{s}_{-i}} \prod_{j \neq i} y_j(s_j, \hat{s}_j, t) V_{t+1}(\hat{s}_i, \hat{s}_{-i})$ is the expected future reward for player i if it makes the transition from s_i to \hat{s}_i at time t . Furthermore, consider (18) without the expected future reward $\sum_{\hat{s}_{-i}} y_{-i}(s_{-i}, \hat{s}_{-i}, t) V_{t+1}(\hat{s}_i, \hat{s}_{-i})$. It becomes

$$\sum_{s_{-i}} \rho_{-i}(s_{-i}, t) \prod_{j,\ell} Y_{ij}(s_i, s_j). \quad (19)$$

Algorithm 1 Retrieving density trajectory from a policy

Require: P^i, p_i, π_i .
Ensure: $\{\rho : \mathcal{S} \times [T] \mapsto [0, 1]\}$
 $\rho(s, t) = 0, \forall t \in [T], s \in \mathcal{S}$
 $\rho(s, 0) = p_i(s) \forall s \in \mathcal{S}$
for $t = 0, \dots, T - 1$ **do**
 for $s \in \mathcal{S}$ **do**
 $\rho(s, t + 1) = \sum_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P_{t, ss'a} \rho(t, s') \pi_i(s', a, t)$
 end for
end for

Since each players' state trajectory is determined by an MDP, the obstacles' states are random variables rather than deterministic locations at each time step. Since $\rho_{-i}(s_{-i}, t)$ are the probability density distributions of all players $[N]/\{i\}$, and player i 's state s_i at time t is given as an input to V in (18), the expression in (19) is equivalent to the likelihood of all players avoiding each other at time step t under policies π_{-i} and conditioned on player i being at state s_i , i.e.,

$$\mathbb{P}[s_j(t) \neq s_\ell(t), \forall j, \ell \in [N] | s_i(t) = s_i, \tau_j \sim h_j(\pi_j), \forall j \neq i]. \quad (20)$$

In particular, if h_j are deterministic processes, such that each player j has deterministic states $s_j(t) = s_j$, then (20) recovers the indicator function $\prod_{j, \ell} \mathbb{1}(s_j(t) \neq s_\ell(t))$. We can conclude that (19) is a probabilistic relaxation of the indicator function when players j 's states are given by deterministic states:

$$\prod_{j, \ell} \mathbb{1}(s_j(t) \neq s_\ell(t)) \rightarrow \mathbb{P}\left[\prod_{j, \ell} s_j(t) \neq s_\ell(t) | s_i(t) = s_i\right].$$

Multiplicative Best Response. Proposition 1 and (18) enable us to directly adapt multiplicative dynamic programming from [9], [10] to perform a best response scheme for reach-avoid Markov potential games (8). The resulting algorithm is shown in Algorithm 3.

We formulate the best response value function $W^* \in \mathbb{R}^{\mathcal{S}(T+1)}$ as

$$\begin{aligned} W_T^*(s_i) &= \sum_{s_{-i}} \rho_{-i}(s_{-i}, T) V_T^{\pi_i, \pi_{-i}}(s_i, s_{-i}), \quad \forall s_i \in \mathcal{S}^N \\ W_t^*(s_i) &= \max_{\pi_i \in \mathcal{X}_{\mathcal{A}_i}} \sum_{\hat{s}_i} y_i^{\pi_i}(s_i, \hat{s}_i, t) \prod_{j, \ell} Y(s_j, s_\ell) \\ &\quad \sum_{s_{-i}, \hat{s}_{-i}} \prod_{j \neq i} y_j(s_j, \hat{s}_j) \rho_j(s_j, t) V_{t+1}^*(\hat{s}_i, \hat{s}_{-i}), \quad \forall s_i \in \mathcal{S}^N, \end{aligned} \quad (21)$$

and $\pi_i^*(s_i, t)$ as an argmax policy that achieves $W_t^*(s_i)$ for all $t, s_i \in [T] \times \mathcal{S}$. Then π_i^* is player i 's best response policy against opponent policy π_{-i} , and $\mathbb{E}[W_0^*(s_i) | s_i \sim p_i]$ is the maximum expected multi-player reach-avoid objective (7) that player i can achieve against opponent policy π_{-i} .

Unlike standard dynamic programming approaches to compute the optimal global policy [9]–[11], player i 's value function is not recursive by itself—i.e., W_t is not recursively defined by W_{t+1} . Instead, we “average” out the effect of other players' state on the global value function using their occupancy measure. The resulting value function, which we

Algorithm 2 Individual player best response

Require: $\pi_{-i}, P_{-i}, p_{-i}, \mathcal{T}_{-i}$.
Ensure: π_i^*
1: **for** $j \in [N]/\{i\}$ **do**
2: $\rho_j = \text{Alg. 1}(P_j, p_j, \pi_j)$
3: **end for**
4: **for** $s \in \mathcal{S}^N$ **do**
5: $V_T(s) = \prod_i X_i(s_i) \prod_{j \neq i} Y(s_i, s_j)$
6: **end for**
7: **for** $t = T - 1, \dots, 0$ **do**
8: **for** $s_{-i}, \hat{s}_{-i} \in \mathcal{S}^{N-1}$ **do**
9: $\rho(s_{-i}, \hat{s}_{-i}) = \prod_{j \neq i} P_j(\hat{s}_j | s_j, \pi_j(s_j, t)) \rho(s_{-i}, t)$
10: **end for**
11: **for** $s_i \in \mathcal{S}$ **do**
12: $\pi_i(s_i; t) = \operatorname{argmax}_{a_i} \prod_{j, \ell} Y_{j\ell}(s_j, s_\ell)$
13: $\sum_{\hat{s}_i} P_i(\hat{s}_i | s_i, a_i) \sum_{s_{-i}, \hat{s}_{-i}} \rho(s_{-i}, \hat{s}_{-i}) V_{t+1}(\hat{s}_i)$
14: **for** $\hat{s}_i \in \mathcal{S}$ **do**
15: $\rho_i(s_i, \hat{s}_i) = P_i(\hat{s}_i | s_i, \pi_i(s_i, t))$
16: **end for**
17: **end for**
18: **for** $s \in \mathcal{S}^N$ **do**
19: $V_t(s) = \prod_{j, \ell} Y_{j\ell}(s_j, s_\ell) \sum_{\hat{s}} \rho_i(s_i, \hat{s}_i) \rho(s_{-i}, \hat{s}_{-i}) V_{t+1}(\hat{s})$
20: **end for**
21: **end for**

denote by W_t , remains an under approximation to the multi-player reach-avoid objective, since a global policy may be able to coordinate the players for better performance at the cost of additional communication and memory overhead.

Policy memory complexity. The output policy of Algorithm 2 requires ST memory units for storage and does not scale with increasing number of players. Achieving this complexity is a key motivation for using local feedback policies and the game-theoretical framework for evaluating the multi-player reach-avoid objective.

Computation complexity. Algorithm takes 1) SNT steps to perform forward propagation, 2) $(T + 1)S^{2N}$ to compute two-time step occupancy measure $\rho(s_{-i}, \hat{s}_{-i})$, 3) AS^2 , 4) TS^N to compute the previous time step value functions. Exploring parallel computing extensions as well as other computational speed ups of Algorithm 2 will be a topic of future work.

One factor that controls Algorithm 2's computation complexity is the occupancy measure at each state. For MDPs with sparse transitions—i.e., most of the player's occupancy measures transition predominantly to a small subset of states—may be faster to evaluate than the worst-case computation complexity. Furthermore, we can eliminate states that have minimal occupancy measure and therefore minimal contribution to the reach-avoid objective, as to trade off computation efficiency for accuracy. The impact of occupancy measure is also time-dependent—the later on in the MDP

time horizon, the more smaller occupancy measures matter. Therefore, we propose using the following heuristic to approximate the two-time step occupancy measure $\rho(s_{-i}, \hat{s}_{-i})$ in Alg. 2 line 9 to reduce the computation complexity.

$$\rho(s_{-i}, \hat{s}_{-i}) = \begin{cases} 0 & \exists j \neq i, \rho(s_{-i}) \leq \epsilon \\ \prod_{j \neq i} P_j(\hat{s}_j | s_j) \rho(s_{-i}, t) & \text{otherwise} \end{cases} \quad (22)$$

From Algorithm 2, we can derive the following multi-player update scheme that converges to the Nash equilibrium in polynomial time. We note that in addition to Algorithm 3, other gradient-based methods such as Frank-Wolfe [22] and gradient play [23] can also be used in conjunction with Algorithm 2 to compute the Nash equilibrium.

Algorithm 3 Iterative Best Response

Require: $\mathcal{T}_i, \mathcal{O}_{ik}, P^i$.

Ensure: π_1^*, \dots, π_N^*

```

1: while  $k = 1, \dots$  do
2:    $i = k \bmod N$ 
3:    $V_i, \pi_i = \text{Alg. } 2(\pi_{-i}^{k-1}, P_{-i}, p_{-i}, \mathcal{T}_{-i})$ 
4:    $\pi_i^k = \pi_i; \pi_{-i}^k = \pi_{-i}^{k-1}$ 
5:   if  $V^i = V^j, \forall i, j \in [N]$  then
6:      $\pi_i^* = \pi_i^k, \forall i \in [N]$ 
7:     Exit
8:   end if
9: end while

```

Theorem 1. *Algorithm 3 converges to a pure-strategy Nash equilibrium in polynomial time [21].*

Whether Algorithm 3 converges to an optimal joint policy that maximizes the multi-player reach-avoid objective (11) over the joint policy space $\Pi_1 \times \dots \times \Pi_N$ is unknown. However, it does converge to a coordinate-wise optimal solution that under-approximates the best multi-player reach-avoid objective achieved by joint policies with global state feedback in (11).

V. MULTI-PLAYER MOTION PLANNING

We evaluate Algorithm 3's efficacy at finding collision-free trajectories in a multi-agent motion planning problem on a grid-world MDP. The grid world has dimensions $M_R \times M_C$ and is executed for T number of steps for N players. Players receive randomized initial and final assigned target squares on the far left and far right columns of the grid world, respectively, and attempt to reach their randomly assigned target squares while avoiding each other. To ensure that the players must deal with collisions, the player assigned the top-left initial state is also assigned the bottom-right destination. Each player's action is to go up, down, left, or right subjected to world boundaries. Each action has an associated stochasticity p : instead of reaching the action's target destination deterministically, the target is reached with probability $1 - p \in [0, 1]$ and a neighbor at random is reached with probability p . We evaluate Algorithm 3 output

| Figure | State size ($M_R M_C$) | Horizon (T) | Players (N) | Stochasticity (p) | Trial size (K) |
|--------|-----------------------------|--------------------|--------------------|--------------------------|-----------------------|
| 2 | 40 | 15 | 3 | $[0.75, 0.95]$ | 50 |
| 3 | $[30, 70]$ | 20 | 2 | 0.95 | 50 |

TABLE I: Simulation hyper-parameters.

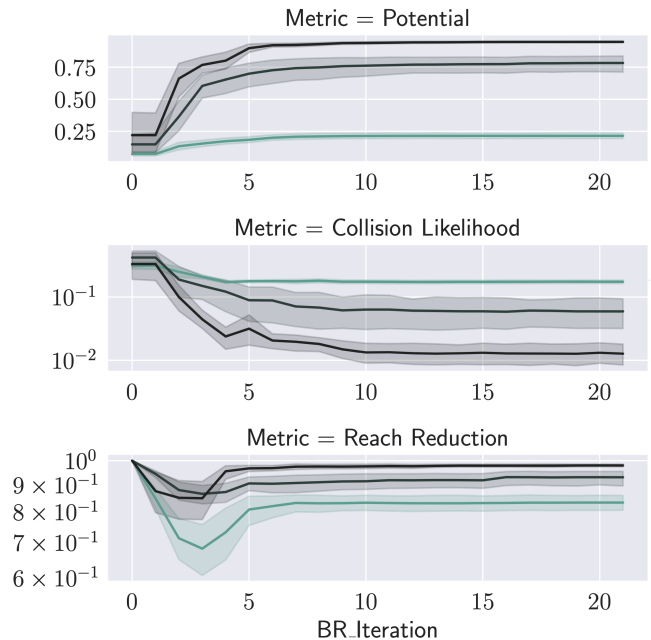


Fig. 1: Reach-avoid metrics over different action stochasticity values (green to black and corresponds to $p = 0.95$ to $p = 0.75$).

optimality and computation efficiency in two test scenarios via K Monte carlo trials, the hyper-parameters of each test scenario is given in Table I and the results are shown in Figures 1 and 2.

Reach-avoid performance. In Figure 1, we visualize three metrics over each best response iteration K in subplots from top to bottom: 1) potential value: the reach-avoid probability (7), 2) collision likelihood: the collision probability among any two players at any time $t \in [T]$,

$$\mathbb{E} \left[1 - \prod_{t=0}^T \prod_{i,j \in [N]} Y_{ij}(s_i(t), s_j(t)) \mid \tau_j \sim h_j(\pi_j^k), \forall j \in [N] \right], \quad (23)$$

and 3) reach reduction: the probability of all players reaching their destination at time T , divided by the probability that each player reaches their destination on their shortest path,

$$\frac{\mathbb{E} \left[\prod_{j \in [N]} X_j(s_j(T)) \mid \tau_j \sim h_j(\pi_j^k), \forall j \in [N] \right]}{\prod_{j \in [N]} \mathbb{E} \left[X_j(s_j(T)) \mid \tau_j \sim h_j(\pi_j^*), \forall j \in [N] \right]}, \quad (24)$$

where π_j^* denotes player j 's shortest path policy. We observe that on average, all three metrics stabilize to their asymptotic values between 5 and 10 best response steps. Furthermore, because each player always initiates the iterative best response with their shortest path policy π_j^* , the initial reduction in reaching player targets is always none. However, these policies also incur collision likelihoods averaging around 50%. As players maneuver around each other to reduce this collision likelihood, the reach reduction first decreases

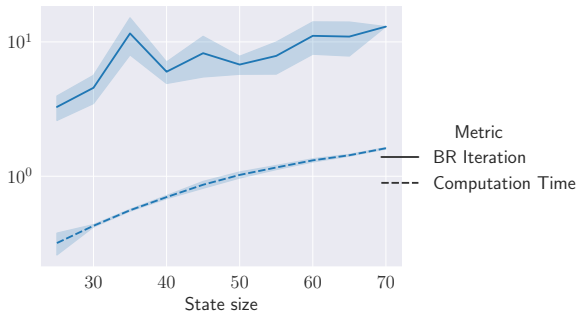


Fig. 2: Computation time (seconds) and best response iteration (k) vs state sizes.

but then gradually increases, while the collision likelihood decreases asymptotically. We note that higher action stochasticity p leads to more unavoidable collisions. This is reflected by the asymptotic trends observed in Figure 1.

Computation Efficiency. Next, we evaluate the computation efficiency as a function of the number of players and state size. We visualize two metrics: 1) the computation time for a best response iteration and 2) the number of best response iterations before the change in potential value decreases below $1e-5$. The results for different state sizes is shown in Figure 2. We observe that the computation time increases approximately linearly over the increasing state space, and this observation is true whether or not we use the approximation (22) as described in Section IV to reduce number of state densities tracked. We use $\epsilon = (1e-2)^{0.75t+3}$ where t is the MDP time step. Additionally, the number of best response iterations also increases approximately linearly over increasing state size.

The average computation time for [2, 3, 4] players over 10 monte carlo simulations is [0.72, 55, 1660] seconds. We observe that the computation does scale exponentially as the number of players increases. With 4 players taking up to ~ 25 minutes for one best response iteration. This exponential increase in player complexity can be mitigated via distributed computing and distributed learning dynamics, which we will explore in future research. We did not observe significant changes in the number of best response iterations between different number of players.

VI. CONCLUSION

Towards a traffic management framework for heterogeneous vehicles, we formulated a game-theoretic extension of a single agent reach-avoid MDP, provided a game-theoretic interpretation of the optimal decision for resource-sharing players, and simulation verified a multi-player extension of multiplicative dynamic programming for finding Nash equilibrium policies.

REFERENCES

[1] L. A. Garrow, B. German, N. T. Schwab, M. D. Patterson, N. Mendonca, Y. O. Gawdiak, and J. R. Murphy, "A proposed taxonomy for advanced air mobility," in *AIAA Aviation 2022 Forum*, 2022, p. 3321.

[2] R. Goyal, C. Reiche, C. Fernando, and A. Cohen, "Advanced air mobility: Demand analysis and market potential of the airport shuttle and air taxi markets," *Sustainability*, vol. 13, no. 13, p. 7421, 2021.

[3] R. E. Weibel and R. J. Hansman, "Safety considerations for operation of unmanned aerial vehicles in the national airspace system," Tech. Rep., 2006.

[4] J. P. McGee, A. S. Mavor, and C. D. Wickens, *Flight to the future: Human factors in air traffic control*. National Academies Press, 1997.

[5] R. Shone, K. Glazebrook, and K. G. Zografos, "Applications of stochastic modeling in air traffic management: Methods, challenges and opportunities for solving air traffic problems under uncertainty," *Euro. J. Oper. Res.*, pp. 1–26, 2021.

[6] D. Hentzen, M. Kamgarpour, M. Soler, and D. González-Arribas, "On maximizing safety in stochastic aircraft trajectory planning with uncertain thunderstorm development," *Aerospace Science and Technology*, vol. 79, pp. 543–553, 2018.

[7] D. Calderone and S. Sastry, "Markov decision process routing games," in *Int'l Conf. Cyber-Phys. Syst.*, 2017, pp. 273–279.

[8] A. P. Vinod, S. Yamazaki, A. Chakrabarty, N. Yoshikawa, and S. Di Cairano, "Aircraft approach management using reachability and dynamic programming," in *2024 American Control Conference (ACC)*. IEEE, 2024, pp. 318–324.

[9] A. Abate, M. Prandini, J. Lygeros, and S. Sastry, "Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems," *Automatica*, 2008.

[10] S. Summers and J. Lygeros, "Verification of discrete time stochastic hybrid systems: A stochastic reach-avoid decision problem," *Automatica*, vol. 46, no. 12, pp. 1951–1961, 2010.

[11] A. Vinod and M. Oishi, "Stochastic reachability of a target tube: Theory and computation," *Automatica*, 2021.

[12] N. Schmid, M. Fochesato, S. H. Li, T. Sutter, and J. Lygeros, "Computing optimal joint chance constrained control policies," *arXiv preprint arXiv:2312.10495*, 2023.

[13] 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, 2016.

[14] J. F. Fisac and S. S. Sastry, "The pursuit-evasion-defense differential game in dynamic constrained environments," in *2015 54th IEEE conference on decision and control (CDC)*. IEEE, 2015, pp. 4549–4556.

[15] M. Chen, Q. Hu, C. Mackin, J. F. Fisac, and C. J. Tomlin, "Safe platooning of unmanned aerial vehicles via reachability," in *2015 54th IEEE conference on decision and control (CDC)*. IEEE, 2015, pp. 4695–4701.

[16] K. Margellos and J. Lygeros, "Hamilton-jacobi formulation for reach-avoid differential games," *IEEE Transactions on automatic control*, vol. 56, no. 8, pp. 1849–1861, 2011.

[17] J. F. Fisac, E. Bronstein, E. Stefansson, D. Sadigh, S. S. Sastry, and A. D. Dragan, "Hierarchical game-theoretic planning for autonomous vehicles," in *2019 International conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 9590–9596.

[18] M. J. Smith, "The existence, uniqueness and stability of traffic equilibria," *Transp. Res. Part B: Method.*, pp. 295–304, 1979.

[19] T. Roughgarden, *Selfish routing and the price of anarchy*. MIT press, 2005.

[20] S. H. Li, D. Calderone, and B. Açıkmeşe, "Congestion-aware path coordination game with markov decision process dynamics," *IEEE Control Systems Letters*, vol. 7, pp. 431–436, 2022.

[21] D. Monderer and L. S. Shapley, "Potential games," *Games and economic behavior*, 1996.

[22] S. H. Li, Y. Yu, N. I. Miguel, D. Calderone, L. J. Ratliff, and B. Açıkmeşe, "Adaptive constraint satisfaction for markov decision process congestion games: Application to transportation networks," *Automatica*, vol. 151, p. 110879, 2023.

[23] J. R. Marden, "State based potential games," *Automatica*, vol. 48, no. 12, pp. 3075–3088, 2012.

APPENDIX

A. Proof of Proposition 1

Proof. For each $s \in \mathcal{S}^N$, we prove the following recursive identity for (15): if $V_{t+1}^\pi(s)$ satisfies (17), then $V_t^\pi(s)$ satisfies (17).

If $V_{t+1}^\pi(s)$ satisfies (17), it is equivalent to

$$V_{t+1}^\pi(s) = \sum_{\tau} R_{t+1}^T((s, \tau)) \prod_j \prod_{\hat{t}=t+2}^T \mathbb{P}[\tau_j(\hat{t}+1)|\tau_j(\hat{t}), \pi_j], \quad (25)$$

for all $s \in \mathcal{S}^N$, where the product $\prod_{\hat{t}=t}^T \mathbb{P}[\tau_j(\hat{t}+1)|\tau_j(\hat{t}), \pi_j]$ is the probability of realizing the trajectory $\tau_j(t+1), \dots, \tau_j(T)$ when $\tau_j(t+1) = s_j$ for all $j \in [N]$. We use (25) to define $V_{t+1}^\pi(\hat{s})$ and (15) to evaluate $V_t^\pi(s)$ as

$$V_t^\pi(s) = \prod_{j,\ell} Y(s_j, s_\ell) \sum_{\hat{s}_1, \dots, \hat{s}_N} \prod_j \mathbb{P}[\hat{s}_j | s_j, \pi_j] \sum_{\tau_{t+2}} R_{t+1}^T((\hat{s}, \tau_{t+2})) \prod_{\hat{t}=t+2}^T \prod_j \mathbb{P}[\tau_j(\hat{t}+1)|\tau_j(\hat{t}), \pi_j] \quad (26)$$

We can combine the summations $\sum_{\hat{s}}$ and $\sum_{\tau_{t+2}}$ to $\sum_{\tau_{t+1}}$ by noting that $\sum_{\hat{s}} \sum_{\tau_{t+2}}$ is equivalent to a single summation over $(\hat{s}, \tau_{t+2}) \in \mathcal{S}^{N(T-t)}$, which we define as τ_{t+1} . Under this definition of τ_{t+1} , $\prod_j \mathbb{P}[\hat{s}_j | s_j, \pi_j] \prod_j \prod_{\hat{t}=t+2}^T \prod_j \mathbb{P}[\tau_j(\hat{t}+1)|\tau_j(\hat{t}), \pi_j] = \prod_{\hat{t}=t+1}^T \prod_j \mathbb{P}[\tau_j(\hat{t}+1)|\tau_j(\hat{t}), \pi_j]$.

For the trajectory (s, τ_{t+1}) , the reach-avoid objective $R_{t+1}^T((s, \tau_{t+1}))$ also satisfies the recursive relationship

$$R_t^T((s, \tau_{t+1})) = \prod_{j,\ell} Y(s_j, s_\ell) R_{t+1}^T(\tau_{t+1}).$$

Therefore, we can conclude that for all joint states $s \in \mathcal{S}^N$.

$$V_t^\pi(s) = \sum_{\tau_{t+1}} R_t^T((s, \tau_{t+1})) \prod_{\hat{t}=t+1}^T \prod_j \mathbb{P}[\tau_j(\hat{t}+1)|\tau_j(\hat{t}), \pi_j]. \quad (27)$$

Finally, since V_T^π satisfies the expectation evaluation (17), $V_{T-1}^\pi, \dots, V_0^\pi$ all satisfies (17). \square