Context-Aware Inefficiency of Self-Interested Agents in UAV Wildfire Fighting*

Niket Parikh† and Rupal Nigam‡
Aerospace Engineering, University of Illinois Urbana-Champaign, Urbana, IL 61820

Max Z. Li §
Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109

Huy T. Tran¶
Aerospace Engineering, University of Illinois Urbana-Champaign, Urbana, IL 61820

Multi-agent systems are present in many real-world aerospace applications, such as firefighting with unmanned aerial vehicles, making the development of autonomous agents that can solve complex tasks an active area of research. We focus on systems where a set of self-interested agents interact with each other and their environment in a manner that affects a desired social outcome, such as minimizing the number of burned trees in the firefighting example. Given such systems, our goal is to measure and understand how these behaviors might induce inefficiencies in the social outcome—we specifically aim to do so in a manner that considers the *context within which these inefficiencies occur.* Understanding this context could, for example, help a system operator intervene when the system approaches a situation that is predicted to have high inefficiency or help a system designer ensure such situations are unlikely to occur. We address this problem by introducing a state-dependent price of anarchy (sPoA) metric that can be used within a sequential decision-making problem formalized as a Markov game. We then outline a computational approach for estimating this metric and understanding observed trends from empirical experiments. We demonstrate our approach in a multi-agent unmanned aerial vehicle firefighting task. Our empirical findings show that sPoA identifies non-trivial trends in inefficiencies due to self-interested behaviors across different states, highlighting the importance of considering state dependence in this problem.

I. Introduction

Design and analysis of multi-agent systems, which consist of multiple agents co-existing in a shared environment, is a pivotal research direction for many real-world aerospace applications, such as air traffic management [1–3], urban air mobility (UAM) [4, 5], unmanned aerial vehicle (UAV) collision avoidance [6, 7], UAV-based wildfire fighting [8–10] and search and rescue [11, 12], and satellites [13–15]. Many of these systems include some system-level social objective that we may wish the agents to optimize [16]. For example, we may hope for a team of UAVs to minimize the total number of burnt trees in a wildfire fighting scenario or a set of autonomous vehicles to minimize total congestion in a UAM system. However, the agents operating in these systems may act according to individual objectives that differ from the overall social objective. Returning to the UAV firefighting example, as wildfires grow and cross into areas under different jurisdictions, firefighting agents may have different areas of responsibility, which may result in slight different (and potentially conflicting) objectives. Self-interested behaviors are also likely to occur in a UAM system, where vehicles could aim to minimize their own travel time irrespective of the impact on overall congestion. It is therefore important to be able to measure and understand the inefficiencies induced by self-interested behaviors, as this understanding could improve our ability to design efficient and equitable multi-agent systems.

*A version of this paper titled "Inefficiency of Self-Interested Behaviour in Markov Games: State-Dependent Price of Anarchy" (Paper AIAA-2025-1929) was presented at the 2025 AIAA SciTech Forum, Orlando, FL, January 6-10, 2025.

†Corresponding author: niketnp2@illinois.edu, Graduate Student, The Grainger College of Engineering, Department of Aerospace Engineering, University of Illinois Urbana-Champaign

‡Graduate Student, The Grainger College of Engineering, Department of Aerospace Engineering, University of Illinois Urbana-Champaign

§Assistant Professor, College of Engineering, Department of Aerospace Engineering, Department of Civil and Environmental Engineering, Department of Industrial and Operations Engineering, University of Michigan

¶Assistant Professor, The Grainger College of Engineering, Department of Aerospace Engineering, University of Illinois Urbana-Champaign

Game theory is a common framework for modeling multi-agent systems and offers a well-known metric, the coordination ratio [17], now known as price of anarchy (PoA), for measuring inefficiencies caused by self-interested behavior. PoA is defined as the ratio between the worst efficiency of a Nash equilibrium and the optimal efficiency in a game, where efficiency is computed with respect to a social objective function [18]. Since its introduction, PoA has received great attention in the game theory community and has rich literature devoted to the concept. For example, [19] models noncooperative satellite range scheduling problems using games, proposes an algorithm that tractably converges to a solution in such games, and uses PoA to compare the outcome in different practical scenarios. [20] studies the space debris removal problem by using PoA to compare a centralized solution approach to a decentralized approach with self-interested agents. They show that decentralized decision-making can be significantly costly and thus recommend minimizing the number of competing agents in this problem. [21] analyzes PoA for traffic networks where the social objective is to route all traffic so as to achieve minimum latency (sum of all travel times) but users minimize personal latency. Their findings can help design efficient networks.

However, most prior work on PoA has been restricted to single-stage static game settings [22], where players choose their actions simultaneously, act only once, and the game effectively has just one state. This formulation is limited in its ability to model many real-world aerospace systems, like UAV firefighting and UAM, as agents typically act sequentially over time in a stochastically evolving environment. It is therefore of practical importance to consider extensions of PoA that can account for sequential decision-making under uncertainty. Furthermore, the inefficiencies caused by self-interested behaviors will likely depend on the state of the system. Thus, there is a need for state-dependent—i.e., context-aware—measurements of such inefficiencies. For example, being able to predict the inefficiency of particular states could enable a system-level operator to intervene and drive the behaviors of self-interested agents to more desirable ones when the system reaches a state predicted to have high inefficiency. More specifically, in the UAV firefighting scenario, a state-level government agency could intervene and provide additional resources or take over management of operations if it was predicted that locally managed operations for a given fire would result in highly inefficient outcomes. Prior work [23], [24] deals with the presence of sequential decisions and uncertainties by extending PoA to a more general class of games known as Markov games (MGs). However, these methods then calculate an expectation of the social objective function over a state distribution defined by the MG—we refer to this approach as a state-aggregated one. While this approach considers context through the state distribution used, it does not allow one to predict or understand the relationship between *specific states* and the inefficiency of agents, and thus would not enable the state-dependent intervention example discussed above.

A. Main Contributions

Our key idea to measure context-aware inefficiencies in multi-agent systems is to incorporate explicit state dependence into PoA through the definition of a state-dependent PoA (sPoA). We then propose to leverage tools from multi-agent reinforcement learning (MARL) to approximate this metric in complex environments involving sequential decision-making under uncertainty. We demonstrate our method in a UAV firefighting case study and empirically show the importance of incorporating explicit state dependence when assessing potential inefficiencies due to self-interested agents. We also provide a set of heuristic metrics to help understand possible underlying reasons for the observed sPoA trends in our case study.

B. Organization of the Paper

The remainder of our paper is organized as follows. Section II introduces our UAV firefighting case study. Section III then presents relevant background concepts and terminology for our work. Section IV presents our proposed approach for measuring the inefficiency of self-interested agents. Finally, Section V presents our experimental setup and results and Section VI presents general conclusions.

II. UAV Firefighting: A Case Study

Over the past few years, there has been burgeoning interest in investigating the use of UAVs to enhance wildfire management [25]. Within this broad effort, studies on the utilization of artificial intelligence to support active-fire tasks, comprising of fire detection, monitoring, and control, form a growing body of work [26, 27]. Motivated by these examples, we focus on a UAV firefighting case study in this paper. Prior work has proposed reinforcement learning (RL) for path planning of multiple UAVs to monitor [28–31] and control [32] wildfires. These works design control methods for a (fully) cooperative group of UAVs. We extend the ongoing investigation of autonomous UAVs for firefighting

by considering how self-interested behaviors, which could arise among such a group of UAVs in practice, affect the firefighting outcome.

Specifically, we consider a scenario where autonomous UAVs are deployed from different regions (e.g., different townships) or agencies to help combat a wildfire by dropping fire retardant on burning trees. We define the social objective to be minimizing the total number of burnt trees—this objective could represent, for example, the goal of the state or national organization overseeing the overall region in which the fire is located. Despite this social objective, it is possible that the UAVs deployed prioritize fire mitigation in different regions or areas of responsibility, such as their own township or areas with certain topographic features. In the USA, inter-agency coordination is managed through a hierarchical structure from national groups (like NMAC) down to federal, state, tribal, and local agencies [33]. Jurisdictional responsibility is primarily determined by the wildfire's origin, though formal agreements, contracts, and compacts also contribute to defining resource sharing, roles, and cost allocation [34]. Resource deployment prioritizes either the closest unit or pre-identified areas of responsibility. While all agencies share the same goal of wildfire suppression, coordination can face friction due to budget concerns, differing organizational missions, and overlapping responsibilities [35]. Self-interested behaviors may also arise due to implicit biases present within the operators or system designers who created the UAV decision policies, different operating procedures used by various townships, or different equipage. A related study on wildfire management [36] finds evidence that "indicates that decisions made by fire managers favor particular groups and materially affect outcomes for those groups," based on data for fire spread in the western United States. For example, they "find evidence that fire managers commit greater effort to combat the spread of fires toward high-value homes as well as that they preferentially protect wealthier neighborhoods."

Thus, we model self-interested UAVs in this scenario as those which preferentially protect specific regions, which we call "selfish regions." As mentioned in Section I, identifying states where these self-interested UAVs may produce high inefficiencies could, for example, help a state-level government agency know when to intervene and provide additional resources or take over management of operations. Such knowledge could also help one design a more efficient overall system through policy or long-term resource allocation. We model the UAVs in our scenario as autonomous agents that act according to a decision policy designed to maximize some reward function (e.g., one which associates a cost with the number of burnt trees). We then model the problem of determining these decision policies as one of sequential decision-making under uncertainty, due to the fact that wildfires spread in a highly uncertain manner and typically do so over an extended period of time. Finally, we use MARL to optimize these decision policies, as it has recently shown promising results for such problems [37–39]. The next section formally introduces key concepts for MARL and PoA. Section V.A formalizes our UAV firefighting model within a MARL framework.

III. Preliminaries

This section introduces relevant technical background and concepts for our work. We denote by $\Delta(X)$ the probability simplex over set X, i.e., the set of all possible distributions over the elements of set X. Variables denoting joint quantities (such as joint actions) are written in boldface.

A. Multi-Agent Reinforcement Learning

RL is a machine learning paradigm where an agent learns to accomplish its objective using data collected from repeated interactions with the environment. The underlying stochastic process in (single-agent) RL is modeled as a discrete-time MDP, formally defined as a tuple $\langle \mathcal{S}, \mathcal{A}, P, R, d_0, \gamma \rangle$, where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ denotes the action space of the agent, $P(s'|s,a) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1]$ denotes the probability of transitioning into state s' when the agent takes action a in state s, $R(s,a,s') : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ denotes the reward received by the agent upon taking action a in state s and the environment transitioning to state s', $d_0(s) : \mathcal{S} \to \Delta(\mathcal{S})$ denotes the initial state distribution, and $\gamma \in [0,1)$ denotes the discount factor. At each time step $t \in \{0,1,2,\dots\}$, the agent takes an action a_t given the current state s_t, after which the environment transitions to state $s_{t+1} \sim P(\cdot|s_t, a_t)$ and the agent obtains a reward $R(s_t, a_t, s_{t+1})$. The agent can observe the state at every time step, i.e., the environment is fully observable.

The decision-making strategy of the agent is denoted by policy $\pi(\cdot|s) : \mathcal{S} \to \Delta(\mathcal{A})$. Given a policy π, the value function $V^\pi(s) : \mathcal{S} \to \mathbb{R} := \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1}) | s_0 = s\right]$ denotes the expected discounted sum of rewards, or the expected return, accrued by the agent when starting in state s and having future transitions driven by actions sampled from policy π. Policy evaluation refers to the problem of finding the value function of a given policy. The goal in RL is to find the policy that has the maximum value at every state $s \in \mathcal{S}$ among the set of possible policies. That is, we aim to

find the optimal policy π^* that satisfies,

$$\pi^* = \arg\max_{\pi \in \Pi} V^{\pi}(s), \ \forall s \in \mathcal{S}, \tag{1}$$

where Π is the set of all possible policies in the MDP. For more details, we refer the reader to [40].

To accommodate multiple agents in the MDP framework, the single-agent definition is augmented with the set of agents and their action spaces and reward functions, both of which may differ for each agent. This framework is formalized as a Markov game (MG). A multi-agent MDP (MMDP) captures the special case where all agents have a shared reward function and receive the same reward at each time step. As our method uses MG and MMDP formulations, we formally define both below.

1. Markov Game

An MG [41], $\mathcal{G}$, is defined as a tuple $\langle \mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}_{i\in\mathcal{N}}, P, \{R^i\}_{i\in\mathcal{N}}, d_0, \gamma \rangle$, where $\mathcal{N} = \{1, \ldots, N\}$ denotes the set of agents, $\mathcal{S}$ denotes the state space, $\mathcal{A}^i$ denotes the action space of the i^{th} agent, $P(s'|s, \mathbf{a}) : \mathcal{S} \times \boldsymbol{\mathcal{A}} \times \mathcal{S} \to [0, 1]$ denotes the probability of transitioning into state s' when the agents take a joint action $\mathbf{a} \in \boldsymbol{\mathcal{A}} = \prod_{i=1}^{N} \mathcal{A}^i$ in state s, $R^i(s, \mathbf{a}, s') : \mathcal{S} \times \boldsymbol{\mathcal{A}} \times \mathcal{S} \to \mathbb{R}$ denotes the reward received by the i^{th} agent upon the agents taking joint action $\mathbf{a}$ in state s and the environment transitioning to state s', $d_0(s) : \mathcal{S} \to \Delta(\mathcal{S})$ denotes the initial state distribution, and $\gamma \in [0, 1)$ denotes the discount factor. At each time step $t \in \{0, 1, 2, \ldots\}$, each agent takes an action a^i given the current state s_t, after which the environment transitions to state $s_{t+1} \sim P(\cdot|s_t, \mathbf{a}_t)$ and each agent obtains a reward $R^i(s_t, \mathbf{a}_t, s_{t+1})$. Each agent can observe the state at every time step.

The policy of agent i is denoted by $\pi^i(\cdot|s) : \mathcal{S} \to \Delta(\mathcal{A}^i)$ and the resulting joint policy is denoted by $\boldsymbol{\pi}(\cdot|s) : \mathcal{S} \to \Delta(\boldsymbol{\mathcal{A}}) \coloneqq \prod_{i=1}^{N} \pi^i(a^i|s)$. Given a joint policy $\boldsymbol{\pi}$, the value function $V_i^{\boldsymbol{\pi}}(s) : \mathcal{S} \to \mathbb{R} \coloneqq \mathbb{E}_{\boldsymbol{\pi}}\big[\sum_{t=0}^{\infty} \gamma^t R^i(s_t, \mathbf{a}_t, s_{t+1})|s_0 = s\big]$ denotes the expected return for agent i when starting at state s. Naively, the goal for the agent i is still to find a policy $\pi^{i,*}$ that results in maximum value $V_i^{\boldsymbol{\pi}}$, similar to the single-agent case. However, since the value function now also depends on the policies of other agents, there may not be a set of agent policies that simultaneously maximizes the value function of each agent. Depending on the assumptions made on agent behavior, a variety of optimality notions have therefore been proposed. The most commonly used one is based on the concept of Nash equilibrium. A joint policy $\boldsymbol{\pi}^{NE}(\cdot|s)$ is a Nash equilibrium if, for every agent i, its current policy $\pi^{NE,i}$ maximizes $V_i^{\boldsymbol{\pi}}$, given that the other agents' policies are fixed. There exists a Nash equilibrium for every (finite) MG [42] and in general, there may exist multiple Nash equilibria. For more details on MGs, we refer the reader to [43].

2. Multi-Agent Markov Decision Process

An MMDP is a special case of an MG where the agents have an identical reward function, i.e., $R^1 = \ldots = R^N$. This can be interpreted as a fully cooperative scenario where all the agents learn to act optimally with respect to a shared objective. More formally, an MMDP [44], $\mathcal{M}$, is defined as a tuple $\langle \mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}_{i\in\mathcal{N}}, P, R, d_0, \gamma \rangle$, where the notation carries over from the definition of an MG.

The (shared) goal for the agents in an MMDP is to find the optimal policies $\{\pi^{i,*}\}_{i=1}^{N}$ such that the resulting joint policy $\boldsymbol{\pi}^*$ maximizes the value function $V^{\boldsymbol{\pi}}$, i.e.,

$$\boldsymbol{\pi}^* = \arg\max_{\boldsymbol{\pi} \in \boldsymbol{\Pi}} V^{\boldsymbol{\pi}}(s), \ \forall s \in \mathcal{S}, \tag{2}$$

where $\boldsymbol{\Pi}$ is the set of possible joint policies in the MMDP and $V^{\boldsymbol{\pi}}(s) : \mathcal{S} \to \mathbb{R} \coloneqq \mathbb{E}_{\boldsymbol{\pi}}\big[\sum_{t=0}^{\infty} \gamma^t R(s_t, \mathbf{a}_t, s_{t+1})|s_0 = s\big]$.

3. Independent Proximal Policy Optimization

Policy gradient (PG) methods are a common approach for solving MARL problems [43]. A commonly used PG algorithm is independent proximal policy optimization (IPPO) [45]. IPPO is a multi-agent variant of the popular proximal policy optimization (PPO) algorithm [46]. IPPO extends PPO to the multi-agent setting by having each agent learn a PPO policy that maximizes their individual reward, while treating the rest of the agents as part of the environment. Despite the violation of the Markov property in such a scenario, IPPO has shown strong empirical results in many multi-agent settings like the StarCraft Multi-Agent Challenge (SMAC) [47]. Next, we give a high-level overview of the IPPO algorithm. Full details have been omitted, and we refer the interested reader to [43] for a detailed exposition.

Consider an MG $\mathcal{G}$, a parameterized policy $\pi^i_{\theta_i}$ for each agent i, and the joint set of policy parameters denoted by $\boldsymbol{\theta} = \{\theta_i\}_{i=1}^{N}$. Most commonly, the policy π^i is parametrized as a neural network, where the parameters θ_i represent

the weights of the neural network. For agent i, an independent PG method aims to find the policy parameters, θ_i, that maximize the objective function, $J_i(\boldsymbol{\pi}_\theta) = \mathbb{E}_{s\sim d_0}\left[V_i^{\boldsymbol{\pi}}(s)\right]$, via gradient ascent. The gradient of the objective, denoted by $\nabla_{\theta_i} J_i(\pi_\theta)$, can be computed by using the multi-agent policy gradient theorem [43]. A commonly used version of the policy gradient is given by,

$$\nabla_{\theta_i} J_i(\boldsymbol{\pi}_{\boldsymbol{\theta}}) = \frac{1}{1-\gamma}\mathbb{E}_{(s,\mathbf{a})\sim d^{\boldsymbol{\pi}}}\left[\nabla_{\theta_i} \log \pi^i_{\theta_i}(a^i|s) A_i^{\boldsymbol{\pi}}(s,\mathbf{a})\right], \tag{3}$$

where $A_i^{\boldsymbol{\pi}}(s,\mathbf{a})$ is the advantage function of agent i, which measures how much better it is to take action $\mathbf{a}$ in state s compared to an action sampled from joint policy $\boldsymbol{\pi}$, and $d^{\boldsymbol{\pi}}$ is the state-action occupancy distribution of $\boldsymbol{\pi}$. Finally, the policy parameter update for agent i at the k^{th} iteration of gradient ascent is given by,

$$\theta_{i,k} = \theta_{i,k-1} + \alpha \nabla_{\theta_i} J_i(\boldsymbol{\pi}_{\boldsymbol{\theta}}), \tag{4}$$

where α is the learning rate. PPO (and thus, IPPO) uses a modified version of the objective function, $J_i(\theta_i)$, to improve training stability. The IPPO objective for policy optimization of agent i is given by,

$$\mathcal{L}_i(\boldsymbol{\theta}) = \mathbb{E}_{(s,\mathbf{a})\sim d^{\boldsymbol{\pi}}}\left[\min\left(\frac{\pi^i_{\theta_i}(a^i \mid s)}{\pi^i_{\theta_{i,old}}(a^i \mid s)} A_i^{\boldsymbol{\pi}}(s,\mathbf{a}), \text{clip}\left(\frac{\pi^i_{\theta_i}(a^i \mid s)}{\pi^i_{\theta_{i,old}}(a^i \mid s)}, 1-\epsilon, 1+\epsilon\right) A_i^{\boldsymbol{\pi}}(s,\mathbf{a})\right)\right], \tag{5}$$

where ϵ is the clipping parameter (typically ranging from 0.1 to 0.3), $\theta_{i,old}$ are the policy parameters at the end of previous iteration, and $A_i^{\boldsymbol{\pi}}(s,\mathbf{a})$ is computed using generalized advantage estimation (GAE) [48].

B. Price of Anarchy

Consider a (single-stage) static game consisting of N players with player i's strategy given by $\pi^i \in \Pi^i$, where Π^i is the set of possible strategies for player i. Let $\boldsymbol{\pi} = \{\pi^1, \ldots, \pi^N\} \in \boldsymbol{\Pi}$ denote the joint strategy, where $\boldsymbol{\Pi}$ is the set of possible joint strategies. The game assigns a utility $R^i : \boldsymbol{\Pi} \to \mathbb{R}$ to player i for a given joint strategy. Finally, the goal of each player is to choose a strategy that maximizes their reward. Note that such a game is equivalent to an MG with time horizon one and a singleton state space.

Let the efficiency of a joint strategy be measured by a social cost $C(\boldsymbol{\pi}) : \boldsymbol{\Pi} \to \mathbb{R}$ (lower values are better) or a social welfare $W(\boldsymbol{\pi}) : \boldsymbol{\Pi} \to \mathbb{R}$ (higher values are better) function. Henceforth, we will refer to a social cost or welfare function as the social objective function and only make the distinction clear where necessary. For a given social objective, [18] defines PoA as the ratio of the efficiency of the worst Nash equilibrium to the efficiency of the optimal solution. That is, the PoA of a game with respect to a social cost function C is given by,

$$\frac{\max_{\boldsymbol{\pi}^{NE}\in\boldsymbol{\Pi}^{NE}} C(\boldsymbol{\pi}^{NE})}{\min_{\boldsymbol{\pi}\in\boldsymbol{\Pi}} C(\boldsymbol{\pi})} \geq 1, \tag{6}$$

and the PoA of a game with respect to a social welfare function W is given by,

$$\frac{\max_{\boldsymbol{\pi}\in\boldsymbol{\Pi}} W(\boldsymbol{\pi})}{\min_{\boldsymbol{\pi}^{NE}\in\boldsymbol{\Pi}^{NE}} W(\boldsymbol{\pi}^{NE})} \geq 1, \tag{7}$$

where $\boldsymbol{\Pi}^{NE}$ is the set of Nash equilibrium joint strategies.

IV. State-Dependent Price of Anarchy

In this section, we present our proposed methodology for context-aware measurement of the inefficiency of self-interested agents. We first define our proposed metric, named sPoA (Section IV.A), and then present our approach for approximating this metric (Section IV.B). Our overall process is summarized in Figure 1.

A. State-Dependent Price of Anarchy

Consider an MG, $\mathcal{G} = \langle \mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}_{i\in\mathcal{N}_\mathcal{G}}, P, \{R^i\}_{i\in\mathcal{N}_\mathcal{G}}, d_0, \gamma\rangle$. For a given (state-dependent) social objective function, we define the sPoA of $\mathcal{G}$ at a state $s \in \mathcal{S}$ as the ratio of the efficiency of the worst Nash equilibrium to the

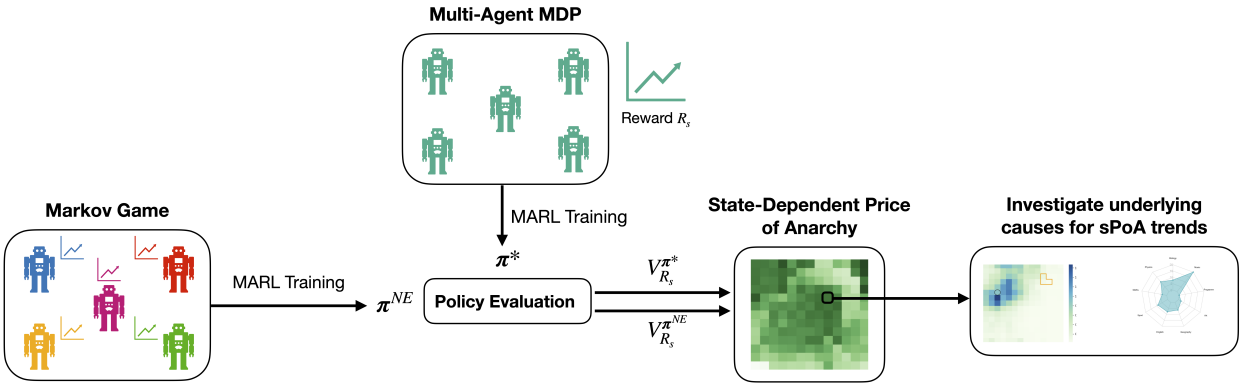


Fig. 1 High-level view of our proposed approach for understanding the inefficiency of self-interested agents.

optimal efficiency when starting from state s. That is, the sPoA of an MG with respect to a social welfare function $W(s, \pi) : \mathcal{S} \times \mathbf{\Pi} \to \mathbb{R}$ is given by the function,

$$sPoA(s) : \mathcal{S} \to [1, \infty) := \frac{\max_{\pi \in \mathbf{\Pi}} W(s, \boldsymbol{\pi})}{\min_{\boldsymbol{\pi}^{NE} \in \mathbf{\Pi}^{NE}} W(s, \boldsymbol{\pi}^{NE})}, \tag{8}$$

and sPoA of an MG with respect to a social cost function $C(s, \pi) : \mathcal{S} \times \mathbf{\Pi} \to \mathbb{R}$ is given by the function,

$$sPoA(s) : \mathcal{S} \to [1, \infty) := \frac{\max_{\boldsymbol{\pi}^{NE} \in \mathbf{\Pi}^{NE}} C(s, \boldsymbol{\pi}^{NE})}{\min_{\boldsymbol{\pi} \in \mathbf{\Pi}} C(s, \boldsymbol{\pi})}, \tag{9}$$

where $\mathbf{\Pi}^{NE}$ is the set of Nash equilibrium joint policies and $\mathbf{\Pi}$ is the set of all joint policies in $\mathcal{G}$. For an MG, the sPoA function is fully determined by the choice of the social objective function.

Choosing a social objective function can be non-trivial for many multi-agent systems since it must capture the desired system-level outcomes and avoid misspecification to serve as a reliable measure of efficiency. There is no general set of criteria that can guide this choice. While the social objective can be any function, in the absence of a clear candidate, the classic choice is to write it in terms of player utilities or in an MG, value functions with respect to agent rewards. Note that utilities are replaced by value functions instead of agent rewards to account for the extended time horizon in MGs. Two classic social objectives are sum social welfare and max-min social welfare. In an MG, these objectives would be given by $\sum_{i \in \mathcal{N}} V_i^{\boldsymbol{\pi}}(s)$ and $\min_{i \in \mathcal{N}} V_i^{\boldsymbol{\pi}}(s)$, respectively. These objectives effectively measure social outcomes in terms of agent rewards. However, this approach is limiting because the best possible social outcome when a set of N-agents interact does not need to be defined by agent rewards. For example, consider a traffic network where each agent's objective is to minimize personal travel time. If the social objective is to minimize the total number of accidents, there is no clear way to express such an objective as a function of agent rewards.

Instead, we consider an arbitrary social reward function, R_s, that captures desired social preferences and allows for the definition of a more generalized class of social objectives. Given that social reward, we define the social objective function as the value function, $V_{R_s}^{\boldsymbol{\pi}}$, for a joint policy $\boldsymbol{\pi}$ with respect to R_s. Then the optimal efficiency for MG agents is $\max_{\pi \in \mathbf{\Pi}} V_{R_s}^{\boldsymbol{\pi}}$, which is attained when the joint policy is chosen to optimize R_s. This policy is precisely the optimal joint policy of the MMDP generated from the MG $\mathcal{G}$ with $R^i = R_s$ for all agents $i \in \mathcal{N}$; that is, it is the optimal joint policy for MMDP $\mathcal{M} = \langle \mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}_{i \in \mathcal{N}_\mathcal{M}}, P, R_s, d_0, \gamma \rangle$. Thus, for a given non-negative social reward, sPoA of an MG is given by,

$$sPoA(s) = \frac{\max_{\pi \in \mathbf{\Pi}} V_{R_s}^{\boldsymbol{\pi}}(s)}{\min_{\boldsymbol{\pi}^{NE} \in \mathbf{\Pi}^{NE}} V_{R_s}^{\boldsymbol{\pi}^{NE}}(s)}, \tag{10}$$

and for a non-positive social reward, sPoA of an MG is given by,

$$sPoA(s) = \frac{\min_{\boldsymbol{\pi}^{NE} \in \mathbf{\Pi}^{NE}} V_{R_s}^{\boldsymbol{\pi}^{NE}}(s)}{\max_{\pi \in \mathbf{\Pi}} V_{R_s}^{\boldsymbol{\pi}}(s)}. \tag{11}$$

As in the case of the classical PoA metric, higher sPoA values indicate more inefficiency (with respect to the social reward function) due to self-interested behaviors in a multi-agent system.

B. Approximating the State-Dependent Price of Anarchy using MARL

1. Training Agents

Our sPoA formulation (Equations (10) and (11)) requires estimation of the efficiency of the worst Nash equilibrium for the considered MG, $\min_{\boldsymbol{\pi}^{NE} \in \boldsymbol{\Pi}^{NE}} V_{R_s}^{\boldsymbol{\pi}^{NE}}(s)$, and the optimal efficiency for the considered MMDP, $\max_{\boldsymbol{\pi} \in \boldsymbol{\Pi}} V_{R_s}^{\boldsymbol{\pi}}(s)$. To estimate these efficiencies, we first need to obtain a joint policy representing the worst Nash equilibrium in the MG $\mathcal{G}$ and an optimal joint policy for the MMDP $\mathcal{M}$. We use MARL to approximate these policies. Despite a lack of convergence guarantees, modern MARL algorithms have been empirically shown to often converge to some Nash equilibrium. For example, [49] empirically observes that IPPO shows convergence patterns to either a Nash equilibrium joint policy or a mixed equilibrium policy (where individual agent policies belong to different Nash equilibria, albeit the resulting joint policy is not necessarily an equilibrium). IPPO has also demonstrated empirical success in MMDP settings [45]. We therefore use IPPO to train agents for the MG setting, where we assume that the algorithm converges to a Nash equilibrium joint policy $\boldsymbol{\pi}^{NE}$, and the MMDP setting, where we assume that the algorithm converges to an optimal joint policy $\boldsymbol{\pi}^{*}$. As discussed in Section IV.B.2, we will then use $\boldsymbol{\pi}^{NE}$ and $\boldsymbol{\pi}^{*}$ to approximate sPoA for a non-negative social reward as,

$$sPoA(s) \approx \frac{V_{R_s}^{\boldsymbol{\pi}^{*}}(s)}{V_{R_s}^{\boldsymbol{\pi}^{NE}}(s)}, \tag{12}$$

and for a non-positive social reward as,

$$sPoA(s) \approx \frac{V_{R_s}^{\boldsymbol{\pi}^{NE}}(s)}{V_{R_s}^{\boldsymbol{\pi}^{*}}(s)}. \tag{13}$$

Note that our sPoA formulation requires an estimate of the worst Nash equilibrium for MG $\mathcal{G}$. Since we simply assume convergence to any Nash equilibrium for $\boldsymbol{\pi}^{NE}$, we effectively approximate a lower bound on sPoA. Finally, in principle, any MARL algorithm can be used for this process—one which gives better convergence guarantees and performance will result in improved approximations of sPoA. Creating MARL algorithms that provide improved or guaranteed convergence to certain Nash equilibria is an open area of research. We leave such study for future work.

2. Evaluating Policies

Once trained policies are available, their value functions with respect to R_s, $V_{R_s}^{\boldsymbol{\pi}^{NE}}$ and $V_{R_s}^{\boldsymbol{\pi}^{*}}$, have to be estimated. We assume that the MARL task is episodic, where every episode terminates in a finite number of time steps, and use the Monte Carlo (MC) policy evaluation method, where the value at each state is estimated by averaging sampled returns. In this work, we limit ourselves to estimating sPoA for states that are in the support of the initial state distribution to simplify computation requirements; that is, we consider the set of states $\mathcal{S}_0 \coloneqq \{s \in \mathcal{S} | d_0(s) > 0\}$. Algorithm 1 shows the pseudocode for MC policy evaluation. We refer the reader to [40] for more details. The number of episodes, M, is the only hyperparameter in this algorithm. Given a policy π, an upper bound on the absolute expected value estimation error, $\left|\mathbb{E}_{s\sim d_0}\left[\widehat{V}_{\mathcal{M}}^{\pi}(s) - V_{\mathcal{M}}^{\pi}(s)\right]\right|$, and a desired probability value, Hoeffding's inequality [50] can be used to compute the number of episodes required to guarantee that the error bound holds, with probability of at least the desired probability value. After the value functions are obtained, we compute sPoA for each state $s \in \mathcal{S}_0$ using Equation (12) or Equation (13).

V. Experimental Results

In this section, we present results from simulated experiments implementing our proposed methodology. We focus on investigating two main research questions:

1) Does incorporating explicit state dependence into PoA (via sPoA) lead to additional insights into self-interested behavior as compared to a state-aggregated variant of PoA?
2) How can we investigate agent behavior to gain deeper insights into the underlying reasons for observed sPoA trends?

A. UAV Firefighting Environment

All experiments are performed using a UAV firefighting environment that models the UAV firefighting case study described in Section II. The adopted wildfire dynamics model was originally proposed in [51]. We adapt that model to

Algorithm 1 MC Policy Evaluation

Input: Trained joint policy π, MMDP $\mathcal{M}$, number of MC episodes M
Output: Estimate of value function $\widehat{V}^{\pi}_{R_s}(s)$ for states $s \in \mathcal{S}_0$
Initialize `values` array to store value for every initial state
for each state s_1 in $\mathcal{S}_0$ **do**
  Initialize `data` array to store episode returns
  for episode = 1, M **do**
    Initialize episode return $G = 0$
    Reset the environment to state s_1
    Store agent observations $(o_{1,i})_{i=1}^{N}$
    Initialize done flag to False
    while not done **do**
      Sample joint action from trained joint policy $a_t \sim \pi((o_{t,i})_{i=1}^{N})$
      Execute joint action a_t and observe reward r_t, state s_{t+1}, agent observations $(o_{t+1,i})_{i=1}^{N}$, and done flag
      $G \leftarrow G + \gamma^{t-1} r_t$
      if done **then**
        break
      end if
    end while
    Append episode return G to `data` array
  end for
  Store the estimate of value function at state s_1 in `values` array, as the mean of elements in `data` array
end for
Return `values`

an MG framework in a similar manner to [32], but with slight modifications, as discussed below. The environment is available on GitHub* and PyPI†.

1. State Space, Action Spaces, and the Initial State Distribution

The environment is an $n \times n$ grid-world representing a forest, where each cell contains a tree, with N UAV agents. Each tree can be in one of three states defined by $s_{\text{tree}} \in \{0\,(\text{healthy}), 1\,(\text{on fire}), 2\,(\text{burnt})\}$. We model self-interested behaviors by defining a rectangular selfish region SR^i of size $l^i \times b^i$ for each agent $i \in \mathcal{N}$. We assume l^i and b^i to be odd-valued integers to ensure there exists a geometric center for each selfish region, which we denote by $c^i = (c^i_x, c^i_y)$. However, our method would apply equally well to a problem with even-valued l^i and b^i. The agents can move on the grid one cell at a time in either of the four cardinal directions or stay still. The action space for each agent is thus $\mathcal{A}^i = \{0\,(\text{still}), 1\,(\text{north}), 2\,(\text{east}), 3\,(\text{south}), 4\,(\text{west})\}$, for all $i \in \mathcal{N}$. When located in the same cell as a tree on fire, the agent automatically discharges fire retardant. The effect of fire retardant is incorporated into the system dynamics. The agent movements are deterministic; for example, taking the 'north' action will always move the agent one cell upward except when a wall or another agent is present.

The initial position of each agent $i \in \mathcal{N}$ is the geometric center c^i of its selfish region. The initial state contains one initial fire region, modeled as a square region of size $l^{fire} \times l^{fire}$, where every tree in the region is in an 'on fire' state. We assume l^{fire} to be an odd-valued integer to ensure that the initial fire has a central cell denoted by $c^{fire} = (c^{fire}_x, c^{fire}_y)$. The set of selfish regions $\{SR^i\}_{i\in\mathcal{N}}$ and the width of the initial fire l^{fire} are fixed for a given MG $\mathcal{G}$. The initial state distribution d_0 is thus defined by the initial fire location, which is selected uniformly at random from all possible locations within the grid.

2. System Dynamics

At time step t, the environment is in a state s_t characterized by tree states and agent positions, and each agent chooses an action to move or remain still, after which the environment transitions to a new state s_{t+1}, based on the

*Source code for gym-based multi-agent UAV firefighting environment: https://github.com/npnike10/wildfire-environment.
†Packaged version of the environment: https://pypi.org/project/wildfire-environment/.

Table 1 Tree state transition probabilities in the UAV firefighting environment

s_{tree}	0 (healthy)	1 (on fire)	2 (burnt)
0 (healthy)	$(1-\alpha)^{f_t(x^i)}$	$1-(1-\alpha)^{f_t(x^i)}$	0
1 (on fire)	0	$\beta - \sum_{j=1}^{N} \mathbb{I}_{x^i}(z_t^j)\Delta\beta$	$1-\beta+\sum_{j=1}^{N} \mathbb{I}_{x^i}(z_t^j)\Delta\beta$
2 (burnt)	0	0	1

following wildfire dynamics. Let x^i denote the position of tree i. If tree i is in a 'healthy' state at time t, it transitions to an 'on fire' state at time $t+1$ with probability,

$$p_{01}(x^i) = 1-(1-\alpha)^{f_t(x^i)}, \tag{14}$$

where $f_t(x^i)$ is the number of neighboring trees on fire at time t and $\alpha \in [0,1]$ is a parameter which controls the rate of fire spread (higher α results in faster spread). A tree has up to four neighbors, the adjacent trees in the north, east, south, and west directions. Let $\{z_t^j\}_{j=1}^N$ denote the positions of the agents at time t. If tree i is in an 'on fire' state at time t, it transitions to a 'burnt' state at time $t+1$ with probability,

$$p_{12}(x^i, \{z_t^j\}_{j=1}^N) = 1-\beta+\sum_{j=1}^{N} \mathbb{I}_{x^i}(z_t^j)\Delta\beta, \tag{15}$$

where $\mathbb{I}_{x^i}(z)$ is an indicator function (i.e., $\mathbb{I}_{x^i}(z)=1$ if $z=x^i$, else $\mathbb{I}_{x^i}(z)=0$), $\beta \in [0,1]$ is a parameter which controls the time length for which fire on a tree subsists (higher β results in longer subsistence), and $\Delta\beta \in [0,\beta]$ is a parameter which controls the strength of the fire retardant (higher $\Delta\beta$ results in higher retardant strength). Note that two agents cannot occupy the same cell simultaneously, so at most one term in the summation in Equation (15) can be non-zero. Table 1 shows all the tree state transition probabilities, with each value representing the probability of transitioning from the state in corresponding row to the state in corresponding column. The goal for the agents is to move to cells containing trees on fire so they can drop fire retardant and prevent further spread of the fire.

3. Reward Functions

The individual agent reward at each time step is such that an agent is penalized for every tree that transitions from a 'healthy' to an 'on fire' state during the time interval between the previous and the current time step, with a higher penalty for trees inside the agent's selfish region. More formally, let $n_{01,t}(s_t, \mathbf{a}_t, s_{t+1})$ be the total number of trees that transition from a 'healthy' to an 'on fire' state between time step t and $t+1$. Let $n_{01,t}^{SR^i}(s_t, \mathbf{a}_t, s_{t+1})$ be the number of such trees in selfish region, SR^i. Then, $(n_{01,t} - n_{01,t}^{SR^i})$ is the number of such trees outside the selfish region, SR^i, and the reward for agent i at time is given by,

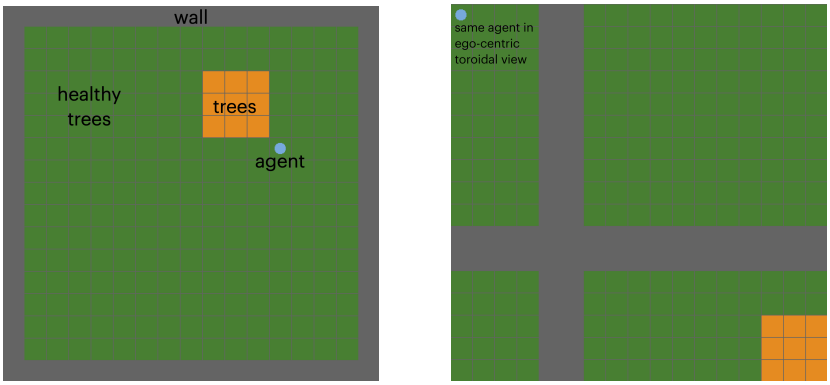
$$R^i(s_t, \mathbf{a}_t, s_{t+1}) = -0.5n_{01,t}^{SR^i} - 0.1(n_{01,t} - n_{01,t}^{SR^i}),\ t \in \{0,1,2,\dots\}. \tag{16}$$

The social reward is defined to ignore any preferences to selfish regions as follows,

$$R_s(s_t, \mathbf{a}_t, s_{t+1}) = -0.5n_{01,t},\ t \in \{0,1,2,\dots\}. \tag{17}$$

4. Agent Observations

A natural state representation is one containing $N+4$ channels, one for each agent, one for each tree state, and one for walls, with each channel consisting of a one-hot encoding of length n^2. However, to improve learning [52], we give each agent its own ego-centric and toroidal observation, analogous to those in [53]. Note that while each agent receives a different observation, each observation contains enough information to maintain full observability of the environment state. More specifically, we define an ego-centric observation by translating the grid such that the ego agent is always in the top left corner of the observation. This observation is also toroidal, in that we wrap around the parts of the grid, which would otherwise be cut off due to the above translation, to the other end of the grid. This process removes the



(a) Actual view **(b) Ego-centric toroidal view**

Fig. 2 An example illustration of an ego-centric toroidal observation in the UAV firefighting environment.

need to maintain a channel for the ego agent, resulting in agents having observations with $(N-1)+4$ channels. Figure 2 shows an example illustration of an ego-centric toroidal observation in our environment. We refer the reader to the supplementary material of [53] for more details. Finally, we append the value of the current time step at the end of each agent's observation vector. Given that we truncate episodes during training (as we discuss in the next subsection), the addition of the time step to states (and observations) is necessary to preserve the Markov property of the MG or MMDP we are training in.

B. Experimental Setup

Our experiments use a grid size of 15×15 with $N = 2$ agents (unless otherwise noted), each having its own selfish region. Each episode is truncated at 100 time steps or when the environment reaches a state that does not contain any trees on fire, whichever is earlier. The initial fire is of size 3×3 cells. We consider five scenarios, each differing in the location and/or size of the selfish regions or fire parameter values. Scenario 1 contains two selfish regions of size 3×3 on the same side of the map. Scenario 2 contains bigger selfish regions (of size 5×5) with the same central cell locations. Scenario 3 contains two selfish regions of size 3×5 and 5×3 located on opposite sides of the map. Scenarios 4 and 5 are the same as scenario 3, but with less effective fire retardant and more challenging fire parameters, respectively. The fire parameter values for scenarios 1, 2, and 3 are $\alpha = 0.15, \beta = 0.9$, and $\Delta\beta = 0.7$. The fire retardant parameter for scenario 4 is $\Delta\beta = 0.5$ and the fire parameters for scenario 5 are $\alpha = 0.17$ and $\beta = 0.91$.

We used the IPPO implementation provided in MARLlib [54] to train agents for all scenarios. Policy network parameters are not shared among the MG agents, whereas they are shared among the MMDP agents. The length of each training run is 20 million time steps. For reproducibility, we list the values of common IPPO hyperparameters used in our experiments in Table 2. MC policy evaluation is done using 5000 episodes for each initial state. The code for all computational experiments in this work is available on GitHub[‡].

C. Experiment Visualizations and Analysis Metrics

We analyze sPoA trends using heatmaps to visualize sPoA for different initial states within a given scenario. Figure 3 shows an example heatmap for scenario 1, where the blowout on the right shows the initial state whose sPoA is represented by the circled cell of the heatmap. The intensity of color at each cell of the heatmap represents sPoA for an initial fire whose geometric center is located at that cell. The cells along the edges of the heatmap are white because they do not represent valid initial states, since their corresponding initial fire regions would extend beyond the map boundary.

We also introduce a variant of sPoA, expected PoA (ePoA), that aggregates over states by taking an expectation over the initial state distribution, following prior work [23, 24]. This metric serves as a baseline to compare sPoA with. Such a comparison also helps us understand the significance of incorporating explicit state dependence. For a given social objective function, the ePoA of an MG $\mathcal{G}$ is given by,

$$ePoA = \mathbb{E}_{s\sim d_0}\left[sPoA(s)\right]. \tag{18}$$

[‡]Code for MARL training, policy evaluation, and computing agent metrics: will be available upon acceptance of the paper

Table 2 Values of common IPPO hyperparameters used in our experiments

Hyperparameter	Value
policy clip parameter (ϵ)	0.3
entropy coefficient	0.01
KL coefficient	0.2
GAE lambda	0.95
optimizer	Adam
optimizer epsilon	1e-5
learning rate	0.0005
batch size	128
number of mini batches	1
epochs	10
value function clip parameter	20
value function coefficient	1
network core architecture	mlp
number of fully connected (fc) layers	4
fc layer dimensions	agent observation size - 256 - 256 - agent action space size
activation function	Tanh
network initialization	random (using standard normal distribution)
discount factor (γ)	0.99

Directly comparing sPoA with ePoA is difficult, since sPoA is a function of state and ePoA produces a scalar value for a given scenario. We therefore calculate the difference between sPoA and ePoA for each initial state, $\Delta\text{PoA}(s) = |\text{sPoA}(s) - \text{ePoA}|$ and present statistical summaries of $\Delta\text{PoA}(s)$, like the expected value and standard deviation. The expected value gives an idea of the overall difference between sPoA and ePoA and the standard deviation gives an idea of the variation of this difference across states.

Finally, we propose a set of heuristic metrics to help understand the underlying reasons for observed sPoA trends. For a given scenario, agent, and initial state, the metrics are as follows. All metrics lie in the range [0,1].

1) State visitation: We visualize state visitation as a heatmap, where the color intensity of each cell shows the proportion of time steps an agent spent at that cell location during an episode.
2) Boundary attack (BA): Let $d_{M,t}$ denote the Manhattan distance of the agent to the closest tree on the boundary of the fire (i.e., the set of trees whose state is 'on fire' with at least one neighbor whose state is 'healthy') at time t. Let the one-step change in Manhattan distance, $u_t \in \{0, 1\}$, be given by,

$$u_t = \begin{cases} 1, & \text{if } d_{M,t+1} - d_{M,t} < 0 \text{ or } d_{M,t+1} = d_{M,t} = 0, \\ 0, & \text{if } d_{M,t+1} - d_{M,t} \geq 0. \end{cases} \tag{19}$$

Then, the BA metric is the cumulative discounted sum of the one-step changes in Manhattan distance over an episode, given by

$$BA = \frac{1-\gamma}{1-\gamma^H}\Big(\sum_{t=0}^{H-1} \gamma^{t-1} u_t\Big), \tag{20}$$

where H is the episode horizon. A higher BA value indicates that the agent is moving closer to the fire boundary

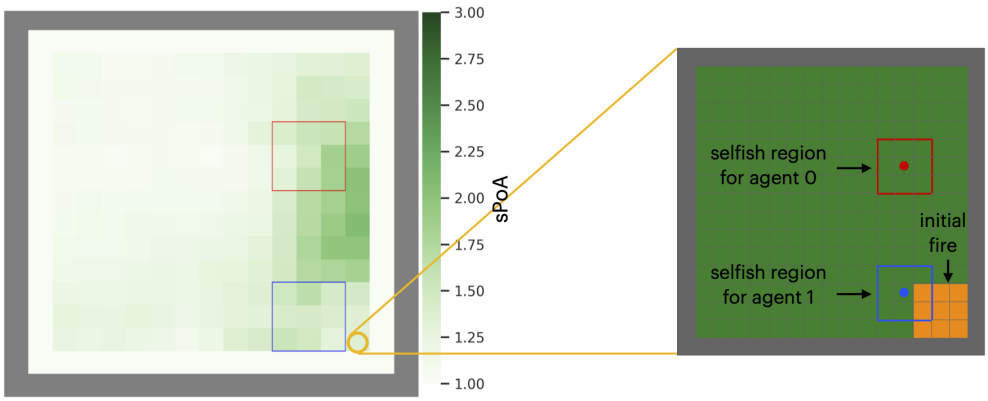


Fig. 3 A visual explanation for how to interpret an sPoA heatmap.

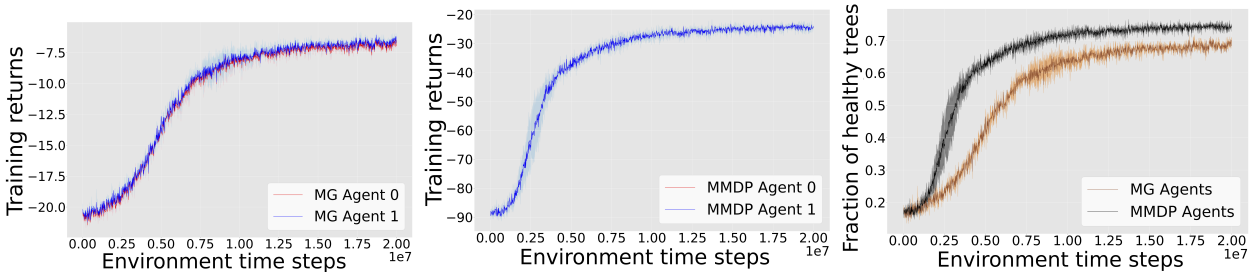
or if it is already on the fire boundary, then continuing to stay on it. This metric is motivated by the fact that an effective real-life wildfire fighting strategy is to first attack the boundary to control the fire.

3) Distance between agents (d_{agents}): At each time step, the mean of the pairwise Manhattan distances between the agent and every other agent is calculated. Then, d_{agents} is the average of the preceding quantity over an episode. The metric is normalized by dividing by $2(n-1)$ for a grid of size $n \times n$. A higher d_{agents} value indicates longer inter-agent distances, averaged over all agent pairs, across an episode.
4) Distance from selfish region (d_{SR}): At each time step, the Manhattan distance between the agent and its selfish region is calculated. Then, d_{SR} is the average of the preceding quantity over an episode. The metric is normalized by dividing by $2(n-1)$ for a grid of size $n \times n$. A higher d_{SR} value indicates longer distances between the agent and its selfish region, on average, across an episode.
5) Time over fire (TOF): This metric is defined as the proportion of timesteps in an episode where the agent is located over a tree on fire. A higher TOF value indicates more time spent over the fire by the agent.
6) Time over fire in selfish region (TOF_{SR}): This metric is defined as the proportion of timesteps in an episode where the agent is located over a tree on fire inside its selfish region. A higher TOF_{SR} value indicates more time spent over the fire inside its selfish region, by the agent.

D. Results

1. MARL Training and Policy Evaluation

As discussed in Section IV.B, we first train MARL agents in MG and MMDP settings to obtain joint policies, $\boldsymbol{\pi}^{NE}$ and $\boldsymbol{\pi}^{*}$. The performance of MARL agents during training is typically evaluated using learning curves, which plot the performance of the agent over the duration of training. A standard choice for measuring performance during training is the episodic return, and is the metric we use to evaluate the training of agents in our experiments. Due to the reward functions being different for the MG and MMDP agents, we cannot directly compare their training return curves, so, we also record a secondary episodic performance metric that is the fraction of healthy trees among total trees at the end of an episode. Let H be the episode horizon and $n_{0,H}$ be the total number of healthy trees at time H. Then, the secondary performance metric is given by $n_{0,H}/n^2$. Figure 4 shows the episodic return and secondary performance metric during training for scenario 1. The plotted values are obtained by first computing a smoothed average of the metric over the last 100 completed episodes, and then averaging this quantity over three independent training runs, initialized with different random seeds. The shaded region in the plots represents the standard deviation of the smoothed average of the metric over the three independent training runs. The training return curves for MMDP Agent 0 and 1 are identical because they have a shared reward. We see that the training curves converge, providing empirical support for the use of these agents in our experiments. Note that while we show results for three independent training runs, we only use one agent of each type for each scenario. We also see that MMDP agents perform better than MG agents with respect to the secondary performance metric, as expected. We then perform MC policy evaluation to estimate $V_{R_s}^{\boldsymbol{\pi}^{NE}}$ and $V_{R_s}^{\boldsymbol{\pi}^{*}}$, which we use to approximate sPoA following Equation (13).



(a) Training returns for MG Agents **(b) Training returns for MMDP Agents** **(c) Training performance metric**

Fig. 4 Learning curves, averaged over three independent training runs, for the episodic return and performance metric in scenario 1.

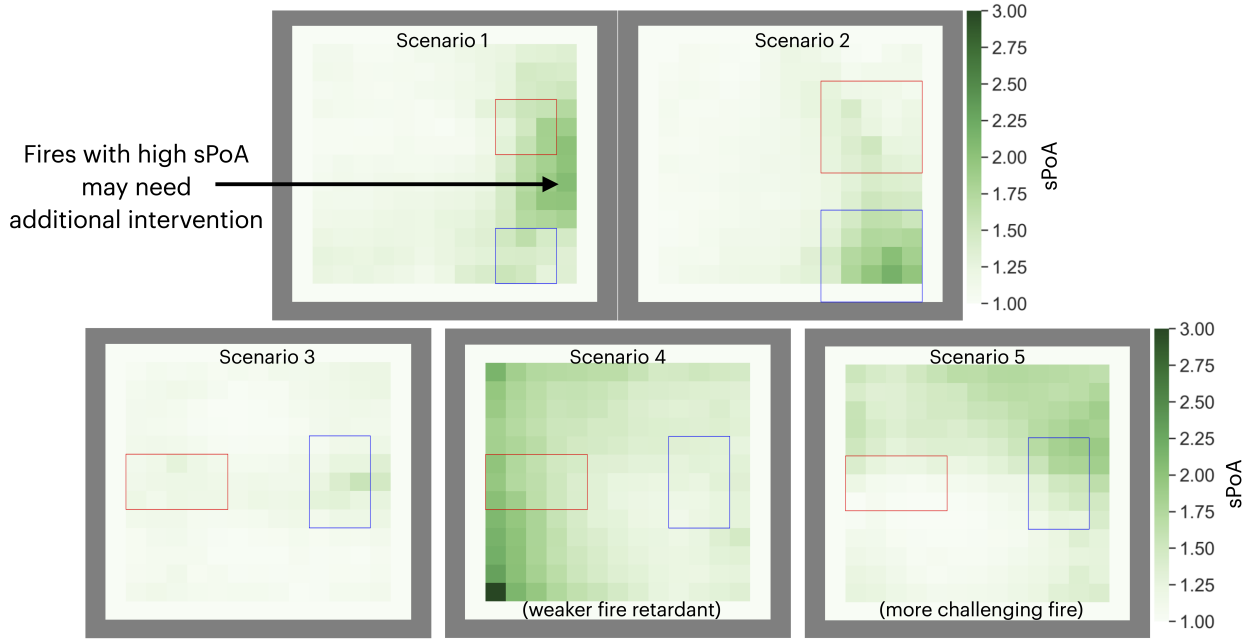


Fig. 5 sPoA heatmaps for five scenarios. Colored outlines represent selfish regions, and agents start at their centers.

2. sPoA for Different Scenarios

Figure 5 shows sPoA heatmaps for the five scenarios mentioned in Section V.B. We highlight two important takeaways in these results. First, for a given scenario, sPoA changes across initial states (i.e., initial fire locations), suggesting that our method could be used to identify potential states where inefficiencies due to self-interested agents are likely to be high. As discussed in Section II, such knowledge could be used to enable more efficient firefighting during a wildfire or inform policy design or long-term resource allocation. Considering scenario 1, we see that initial fires located between the two selfish regions have high predicted sPoA, suggesting that additional interventions may be required if such a fire occurs. One possible explanation for this result is that the agents may be focused on preventing these fires from spreading into their own regions and therefore not preventing the spread of the fire to the west, after which the fire spreads broadly throughout the map.

Our second takeaway is that these sPoA trends change across scenarios, often in non-trivial or un-intuitive ways. That is, the regions which produce high inefficiencies change as the scenario changes. For example, scenario 2, which simply increases the size of the selfish regions relative to scenario 1, shows high sPoA within the selfish region of Agent 1, rather than between the selfish regions. One possible explanation for this trend is that the larger selfish region for Agent 0 requires it to prevent the westward spread of fires starting between the selfish regions, resulting in less

Table 3 ePoA and expected value, standard deviation, and maximum value of ΔPoA for all scenarios

Scenario	ePoA	Expected ΔPoA	Std. dev. ΔPoA	Max. ΔPoA
1	1.22	0.1	0.16	0.85
2	1.18	0.085	0.15	0.99
3	1.11	0.035	0.05	0.45
4	1.47	0.12	0.18	1.55
5	1.3	0.13	0.15	0.61

inefficiencies. We also see changes to sPoA trends across scenarios 3, 4, and 5, which primarily change the fire retardant and fire spread parameters. Some of these changes are more difficult to predict, such as that between scenarios 4 and 5, where the region of high sPoA changes from the southwest corner of the map to the northeast. The magnitude of the maximum predicted sPoA is also significantly higher for scenario 4 than in any of the other scenarios, potentially due to the less effective fire retardant exacerbating performance gaps between self-interested and socially-optimal agents.

The heatmaps shown in Figure 5 partially answer our first research question, as they show that sPoA uncovers different and often non-trivial trends in agent inefficiencies within and across scenarios. To fully answer our question, Table 3 compares our aggregate baseline ePoA, which aggregates sPoA over states, with various statistics for ΔPoA, which explicitly considers state dependence. The first observation here is that, for every scenario, the expection and standard deviation of ΔPoA are small relative to ePoA (each is within 13% of ePoA). This suggests that sPoA is close to ePoA for most states. The second observation, however, is that the maximum value of ΔPoA is significant in all scenarios and can be as large as 105% of ePoA. This result suggests that, while sPoA is generally close to ePoA, there are particular states where sPoA is significantly higher than ePoA. Being able to identify such states was the exact motivation of this work. Hence, the key takeaway here is that sPoA, which incorporates explicit state dependence into PoA, can give critical information about potential inefficiencies of self-interested agents at particular states that would be hidden if using an aggregate variant, such as ePoA.

3. sPoA with Increasing Number of Agents

We also consider experiments with increasing number of agents and larger grid sizes. We specifically consider scenarios with $N = 4$ agents (two teams of two agents each), $N = 6$ agents (three teams of two agents each), and $N = 8$ agents (two teams of four agents each), with a grid size of 15×15 for the $N = 4$ scenario and 20×20 for $N = 6, 8$ scenarios. The fire spread parameter values are $\alpha = 0.17$ for $N = 4, 6$ and $\alpha = 0.19$ for $N = 8$, with all other parameters values as defined in Section V.B. For each two agent team, agents start at the left and right adjacent cells to the center cell of the team's selfish region. For each four agent team, agents start at the four diagonally-adjacent cells to the center cell of the team's selfish region.

Figure 6 shows the sPoA heatmaps for these scenarios. We consistently observe that within each heatmap, the highest inefficiency is for fires starting in and around the selfish regions. We also see that the highest inefficiency is in the $N = 6$ scenario, likely due to the presence of three separate teams of agents (other scenarios consider two teams of agents), with the second highest inefficiency being in the $N = 8$ scenario, likely due to the presence of more agents within a team. This result suggests that increasing the number of teams within a scenario and the number of agents within a team can significantly increase inefficiencies in a system, with the number of teams having a larger impact.

4. sPoA in a Windy Scenario

We also consider experiments with a wind model incorporated into the wildfire dynamics to understand how wind, an important factor in real-world wildfires, might affect sPoA trends. We choose a spatially-varying deterministic wind model, inspired by the fact that wind and terrain can strongly influence the spread of wildfires [55]. Our modeled wind pattern, visualized in Figure 7a, consists of a predominantly eastward flowing wind, with a central region consisting of westward flowing wind and the top and bottom edge of the grid consisting of northward flowing wind. Wind is incorporated into our wildfire model by making the spread rate parameter, $\alpha(x, v)$, vary spatially with each cell location x and direction v. Let the wind direction at x be denoted by wind field, $W(x)$. Given a tree on fire at x^j, our model

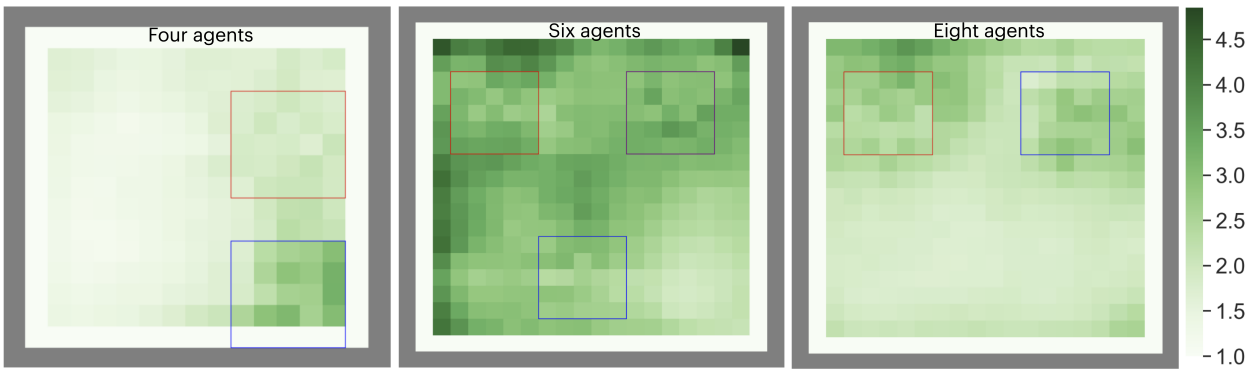
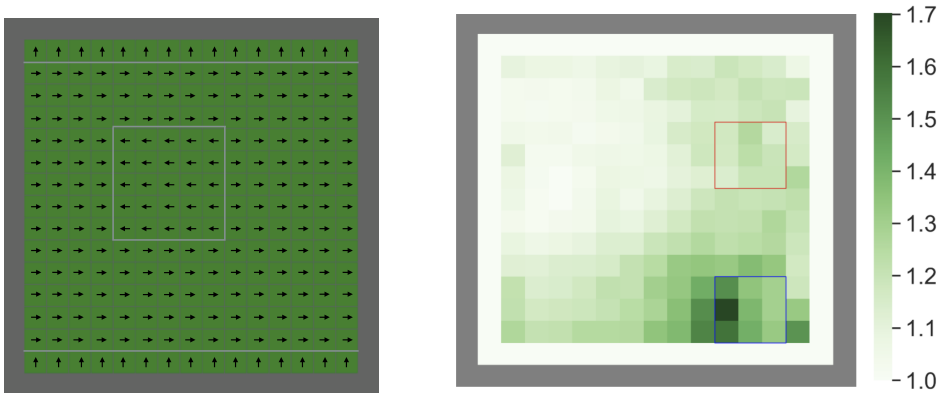


Fig. 6 sPoA heatmaps for scenario 1 with $N = 4, 6, 8$ agents.



(a) Chosen wind pattern **(b) sPoA heatmap**

Fig. 7 Results from adding a wind model to scenario 1. Gray outlines represent the boundaries where wind direction shifts.

assumes that the fire spreads fastest in the downstream direction (i.e., $W(x^j)$), slower in the cross-wind direction (i.e., any direction orthogonal to $W(x^j)$), and slowest in the upstream direction (i.e., $-W(x^j)$). We define these spread rates through three variables, α_{upstream} (the spread rate for fire in upstream wind direction), $\alpha_{\text{cross-wind}}$ (the spread rate for fire in directions orthogonal to the wind), and $\alpha_{\text{downstream}}$ (the spread rate for fire in downstream wind direction), such that $\alpha_{\text{downstream}} > \alpha_{\text{cross-wind}} > \alpha_{\text{upstream}}$.

We then modify the system dynamics as follows. Let the location of neighboring trees of tree i be denoted by $f_t^{north}(x^i)$, $f_t^{south}(x^i)$, $f_t^{east}(x^i)$, and $f_t^{west}(x^i)$. Let $x_{j,i}$ denote the vector (direction) from x^j to x^i. Then Equation (14) is modified for wind as,

$$p_{01}^{wind}(x^i) = 1 - \Pi_{j \in \{north,south,east,west\}}(1 - \alpha(f_t^j(x^i), x_{f_t^j(x^i),i})), \tag{21}$$

where $\alpha(x, v)$ is given by,

$$\alpha(x,v) = \begin{cases} \alpha_{\text{downstream}} & \text{if } \langle W(x), v\rangle = 1, \\ \alpha_{\text{cross-wind}} & \text{if } \langle W(x), v\rangle = 0, \\ \alpha_{\text{upstream}} & \text{if } \langle W(x), v\rangle = -1, \end{cases} \tag{22}$$

with $\langle x, y\rangle$ denoting the dot product of vectors x and y.

We test this wind model using scenario 1 with $\alpha_{\text{downstream}} = 0.2$, $\alpha_{\text{cross-wind}} = 0.15$, and $\alpha_{\text{upstream}} = 0.1$. All other fire parameters are as in Section V.B. Figure 7b shows the sPoA heatmap for this windy scenario. We see that the inefficiency found is highest for fires starting inside the lower selfish region, likely due to those fires tending to move eastward, which increases the conflict between the objectives of MMDP and MG agents. We also see that inefficiency is

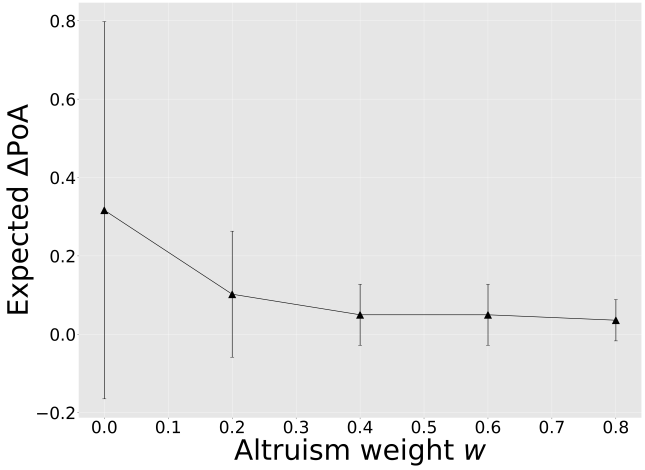


Fig. 8 Expected ΔPoA with 1σ error bars for varying levels of selfishness in scenario 1.

lowest for fires starting in the upper left quadrant of the grid, likely due to the westward winds of the central region, which result in fires having a lower likelihood of spreading towards the selfish regions. Overall, we see that there is a non-trivial change in the observed sPoA trends compared to scenario 1, due to the incorporation of a wind model, further supporting the value of our metric.

5. sPoA for Different Amounts of Self-interest

We also explore the sPoA trends as the level of self-interest of the agents changes. We vary the level of self-interest by modifying the reward function of MG agents, specifically, by altering their level of preference for protecting trees in their own selfish regions, over the other trees in the forest. We achieve this by defining an altruism weight, $w \in [0, 1]$, where lower w results in higher selfishness. The MG rewards degenerate to the MMDP reward when $w = 1$. The reward for MG agent i with altruism weight, w, is given by,

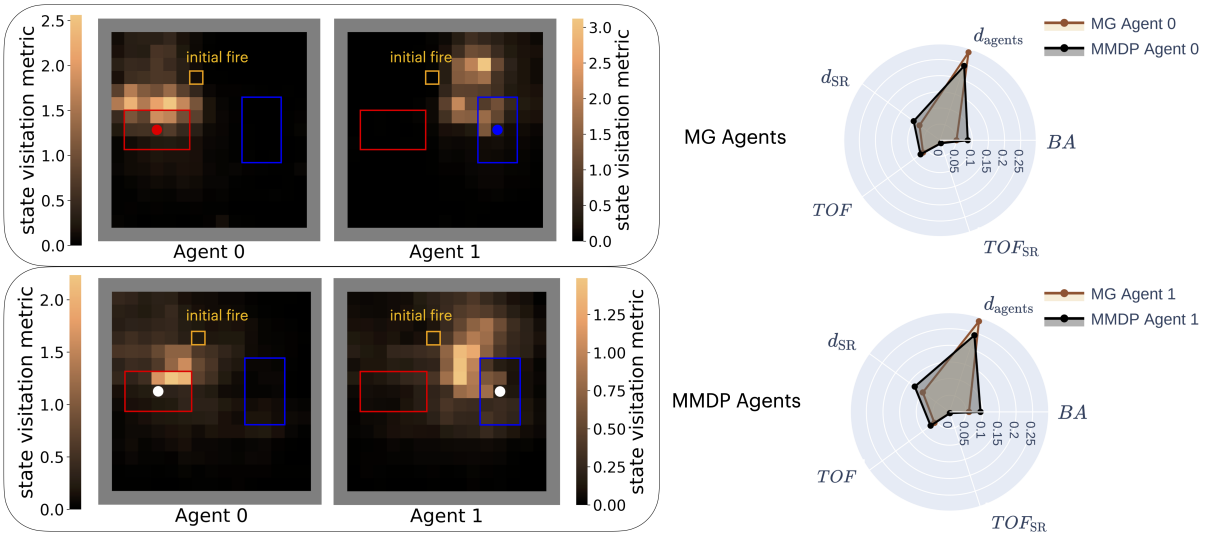
$$R^i(s_t, \mathbf{a}_t, s_{t+1}) = -0.5[n_{01,t}^{SR^i} + w(n_{01,t} - n_{01,t}^{SR^i})],\ t \in \{0, 1, 2, \ldots\}. \tag{23}$$

In all our experiments, we assume that the altruism weight for each MG agent is equal. The MG reward in Equation (16), which was used to train the previously discussed MG agents, corresponds to $w = 0.2$. Here, we additionally present sPoA results with $w = 0, 0.4, 0.6$, and 0.8 in scenario 1. Figure 8 shows the expected ΔPoA with 1σ error bars. We see that the expected ΔPoA decreases as altruism weight increases, which implies that as the level of self-interest in agents decreases, sPoA moves closer to ePoA. Thus, sPoA can give additional insights over ePoA for systems with higher levels of self-interest. But in systems with lower levels of self-interest, ePoA may be a better choice for analysis because the use of sPoA incurs a higher computational cost, which may not be justified given that the differences between sPoA and ePoA are likely to be small.

6. Investigating Underlying Causes for sPoA Trends

Finally, we answer our second research question by using our proposed analysis metrics to understand observed sPoA trends, focusing on scenario 3. Figure 9 shows these metrics for scenario 3 at an initial state with low sPoA (cell (7, 4)), while Figure 10 shows the same for an initial state with high sPoA (cell (13, 8)). All metrics are averaged over 500 episodes.

Beginning with state visitation maps, we first see that for the low sPoA state (Figure 9), MG agents show similar visitation patterns to MMDP agents. For example, in both cases, both agents spend most of their time between their respective selfish regions and the initial fire location. However, for the high sPoA state (Figure 10), Agent 1 shows notably different behaviors between the MG and MMDP cases. More specifically, in the MG case, Agent 1 spends a significant amount of time towards the northeast quadrant of the grid, while in the MMDP case, Agent 1 spends most of



(a) Visitation Maps **(b) Spider Charts**

Fig. 9 Heuristic analysis metrics for an initial fire with low sPoA (i.e., low inefficiency due to self-interested behavior) in scenario 3.

its time near its selfish region—this is a somewhat unintuitive result, as one would have expected MG Agent 1 to spend most of its time near its selfish region. One potential reason for this result is that the selfishness of the MG agents leads to inefficient fire containment which allows the fire to spread farther north than in the MMDP case, which then requires MG Agent 1 to actually move north to prevent the fire from returning to its selfish region.

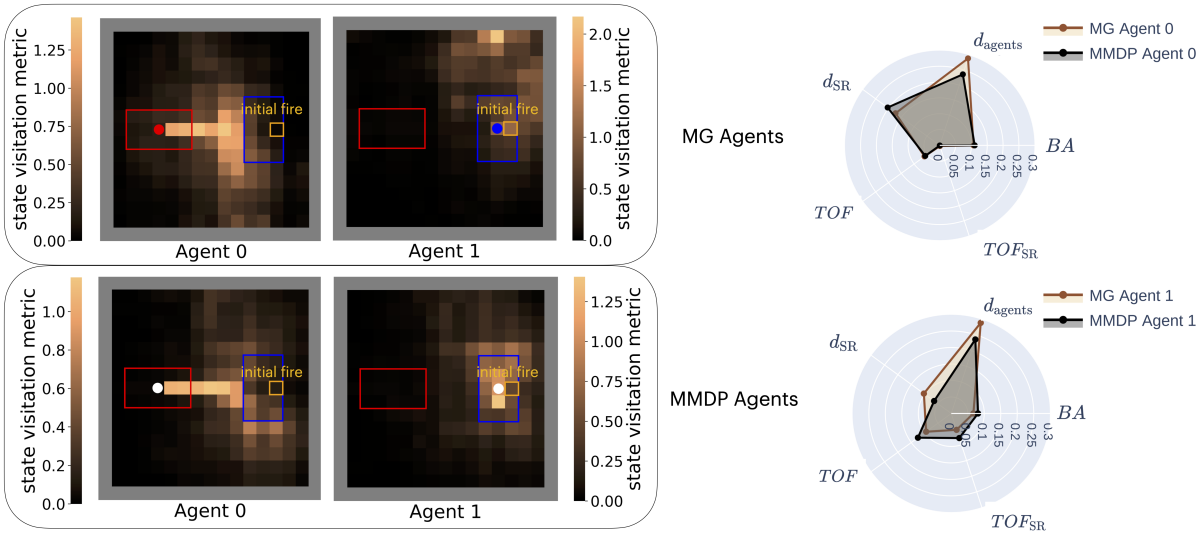
Next, we consider the remaining proposed analysis metrics, which we visualize in spider charts. First, we observe that values for the boundary attack metric, BA, are higher for the MMDP agents than the MG agents for the low and high sPoA states. Given that boundary attack quantifies the extent to which agents adopt a successful real-world firefighting tactic, this was an expected result, since MMDP agents show better overall task performance. However, this trend is not consistent across other states and scenarios, as there are some scenarios where MG agents possess higher boundary attack values, despite having worse task performance than MMDP agents. Thus, while our boundary attack metric can provide insight into how closely trained agents mimic a real-world firefighting tactic, it is not a reliable indicator for task performance. This may be due to the agents learning a different, but successful, firefighting strategy.

Regarding the inter-agent distance metric, d_{agents}, we see that MG agents have higher values than MMDP agents. This result suggests that a scenario with high d_{agents} may suggest a high level of self-interest within that set of agents. For the distance from selfish region metric, d_{SR}, we observe that it is typically higher for MMDP agents than MG agents, except for the case of MG Agent 1 in the high sPoA state. This result may again be due to the inefficiency of MG agents in this scenario, which leads to inefficient fire containment and allows the fire to spread farther north, which then requires MG Agent 1 to leave its selfish region.

Finally, we observe that the MG agents and MMDP agents show insignificant differences in the time over fire metric, TOF, again with the exception of MG Agent 1 in the high sPoA state, where its TOF metric is lower than for MMDP Agent 1. The time over fire in selfish region metric, TOF_{SR}, is close to zero except for the high sPoA state, where both MG Agent 1 and MMDP Agent 1 show non-zero values with MG Agent 1 exhibiting a lower value. The TOF and TOF_{SR} results are in agreement with state visitation heatmaps and the d_{SR} results, which all suggest that the performance gap between MG and MMDP agents for the high sPoA state may be due to the selfish behaviors for MG Agent 1. Overall, we see that these metrics can give interesting and often unintuitive insights into the underlying causes for the observed sPoA trends, by, for example, helping to identify which agents are causing inefficiencies and serving as indicators which can point to potential inefficiencies in the system.

E. Main Observations

Here we consolidate and summarize the main observations from our results for sPoA in different scenarios of the wildfire environment. First, we observe that sPoA changes by non-trivial amounts across initial states within a scenario



(a) Visitation Maps **(b) Spider Charts**

Fig. 10 Heuristic analysis metrics for an initial fire with high sPoA (i.e., high inefficiency due to self-interested behavior) in scenario 3.

and this trend changes in un-intuitive ways across different scenarios. Such information can guide system-level decision makers by identifying specific fires that may require additional interventions, as compared to other fires, due to high inefficiencies from self-interested behaviors. Second, we find that sPoA provides additional insights as compared to a state-aggregated alternative, ePoA, where it identifies critical states with high potential inefficiencies that would be missed if simply considering ePoA. Third, we find that sPoA is particularly useful for analyzing systems with high levels of self-interest in agents. For systems with low level of self-interest, ePoA may be a better choice due to its lower computational requirements. Finally, we find that our set of heuristic metrics can identify individual agents that disproportionately contribute to the overall inefficiency of the system, as well as serve as diagnostic tools to predict inefficiency by observing agent behavior, when prior knowledge about the agent objectives is not available.

VI. Conclusion

This work addresses the problem of measuring the inefficiency, with respect to a social objective, of self-interested behavior in multi-agent systems. Our key idea is to estimate this inefficiency using a state-dependent formulation of price of anarchy, which we call sPoA. We propose a computational method for approximating sPoA using MARL. We then demonstrate our method through extensive empirical results for a UAV firefighting case study. Our results show that sPoA can drastically change as the initial state (initial fire location in our case study) changes within a given scenario. Furthermore, these sPoA trends themselves also change as the scenario changes, thus highlighting the benefit of incorporating explicit state-dependence into PoA. We also quantitatively show that that our sPoA metric captures important insights into potential inefficiencies that would have been missed with a baseline ePoA metric that aggregates over states. Finally, we show the potential of utilizing a proposed set of heuristic metrics for understanding the underlying reasons behind observed sPoA trends.

Limitations of our work include the assumption of full observability, only computing sPoA over states lying in the support of the initial state distribution, the computational cost of policy evaluation, and the assumption that our MARL training converges to a Nash equilibrium.

Future directions include addressing these limitations, as well as considering other extensions like further analysis of heuristic metric trends (e.g., discussions with subject matter experts), exploring overlapping regions of interest (which could create interesting mixed-interest scenarios) and creating task-agnostic analysis metrics that can not only give generalizable insights into sPoA trends but could also be leveraged during MARL training to promote cooperation. Other directions include creating a systematic framework for designing a meaningful social objective function (which we currently assume is given) and exploring an action-value function extension that allows one to analyze how specific

actions (for a given state) affect system inefficiency. Finally, we also aim to better integrate our work with the existing literature on wildfire response planning and higher fidelity wildfire models.

VII. Acknowledgments

The authors would like to thank Dr. Husni Idris for insightful and helpful discussions. This work was funded in part by NASA TTT Award 80NSSC23M0221 and ONR N00014-20-1-2249.

References

[1] Aditya, V., Aswin, D. S., Dhaneesh, S. V., Chakravarthy, S., Kumar, B. S., and Venkadavarahan, M., "A review on air traffic flow management optimization: trends, challenges, and future directions," *Discover Sustainability*, Vol. 5, No. 1, 2024, p. 519. https://doi.org/10.1007/s43621-024-00781-7.

[2] Kistan, T., Gardi, A., and Sabatini, R., "Machine Learning and Cognitive Ergonomics in Air Traffic Management: Recent Developments and Considerations for Certification," *Aerospace*, Vol. 5, No. 4, 2018. https://doi.org/10.3390/aerospace5040103.

[3] Waslander, S. L., "Multi-agent systems design for aerospace applications," Ph.D. thesis, Stanford University, 2007. URL https://www.proquest.com/dissertations-theses/multi-agent-systems-design-aerospace-applications/docview/304810405/se-2.

[4] Straubinger, A., Rothfeld, R., Shamiyeh, M., Büchter, K.-D., Kaiser, J., and Plötner, K. O., "An overview of current research and developments in urban air mobility – Setting the scene for UAM introduction," *Journal of Air Transport Management*, Vol. 87, 2020, p. 101852. https://doi.org/10.1016/j.jairtraman.2020.101852.

[5] Bauranov, A., and Rakas, J., "Designing airspace for urban air mobility: A review of concepts and approaches," *Progress in Aerospace Sciences*, Vol. 125, 2021, p. 100726. https://doi.org/10.1016/j.paerosci.2021.100726.

[6] Yasin, J. N., Mohamed, S. A. S., Haghbayan, M.-H., Heikkonen, J., Tenhunen, H., and Plosila, J., "Unmanned Aerial Vehicles (UAVs): Collision Avoidance Systems and Approaches," *IEEE Access*, Vol. 8, 2020, pp. 105139–105155. https://doi.org/10.1109/ACCESS.2020.3000064.

[7] Tang, J., Lao, S., and Wan, Y., "Systematic Review of Collision-Avoidance Approaches for Unmanned Aerial Vehicles," *IEEE Systems Journal*, Vol. 16, No. 3, 2022, pp. 4356–4367. https://doi.org/10.1109/JSYST.2021.3101283.

[8] Aydin, B., Selvi, E., Tao, J., and Starek, M. J., "Use of Fire-Extinguishing Balls for a Conceptual System of Drone-Assisted Wildfire Fighting," *Drones*, Vol. 3, No. 1, 2019. https://doi.org/10.3390/drones3010017.

[9] Saikin, D. A., Baca, T., Gurtner, M., and Saska, M., "Wildfire Fighting by Unmanned Aerial System Exploiting Its Time-Varying Mass," *IEEE Robotics and Automation Letters*, Vol. 5, No. 2, 2020, pp. 2674–2681. https://doi.org/10.1109/LRA.2020.2972827.

[10] Prakasha, P. S., Nagel, B., Kilkis, S., Naeem, N., and Ratei, P., "System of Systems Simulation Driven Wildfire Fighting Aircraft Design," *AIAA AVIATION 2021 Forum*, 2021. https://doi.org/10.2514/6.2021-2455.

[11] Quero, C. O., and Martinez-Carranza, J., "Unmanned aerial systems in search and rescue: A global perspective on current challenges and future applications," *International Journal of Disaster Risk Reduction*, Vol. 118, 2025, p. 105199. https://doi.org/10.1016/j.ijdrr.2025.105199.

[12] Lyu, M., Zhao, Y., Huang, C., and Huang, H., "Unmanned Aerial Vehicles for Search and Rescue: A Survey," *Remote Sensing*, Vol. 15, No. 13, 2023. https://doi.org/10.3390/rs15133266.

[13] Picard, G., Caron, C., Farges, J.-L., Guerra, J., Pralet, C., and Roussel, S., "Autonomous Agents and Multiagent Systems Challenges in Earth Observation Satellite Constellations," *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2021)*, Londres, United Kingdom, 2021, pp. 39–44. URL https://www.ifaamas.org/Proceedings/aamas2021/pdfs/p39.pdf.

[14] Jiang, W., Han, H., He, M., and Gu, W., "When game theory meets satellite communication networks: A survey," *Computer Communications*, Vol. 217, 2024, pp. 208–229. https://doi.org/10.1016/j.comcom.2024.02.005.

[15] Bonnet, G., and Tessier, C., "Multi-agent collaboration: A satellite constellation case," *STAIRS 2008*, IOS Press, 2008, pp. 24–35. https://doi.org/10.3233/978-1-58603-893-9-24.

[16] Wang, Y., Garcia, E., Casbeer, D., and Zhang, F., *Cooperative Control of Multi-Agent Systems: Theory and Applications*, Wiley, 2017. https://doi.org/10.1002/9781119266211.

[17] Koutsoupias, E., and Papadimitriou, C., "Worst-Case Equilibria," *STACS 99*, edited by C. Meinel and S. Tison, Springer Berlin Heidelberg, Berlin, Heidelberg, 1999, pp. 404–413. https://doi.org/10.1007/3-540-49116-3_38.

[18] Nisan, N., Roughgarden, T., Tardos, E., and Vazirani, V. V., *Algorithmic Game Theory*, Cambridge University Press, New York, NY, USA, 2007.

[19] Vazquez, A. J., and Scott Erwin, R., "Noncooperative Satellite Range Scheduling with perfect information," *2015 IEEE Aerospace Conference*, 2015, pp. 1–10. https://doi.org/10.1109/AERO.2015.7119276.

[20] Klima, R., Bloembergen, D., Savani, R., Tuyls, K., Wittig, A., Sapera, A., and Izzo, D., "Space Debris Removal: Learning to Cooperate and the Price of Anarchy," *Frontiers in Robotics and AI*, Vol. 5, 2018. https://doi.org/10.3389/frobt.2018.00054.

[21] Roughgarden, T., "The price of anarchy is independent of the network topology," *Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing*, Association for Computing Machinery, New York, NY, USA, 2002, p. 428–437. https://doi.org/10.1145/509907.509971.

[22] Volij, O., *Static Games*, Springer New York, New York, NY, 2009, pp. 8649–8668. https://doi.org/10.1007/978-1-4614-1800-9_188.

[23] Zhang, R., Zhang, Y., Konda, R., Ferguson, B., Marden, J., and Li, N., "Markov Games with Decoupled Dynamics: Price of Anarchy and Sample Complexity," *2023 62nd IEEE Conference on Decision and Control (CDC)*, 2023, pp. 8100–8107. https://doi.org/10.1109/CDC49753.2023.10383591.

[24] Chen, D., Zhang, Q., and Doan, T. T., "Convergence and Price of Anarchy Guarantees of the Softmax Policy Gradient in Markov Potential Games," , 2022. URL https://arxiv.org/abs/2206.07642.

[25] Yuan, C., Zhang, Y., and Liu, Z., "A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques," *Canadian Journal of Forest Research*, Vol. 45, No. 7, 2015, pp. 783–792. https://doi.org/10.1139/cjfr-2014-0347.

[26] Boroujeni, S. P. H., Razi, A., Khoshdel, S., Afghah, F., Coen, J. L., O'Neill, L., Fule, P., Watts, A., Kokolakis, N.-M. T., and Vamvoudakis, K. G., "A comprehensive survey of research towards AI-enabled unmanned aerial systems in pre-, active-, and post-wildfire management," *Information Fusion*, Vol. 108, 2024, p. 102369. https://doi.org/10.1016/j.inffus.2024.102369.

[27] Giannakidou, S., Radoglou-Grammatikis, P., Lagkas, T., Argyriou, V., Goudos, S., Markakis, E. K., and Sarigiannidis, P., "Leveraging the power of internet of things and artificial intelligence in forest fire prevention, detection, and restoration: A comprehensive survey," *Internet of Things*, Vol. 26, 2024, p. 101171. https://doi.org/10.1016/j.iot.2024.101171.

[28] Julian, K. D., and Kochenderfer, M. J., "Distributed Wildfire Surveillance with Autonomous Aircraft Using Deep Reinforcement Learning," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 8, 2019, pp. 1768–1778. https://doi.org/10.2514/1.G004106.

[29] Venturini, F., Mason, F., Pase, F., Chiariotti, F., Testolin, A., Zanella, A., and Zorzi, M., "Distributed reinforcement learning for flexible UAV swarm control with transfer learning capabilities," *Proceedings of the 6th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*, Association for Computing Machinery, New York, NY, USA, 2020. https://doi.org/10.1145/3396864.3399701.

[30] Viseras, A., Meissner, M., and Marchal, J., "Wildfire Front Monitoring with Multiple UAVs using Deep Q-Learning," *IEEE Access*, 2021, pp. 1–1. https://doi.org/10.1109/ACCESS.2021.3055651.

[31] Shobeiry, P., Xin, M., Hu, X., and Chao, H., "UAV Path Planning for Wildfire Tracking Using Partially Observable Markov Decision Process," *AIAA Scitech 2021 Forum*, 2021. https://doi.org/10.2514/6.2021-1677.

[32] Haksar, R. N., and Schwager, M., "Distributed Deep Reinforcement Learning for Fighting Forest Fires with a Network of Aerial Robots," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1067–1074. https://doi.org/10.1109/IROS.2018.8593539.

[33] NIFC-Multi-Agency Coordinating Group, "National Interagency Standards for Resource Mobilization," , Mar. 2024. URL https://www.nifc.gov/sites/default/files/NICC/3-Logistics/Reference%20Documents/Mob%20Guide/2024%20NATIONAL%20INTERAGENCY%20STANDARDS%20for%20RESOURCE%20MOBILIZATION.pdf.

[34] Riddle, A., "Federal Interagency Wildfire Response Framework," , Jul. 2024. URL https://www.congress.gov/crs-product/IF12384.

[35] Artley, D., "Wildland Fire Protection and Response in the United States," , Aug. 2009. URL https://www.forestsandrangelands.gov/documents/strategy/foundational/wildlandfire_protectresponse_us_20090820.pdf.

[36] Plantinga, A. J., Walsh, R., and Wibbenmeyer, M., "Priorities and Effectiveness in Wildfire Management: Evidence from Fire Spread in the Western United States," *Journal of the Association of Environmental and Resource Economists*, Vol. 9, No. 4, 2022, pp. 603–639. https://doi.org/10.1086/719426.

[37] Ning, Z., and Xie, L., "A survey on multi-agent reinforcement learning and its application," *Journal of Automation and Intelligence*, Vol. 3, No. 2, 2024, pp. 73–91. https://doi.org/10.59214/jai.v3i2.144.

[38] Jiang, W., Zhan, Y., and Fang, X., "Satellite Edge Computing for Mobile Multimedia Communications: A Multi-agent Federated Reinforcement Learning Approach," *ACM Trans. Auton. Adapt. Syst.*, 2025. https://doi.org/10.1145/3715146.

[39] Lee, J. W., Wang, H., Jang, K., Lichtlé, N., Hayat, A., Bunting, M., Alanqary, A., Barbour, W., Fu, Z., Gong, X., Gunter, G., Hornstein, S., Kreidieh, A. R., Nice, M.-T. W., Richardson, W. A., Shah, A., Vinitsky, E., Wu, F., Xiang, S., Almatrudi, S., Althukair, F., Bhadani, R., Carpio, J., Chekroun, R., Cheng, E., Chiri, M. T., Chou, F.-C., Delorenzo, R., Gibson, M., Gloudemans, D., Gollakota, A., Ji, J., Keimer, A., Khoudari, N., Mahmood, M., Mahmood, M., Matin, H. N. Z., Mcquade, S., Ramadan, R., Urieli, D., Wang, X., Wang, Y., Xu, R., Yao, M., You, Y., Zachár, G., Zhao, Y., Ameli, M., Baig, M. N., Bhaskaran, S., Butts, K., Gowda, M., Janssen, C., Lee, J., Pedersen, L., Wagner, R., Zhang, Z., Zhou, C., Work, D. B., Seibold, B., Sprinkle, J., Piccoli, B., Monache, M. L. D., and Bayen, A. M., "Traffic Control via Connected and Automated Vehicles (CAVs): An Open-Road Field Experiment with 100 CAVs," *IEEE Control Systems*, Vol. 45, No. 1, 2025, pp. 28–60. https://doi.org/10.1109/MCS.2024.3498552.

[40] Sutton, R. S., and Barto, A. G., *Reinforcement Learning: An Introduction*, 2nd ed., The MIT Press, 2018.

[41] Shapley, L. S., "Stochastic Games," *Proceedings of the National Academy of Sciences*, Vol. 39, 1953, pp. 1095 – 1100. https://doi.org/10.1073/pnas.39.10.1095.

[42] Fink, A. M., "Equilibrium in a stochastic *n*-person game," *Journal of Science of the Hiroshima University, Series A-I (Mathematics)*, Vol. 28, No. 1, 1964, pp. 89 – 93. https://doi.org/10.32917/hmj/1206139508.

[43] Albrecht, S. V., Christianos, F., and Schäfer, L., *Multi-Agent Reinforcement Learning: Foundations and Modern Approaches*, MIT Press, 2024.

[44] Boutilier, C., "Planning, Learning and Coordination in Multiagent Decision Processes," *Proceedings of the 6th Conference on Theoretical Aspects of Rationality and Knowledge*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1996, p. 195–210.

[45] de Witt, C. S., Gupta, T., Makoviichuk, D., Makoviychuk, V., Torr, P. H. S., Sun, M., and Whiteson, S., "Is Independent Learning All You Need in the StarCraft Multi-Agent Challenge?" , 2020. URL https://arxiv.org/abs/2011.09533.

[46] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O., "Proximal Policy Optimization Algorithms," , 2017. URL https://arxiv.org/abs/1707.06347.

[47] Samvelyan, M., Rashid, T., Schroeder de Witt, C., Farquhar, G., Nardelli, N., Rudner, T. G. J., Hung, C.-M., Torr, P. H. S., Foerster, J., and Whiteson, S., "The StarCraft Multi-Agent Challenge," *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2019, p. 2186–2188. URL https://ifaamas.csc.liv.ac.uk/Proceedings/aamas2019/pdfs/p2186.pdf.

[48] Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P., "High-Dimensional Continuous Control Using Generalized Advantage Estimation," , 2018. URL https://arxiv.org/abs/1506.02438.

[49] Yasir, M., Howes, A., Mavroudis, V., and Hicks, C., "Environment Complexity and Nash Equilibria in a Sequential Social Dilemma," , 2024. URL https://arxiv.org/abs/2408.02148.

[50] Hoeffding, W., "Probability Inequalities for Sums of Bounded Random Variables," *Journal of the American Statistical Association*, Vol. 58, No. 301, 1963, pp. 13–30. https://doi.org/10.2307/2282952.

[51] Somanath, A., Karaman, S., and Youcef-Toumi, K., "Controlling stochastic growth processes on lattices: Wildfire management with robotic fire extinguishers," *53rd IEEE Conference on Decision and Control*, 2014, pp. 1432–1437. https://doi.org/10.1109/CDC.2014.7039602.

[52] Charniak, E., "Extrapolation in Gridworld Markov-Decision Processes," , 2020. URL https://arxiv.org/abs/2004.06784.

[53] Barreto, A., Hou, S., Borsa, D., Silver, D., and Precup, D., "Fast reinforcement learning with generalized policy updates," *Proceedings of the National Academy of Sciences*, Vol. 117, No. 48, 2020, pp. 30079–30087. https://doi.org/10.1073/pnas.1907370117.

[54] Hu, S., Zhong, Y., Gao, M., Wang, W., Dong, H., Liang, X., Li, Z., Chang, X., and Yang, Y., "MARLlib: A Scalable and Efficient Multi-agent Reinforcement Learning Library," *Journal of Machine Learning Research*, Vol. 24, No. 315, 2023, pp. 1–23. URL http://jmlr.org/papers/v24/23-0378.html.

[55] Sharples, J., McRae, R., and Weber, R., "Wind characteristics over complex terrain with implications for bushfire risk management," *Environmental Modelling Software*, Vol. 25, No. 10, 2010, pp. 1099–1120. https://doi.org/10.1016/j.envsoft.2010.03.016.