Prize Collecting Multi-Agent Orienteering: Price of Anarchy Bounds and Solution Methods

Timothy Murray, Jugal Garg, and Rakesh Nagi

***Abstract*—We propose and address a new variation of the Team Orienteering Problem (TOP) in which all members of the team are independent self-interested agents. The prize-collecting nature emanates from the fact that the prize available at a node of the traversal graph can be collected only by a single visiting agent. The problem is motivated by situations in which team members (agents) must accomplish tasks towards a common goal but are unable to communicate, such as a fleet of surveillance drones operating in a communication denied area or with remote pilots operating independently. We explore three policies for minimizing the amount of inefficiency these self-interested agents can bring into the situation. We analyze these policies in a game theoretic framework and show upper bounds on the Price of Anarchy (PoA) ranging from ≈ 1.582 to unbounded depending on the policy, network type and number of players k. This is done by extending well-known PoA bounds for valid utility systems to a leader-follower setting. The PoA also depends on the behavior of the agents, which could not have goodwill towards other agents. In many cases we are able to provide examples that establish the tightness of the bounds. Finally, solution methods are provided for each of these policies. Numerical results computed by these solution methods are then presented and compared to the optimal centrally coordinated solutions.**

***Note to practitioners*–Unmanned Aerial Vehicles (UAVs) are becoming increasingly popular for information collection tasks in defense and civilian applications alike. When the collection area is large, it is not unusual that a fleet of UAVs is deployed. Routing of a fleet can be performed in a centralized or decentralized manner. Decentralized routing might be the only possibility when centralized situational awareness is not possible due to bandwidth limitations and when centralized optimal routes for each UAV are too complex to compute. For managers of UAV systems, our work provides a theoretical bound on how bad decentralized routing could be in the context of a prize-collecting game. Under a game theoretic framework, we prove that the fleet will collect at least 50% of the prizes collected by the optimal centralized solution. Empirically we show that the performance of the fleet is much better, usually providing at least 90% of the optimal centralized solution. Our routing strategies provide valuable guidance to the practicing engineer or manager of a UAV fleet.**

***Index Terms*—Multi-agent Systems; Prize Collection; Game Theory; Price of Anarchy**

I. Introduction

Team Orienteering Problems (TOPs) began as outdoor games: players arrive in the woods and are equipped with compasses, maps, and instructions for finding checkpoints, and must visit as many checkpoints as possible within a given time limit. These games have a natural link to the classical Vehicle Routing Problem (VRP), but are distinct from the VRP in that it may not be possible to visit all checkpoints; the players must decide where they should go in order to collect the maximum number of points given that not all checkpoints are equal. The TOP is NP-hard even in the single agent case; see [1].

The TOP arises naturally in logistics as an extension of the VRP: direct-to-customer shipping companies must decide which of their orders should be filled today rather than tomorrow when it is infeasible to fill all orders, and retail companies must determine which outlets need to be resupplied immediately to maintain positive inventory and which can wait until next week. Fittingly for such an important problem, it has been well-studied and heuristics have been proposed with empirically satisfying results (more on this in Section I-A).

The situation which we were interested in modeling as a TOP is an Unmanned Aerial Vehicle (UAV) or drone Intelligence Surveillance and Reconnaissance (ISR) network. However, we found that the traditional TOP setup could not realistically model this scenario: TOP problems assume centralized solutions or communication between teammates, as all are working toward a common goal. This is not a realistic assumption for a drone intelligence-gathering network, particularly in an adversarial setting where incoming and outgoing communications may be observed or denied by jamming, and team members may be out of communication with the central authority for extended periods of time. Agents may be aware of their teammates' locations through passive sensing techniques such as visual detection but are unable or unwilling to communicate more actively by broadcasting information. Therefore, we consider a natural approach to this problem, allowing each team member to act as an independent agent seeking to maximize its own score. By reasoning about their fellow agents' locations, agents attempting to maximize their own scores will naturally try to minimize their overlap. We designate this setting as the Prize-Collecting Multi-Agent Orienteering Problem (PCMOP), a new variation on the TOP. The PCMOP is distinct from the Multiagent Orienteering Problem (MOP) formulated by [2] as prizes can be collected by only one agent, with rules regarding which agent may collect them determined by a policy over the game. More detail on these distinctions will be given in Section I-A. We propose three such policies to determine how prizes can be distributed among the agents, and examine the resulting total prize collection. We consider the equilibria resulting from our policies on different graphs such as general, undirected, and directed acyclic, and calculate the theoretical Price of Anarchy (PoA) related to them as a measure of the maximum inefficiency of these equilibria compared to a centrally coordinated optimal solution. The goal of the fleet operator is to have agents collect

T. Murray, J. Garg, and R. Nagi are with the Department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, 61801 USA e-mail: {tsmurra2, jugal, nagi}@illinois.edu.

a maximum value set of prizes through the selection of an appropriate policy.

The rest of the paper is organized as follows: the following subsection comprises an in-depth literature review of work related to the traditional TOP, as well as the few papers which address situations with self-interested agents. Section II details the setting of our problem, and provides a full description of the three policies we propose and examine. In §III, we analyze our policies for different network types for 2 agents and develop tight bounds on the PoA for each policy, as well as an extension to the result on the PoA of simultaneous games over *valid utility systems* [3], showing that they display a PoA of at most 2. In §IV, we repeat the analysis for an arbitrary number of players k. In §V, we develop methods for solving the Stackelberg games resulting from each of our proposed prize-division policies. In §VI, we numerically analyze on test cases by generating approximate $\mathcal{R}^2$ and $\mathcal{R}^2$ planar networks. In §VII, we present a summary and discussion of our results. A full summary of our theoretical results is given in Table II organized by network type, policy type, number of players, range of players, and whether players represent a homogeneous fleet (the details of the policies are defined in §II).

A. Literature Review

As mentioned in the introduction, the TOP is well-studied and several heuristic and exact solution approaches have been proposed and empirically tested in papers such as [4], [5], [6], [7], [8], [1], [9], [10]. [11] performs an in-depth analysis of the single-agent case. [12] examines the applicability of the TOP to drone-related military situations and [13] addresses a UAV variant in which each prize is information, and depending on UAV configuration a single agent may only collect certain types of information from each location. Additionally, [14] addresses the problem of managing a UAV fleet in a communication denied area where time to complete actions and the reward for doing so are uncertain.

Many works, such as [15], [16], [17], [18], [19], [20], [21], [22], propose frameworks and algorithms for coordination of UAV fleets in different settings, such as combating wildfires, surveillance, and target removal, and do so under both online and *a priori* knowledge settings.

Another work somewhat closely related to ours is [2], which formulates and addresses the MOP, and relies on a game-theoretic framework for analysis. In particular, a node i with a prize p_i takes time t_i to deliver that prize to any player who visits it. The node i can only service k_i players at a time, leading to a queue if more than k_i players are present. However, while the number of players who may receive a prize at once on a node is a limiting condition, once an earlier player leaves, the next queued player can still receive a prize. This game models theme-park or tourist routing situations, in which players want to visit the most attractions with the fewest lines, but does not adapt to the situation of limited prizes which we consider, necessitating our PCMOP model. However, the work from which this paper most draws inspiration is the recent article [23], which considers the problem of area surveillance with self-interested agents. In it, two or more UAVs greedily consider choosing routes for the next l time-steps. Additionally, different types of information/prizes are available at each site, which can only be collected by a specific UAV if it has the appropriate sensor type (audio, video, thermal, etc.). The sensors have varying levels of effectiveness so that some portion of the collectible information type is captured and the residual information of that type may be collected by another UAV. However, there are two fundamental differences between [23] and our work: [23] considers simultaneous movements in the game, while our paper considers variations on leader-follower strategies. We will see in Sections III and IV that the leader-follower setting introduces several new complications. Additionally, [23] focuses on games over spacial grid graphs, while we consider games over more general networks.

The proposed PCMOP model has the following similarities with congestion games [24]: both consist of multiple self-interested agents attempting to get from source to destination. However, congestion games focus on the edges of the graph as the source of the delay, while the PCMOP focuses on the nodes of the graph as the locations of prizes. Further, unlike in a congestion game where each player of the same type traveling along an edge experiences the same delay, variations on the prize collecting problems provide inherently unequal payoffs as prizes can only be collected by one individual. However, the two classes of games are similar enough that we look to similar tools in order to analyze them: [25], [26] and [27] all describe variations on Stackelberg (leader-follower) strategies for player decision-making in the congestion game which we consider in our analysis of the PCMOP. In this paper, we focus on a Stackelberg setting in which a pure equilibrium is guaranteed to exist, as opposed to the simultaneous setting (meaning that both players choose their routes simultaneously rather than in leader-follower ordering) in which a PNE does not necessarily exist; see Lemma 2. However, a leader-follower setting also introduces complications to decision-making and analysis, which we detail in Sections III and IV.

II. Setting

A Prize-Collecting Multi-Agent Orienteering Problem is defined by a graph $G(V, E, R)$ and agents $P(D, \mathcal{S}, \mathcal{T})$ where $|P(D, \mathcal{S}, \mathcal{T})| = k$ is the number of agents. Here,

- $V :=$ the set of vertices/nodes in the network, $|V| = n$.
- $E :=$ the set of directed or undirected edges in the graph. Edge e has a nonnegative length l_e.
- $R :=$ the set of prizes at each vertex.
- $D :=$ the set of maximum distances each player can travel before they must reach their destination.
- $\mathcal{S} :=$ the set of source nodes for each player, i.e. where they begin their route.
- $\mathcal{T} :=$ the set of terminal nodes for each player, i.e. where they must end their route.
- $\Sigma :=$ the set of strategy spaces of each player, with Σ_i the set of mixed strategies of player i; a mixed strategy is a probability distribution over the pure strategies, which in this context are the set of all paths from s_i to t_i.

We consider positive edge lengths and non-negative prizes as we are concerned primarily with the drone surveillance

network use-case, and a graph network is a convenient abstraction from the $\mathcal{R}^2$ setting that prizes (areas of interest for surveillance) are likely to be in. Graphs can be constructed as a grid network, or solely as edges between the prize locations (nodes) depending on operator preference.

We consider the general case of this problem, where player sources, terminals, and maximum distances may vary, although traversal speed is assumed to be equal for all players. However, special consideration will also be given to the case of a homogeneous fleet, in which each player has the same maximum distance d, a common source S, and a common terminal T. We will consider the homogeneous case specifically due to its relation to our underlying use case: operators of a fleet of surveillance drones will likely have a homogeneous fleet, and may well be operating from the same origin/destination ($s_i = s_j, t_i = t_j \ \forall i, j$). A natural question for a manager or fleet commander in this setting is how to deploy its agents such that the inefficiency due to agent selfishness is minimized. That is, the manager wants its agents to collect a set of prizes so that the net value of prizes collected by all agents is maximized. This is in contrast to the agents, who want to collect as set of prizes of maximum value for themselves. We will consider three potential deployment policies.

A. Policies to explore

We propose and explore the efficiencies of three natural policies. We assume an arbitrary ordering of the players $1, 2, ..., k$ to indicate turn-ordering in a leader-follower setting. We believe both the policies and the ordering to be natural to large organizations, which frequently display hierarchies based on seniority (e.g., nurses or flight attendants picking schedules). The policies are:

1) Reserved path policy: A player i declares its path from s_i to t_i. All prizes it claims cannot be picked up by another player $j > i$, even if j arrives first.
2) Unreserved path policy (priority for ties): A player i declares its path from s_i to t_i, and may not deviate from it. If another player j with lesser priority (i.e., $j > i$ in the ordering) arrives at a node first, j collects the prize. If both i and j arrive simultaneously, i collects the prize in its entirety and j receives nothing.
3) Turn-based policy: Players take turns moving through the network one node at a time. This corresponds to players moving simultaneously through the network but only having to commit to the next node, rather than their whole route. In the event that two players i and $j > i$ arrive at a node simultaneously, i collects the prize in its entirety and j receives nothing.

We use the term "full-path" policies to refer to the reserved and unreserved path policies, as each player i's strategy set consists of full paths from s_i to t_i. Additionally, we assume that agents are not capable of waiting in a single location, i.e. choosing not to move, as areas which need constant surveillance will have permanent cameras installed. However, waiting can be easily incorporated by adding a self-loop on each node in the network. Waiting will only occur in the turn-based policy though, as it is sub-optimal in the reserved and unreserved path policies. We choose these policies because, due to the successive manner in which players choose their moves, each can be considered an instance of a Stackelberg game in which players take turns selecting their strategies, and the actions of earlier players can be observed by later players. This guarantees the existence of a pure equilibrium. While Stackelberg defined the equilibrium for games of 2-players, the concept can be generalized to k players. More formally:

Definition 1. *A k-player leader-follower (Stackelberg) game with a leader-follower ordering of players $\{1, 2, ..., k\}$ is said to display a pure Stackleberg equilibrium when no deviation by player i will result in a higher payoff for player i, taking into account the changes that players $i + 1$ through k will make to their strategies in response.*

We adopt this k-player extension of the Stackelberg equilibrium from [28].

Lemma 1. *Any k-player full-knowledge leader-follower game displays a pure equilibrium provided the maximum total number of strategy decisions for the game is finite and players have a fixed rule for breaking ties between strategies with equivalent payoffs.*

Proof. Consider player k, the last player to move. Player k must pick the strategy which benefits it the most with full knowledge of the strategies chosen by the first $k - 1$ players. Therefore, the choice which maximizes its payoff given those strategies is a pure equilibrium choice. In the event of an equal maximal payoff between multiple strategies, player k picks according to the fixed tie-breaking rule. Player $k-1$ can predict exactly how player k will react to its own strategy and has full knowledge of the first $k - 2$ players. The use of a fixed-rule for choosing between ties ensures this. Therefore, the choice which maximizes its payoff given those strategies and player k's response is a pure equilibrium choice. Similarly, player i knows the strategy choices of the first $i - 1$ players, and can predict how player $i + 1$, and by extension all players after it, will react to its own strategy. Thus, picking the strategy which maximizes i's payoff given the chosen strategies and coming (predictable) reactions is a pure equilibrium for i. Therefore, under this policy, every player has a pure equilibrium choice regardless of what the previous players did. □

We note that we require the maximum total number of choices made to be finite, not the total number of choices available: On an undirected graph, an agent with infinite range has an infinite number of routes, as it may cycle indefinitely. However, it has only a finite, albeit exponentially large in $|E|$, number of routes it should consider as cycling will not result in any increase in payoff. We note that when each agent i has finite range $d_i \in D$, this is not a problem as there are only a finite number of choices available. With non-finite d_i we resolve this issue by assuming that given two strategies with equal payoff from the same set of nodes, an agent will pick the one corresponding to a shorter route (i.e. avoiding needless cycling). This removes the possibility of non-terminating routes in the reserved and unreserved path policies. However, it does become a problem in the turn-based

policy when agents have unrestricted range, as it may result in non-terminating routes. Theorem 5 shows this in more detail. We also note that having an agent with non-finite range is not feasible in our motivating use case, but we believe it important to consider how agents behave in extreme settings.

While the term PoA traditionally refers to the ratio of the optimal centrally coordinated solution to the worst Nash equilibrium when discussing utility maximization games, we will use it here to refer to the ratio of the optimal centrally coordinated solution to the worst Stackelberg equilibrium where PoA ≥ 1. More formally, for a game g, $PoA(g) = \max_{\sigma \in SE(g)} \frac{u(\sigma^*)}{u(\sigma)}$ where σ^* is the centrally coordinated solution. It is also standard practice to denote the PoA of a set of games G as the supremum of the PoAs of the games in the set, $PoA(G) = \sup_{g \in G} PoA(g)$. Finally, we will be interested in two specific fixed tie-breaking rules in this paper, *Goodwill* and *Sadism*. The rule of Goodwill will limit or reduce some of the performance inefficiencies of our three policies, and the corresponding rule of Sadism will increase these inefficiencies.

Definition 2. *A player i is said to display goodwill to a player j if, given a set of strategies Σ_i all resulting in equal (maximal) payoff for i, player i picks the one which allows j to achieve the maximum payoff.*

Definition 3. *A player i is said to display sadism to a player j if, given a set of strategies Σ_i all resulting in equal (maximal) payoff for i, player i picks the one which forces j to achieve the minimum payoff.*

Under any fixed rule which fails to break a tie, we assume that the player choosing makes its choice according to some second arbitrary rule, such as a lexicographic ordering of routes, which will not fail.

Another policy to consider would be one of simultaneous route picking, in which each agent simultaneously picks its entire route and proceeds through the network, collecting any prizes it comes across first. However, we have largely neglected to explore this policy because it does not necessarily possess a PNE as shown in the following lemma.

Lemma 2. *Consider a 2-player game in which both players simultaneously pick their entire routes and split any prizes they arrive at simultaneously according to some fixed proportion $\lambda \in [0, 1]$. This game does not necessarily contain a PNE.*

Proof. We show this by a counter-example. Consider the network in Figure 1. There are 3 routes between S and T: ABC, AC, and D. The value underneath each node label is the prize associated with that node, i.e. node C has a prize of $2 - 4\varepsilon$, where $0 < \varepsilon \ll 1$. Players are identical: both start from S and go to T, with $d_1 = d_2 \geq 4$ and $l_e = 1$ for all edges $e \in E$. If both players arrive simultaneously at a node, player 1 receives $\lambda \in [0, 1]$ of the prize and player 2 receives $\mu = (1 - \lambda)$. Due to the small number of routes, we construct the payoff matrix for both players picking their entire route simultaneously in Table I, where player 1 is the row player and player 2 is the column player. There is no value $\lambda \in [0, 1]$ in which causes a cell in the matrix to be a PNE. □

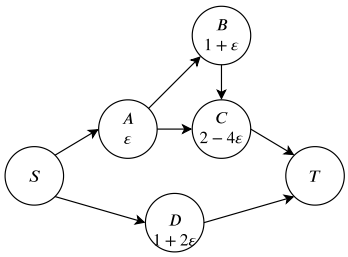

Fig. 1. Network with no pure Nash equilibrium for a 2-Player full-route simultaneous game

TABLE I
PAYOFF MATRIX FOR A FULL-ROUTE SIMULTANEOUS GAME ON FIGURE 1

Route	ABC	AC	D
ABC	$\lambda(3-2\varepsilon), \mu(3-2\varepsilon)$	$1+(1+\lambda)\varepsilon, 2-(3+\lambda)\varepsilon$	$3-2\varepsilon, 1+2\varepsilon$
AC	$2-(4-\lambda)\varepsilon, (2-\lambda)\varepsilon$	$\lambda(2-3\varepsilon), \mu(2-3\varepsilon)$	$2-3\varepsilon, 1+2\varepsilon$
D	$1+2\varepsilon, 3-2\varepsilon$	$1+2\varepsilon, 2-3\varepsilon$	$\lambda(1+2\varepsilon), \mu(1+2\varepsilon)$

We conclude this section with a compilation of our theoretical results, presented in Table II. We use $\mathfrak{e}$ to represent Euler's number, ≈ 2.718, as e is already used for edge representation.

III. RESULTS: THE 2-PLAYER PCMOP

We now provide the proofs of the results for 2-player games.

A. The General 2-Player PCMOP

Before providing results related to the PoAs of the reserved and unreserved path policies, we first revisit the idea of simultaneous games under these full-path policies. Lemma 2 shows that a PNE may not exist in all games for this setting. Therefore, we now derive PoA bounds under mixed Nash equilibrium, which is guaranteed to exist [29], and we show that the PoA under the reserved and unreserved path policies is at most 2. We will do so by drawing upon the concept of a *valid utility system* from [3].

A *utility system* is defined with the following structure: Non-cooperative agents whose action spaces are subsets of an underlying groundset make decisions which induce some social utility, measured by a set function on the actions taken. The agents attempt to maximize their own private utility rather than the social utility. Additionally, the following three conditions hold:

1) The social utility function u and the private utility functions u_i are measured in the same standard unit.
2) The social utility set-function u is submodular. Mathematically, for $A \subseteq B$ and $x \notin B$, we have $u(A \cup \{x\}) - u(A) \geq u(B \cup \{x\}) - u(B)$.
3) The private utility of an agent i is at least the change in social utility which would occur if the agent did not participate in the game. For a strategy set σ and σ_{-i}, the actions of all other agents, we have that $u_i(\sigma) \geq u(\sigma) - u(\sigma_{-i})$.

A utility system is *valid* if and only if

4. $\sum_{i=1}^{k} u_i(\sigma) \leq u(\sigma)$ for all strategy profiles σ.

[3] shows that any game over a valid utility system has a PoA of at most 2.

TABLE II
POA BOUNDS FOR STUDIED POLICIES ON DIFFERENT NETWORK TYPES

		Homogeneous Fleet		General Fleet	
Network	Policy	2-Player	k-Player	2-Player	k-Player
Directed Acyclic	Reserved Path	$\frac{4}{3}$*	$\frac{k^k}{k^k-(k-1)^k} \to \frac{\mathfrak{e}}{\mathfrak{e}-1}$*	2*	2*
	Unreserved Path	2*	$\frac{k^2}{k-1}$	2*	$\frac{k^2}{k-1}$
	Turn-Based	Unbounded	Unbounded	Unbounded	Unbounded
General	Reserved Path	$\frac{4}{3}$*	$\frac{k^k}{k^k-(k-1)^k} \to \frac{\mathfrak{e}}{\mathfrak{e}-1}$*	2*	2*
	Unreserved Path	2*	$\frac{k^2}{k-1}$	2*	$\frac{k^2}{k-1}$
	Turn-Based	Unbounded†	Unbounded†	Unbounded*	Unbounded*
Undirected	Reserved Path	$\frac{4}{3}$*	$\frac{k^k}{k^k-(k-1)^k} \to \frac{\mathfrak{e}}{\mathfrak{e}-1}$*	2*	2*
	Unreserved Path	2*	$\frac{k^2}{k-1}$	2*	$\frac{k^2}{k-1}$
	Turn-Based	Unbounded	Unbounded	Unbounded	Unbounded
Undirected (Unrestricted Range)	Reserved Path	1*	1*	1*	1*
	Unreserved Path	1*	1*	1*	1*
	Turn-Based	1*	1*	1*	1*

* Bound is tight
† Non-Terminating

Theorem 1. *Any simultaneous game under the reserved or unreserved path policies has a PoA of at most 2.*

Proof. This proof will proceed by showing that in the simultaneous setting, the game under the reserved and unreserved path policies is a game over a valid utility system, as defined in [3]. This is sufficient, as [3] also shows that any k-player game over a valid utility system has a PoA of at most 2.

1) The social utility function and player utility functions are both measured in the same units: the value of the prizes collected.
2) The social utility function is submodular. To see this, suppose we have two sets of player paths, S and S' such that $S \subseteq S'$. For some path p, we have that $u(S \cup \{p\}) - u(S) \geq u(S' \cup \{p\}) - u(S')$ as the set of prizes on p which are uncollected in S' must be a subset of the set of prizes on p which are uncollected in S.
3) Private utility of each player is at least as much as the change in the social utility if that player was not present and all other players played the same strategy: The change in the social utility from player i being present is exactly the value of the prizes which are on its path p_i and not any of the other paths p_{-i}, and player i receives at least this set of prizes under both policies.
4) The sum of the players' private utilities is at most the value of the social utility function. This is equivalent to saying $\sum_{i=1}^{k} u_i(S) \leq u(S)$ for any set of paths S. Here the social utility is defined to be the value of all prizes obtained which means that $\sum_{i=1}^{k} u_i(S) = u(S)$.

This completes our proof. □

Next, we provide an extension of [3]'s proof to show that any 2-player Stackelberg equilibrium over a valid utility system also has a PoA of at most 2 when the social and private utility functions are $u_i(S_{-i} \cup \emptyset_i) = 0$ for all $S_{-i} \in \Sigma_{-i}$ for all players i. Here $\emptyset_i$ is equivalent to player i taking no action.

Theorem 2. *Given a 2-player leader-follower game over a valid utility system in which* $\sum_{i=1}^{2} u_i(S) = u(S)$ *and* $u_i(\emptyset_i) = 0$*, the PoA is at most 2.*

Proof. This will be proven by constructing a new simultaneous game in which there is a PNE equivalent to the leader-follower equilibrium, then showing that the setting over which the new game is played is still a valid utility system. Let $S^{st} = \{s_1^{st}, s_2^{st}\}$ be the Stackelberg equilibrium. Define $BR_2(s_1) = \arg\max_{s_2 \in \Sigma_2} u_2(s_1, s_2)$ as the second player's best response to the first player playing s_1. We next introduce a new pure strategy $*_2$ for player 2, where $u_1(s_1, *_2) = u_1(s_1, BR_2(s_1))$ and $u_2(s_1, *_2) = u_2(s_1, BR_2(s_1))$ for all $s_1 \in \Sigma_1$. Thus playing $*_2$ is equivalent to player 2 playing its best response to s_1 after observing s_1. Because $*_2$ is the best response to every pure strategy s_1, it is also the best response to every mixed strategy σ_1. $*_2$ is thus a (possibly weakly) dominant strategy for player 2, and therefore there exists at least one PNE $\{s_1^{st}, *_2\}$. This equilibrium is equivalent to the Stackelberg equilibrium S^{st}. As an example, consider playing the simultaneous prize-collecting game over the network in Figure 1 under the unreserved policy: After the introduction of $*_2$ there is a pure equilibrium of $(AC, *_2)$, which is equivalent to the Stackelberg equilibrium of (AC, D).

Next, we show that the new game still represents a game over a valid utility system as defined by [3]. As the original game was over a valid utility system, we only need to consider

what happens when the second player plays $*_2$. However, we first note that if player i takes action s_i and player $-i$ takes no action, then $u(s_i) = u(s_i, \emptyset_{-i}) = u_i(s_i, \emptyset_{-i}) + u_{-i}(s_i, \emptyset_{-i}) = u_i(s_i, \emptyset_{-i}) = u_i(s_i)$.

1) The social utility and players' personal utilities are still measured in the same units. This is because the original game was over a valid utility system, and the utility functions have not changed with the introduction of $*_2$.
2) The private utility each player receives is at least as much as the change in social utility from their action. To demonstrate this, we must show $u_2(s_1, *_2) \geq u(s_1, *_2) - u(s_1)$ and $u_1(s_1, *_2) \geq u(s_1, *_2) - u(*_2)$ where s_1 is any action taken by player 1. $u_2(s_1, *_2) \geq u(s_1, *_2) - u(s_1)$ follows from the fact that the original game was over a valid utility system and playing $*_2$ is equivalent to playing $BR_2(s_1)$, player 2's best response to s_1. For the second, let $s_2 = BR_2(s_1)$ be the second player's best response to s_1 and let $s_2' = BR_2(\emptyset_1)$ be player 2's best action when under no competition. Clearly $u(s_2') = u_2(s_2') \geq u_2(s_2) = u(s_2)$. Therefore,
$$\begin{aligned} u_1(s_1, *_2) = u_1(s_1, s_2) &\geq u(s_1, s_2) - u(s_2) \\ &\geq u(s_1, s_2) - u(s_2') \\ &= u(s_1, *_2) - u(*_2). \end{aligned} \tag{1}$$
3) The social utility function is submodular. To show this, we note there are only two players and show that $u(*_2) - u(\emptyset) \geq u(s_1, *_2) - u(s_1)$ and $u(s_1) - u(\emptyset) \geq u(s_1, *_2) - u(*_2)$. The first follows from the fact that the original game was over a valid utility system: adding $*_2$ to the set is equivalent to adding $s_2 = BR_2(s_1)$, player 2's best response to s_1, on the right hand side. On the left hand side, it is equivalent to adding $s_2' = BR_2(\emptyset_1)$, player 2's best action when under no competition. Therefore
$$\begin{aligned} u(*_2) - u(\emptyset) = u(s_2') &= u_2(s_2') \\ &\geq u_2(s_2) \\ &= u(s_2) \\ &\geq u(s_1, s_2) - u(s_1) \\ &= u(s_1, *_2) - u(s_1), \end{aligned} \tag{2}$$
where $u_2(s_2') \geq u_2(s_2)$ was established in the previous point. For the second, from the original game we know that $u(s_1) \geq u(s_1, s_2) - u(s_2)$. We also know $u(s_2) \leq u(s_s')$ which implies
$$\begin{aligned} u(s_1) - u(\emptyset) &\geq u(s_1, s_2) - u(s_2) \\ &\geq u(s_1, s_2) - u(s_2') \\ &= u(s_1, *_2) - u(*_2). \end{aligned} \tag{3}$$
4) $\sum_{i=1}^{2} u_i(S) \leq u(S)$. As we have assumed $\sum_{i=1}^{2} u_i(S) = u(S)$, and the utility functions have not changed with the introduction of $*_2$, this is true.

This completes the proof. □

It is immediately apparent that Theorems 1 and 2 together imply an upper bound of 2 on the PoA of 2-player games under the unreserved path policy. In Theorem 4 we will show that this bound is tight.

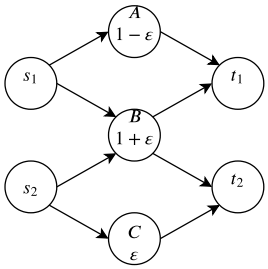

Fig. 2. Network With PoA of 2 under reserved and unreserved path policies

Lemma 3. *The PoA in the general k-Player setting under the reserved path policy is at most 2.*

Proof. Theorem 1 demonstrates that in the simultaneous setting, the k-Player game under the reserved path policy represents a game over a valid utility system. Therefore, [3] implies that under the simultaneous setting, the game has a PoA of at most 2. We make the observation that under the reserved path policy, the leader-follower and simultaneous games are equivalent as for $i < j < l$, player j can ignore the actions of player l and predict the actions of player i, as i can also ignore all players with less "seniority" than it has, even in the simultaneous setting. Therefore, the reserved path policy has a PoA of at most 2 in the general setting, as it is equivalent to a simultaneous game over a valid utility system. Figure 2 shows this bound to be tight using $k = 2$ players: The first player will go to B and the second will go to C before continuing to t_1 and t_2, respectively. A total of $1 + 2\varepsilon$ in prizes will be collected, when the centrally coordinated solution would collect a total of 2 in prizes, from nodes A and B. □

B. The Homogeneous 2-Player PCMOP

Previously, we considered the most general form of the 2-Player PCMOP. Now we consider the homogeneous fleet PCMOP, where $s_i = s_j = S$, $t_i = t_j = T$, and $d_i = d_j = d$. We refer to such a setting as a *homogeneous game* and we will see that although the PoA for the 2-player game under the unreserved path policy remains 2, the PoA for the reserved path policy will be reduced to $\frac{4}{3}$.

In order to prove several of our results in this section, we first let $A_{[i]}$ denote the total value of the prizes that are on the routes planned by players 1 through i. We let $A^*_{[k]}$ denote the total value of the prizes collected by k players in the optimal centrally coordinated solution. We will typically normalize $A^*_{[k]} = 1$ when proving theorems.

Lemma 4. *In a k-player homogeneous game in which players 1 through i have planned their routes to collect a total of $A_{[i]}$ prizes, there is a path containing at least $\frac{1}{k}(A^*_{[k]} - A_{[i]})$ prizes which none of the first i players will collect in the unreserved and reserved path policies.*

Proof. If the first i players have set their routes so that their paths contain a set of prizes valued at $A_{[i]}$ in total, then the optimal paths of the k players must still retain at least $(A^*_{[k]} - A_{[i]})$ in ignored prizes. There are k optimal paths, so at least one must contain a set of prizes with minimum value

of $\frac{1}{k}(A^*_{[k]} - A_{[i]})$ which is non-overlapping with the set of prizes in the first i players' paths. □

Theorem 3. *Under the reserved path Policy, the PoA of 2-player homogeneous games has a tight upper bound of* $\frac{4}{3}$.

Proof. Lemma 4 directly provides an upper bound of $\frac{4}{3}$ on the PoA: If we normalize $A^*_{[2]} = 1$, the value of the prizes collected by two players in the centrally coordinated solution, there is a path containing $a_1 \geq \frac{1}{2}$ in prizes which the first player collects. With that in mind, the Lemma then shows there is a path containing $a_2 \geq \frac{1}{2}(1-a_1)$ in uncollected prizes which is taken by the second player, as it cannot steal any prizes from the first player. The total collected is $a_1 + a_2 \geq \frac{3}{4}$. We show this bound is tight by example, using the network in Figure 3. The network displays a PoA of $\frac{4}{3}$, so this is a tight bound for the reserved path policy in a directed acyclic, and therefore general, graph: In the figure, the first player will maximize its payoff by going to the two nodes containing prizes of $1+\varepsilon$, leaving the second player able to collect only one of the remaining prizes. We also show it to be tight on an undirected graph with restricted range d using the same example. We do so by setting the range to $d = 3$ and changing the edges in Figure 3 to be undirected. Player 1 again collects the two $1+\varepsilon$ prizes and Player 2 again collects only one of the two remaining prizes. Therefore, the PoA is $\frac{4}{3}$, so this is a tight bound for undirected graphs with limited range as well. □

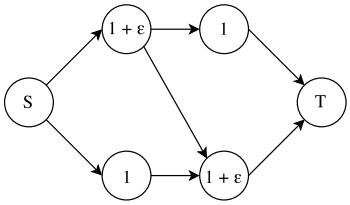

Fig. 3. Network with a 2-Player PoA of $\frac{4}{3}$ under the reserved path Policy

Theorem 4. *Under the unreserved path Policy, the PoA of 2-player homogeneous games has a tight upper bound of* 2.

Proof. Theorems 1 and 2 together imply that in the general 2-player game the unreserved path policy has a PoA of at most 2 for all networks. Now we show by example that this bound is tight for identical 2-player games: Figure 4 demonstrates a PoA of 2 on a directed acyclic graph (DAG) and hence a general graph, with the first player choosing $B \to C \to T$ and the second choosing $C \to T$. If the network in the figure is undirected and each player has a maximum range of 4, then the first player again chooses $B \to C \to T$ while the second player now chooses $C \to T \to E \to T$. Thus the bound of 2 also applies to undirected networks with limited range. □

Thus far we have not addressed the turn-based policy. This is because even in the 2-player setting it must be considered as an extensive-form game, something which we can avoid in the reserved and unreserved path policies. Therefore, most of our work with the turn-based path policy is presented in Sections V and VI, as an empirical study. However, we will at this time provide one theoretical result:

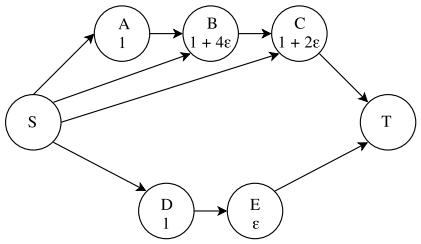

Fig. 4. Network with a 2-Player PoA of 2 for Turn-Based and Unreserved Policies

Theorem 5. *Under the turn-based policy, the Price of Anarchy for a k-player homogeneous game may be unbounded for an arbitrary fixed tie-breaking rule.*

Proof. This will be a proof by example using a game with $k = 2$. Suppose there is a bound r on the PoA of turn-based games in general graphs. We will construct a game which has a PoA greater than this. Consider the graph in Figure 5 and a 2-player game where each player has range $d = r + 3$ and the directed ring of 1-prize nodes is of length greater than $2(r+1)$. There are directed edges from both A and B to every node in the ring, and every ring node also has a directed edge going to T. Centrally coordinated, each player should move to one of the staging nodes (A and B) and then move to the ring in such a way that they can each collect $r + 1 + \varepsilon$ prizes before moving to T, resulting in $2(r+1+\varepsilon)$ prizes collected in total.

Now we consider the game when each player displays *sadism* toward the other. After both players' initial moves, each will be at A or B. Without loss of generality, assume player 1 is at A and player 2 is at B. The first player must decide whether to go to B or go to one of the nodes on the ring. If it goes to one of the nodes on the ring, the first player will collect a prize of one and the second player, being sadistic, will move directly in front of it resulting in the first player obtaining $1+\varepsilon$ prizes. The first player is able to do this at any point, as the second player cannot collect all the ring prizes, so it obtains the same value in prizes by waiting. Additionally, it knows that the second player will not venture into the ring and so will reduce the number of prizes it can collect. Because the first player is sadistic, it therefore moves to B. The second player is then faced with the same choice and, as it is also sadistic, it moves from B to A. The two players cycle back and forth until each has two moves left, then each will visit one ring node before proceeding immediately to T. The value of the prizes collected is $2(1+\varepsilon)$, giving a PoA of $\frac{2(r+1+\varepsilon)}{2(1+\varepsilon)} > r$. □

Note that Theorem 5 only implies that an arbitrarily chosen fixed tie-breaking rule may have an unbounded PoA, not that every fixed tie-breaking rule has one.

Note that as this result applies to homogeneous games, it also applies to the more general PCMOP setting. It is also worth noting that in the setting described in the proof of Theorem 5, if both players have infinite range they may cycle indefinitely, and the game will not terminate.

Despite the lack of formal theoretical bounds on the performance of the turn-based policy, we show in Section VI that empirically it results in a lower average PoA across nearly all tested problem classes and sizes than the unreserved policy.

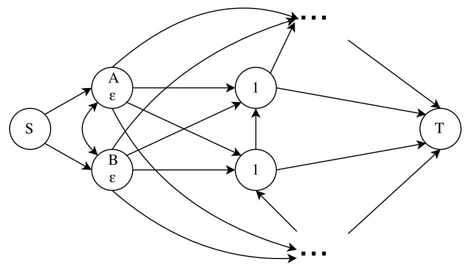

Fig. 5. Network with an unbounded 2-Player PoA for Turn-Based Policy

IV. The k-Player PCMOP Game

We now wish to consider the PCMOP with an arbitrary number of players k. For the k-player game, we will confine our discussion to the homogeneous setting.

From Lemma 3, we know the k-player reserved path policy has a PoA of at most 2, and that bound is tight. However, Theorem 3 shows the bound improves to $\frac{4}{3}$ in 2-player homogeneous games, leading to the following Theorem:

Theorem 6. *Under the reserved path Policy, the Price of Anarchy for homogeneous k-player games has a tight upper bound of $\frac{k^k}{k^k-(k-1)^k}$, with a limit of $\frac{\mathfrak{e}}{\mathfrak{e}-1}$ as $k \to \infty$.*

Proof. Suppose $A^*_{[k]} = 1$. Using Lemma 4 we know that the first player captures prizes with a value of $a_1 \geq \frac{1}{k}$. The second player captures prizes with a minimum value of $a_2 \geq \frac{1}{k}(1-a_1)$ after player 1 plays, so players 1 and 2 together capture prizes with a minimum value of $\frac{2k-1}{k^2}$. Table III shows the results of continuing this line of reasoning. We next show that a lower bound on the value of the prizes captured in a k player game is $\frac{\sum_{j=0}^{k-1}\binom{k}{j}*k^j(-1)^{k-1-j}}{k^k} = \frac{k^k-(k-1)^k}{k^k}$.

Suppose for games with k players, the first i players capture a minimum of $A_{[i]} \geq \frac{\sum_{j=0}^{i-1}\binom{i}{j}*k^j(-1)^{i-1-j}}{k^i}$ of what they optimally could. Now consider player $i+1$. It captures a minimum of $a_{i+1} \geq \frac{1}{k}(1-A_{[i]})$ where

$$\frac{1}{k}(1-A_{[i]}) \leq \frac{1}{k}\left(1-\frac{\sum_{j=0}^{i-1}\binom{i}{j}*k^j(-1)^{i-1-j}}{k^i}\right) = \frac{\sum_{j=0}^{i}\binom{i}{j}*k^j(-1)^{i-j}}{k^{i+1}}. \tag{4}$$

We then have $A_{[i+1]} = A_{[i]} + a_{i+1}$, and note that the minimum value of $A_{[i+1]}$ can occur only if the minimum value of $A_{[i]}$ occurred (and thus we have the maximum guarantee on the value of a_{i+1}) as an increase of δ to $A_{[i]}$ results in a decrease of only $\frac{\delta}{k}$ to the guaranteed minimum of a_{i+1}. Thus

$$\begin{aligned} A_{[i+1]} &= A_{[i]} + a_{i+1} \\ &\geq \frac{\sum_{j=0}^{i-1}\binom{i}{j}*k^j(-1)^{i-1-j}}{k^i} + \frac{\sum_{j=0}^{i}\binom{i}{j}*k^j(-1)^{i-j}}{k^{i+1}} \\ &= \frac{\sum_{j=0}^{i}\binom{i+1}{j}*k^j(-1)^{i-j}}{k^{i+1}}, \end{aligned} \tag{5}$$

which gives $A_{[k]} \geq \frac{\sum_{j=0}^{k-1}\binom{k}{j}*k^j(-1)^{k-1-j}}{k^k} = \frac{k^k-(k-1)^k}{k^k}$. As $k \to \infty$, we have

$$\begin{aligned} \lim_{k\to\infty} \frac{k^k-(k-1)^k}{k^k} &= \lim_{k\to\infty} 1-\left(\frac{k-1}{k}\right)^k \\ = \lim_{k\to\infty} 1-\left(\left(\frac{k}{k-1}\right)^k\right)^{-1} &= 1-\mathfrak{e}^{-1} = \frac{\mathfrak{e}}{\mathfrak{e}-1}. \end{aligned} \tag{6}$$

We can show that this is a tight lower (upper) bound on the efficiency (PoA) of the reserved path policy by constructing examples of networks with these efficiencies (PoAs). We do so by creating a graph with k parallel paths of length k^{k-1} and add edges in a way so that each player collects the minimum it is guaranteed, and visits all paths. Figure 3 displays this PoA of $\frac{4}{3}$ for $k=2$, and Figure 6 displays a PoA of $\frac{19}{27}$ for $k=3$, but due to exponential size of these graphs in k we have not provided images here for $k \geq 4$. □

TABLE III
Reserved Path Bounds

i	Min Captured	Min Captured, $k=i$
1	$\frac{1}{k}$	1
2	$\frac{2k-1}{k^2}$	$\frac{3}{4}$
3	$\frac{3k^2-3k+1}{k^3}$	$\frac{19}{27}$
4	$\frac{4k^3-6k^2+4k-1}{k^4}$	$\frac{175}{256}$
...	...	...
k	$\frac{\sum_{j=0}^{k-1}\binom{k}{j}*k^j(-1)^{k-1-j}}{k^k}$	$\frac{k^k-(k-1)^k}{k^k}$

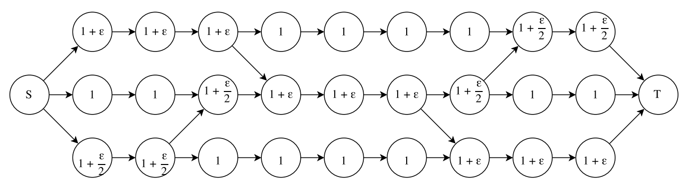

Fig. 6. Network with a 3-Player PoA of $\frac{19}{27}$ for all policies

We now consider the unreserved path policy. From Theorem 1, we know that the k-player game under the unreserved path policy represents a game over a valid utility system when played simultaneously. Therefore, we know that in the simultaneous setting the PoA of the k-player game is at most 2. However, we do not have a theorem concerning the PoA of k-player leader-follower games over valid utility systems. Because of this, we establish a loose bound on the PoA of the unreserved path policy.

Lemma 5. *Under the unreserved path Policy, the Price of Anarchy for a k-player homogeneous game on a general graph is less than or equal to $\frac{k^2}{k-1}$.*

Proof. We will normalize $A^*_{[k]} = 1$. Consider the case where $k-1$ players have laid out their routes and collected a total of $A_{[k-1]}$. If $A_{[k-1]} \geq \frac{1}{k}$ then at least $A_{[k-1]}$ will be collected no matter what player k does. Therefore, assume $A_{[k-1]} < \frac{1}{k}$. By Lemma 4, there is a path q_k containing at least $\frac{1}{k}\left(1-A_{[k-1]}\right) \geq \frac{k-1}{k^2}$ uncollected prizes, so we know that the k^{th} player will collect at least this many. □

Lemma 6. *Under all policies on an undirected graph with unlimited range, the general game has a PoA of 1.*

Proof. The reserved path policy is trivial: The first player will plot the shortest route which visits all nodes and collect all prizes. The unreserved path policy is similarly easy: The first player must plot a path which ultimately ends at t_1, as must the second player and so on. If by the time player k must select its route, there are still prizes which will remain uncollected, then player k will make sure to obtain all of them before terminating at t_k. The turn-based policy is slightly more difficult: Each player has determined its entire route prior to making its first move. Each infinite route is dominated by a finite route (unlike in the directed cyclic case seen in Figure 5) because after some time, none of the routes plan to collect anymore prizes. Since all players plan to end at t_i in a finite number of turns, some player i plans to finish last. i will not move to T until all remaining prizes are collected, because range is unlimited and the undirected graph has full bi-directional connectivity between all collectible prize nodes.

In the case where the graph has disconnected components, each component represents its own subgame where the above scenarios play out. □

V. Solution Methods

Having introduced the PCMOP and providing theoretical analysis of the PoA under our three policies in Sections II through IV, we next develop solution methodologies to solve a 2-player games with integer length edges. This section illustrates how to solve a game for each policy on directed acyclic graphs, beginning with an integer program formulation for the original TOP. Formulating the TOP is necessary for two reasons: First, to measure the PoA of a given problem to compare to our theoretical bounds, we must compute the optimal solution to the TOP. Second, all of our policy solution methods incorporate the TOP formulation as a helper function. Our reserved path solution method uses the TOP formulation repeatedly to iteratively construct each player's path, the solution method for the unreserved path approach uses it to iterate over multiple high value paths and expands it to compute best responses for the second player to the first player's path, and our solution method for the turn-based policy, which iterates over the game tree, will use the TOP formulation to avoid non-optimal leaves in the tree. All algorithms will be discussed in depth in the remainder of this section, with pseudocode for each provided in Appendix A.

TOP Formulation: We begin by formulating the TOP because each of our solution methods will either build on it or solve the formulation as a helper method. We model the problem as an Integer Program patterned on the integer-valued flow-conservation formulation given in [30], given here as IP1 in Appendix A. x_{ij}^m is binary to indicate whether agent m traversed edge (i, j), $z_i \in \{0, 1\}$ indicates whether the prize at node i is collected, l_{ij} is the length of an edge, and r is the common maximum range for all players. Finally, $d_j = 0$ except for $d_1 = 1$, $d_n = -1$ to establish the flow conservation constraints, and c_i is the value of the prize at node i.

Reserved Path Analysis: The reserved path policy is the simplest policy to derive a solution for. Because each player does not need to be concerned with the actions of any later player, each need only solve a single-player orienteering problem where the prizes that earlier players will collect are set to zero. Pseudocode is given in Algorithm 1 in Appendix A, where SolveIP1 is a helper function which solves the integer program formulation (IP1) of the TOP presented in the previous subsection.

Unreserved Path Analysis: Given the complexity of this problem, we only consider the two-player game. We consider the unreserved path as a two-stage game tree. While the players may no longer be able to observe each other after they begin moving through the network, each is required to declare its path prior to setting out. However, for a directed acyclic graph of n nodes there may be as many as 2^{n-2} paths going from node 1 to node n, meaning the tree may have as many as $2^{2(n-2)}$ leaf nodes. To reduce computation time we make the following observation: if the first player has found a path that gives v in prizes after the second player makes its best response, then any path containing less than v total prizes must be strictly worse for the first player than the path it has already found. Thus we can order the paths in the network according to the value of their prizes and stop searching paths once their value drops below the current best value v that the first player is able to obtain after player 2's best response. We refer to v and the corresponding path as the first player's best strategy so far. Therefore, we adopt a computation strategy based on computing the k'-best paths. Note that k' is distinct from k, the number of players in the game.

In order to compute the second player's best response, we use the integer program IP2 in Appendix A, where M is defined to be a large, positive constant. t_i^j is a variable that represents the time-step at which player j arrives at node i, and is 0 if i is not visited by j. v_i^j is a binary variable which is 1 if and only if player j visits node i. Because we are calculating a best response to player 1's path, all t_i^1 and v_i^1 are set to their appropriate value to represent the path, and are capitalized to T_i^1 and V_i^1 to indicate they are constants. We offset arrival times for visited nodes so that if the first player reaches node i at time j, then $T_i^1 = j - 0.5$, in order to represent the fact that the first player obtains the prize in any tied arrivals. Again, it is important note that IP2 is made for Directed Acyclic Graphs, as constraint (19) will not allow a node to be visited by the second player more than once.

The problem of finding the top k' paths through a network is well studied, with efficient algorithms proposed as far back as the 1970's in [31] and [32]: these still form the basis for many algorithms used today. While these approaches are developed towards finding the k'-shortest or cheapest paths, it is possible that algorithms patterned after them may be developed in the case of directed acyclic graphs. In an enterprise-level solution this should be attempted, but we chose to solve the TOP iteratively with $k' = 1$ for convenience as we already developed the framework (IP1) to do so by setting $k = 1$. Paths are constructed iteratively, then compared the value of the path to the value v of the first player's current best strategy.

If the value of the current path is less, we terminate and return the current best strategy. Otherwise, we compute the second player's best response to the path using IP2. Following the best response of the second player, the first player's best strategy is updated if a new best strategy is found. We then add a constraint to IP1 to disallow the current path, and continue to the next iteration. The pseudocode in Algorithm 2 in Appendix A illustrates this method, where SolveIP2 computes the second player's best response to the first player's current path.

Turn-Based Analysis: For the turn-based policy, we again consider the game tree. The game tree has the same number of leaf nodes as the two-level game tree from the unreserved path policy, but resists an easy ordering of them. However, there is a great deal of commonality between several nodes in the game tree: For a DAG, if the next player to take a turn is at node i and the other player is at node j, then it does not matter where they were before, only which prizes have been collected on the nodes between i and j and the remaining range of each player. Because of this, we maintain a hashset with states of the game as keys, and the next move (i.e., next state) along with the value for each player of the current state. If a state is not already in the set then it has not been solved yet, and its solution is the successor state that leads to the maximum value. The recursion stops when it reaches a state where one player is at n. It then solves IP1 with $k = 1$ starting from the position of the unfinished player and returns. Algorithm 3 illustrates this approach, using Algorithm 4 as a helper function. The helper function fills in the game tree (StateSpace in Algorithm 3) for each feasible pair of positions for the two players, including which nodes between their positions have already been visited. Finally, a second helper function ConstructPaths follows the pointers through the game tree to return the paths of each player as a sequence of nodes. Pseudocode versions of both algorithms are given in Appendix A.

Of our solution methods, the turn-based solution is the only one which requires integer length edges. To represent the fact that players are moving and making decisions near simultaneously, we convert the integer edge-length graph into an unweighted graph, dividing a length l edge into l length-1 edges with $l - 1$ nodes between them, containing no prizes.

VI. Numerical Analysis

This section presents numerical experiments using the solution methods detailed in Section V. Section VI-A details our generation of test-cases for these methods, while Section VI-B presents the results of games played on these test cases.

A. Test-Case Generation

1) Approximate $\mathcal{R}^2$ Networks: Because our motivation for considering the PCMOP is for its applications to UAV surveillance networks, we chose to generate networks which resemble real geography. We did so as follows: consider a $l \times l$ box in the plane ($\mathcal{R}^2$). To generate an n node DAG, place n points uniformly at random within the box and label the points so that $x_1 \leq x_2 \leq ... \leq x_n$. Compute the Euclidean distance between each pair of points. Let $D \in \mathcal{Z}^+$ be the maximum length permitted for any edge within the network. For $i < j$, if the distance from point i to point j is $dist(i, j) \leq D$, add an edge from node i to j of length $\lceil dist(i, j) \rceil$ to the DAG. We take the ceiling of the distance because, as noted in the previous section, it is easier to compute the turn-based game with integer length edges. Because we only consider adding an edge (i, j) if $i < j$, the resulting network will be a DAG. Having constructed the network, we then ran tests using two different prize distributions for the nodes: In the first, each node is assigned a prize drawn from the geometric distribution with $p = \frac{1}{2}$. In the second, each node is assigned a prize of value 1, 2, or 3, uniformly at random. Therefore both distributions generate the same mean prize value. Additionally, the prizes on $S = 1$ and $T = n$ are set to 0.

It should be noted that the resulting network may not have a path from 1 to n, or may not have one of sufficiently short length. In this case, a new network is generated.

2) Planar Networks: We were also interested in studying planar networks, as they are another common type of real network in $\mathcal{R}^2$. In particular, they better model a surveillance fleet which is restricted to roads. In order to generate these networks we again generate a set of n nodes uniformly in a box on the plane ($\mathcal{R}^2$). We then use the method of Delaunay triangulation to produce planar networks from these points. If an edge exists between nodes i and j such that $i < j$ then the edge is assigned to be (i, j) rather than (j, i), which guarantees the resulting network is a DAG. After constructing the network, we use the same uniform prize distribution we used previously and assign all nodes other than $S = 1$ and $T = n$ a prize of 1, 2, or 3 uniformly at random. We chose not to run trials for the geometric prize distribution because while we are interested in planar networks as a subset of networks in $\mathcal{R}^2$, they are less applicable to our use case as UAVs are not bound to roadways and infrastructure.

B. Empirical Results

Table IV at the end of this section presents the full details of our numerical experiments, both in our approximate $\mathcal{R}^2$ and planar networks. Max Edge refers to the maximum length edges inserted into the graph as described in Section VI-A. Box refers to the edge length l of the box used to generate the graph. All entries related to Computation Time are in seconds. Tests for uniformly distributed prizes in both planar and approximate $\mathcal{R}^2$ networks were run on machines using the Windows 10 Enterprise OS and 16GB RAM with Intel Xeon(R) v6 CPUs at 3.30GHz. Tests for the geometrically distributed prizes for the approximate $\mathcal{R}^2$ networks were run with Windows 10 Pro OS and 32GB RAM with Intel(R) i7 CPUs at 2.6GHz.

To test how PoA changes with the average size of the network, we generated instances of approximate $\mathcal{R}^2$ networks with varying numbers of nodes n. These trials are detailed in Table IV. Additionally, Figures 7 and 9 show the average PoA for approximate $\mathcal{R}^2$ networks as function of n when these graphs are generated in an $l = 10 \times 10$ box, with max edge length 3, range 15, and prizes drawn geometrically with $p = \frac{1}{2}$ and uniformly at random from 1, 2, 3, respectively. Figures

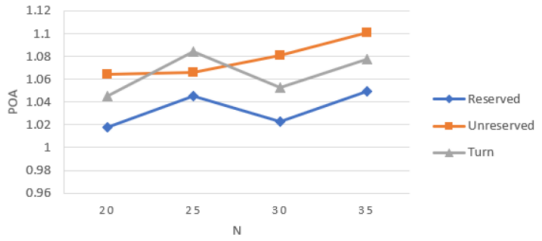

Fig. 7. Average PoA for Approximate $\mathcal{R}^2$ Networks with Geometric Prize Distribution

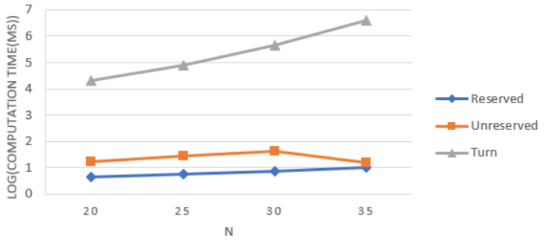

Fig. 8. Average Computation time (log-scaled) for Approximate $\mathcal{R}^2$ Networks With Geometric Prize Distribution

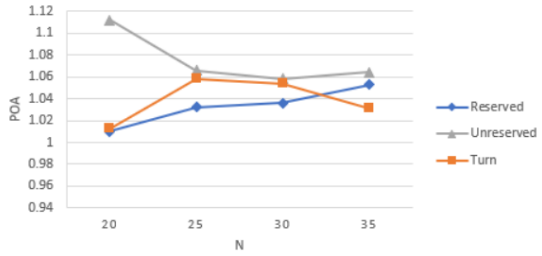

Fig. 9. Average PoA for Approximate $\mathcal{R}^2$ Networks with Uniform Prize Distribution

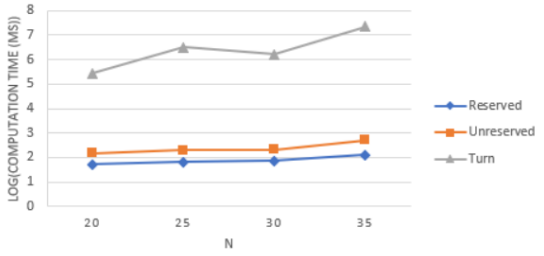

Fig. 10. Average Computation time (log-scaled) for Approximate $\mathcal{R}^2$ Networks With Uniform Prize Distribution

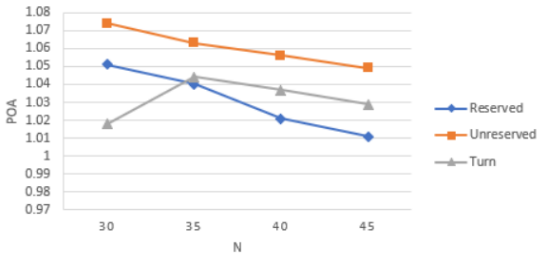

Fig. 11. Average PoA for Planar Networks with Uniform Prize Distribution

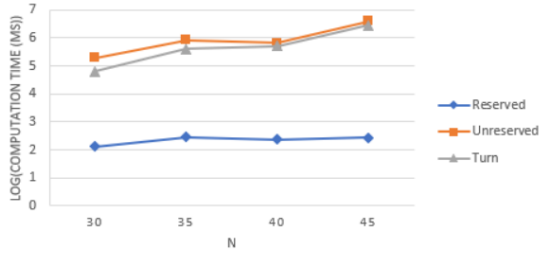

Fig. 12. Average Computation time (log-scaled) for Planar Networks with Uniform Prize Distribution

8 and 10 show the corresponding average computation times for solving these networks, given as the log of the average milliseconds (ms) required.

When we consider the figures, we see that the computation times for the two prize distributions display similar behavior, with an approximately linear increase in log-scaled mean computation time as a function of the number of nodes n. The unreserved case for $n = 35$ and geometrically distributed prizes is the main contradiction to this statement, however this is likely due to the fact that only 5 trials were run for this instance: Considering the higher sample variance for the unreserved policy when $n = 25, 30$ especially when compared to their sample means, it is evident that when test cases are generated according to this distribution there is a tendency to produce very difficult outlying instances, which may not occur with a smaller sample size of test cases.

The more interesting figures to consider are Figures 7 and 9. While test cases generated from the uniform distribution in most cases demonstrate a lower average PoA, the results are empirically very close. However, while we see a possible upward trend with respect to n in the average PoA for all policies when prizes are geometrically distributed, the same is not true of the networks when prizes are uniformly distributed: While the reserved policy seems to display an upward trend, it is difficult to say what if any regular behavior the average PoAs of the unreserved and turn-based policies display.

While we are interested in planar networks as they are an important subset of graphs in $\mathcal{R}^2$, we have already noted that they are less relevant to our UAV use-case as UAVs are not bound to existing infrastructure. Because of this, and the fact that computing the equilibrium in the turn-based game is the most computationally expensive portion of the approximate $\mathcal{R}^2$ networks, we used our planar test cases to consider what happens to the turn-based game when some of the computational complexities disappear. In particular, all tests were run with unit edge lengths and unlimited range, with prizes again drawn uniformly at random from 1, 2, and 3 for each node. The resulting average PoA's as a function of n can be seen in Figure 11, with Figure 12 containing the associated average computation times. Unsurprisingly, this reduces the computation time immensely for the turn-based game, bringing computation time down by a factor of approximately e compared to a uniform distribution of prizes over a similarly sized approximate $\mathcal{R}^2$ network which can be seen

TABLE IV
NUMERICAL RESULTS RELATED TO POA AND COMPUTATION TIMES

$\mathcal{R}^2$ Approximate Networks Geometric Prize Distribution					PoA						Computation Time (s)					
					Reserved		Unreserved		Turn		Reserved		Unreserved		Turn	
n	Box	Max Edge	Range	Trials	Mean	Var	Mean	Var	Mean	Var	Mean	Var	Mean	Var	Mean	Var
20	10	3	15	5	1.018	.002	1.064	.012	1.045	0.004	0.042	1E-4	.168	.056	201.56	1.30E5
25	10	3	15	15	1.045	.004	1.066	.008	1.084	.006	.056	3E-4	.280	.474	763.14	1.57E6
30	10	3	15	15	1.023	.001	1.081	.007	1.053	.002	.074	4E-4	.413	.287	4.53E3	1.18E8
35	10	3	15	5	1.049	.001	1.101	.007	1.078	.007	.097	4E-4	.159	.006	4.06E4	4.85E9
20	7	2	12	5	1.018	6E-4	1.12	.022	1.038	.002	.031	1E-4	.186	.027	57.839	5.85E3
30	5	2	∞	2	1.009	1E-4	1.074	.011	1.06	.003	.109	5E-10	329.5	1.39E5	1.05E6	1.43E10

$\mathcal{R}^2$ Approximate Networks Uniform Prize Distribution					PoA						Computation Time (s)					
					Reserved		Unreserved		Turn		Reserved		Unreserved		Turn	
n	Box	Max Edge	Range	Trials	Mean	Var	Mean	Var	Mean	Var	Mean	Var	Mean	Var	Mean	Var
20	10	3	15	5	1.010	5E-4	1.112	.040	1.013	.001	.054	2E-4	.154	.006	285.30	1.71E5
25	10	3	15	15	1.032	.002	1.066	.005	1.058	.006	.066	4E-4	.201	.071	3.27E3	1.02E8
30	10	3	15	15	1.036	.003	1.058	.004	1.054	.007	.072	3E-4	.215	.051	1.63E3	1.46E7
35	10	3	15	5	1.053	.004	1.064	.007	1.031	.005	.128	.001	.503	.076	2.23E4	2.50E8
20	7	2	12	5	1.020	.001	1.024	.001	1.012	.001	.044	1E-4	.425	.258	69.074	1.25E4
30	5	2	∞	2	1.010	2E-4	1.039	.001	1	0	.115	.001	756.35	1.00E6	2.91E4	8.90E8

Planar Networks					PoA						Computation Time (s)					
					Reserved		Unreserved		Turn		Reserved		Unreserved		Turn	
n	Box	Max Edge	Range	Trials	Mean	Var	Mean	Var	Mean	Var	Mean	Var	Mean	Var	Mean	Var
30	N/A	1	∞	10	1.051	.001	1.074	.004	1.018	.001	.130	.001	191.84	9.26E4	62.84	596.8
35	N/A	1	∞	10	1.040	.003	1.063	.002	1.044	.002	.283	.005	842.78	3.26E6	406.68	4.22E4
40	N/A	1	∞	10	1.021	.001	1.056	.001	1.037	.001	.234	.007	668.75	1.84E6	509.46	1.09E5
45	N/A	1	∞	5	1.011	1E-4	1.049	.003	1.029	.002	.268	.004	4.00E4	1.79E9	2.82E3	8.54E6

by comparing Figures 10 and 12. What is initially surprising though is the degree to which it raises the computational effort for calculating the unreserved equilibrium. However, upon further consideration the reason becomes clear: As what is essentially a bi-level optimization problem in which the first player needs to calculate its optimal move given the best response of the second player, removing limits on the range of each player, even in a DAG, allows for the potential of exponentially more strategies being available without a method to remove consideration for some of these strategies. Still, the fact that the average computational effort exceeds that of the turn-based game is noteworthy. The variance in computation times sheds some light on this, as Table IV shows in all cases the variance in computation time for the unreserved case is an order of magnitude or more higher than the variance for the turn-based, suggesting that some outlying test networks are particularly difficult to solve.

VII. SUMMARY AND DISCUSSION

In this paper, we introduced the Prize-Collecting Multi-Agent Orienteering Problem and proposed three policies to govern the selfishness of the agents. Given that the PCMOP lies at the intersection of congestion games, shortest path computation, longest path computation, top k paths computation, and the TOP, it is natural that it inherits complexities from each of them, in particular an extreme sensitivity to changes in parameter values. Despite these complexities, we derived theoretical bounds on the maximum inefficiency possible under each of these policies in the form of PoAs. As part of that analysis, we extended [3]'s result related to PoA of games over valid utility systems to a 2-player leader-follower setting. In addition to theoretical bounds, we developed solution methods to solve a PCMOP under three policies. In terms of empirical efficiency, there are relatively small differences in the average PoA of the three policies, as seen in Figures 7, 9, and 11. While the reserved path policy produces the best average PoA in most cases, we see there are some where the average performances of the turn-based policy surpasses it. Also, while the performance of the unreserved policy is the worst on average for all but one of our test cases, there are individual instances where it delivers the best value, albeit not as many as the reserved and turn-based policies.

We have seen that the reserved path policy has the best theoretical guarantees on performance in terms of prize-collecting. However, it may produce poorly distributed prize divisions: consider that with sufficient range on an undirected graph, the first agent will collect all prizes, which defeats the purpose of using multiple agents. Also, while the reserved path policy has the best *guarantees*, it does not always produce the best *results*: A 2-Player game on the directed graph in Figure 13 using either the unreserved or turn-based policies results in all prizes being collected, but the reserved path policy results in a PoA of $\frac{4+2\varepsilon}{3+2\varepsilon}$, its theoretical worst result.

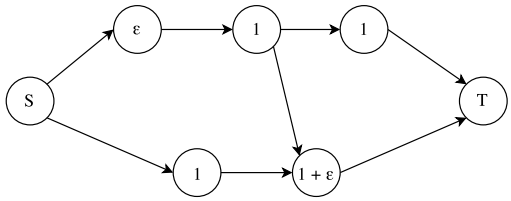

Fig. 13. Network with a 2-Player PoA of $\frac{4}{3}$ for reserved policy and 1 for other policies

The unreserved path and turn-based policies appear to produce the best results in terms of distributions: earlier players have a hierarchical advantage, being able to get any specific prize they want before or at the same time as lower-ranked players and taking it from them, but they must weigh the cost

of the prizes they can no longer guarantee themselves against the ones they want. The turn-based policy seems particularly realistic: if an earlier player makes it clear it is going after a specific large prize, it can make others back off sooner and allow the player to stop and collect additional smaller prizes along the way, rather than racing by them (as in the unreserved policy) even though no other player is targeting the same prize. This is seen in Figure 1, where the prize at node B is collected in the turn-based policy but not the unreserved path policy, resulting in a PoA of $\frac{4}{3}$ for the unreserved case.

The main drawback to the turn-based policy is the computational complexity: The game tree is exponentially large and while there is frequent similarity among branches to reduce computations, it is still a problem. A one player PCMOP/TOP is solved at each leaf node which is not recognized as already solved, which, although likely a smaller problem (since much of the graph is already traversed), is NP-Hard.

The advantage of the reserved path policy is that it is relatively easy to compute, as it allows earlier players to completely ignore the actions of later players, and every player after the first can remove prizes from the network and pretend that it is the first player (for computation). Then each agent only has to solve the single player PCMOP/TOP which, while still NP-Hard, is more tractable than the other versions. While it can cause unbalanced prize distributions in general, it is unlikely to do so when agent ranges divides the workload approximately evenly between agents.

The unreserved path policy seeks to reduce the computational effort of the turn-based policy by considering the best paths first: while computation tends to be longer when there are a number of paths with the same value in prizes, such as when the range is relatively large, in most cases the runtime is shorter than in the turn-based policy. It also appears to produce a more even distribution of prizes than the reserved path policy.

ACKNOWLEDGMENTS

The work of J. Garg was supported by NSF Grant CCF-1942321 (CAREER). T. Murray and R. Nagi were supported in part by the ONR through the Program Management of Drs. D. Wagner and W. Adams under Award N000014-16-1-2245.

APPENDIX A
INTEGER PROGRAM FORMULATIONS, ALGORITHMS, AND NUMERICAL RESULTS

$$\text{IP1: } \max \sum_{i=1}^{n} c_i z_i$$

subject to

$$\sum_{i|(j,i)\in E} x_{ji}^m = \sum_{i|(i,j)\in E} x_{ij}^m + d_j \qquad \forall j \in [n], m \in [k], \tag{7}$$

$$z_j \le \sum_{m=1}^{k} \sum_{i|(i,j)\in E} x_{ij}^m \qquad \forall j \in [n], \tag{8}$$

$$r \ge \sum_{(i,j)\in E} l_{ij} x_{ij}^m \qquad \forall m \in [k], \tag{9}$$

$$x_{ij}^m, z_j \in \{0,1\} \qquad \forall i,j \in [n], m \in [k]. \tag{10}$$

$$\text{IP2: } \min \sum_{i=1}^{n} c_i z_i^2$$

subject to

$$\sum_{i|(j,i)\in E} x_{ji} = \sum_{i|(i,j)\in E} x_{ij} + d_j \qquad \forall j \in [n], \tag{11}$$

$$x_{ij} \le a_{ij} \qquad \forall i,j \in [n], \tag{12}$$

$$v_j^2 \le \sum_{i|(i,j)\in E} x_{ij} \qquad \forall j \in [n], \tag{13}$$

$$v_j^2 \ge \frac{1}{n} \sum_{i|(i,j)\in E} x_{ij} \qquad \forall j \in [n], \tag{14}$$

$$r \ge \sum_{(i,j)\in E} l_{ij} x_{ij}, \tag{15}$$

$$z_i^1 + z_i^2 \le 1 \qquad \forall j \in [n], \tag{16}$$

$$z_i^1 \ge (V_i^1 * t_i^2 - T_i^1)/M \qquad \forall i \in [n], \tag{17}$$

$$z_i^2 \ge (v_i^2 * T_i^1 - t_i^2)/M \qquad \forall j \in [n], \tag{18}$$

$$t_j^2 \ge t_i^2 + (x_{ij} - 1)M + l_{ij} x_{ij} \qquad \forall i,j \in [n], \tag{19}$$

$$x_{ij}, z_j^m, v_j^2 \in \{0,1\} \qquad \forall i,j \in [n], \quad m \in [k]. \tag{20}$$

Algorithm 1 Algorithm 1: ReservedSolution(G, k, prizes)

Require: G,k,prizes
  Initialize $sum \leftarrow 0$;
  Initialize $Paths$ as a size k array;
  for $i = 1$ to k **do**
    $path$, $score \leftarrow$ SolveIP1($G, k \leftarrow 1$,prizes);
    for j in $path$ **do**
      prizes[j] $\leftarrow 0$;
    $sum \leftarrow sum + score$, $Paths[i] \leftarrow path$;
  return $Paths$, sum

Algorithm 2 UnreservedSolution(G, prizes)

  Initialize $score1 \leftarrow 0$, $score2 \leftarrow 0$, $path1$, $path2$;
  Initialize $model \leftarrow$ TOP($G, k = 1$,prizes)
  while $True$ **do**
    $p1$, $s1 \leftarrow$ SolveIP1($model$);
    if $s1 \le score1$ **then**
      return $score1, path1, score2, path2$
    else
      $p2 \leftarrow$ SolveIP2($G, p1$,prizes);
      $s1$, $s2 \leftarrow$ ComputeScores($G, p1, p2$,prizes);
      if $s1 > score1$ **then**
        $score1 \leftarrow s1$, $score2 \leftarrow s2$, $path1 \leftarrow p1$, $path2 \leftarrow p2$;
      $model$.AddConstraint($p1$ is infeasible)

Algorithm 3 TurnSolution(G, prizes)

  Initialize hashset Space $\leftarrow \emptyset$;
  Initialize nCur $\leftarrow 1$, nNext $\leftarrow 1$, rCur $\leftarrow r$, rNext $\leftarrow r$, colBetween $\leftarrow \emptyset$;
  Initialize state0 $\leftarrow$ (nCur, nNext, rCur, rNext, colBetween);
  run SolveState(state0)
  return ConstructPaths(Space)

REFERENCES

[1] T. Tsiligirides, "Heuristic methods applied to orienteering," *The Journal of the Operational Research Society*, vol. 35, no. 9, pp. 797–809, 1984.
[2] C. Chen, S.-F. Cheng, and H. C. Lau, "Multi-agent orienteering problem with time-dependent capacity constraints," *Web Intelli. and Agent Sys.*, vol. 12, no. 4, pp. 347–358, Oct. 2014.
[3] A. Vetta, "Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions," in *The 43rd Annual IEEE Symposium on Foundations of Computer Science*, 2002, pp. 416–425.
[4] C. Archetti, A. Hertz, and M. G. Speranza, "Metaheuristics for the team orienteering problem," *Journal of Heuristics*, vol. 13(1), pp. 49–76, 2007.

Algorithm 4 SolveState(state)

```
n1 ← state.nCur, n2 ← state.nNext, r1 ← state.rCur, r2 ← state.rNext;
curPrize ← prizes, collected ← state.collectedBetween;
for node in collected do
    curPrize[node] = 0;
if state in Space then return
else if n2 = n then
    path, score = SolveIP1(G,curPrize,startFrom←n1, r1);
    if IP1 is infeasible then
        valCur ← −∞;
    else valCur ← score;
    valNext ← 0, nextState ← (n, n, 0, 0, ∅);
    nextStep ← path;
    Space[state] ← (valCur, valNext, nextState, nextStep);
    return
else
    Space[state] ← (−∞, −∞, ∅, n);
    nextMoves ← possible moves for current player;
    for j in nextMoves do
        collectedBetween ← ∅;
        for i ∈ [min (node2, j), max (node2, j)] do
            if i ∈ collected then
                collectedBetween.append(i)
        newState ← (n2,j,r2,r1-1,collectedBetween);
        SolveState(newState);
        valNext ← StateSpace[newState].valNext;
        valCur ← StateSpace[state].valCur;
        p ← curPrize[j] +valNext;
        if p > valCur then
            newValN ← StateSpace[newState].valCur;
            Space[state] ← (p,newValN,newState,j);
```


[5] S. Boussier, D. Feillet, and M. Gendreau, "An exact algorithm for team orienteering problems," *4OR*, vol. 5, no. 3, pp. 211–230, Sep 2007.

[6] I.-M. Chao, B. L. Golden, and E. A. Wasil, "The team orienteering problem," *European Journal of Operational Research*, vol. 88, no. 3, pp. 464 – 474, 1996.

[7] R. El-Hajj, D. Dang, and A. Moukrim, "Solving the team orienteering problem with cutting planes," *CoRR*, vol. abs/1604.02934, 2016.

[8] H. Tang and E. Miller-Hooks, "A TABU search heuristic for the team orienteering problem," *Computers & Operations Research*, vol. 32, no. 6, pp. 1379 – 1407, 2005.

[9] P. Vansteenwegen, W. Souffriau, G. V. Berghe, and D. V. Oudheusden, "A guided local search metaheuristic for the team orienteering problem," *European Journal of Operational Research*, vol. 196, no. 1, pp. 118 – 127, 2009.

[10] G. Best and G. A. Hollinger, "Decentralised self-organising maps for the online orienteering problem with neighbourhoods," in *International Symposium on Multi-Robot and Multi-Agent Systems*, 2019, pp. 139–141.

[11] G. Laporte and S. Martello, "The selective travelling salesman problem," *Discrete Applied Mathematics*, vol. 26, no. 2, pp. 193 – 207, 1990.

[12] V. K. Shetty, M. Sudit, and R. Nagi, "Priority-based assignment and routing of a fleet of unmanned combat aerial vehicles," *Computers & Operations Research*, vol. 35, no. 6, pp. 1813 – 1828, 2008, part Special Issue: OR Applications in the Military and in Counter-Terrorism.

[13] F. Mufalli, R. Batta, and R. Nagi, "Simultaneous sensor selection and routing of unmanned aerial vehicles for complex mission plans," *Computers & Operations Research*, vol. 39(11), pp. 2787 – 2799, 2012.

[14] Y. Xia, R. Batta, and R. Nagi, "Controlling a fleet of unmanned aerial vehicles to collect uncertain information in a threat environment," *Oper. Res.*, vol. 65, no. 3, pp. 674–692, Jun. 2017.

[15] C. Phan and H. Liu, "A cooperative UAV/UGV platform for wildfire detection and fighting," in *System Simulation and Scientific Computing, 2008. ICSC 2008. Asia Simulation Conference-7th International Conference on*. IEEE, 2008, pp. 494–498.

[16] J. Tisdale, A. Ryan, Z. Kim, D. Tornqvist, and J. Hedrick, "A multiple UAV system for vision-based search and localization," in *American Control Conference, 2008*. IEEE, 2008, pp. 1985–1990.

[17] M. Alighanbari and J. How, "Decentralized task assignment for unmanned aerial vehicles," in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*. IEEE, 2005, pp. 5668–5673.

[18] Y. Yang, A. Minai, and M. Polycarpou, "Decentralized cooperative search by networked UAVs in an uncertain environment," in *American Control Conference*, vol. 6. IEEE, 2004, pp. 5558–5563.

[19] J. Tisdale, Z. Kim, and J. Hedrick, "Autonomous UAV path planning and estimation," *Robotics & Automation Magazine, IEEE*, vol. 16, no. 2, pp. 35–42, 2009.

[20] M. Han, Y.-B. Wang, X. Chen, and X. Zhao, "A research on vehicle and UAV routing problem during distribution based on IAMMAS," *IOP Conference Series: Materials Science and Engineering*, vol. 688, p. 044004, 2019.

[21] A. Thibbotuwawa, G. Bocewicz, P. Nielsen, and B. Zbigniew, "Planning deliveries with UAV routing under weather forecast and energy consumption constraints," *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 820 – 825, 2019, 9th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2019.

[22] J. Baek, S. I. Han, and Y. Han, "Energy-efficient UAV routing for wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 1741–1750, 2020.

[23] O. Thakoor, J. Garg, and R. Nagi, "Multiagent UAV routing: A game theory analysis with tight price of anarchy bounds," *IEEE Transactions on Automation Science and Engineering*, vol. 17(1), pp. 100–116, 2020.

[24] R. W. Rosenthal, "A class of games possessing pure-strategy Nash equilibria," *International Journal of Game Theory*, vol. 2, no. 1, pp. 65–67, Dec 1973.

[25] T. Roughgarden, "Selfish routing," Ph.D. dissertation, Cornell University, 2002.

[26] J. Correa and N. E. Stier-Moses, "Stackelberg routing in atomic network games," *Technical Report, Columbia Business School*, February 2007.

[27] G. Karakostas and S. G. Kolliopoulos, "Stackelberg strategies for selfish routing in general multicommodity networks," *Algorithmica*, vol. 53, no. 1, pp. 132–153, Jan 2009.

[28] L. Fang, K. Hipel, and D. Kilgour, *Interactive Decision Making: The Graph Model for Conflict Resolution*, ser. A Wiley-Interscience publication. Wiley, 1993, no. v. 3.

[29] J. Nash, "Non-cooperative games," *Ann. Math.*, vol. 54, no. 2, pp. 286–295, 1951.

[30] P. Vansteenwegen, W. Souffriau, and D. V. Oudheusden, "The orienteering problem: A survey," *European Journal of Operational Research*, vol. 209, no. 1, pp. 1 – 10, 2011.

[31] J. Yen, "An algorithm for finding shortest routes from all source nodes to a given destination in general networks," *Quarterly of Applied Mathematics*, vol. 27, 01 1970.

[32] D. R. Shier, "On algorithms for finding the k shortest paths in a network," *Networks*, vol. 9, no. 3, pp. 195–214, 1979.

Timothy Murray is a graduate student pursuing a Ph.D. in Industrial Engineering from the University of Illinois at Urbana-Champaign. He is broadly interested in Algorithmic and Computational Game Theory, in particular modeling Non-Nash behavior and policy design to minimize inefficiencies resulting from selfish behavior.

Jugal Garg is an Assistant Professor in Industrial and Enterprise Systems Engineering and an Affiliate faculty in Department of Computer Science at the University of Illinois at Urbana-Champaign. He is interested broadly in design and analysis of algorithms, optimization, and mathematical programming.

Rakesh Nagi is Willett Professor in the Department of Industrial and Enterprise Systems Engineering at the University of Illinois, Urbana-Champaign. He has more than 200 journal and conference publications. Dr. Nagi's major research thrust is in the area of production systems, applied/military operations research and data fusion using graph theoretic models.