Stochastic Prize-Collecting Games: Strategic Planning in Multi-Robot Systems

Malintha Fernando, Petter Ögren, Silun Zhang

***Abstract*—The Team Orienteering Problem (TOP) generalizes many real-world multi-robot scheduling and routing tasks that occur in autonomous mobility, aerial logistics, and surveillance applications. While many flavors of the TOP exist for planning in multi-robot systems, they assume that all the robots *cooperate* toward a single objective; thus, they do not extend to settings where the robots must *compete* in reward-scarce environments. We propose Stochastic Prize-Collecting Games (SPCG) as an extension of the TOP to plan in the presence of *self-interested* robots operating on a graph, under *energy constraints* and *stochastic transitions*. A theoretical study on complete and star graphs establishes that there is a unique pure Nash equilibrium in SPCGs that coincides with the optimal routing solution of an equivalent TOP given a rank-based conflict resolution rule. This work proposes two algorithms: Ordinal Rank Search (ORS) to obtain the "ordinal rank" –one's effective rank in temporarily-formed local neighborhoods during the games' stages, and Fictitious Ordinal Response Learning (FORL) to obtain best-response policies against one's senior-rank opponents. Empirical evaluations conducted on road networks and synthetic graphs under both dynamic and stationary prize distributions show that 1) the state-aliasing induced by OR-conditioning enables learning policies that scale more efficiently to large team sizes than those trained with the global index, and 2) Policies trained with FORL generalize better to imbalanced prize distributions than those with other multi-agent training methods. Finally, the learned policies in the SPCG achieved 87% to 95% optimality of an equivalent TOP solution obtained by mixed-integer linear programming.**

I. Introduction

The team orienteering problem (TOP) generalizes the classical traveling salesman problem (TSP) to compute a set of "tours" that visit a *subset* of cities while maximizing a *collective utility* [1]. It provides a unified framework for an array of real-world applications, such as logistic networks, and path planning [2], [3]. While several TOP variants exist, e.g., TOPs with task deadlines [4], they share the notion that all robots must maximize the collective return, thus failing to model settings with heterogeneous or competing objectives, e.g., competing individuals or teams in resource-scarce environments. To address this gap, we propose a game-theoretic variant of the TOP, played by self-interested robots (agents), which we refer to as the *Stochastic Prize-Collecting Games (SPCG)*.

In multi-robot systems, significant research is underway to design robotic teams to automate last-mile delivery and hotspot monitoring, e.g., in surveillance and ocean monitoring.

The authors are with the KTH Royal Institute of Technology, Stockholm, Sweden. E-mail:{malintha, petter, silun} @kth.edu.
Malintha Fernando would like to acknowledge the generous support of the Digital Futures postdoctoral fellowship.

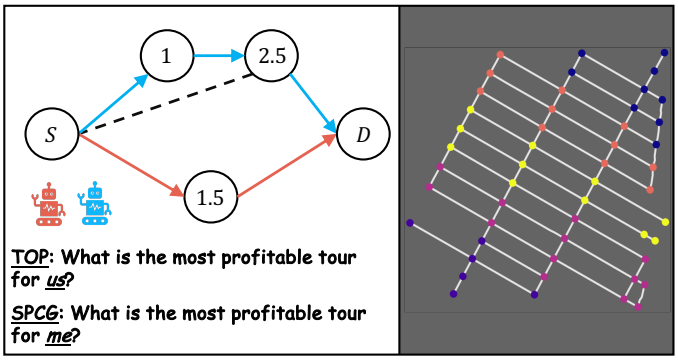


Fig. 1. **Left:** A multi-robot prize-collecting problem on a graph. Here, S, D denote the start and terminal nodes. One robot gets priority during conflicts. In the TOP, the optimal policy (shown) routes the robots to maximizes the *total reward* regardless of their ranks. However, when the robots are *self-interested*, the senior robot must directly move to prize (2.5) ignoring the prize (1) to prevent being cut-in-front, while the other must move to (1.5). Therefore, the maximum total reward in the SPCG (4), is lesser than that of the TOP's (5). **Right:** A Manhattan road-network graph used for the experiments. The mean prize at a node depends on the proximity to the center. Any dead-end node may serve as a terminal, while S is randomized.

However, maintaining a large fleet of mobile robots is often impractical for service providers, who may instead delegate tasks to multiple externally-owned fleets with diverse capabilities, i.e., carrying capacity and sensors. While the fleets can route themselves to service the depots *independently*, they may not coordinate their plans with the other fleets due to the inherent misalignments in mode of operation, capabilities, and the economic incentives. Therefore, conventional TOP-based routing solutions, which assume cooperative robots with a shared objective, are not directly applicable to systems of heterogeneous and self-interested fleets, motivating a TOP variant that accommodates individual objectives and strategic interactions.

In the modern transportation industry, SPCGs also effectively capture the bi-level, profit-driven nature of crowd-sourced applications where rational operators, e.g., delivery drivers, compete to maximize their individual profits among other operators, when the service providers aim to maximize their own profits while competing with similar other providers, e.g., *Uber*, and *Lyft*.

Compared to the TOP, agents in an SPCG aim to maximize their individual rewards rather than a collective utility given a *conflict-resolution rule*: a rule that defines the prize distribution when multiple robots serve the same node. Additionally, in the SPCG formulation, the start and the terminal nodes need not be fixed; robots may initialize randomly and choose between multiple terminal nodes as time progresses. We prove that when the graph is complete, there exists a unique pure Nash equilibrium (PNE) that can be obtained under local informa-

tion and rank-based conflict resolution. We propose two new algorithms to obtain best-response strategies under the multi-agent reinforcement learning (MARL) paradigm, showing that the resulting strategies coincide with the Nash equilibrium in complete graphs. Further, we quantify the *Price-of-Anarchy* (PoA) –the ratio between the worst-case equilibrium and the optimal routing solution obtained by solving a mixed-integer linear program (MILP). In experiments conducted on real-world graphs, we observe that when the robots' local observations are conditioned on their ordinal rank, learned policies achieve 95% total team reward of the optimal routing solution in an equivalent TOP, show better generalizability to previously unseen prize distributions, and scale better with the number of agents. Fig. 1 compares best-response strategies in a prize-collecting game to the optimal strategy of an equivalent TOP.

The key contributions of this work are,

1) introducing SPCGs for distributed decision-making in competitive, stochastic graph-structured environments under energy constraints, while proving the existence of a unique PNE in certain graphs,
2) proposing two algorithms, 1) Ordinal Rank Search (ORS), and 2) Fictitious Ordinal Response Learning (FORL). The former finds the agents' *local rank* within temporally-formed neighborhoods, while the latter computes stationary best-response strategy profiles in SPCGs given the local observations, and
3) proposing a transformer-based architecture for modeling the policies. We also conduct extensive numerical experiments to evaluate the generalizability, and the Nash properties of the policies in real-world road networks.

A. Related Literature

1) The Team Orienteering Problem: The TOP is a multi-agent generalization of the TSP which does not necessitate the salesmen to visit all the cities, rather a subset that maximizes the *collective* reward given a maximum travel budget [4]. In [5], authors formulate the multiple TSP as a Markov decision process, where the salesmen contribute toward a common global objective. The TOP has been extensively studied in robotics for scheduling in stochastic environments. In [3], authors consider a single agent orienteering problem with stochastic prizes, while in [2], authors formulates the multi-robot ocean monitoring under the TOP. The game-theoretic variants of the TOP have only recently gained attention in operational research. In [6], authors introduce a two-agent game where a leader seeks to interdict the follower's revenue in pure competitive settings. A related game-theoretic formulation is considered in [7] where the agents experience a cost proportionate to the number of agents at a chosen node. However, this formulation restricts the agents to observe the same cost during conflicts. In [8], authors studied the multi-robot prize-collecting game on a graph. However, the agents must move one-after-the-other in their their Stackelberg formulation as opposed to simultaneous moves in ours.

2) Game-Theoretic Task Assignment: Several recent works propose MARL-based approaches for multi-robot task assignment (MRTA) in dynamic, stochastic environments in cooperative [9], [10], [11], and competitive settings [12] for manufacturing, transportation and emergency-rescue applications. Market auctions have also been studied as MRTA at the presence of self-interested robotic agents where individuals can propose task bundles they would like to execute by minimizing an individual cost [13]. Oftentimes the auction-based methods require a central arbitrator who evaluates the robots' proposals. This requires iterating through a number of task bundles that is exponential in the number of nodes in the graph –requiring extensive computation, while overlooking the stochastic environmental transitions.

Finally, we draw parallels to resource-selection games (RSG) in operations research [14]. We highlight that the SPCGs differ from RSGs in two crucial ways: firstly, the action space of an agent in SPCGs is constrained by the graph structure, and secondly, SPCGs are an extensive-form game as opposed to the RSG's normal-form nature.

II. Preliminaries

A. Stochastic Games

A stochastic game (SG) is a tuple $\langle \mathcal{N}, \mathcal{S}, \mathbb{T}, R, A, \gamma \rangle$, where $\mathcal{N} = \{1, \dots, n\}$, a set of agents, $\mathcal{S}$ is the state space. The joint action space $A = \times_{i \in [n]} A_i$ where A_i is the action space of the i-th agent, $R = \{R_1, \dots, R_n\}$, the collection of the agents' reward functions such that $R_i : \mathcal{S} \times A \times \mathcal{S} \to \mathbb{R}, \forall i \in \mathcal{N}$. The transition kernel $\mathbb{T} : \mathcal{S} \times A \to \Delta(S)$ defines the probability distribution over S given the current joint state and the joint action. Finally, γ is the discounting factor in i's value function $V_i^{\pi}(s^t)$ that is the *total expected discounted cumulative reward* at any state $s^t \in S$ under a joint strategy $\pi = \pi_i \times \pi_{-i}$. Here, we use $-i$ to denote the set of all other agents except i. The objective of an agent i in an SG is to find the strategy π_i^* that maximizes its value function as

$$\pi_i^* \leftarrow \underset{\pi_i(a_i^t | s^t)}{\arg\max} V_i^{\pi_i, \pi_{-i}}(s^t)$$

for all t using its local observations.

Definition 1. *(Nash Equilibrium) In a stochastic game, a joint strategy π is called a Pure Nash Equilibrium (PNE) if no player has an incentive to unilaterally deviate from it, i.e.;*

$$V_i^{\pi_i, \pi_{-i}}(s^t) \geq V_i^{\pi_i', \pi_{-i}}(s^t) \tag{1}$$

for all $i \in \mathcal{N}$, $s^t \in S$ and $\pi_i' \neq \pi_i$ [15].

An *stage game* is a *one-shot* game played by the agents during a single timestep of the stochastic game.

III. stochastic prize-collecting game

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ define a weighted-undirected graph, where $\mathcal{V} = \{1, \dots, V\}$, $\mathcal{E} \subseteq \{(u, v) : u, v \in \mathcal{V}\}$ are the sets of nodes and edges. Here, $\mathcal{W}$ is a symmetric matrix that defines the travel cost over an edge, without loss of generality, we assume, $\mathcal{W}(u, v) \in [0, w]$, for any $(u, v) \in \mathcal{E}$.

Let $p_u^t(\beta_u) \in \mathbb{R} \geq 0$ denote the prize state at time t for each node $u \in \mathcal{V}$, where $p_u^t(\beta_u)$ is a random variable drawn from distribution $w_u(\beta_u)$ with β_u being some sufficient statistics

for the value distribution. We define the prizes vector $\mathbf{p}^t(\beta) = \big(p_u^t(\beta_u)\big)_{u\in\mathcal{V}}$.

An agent i's state space is $\mathcal{S}_i = \mathcal{V} \times \mathbb{R}^2_{\geq 0}$, containing the tuples $< x_i^t, l_i^t, i >$, whose elements are $x_i^t \in \mathcal{V}$ that is i's location at t, $l_i^t \in [0, L_{\max}]$ which is i's remaining travel budget under a common maximum budget $L_{\max}$ and its index i. Each agent has a starting node s_i and a destination node d_i in $\mathcal{V}$. We consider $d_i = d, \forall i$ in this work. The terminal reward p_d at d is fixed such that $p_d >> p_u, \forall u \neq d$.

Next, we extend the definition of a SG by including $\mathcal{G}$ and a conflict-resolution rule Ξ to define an SPCG, denoted by Γ. The conflict-resolution rule Ξ defines a prize-sharing criteria if multiple agents visit the same node, simultaneously.

An SPCG is represented as, $\Gamma = \langle \mathcal{G}, \mathcal{N}, \mathcal{S}, \mathbb{T}, R, A, \gamma, \Xi \rangle$, where $\mathcal{S} = \times_{i\in N}\mathcal{S}_i \times \mathbb{R}^{|\mathcal{V}|}$ which include the states of the agents and the prizes. For brevity, we ignore the time index and define the action space of an agent as $A_i = \mathcal{V}$. Note that, the actions available for i at t is set of x_i^t's *one-hop* neighbors. We refer this set as i's *reachable set* at t such that $S_i^t = \{v \in \mathcal{V} : (x_i^t, v) \in \mathcal{E}\}$.

We focus on a rank-based conflict-resolution rule that allocates the prize at a node wholly to the agent with the smallest index (highest global rank). Let the index numbering reflect the *global rank* of the agents. Then the highest prize value an agent i receives by visiting a node v at t is p_v^t. The agent may collect it fully given that either it has not already been collected by another agent at a previous timestep, or no agent with rank j $(< i)$ visits v at t. Formally, for some i whose $x_i^t = v$,

$$R_i^{t+1} = \begin{cases} 0 & \text{if } \exists\ j \in \mathcal{N}, j < i, \text{ such that } x_j^t = v, \\ p_v^t & \text{otherwise.} \end{cases} \tag{2}$$

Once collected, the prize gets set to 0, and re-drawn as below.

$$p_v^{t+1} = \begin{cases} \sim \omega_u(\beta_u) & \text{if } p^t = 0 \wedge \exists i \in \mathcal{N} \text{ such that } x_i^t = v, \\ p_v^t & \text{otherwise,} \end{cases}$$

Finally, we define $\mathbb{I}_i^t$ to indicate the availability of an agent i at t, such that $\mathbb{I}_i^{t+1} = 1$ if $l_i^t > 0 \wedge x_i^t \neq d$, or zero otherwise. Then we can write the value function of an agent at $t = 0$ as

$$V_i^\pi(s^0) = \mathbb{E}_{\substack{a_i\sim\pi_i \\ a_{-i}\sim\pi_{-i}}}\Big[\sum_{t=0}^{T} \mathbb{I}_i^t \gamma^t R_i^t(a_i^t, a_{-i}^t, s^t)|s^0\Big] \tag{3}$$

where, $s^0 \in \mathcal{S}$ is the initial state.

IV. Equilibria Analysis

The SPCGs are highly combinatoric in nature as it require countering the others' strategies, rendering them more challenging to solve compared to conventional TOPs. In [8], authors proposed a game where the agents reserve their full routes between s and d in a leader-follower manner. The *Stackelberg* response of an agent in leader-follower formulation is to choose the *optimal* route during its turn by excluding the prizes on previously reserved routes by the senior agents. In contrast, SPCG is a simultaneous-move game in which no agent can reserve a tour; meaning, any pre-determined tour of an agent can be intercepted by another during the current or a future stage of the game.

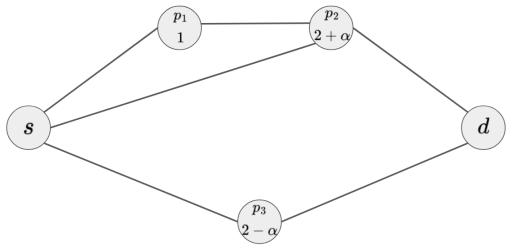


Fig. 2. A two-agent SPCG played on a graph defined by $\mathcal{V} = \{s, 1, 2, 3, d\}$ without self-edges. The players start at s at $t = 0$, share the same destination d. Consider $\mathcal{W}(u, v) = 1, \forall (u, v) \in \mathcal{E}$, and $L_{\max} = 3$. The prize at each node is indicated by p_u. Consider $p_s = 0$ and $p_d >> p_u, \forall u \in \mathcal{V}\backslash d$. This game does not have a PNE for any $\alpha \in (0, 1)$.

	π^1	π^2	π^3
π^1	$(3+\alpha, 0)$	$(1, 2+\alpha)$	$(3+\alpha, 2-\alpha)$
π^2	$(2+\alpha, 1)$	$(2+\alpha, 0)$	$(2+\alpha, 2-\alpha)$
π^3	$(2-\alpha, 3+\alpha)$	$(2-\alpha, 2+\alpha)$	$(2-\alpha, 0)$

TABLE I
Payoff matrix for the 2-agent SPCG shown in figure 2

According to Nash's existence theorem, there is at least one Nash equilibrium in the mixed-strategies in an SPCG, since the number of agents and the action space is finite. However, a more interesting question is whether an SPCG played on an arbitrary graph possesses a pure Nash equilibrium (PNE)? We will show that the answer to this question depends on the structure of the graph and the prize distribution. Here we only focus on a single realization of prizes; $\mathbf{p}^0(\omega_u) \sim \times_{u\in\mathcal{N}}\omega_u$, without re-population.

Theorem 1. *There is a unique pure strategy Nash equilibrium for all $\mathbf{p}^0(\omega_u), \forall \omega_u$ when $\mathcal{G}$ is complete, and a fixed rule is available for breaking ties between multiple maximal rewarding strategies for an agent.*

Proof: Consider some agent $i \in \mathcal{N}$, $|\mathcal{V}| > n$ and the descending order of prizes $\mathbf{p}_{desc} = (p_1, \ldots, p_u, p_{u+1}, \ldots p_{|\mathcal{V}|})$, where $p_u \geq p_{u+1}, \forall u \in \mathcal{N}$. The best action for i during the first stage game is to choose the i-th largest prize from $\mathbf{p}_{desc}$. Because, any deviation made by i to left in $\mathbf{p}_{desc}$, (e.g., $i-1$-th prize), will result in zero rewards since there is always a senior agent, $(i-1)$, who can override i. Any deviations to right will strictly reduce the reward. Therefore no agent can do better than choosing the prize that matches their rank. In case multiple maximal prizes are available, one may invoke a tie-breaking rule that is known to everyone, e.g., *agent with the lower rank choosing the prize with the minimum travel cost*. Then, re-sort $\mathbf{p}_{desc}$, and repeat the same procedure in the subsequent stages. If the i-th prize is 0, or l_i^t is not sufficient to reach another prize, then move to d. This concludes the pure Nash equilibrium strategy for any i for any $\mathbf{p}^0$. The uniqueness of the solution is guaranteed by the uniqueness of $\mathbf{p}_{desc}$ if no prize repeats, and by the tie-breaking rule, otherwise. ■

Remark 1. *It is clear from Theorem 1 that the PNE under a fixed tie-breaking rule also results in the PNEs in the stage games. Also, the total team reward under the PNE is equal to*

that of the optimal solution obtained by solving an equivalent TOP for a given L.

Next, consider $\mathcal{G}_s$ is a star graph, i.e., edge set $\mathcal{E}_s = \{(w,u) : u \in \mathcal{V}, u \neq w\}$, where w is some *staging node* that does not contain a prize. Then Theorem 1 also applies for G_s: the PNE strategy for the i-th agent is moving to the node with the i-th largest prize in every alternating stage.

Remark 2. *It is sufficient for an agent to only know its rank, the number of players, and the graph state to compute the PNE, instead of the full state vector, s^t that includes the private state components of the other agents.*

Theorem 2. *There exists some joint distribution of prizes $\times_{u\in\mathcal{V}}\omega_u(\bar{p}_u)$ and a graph $\mathcal{G}$, for which an SPCG played on an arbitrary $\mathcal{G}$ does not possess a PNE.*

Proof: This is a proof by counter-example. Consider the SPCG shown in figure 2, played on a graph $\mathcal{G}$. Consider two players $i, i+1$, referred to as *senior* and *junior*, are at s, and share the same destination $d_1 = d_2 = d$. Consider the pure strategies available to each player: $\pi^1 = (1,2)$, $\pi^2 = (2)$, $\pi^3 = (3)$ (we use the superscripts to avoid confusing with agents' strategies π_i). Here, a strategy $(u,v), u, v \in \mathcal{V}$ refers to the path from s to d through the nodes in the strategy, i.e., the path corresponding to strategy $(1,2)$ is $s, 1, 2, d$. Given that $L_{\max} = 3$, no strategy that contain a moving-back action will terminate at d, and is not profitable. Therefore, we exclude all such strategies from the payoff matrix (Table I). The rows indicate the strategies of the senior player. Since all the strategies contain the terminal reward p_d, we have also omitted it. Now consider the strategy pairs that contain the best responses of the junior player for any strategy of the senior player: $(\pi^1,\pi^2),(\pi^2,\pi^3),(\pi^3,\pi^1)$. Since no strategy of the senior player in them is a best response to the junior player's strategy, this game does not contain a Nash equilibrium in the pure strategies for any $\alpha \in (0,1)$. ∎

In [8], authors show that the leader-follower route-selection game has a Stackelberg equilibrium for any $\mathcal{G}$ as long as a fixed tie-breaking rule is imposed to handle multiple maximal strategies. However, Theorem 2 shows that such a result does not necessarily carry in SPCGs when agents move simultaneously regardless the graph is undirected or directed.

Next, referring to Remark 2, we generalize ordinal-response playing to the stage games in incomplete graphs. On an incomplete graph however, one's global rank does not directly correspond to the largest prize it can claim: an agent j with a lower rank, $j > 1$, may still claim the highest prize in its reachable set, as long as j is higher than its opponents' ranks in the neighborhood in the current stage game, i.e., immediate opponents. Next, we make an interjection to introduce the concept of *localized-stage game* to better clarify the concept of one's "immediate opponents".

Consider the reachable sets $S_1, \ldots, S_n$ (ignoring the time index since we focus on a single stage). First, we define a *separating graph* as a graph whose *connected components* (partitions) represent a stage game that is localized to some subset of agents. The formal definition follows below.

Definition 2. *(Separating Graph) The separating graph is defined as $\bar{\mathcal{G}} = (\bar{\mathcal{N}}, \bar{\mathcal{E}})$ whose set of nodes $\bar{\mathcal{N}} = \mathcal{N}$, and the edges $\bar{\mathcal{E}} = \{(i,j) : S_i \cap S_j \neq \emptyset, \forall i,j \in \bar{\mathcal{N}}\}$.*

The set of *connected components* in $\bar{\mathcal{G}}$ is $\Lambda = \{\lambda_y : \lambda_y \subseteq \bar{\mathcal{N}}, y = 1,2,\ldots\}$ where each λ_y is the largest connected component such that there is no path between any node in λ_y to any node in λ_z, for any $\lambda_y, \lambda_z \in \Lambda, y \neq z$. Further, for every $i \in \mathcal{N}$, define a mapping $\lambda(i) \in \Lambda$ such that $i \in \lambda(i)$. From the definition of Λ, $\lambda(i)$ is well-defined.

Definition 3. *(Localized-Stage Game) A subset of agents are participating in a localized stage game y if and only if they belong to the same connected component λ_y.*

In other words, the actions of any agent in the subset $\lambda(i)$, except i has an immediate impact on i's actions. Therefore, we may refer to this subset of agents except i as i's "immediate opponents". Finally, we introduce the concept of *ordinal rank*: a concept that effectively summarize valuable information on one's immediate opponents.

Definition 4. *(Ordinal Rank) For each agent $i \in \mathcal{N}$, we define its ordinal rank as*

$$\mathcal{I}_i = 1 + |\{j : j \in \lambda(i), j < i\}|. \tag{4}$$

Implicitly, $\mathcal{I}_i$ embeds the location and rank information of i's immediate opponents. In turn, this allows an agent to choose best-response actions to their immediate opponents even without knowing the global rank, and the locations of the others. Note that when $\mathcal{G}$ is complete, the separating graph has only one connected component at any stage of the SPCG, and thus, one's ordinal rank is equivalent to its global rank. Crucially, this approach helps us choose one's observation vector more carefully without sacrificing the quality of the solution. Further, on an incomplete graph, it allows us see *exactly* which information needs observing to reach a best-response solution for *myopic* agents. In the experiments, we observed that the strategies trained on local information conditioned on the OR significantly outperforms those trained on the global state, or conditioned on the global rank. In the next section, we will provide a graph-search algorithm to calculate the ordinal ranks of the agents.

V. Approach

As supported by Remark 2, we saw that the ordinal ranks carry significant information that enable best-responding to one's immediate opponents. If the agents are *myopic* –meaning they value immediate rewards more than those in future stages, we have shown that incorporating ordinal ranks results in best-responses to immediate opponents. In this section, we will show that the ordinal rank enables learning stationary strategies for the stochastic game, especially on graphs where best-responding to the immediate neighbors coincides with the PNE. When the game does not possess a PNE, we propose an algorithm to find a set of stationary strategies that are best responses to one's higher-ranked opponents via an alternating *fictitious play* approach.

Since an agent is now only required to best-respond to those who engage in the same localized-stage, it is sufficient for one

to know information about its localized game, and the prizes $\mathbf{p}^t$, rendering SPCGs solvable with partial observations. Thus, let $o_i^t \in \mathcal{O}_i$ define a local observation vector obtained from the mapping $\mathcal{O}_i : \mathcal{S} \to O_i$, where by O_i we denote i's *observation space*.

In this section, we will use the term "policy" instead of "strategy" to better align with the RL literature.

A. Ordinal Rank Search

Algorithm 1: ORS: Ordinal Rank Search

Input: Separating graph: $\bar{\mathcal{G}}$
Output: Groups of localized subsets: $\Lambda \subset 2^{\bar{\mathcal{N}}}$

```
1  visited ← {}; Λ ← {};
2  for i ∈ 𝒩 do
3      if i ∉ visited then
4          stack ← {i}; λ ← {};
5          while stack ≠ ∅ do
6              n ← pop(stack);
7              if n ∉ visited then
8                  visited ← visited ∪ {n};
9                  λ ← λ ∪ {n};
10                 stack ← stack ∪
                     {j : ∃(j, n) ∈ ℰ̄, ∀j ∈ 𝒩̄};
11             end
12         end
13         Λ ← Λ ∪ {λ} ;
14     end
15 end
16 return Λ
```

First, we proposes Ordinal Rank Search (ORS), outlined in algorithm 1, for identifying the subsets of agents engaging in localized stage games, and subsequently computing their ordinal ranks. Note that the algorithm here is presented in a centralized form, but it can also be decentralized if agents are able to communicate with their multi-hop neighbors.

The algorithm iterates through all nodes in $\bar{\mathcal{N}}$ by keeping track of the already visited ones in `visited`. For every unvisited node, 1) a depth-first traversal is launched by inserting its neighbors into the `stack` **(line 10)**, 2) removing them one-at-a-time while inserting them into a single connected component λ **(line 9)**. The `pop` operation retrieves and removes the top-most element of the stack. Once all the neighbors are visited, we insert λ into Λ **(line 13)**. The final output Λ contains all the connected components of $\bar{\mathcal{G}}$, each containing the set of agents engaged in an independent localized game whose actions directly depend on one-another. Once computed, it is straightforward to find the ordinal ranks of each agent in an component $\lambda_{\cdot} \in \Lambda$ by sorting their global ranks in the ascending order as per Eq. (4).

B. Fictitious Ordinal Response Learning

This subsection proposes an algorithm for learning stationary policies in SPCGs played on arbitrary graphs. In the proposing method, we sequentially update the agents' policies following their ranks one-at-a-time, assuming the others' policies remain stationary between updates —a concept central to the *fictitious-play* in classical game theory [16]. We will empirically show that this alternating fictitious play process will lead to best response policies against one's senior opponents when $\mathcal{G}$ is incomplete, and also yield PNE policies when $\mathcal{G}$ is complete.

Algorithm 2 outlines the proposed method. During the initial stage (*bootstrapping* stage), we only update the highest-ranked agent's policy, while the others follow the random policy (parameterized by $\theta^0_{\cdot}$). This stems from the intuition that any agent must first learn to solve the underlying single-agent orienteering problem under the environmental stochasticity which incorporates the senior agents' unknown actions, and the distribution of the prizes. The confidence of a policy is measured by its entropy that is, a policy with high entropy has a low confidence about its actions, and vice-versa. Once the highest-ranked agent's policy learns to solve the stochastic TOP with enough confidence, the learned parameters are shared with the other agents, therefore eliminating redundantly learning to solve the single agent OP. In the second stage (fictitious-play stage), the agents update their parameters independently, and sequentially by playing against the opponents' "stationary" policies.

Algorithm 2 outlines the complete procedure. We first initialize the algorithm with a hashmap Θ of random policy parameters of the agents, entropy-decaying step size $\delta h > 0$, an early-stopping entropy $\mathcal{H}_{\text{stop}}$, and an initial *freezing point* (IFP) $\mathcal{H}^0$. Further, $0 < \mathcal{H}_{\text{stop}} < \mathcal{H}^0 < \mathcal{H}_{\max}$, where $\mathcal{H}_{\max}$ is the highest possible entropy induced by the initial uniform random action selection, prize distribution, and the others' random behavior. Ignoring the entropy contribution of the others, and the prizes, it holds that $\mathcal{H}_{\max} \leq \bar{\mathcal{H}}_{\max}$ where $\bar{\mathcal{H}}_{\max} = -\ln \frac{1}{|\mathcal{V}|}$, the entropy of the uniform random policy. Let $t_{\max}$ be the number of timesteps to train. We use $\texttt{step}(a^t, s^t)$, and $\texttt{update}(\theta_j, r_j^t, o_j^{t-1}, a_j^t, o_j^t)$ functions to evolve the environment by a single timestep, and to update the parameters of θ_j with stochastic gradient-descent.

At the beginning, $j = 1$'s policy is updated while the others play randomly, until its entropy $\mathcal{H}[\pi_j^{\theta_j}]$ reaches the IFP that is $\mathcal{H}^0$ **(line 6)**. Then the parameters of the first policy are shared among the others, and the algorithm exits the bootstrapping stage **(line 8)**. Next, agent $j = 2$ enters the fictitious play stage to update its policy playing against the updated policy of $j = 1$, $\Theta[1]$, and the random policies of the others $\pi^{\theta^0}_{N\setminus[1,2]}$. This process repeats until the last agents' policy reaches the IFP **(line 11)**. Then we reduce the entropy at the next freezing point by δh amount **(line 12)**, and repeat the procedure. The algorithm terminates when the last agents' policy reaches the prescribed early-stopping entropy $\mathcal{H}_{\text{stop}}$ or till the total number of training timesteps $t_{\max}$ is reached **(line 14, 20)**. Finally, the hashmap of the most recent policies are returned **(line 21)**.

When the game does not possess a PNE, the entropy of the policies is unlikely to reach 0, assuming $\mathcal{H}_{\text{stop}}$ is set to 0.

The following proposition shows that Algorithm 2 always terminates with a joint policy π^* which is a PNE if one exists, and otherwise provides confidence bounds on the quality of the terminating policy.

Algorithm 2: FORL: Fictitious Ordinal Response Learning

Input: $\mathcal{H}^0$, δh, $\mathcal{H}_{\text{stop}}$, $\Theta = \{i : \theta_i^0; \forall i \in \mathcal{N}\}$

$\kappa(t)$, $j \leftarrow 1$, $\mathcal{H}^{\kappa(t)} \leftarrow \mathcal{H}^0$; ▷ Initialization
while $t \leq t_{\max}$ **do**
  $\theta_i \leftarrow \Theta[i], \forall i \in \mathcal{N}$;
  $a_i^t \leftarrow \pi_i^{\theta_i}[\cdot|O_i], \forall i \in \mathcal{N}$;
  $s^t, r^t \leftarrow \texttt{step}(a^t, s^{t-1})$; ▷ Environment step
  $\Theta[j] \leftarrow \texttt{update}(\theta_j, r_j^t, o_j^{t-1}, a_j^t, o_j^t)$;
  if $\mathcal{H}[\pi_j^{\theta_j}] \leq \mathcal{H}^{\kappa(t)}$ **then**
    if $j = 1 \wedge \kappa(t) = 1$ **then**
      $\Theta[i] \leftarrow \Theta[j]\ \forall i \in \mathcal{N}$; ▷ Bootstrapping
    end
    $j \leftarrow j + 1$; $\kappa(t) \leftarrow t$; ▷ Next policy
    if $j \geq n$ **then**
      $\mathcal{H}^{\kappa(t)} \leftarrow \mathcal{H}^{\kappa(t)} - \delta h$; ▷ Reduce entropy
      $j \leftarrow 1$; ▷ Reset j for next round
      if $\mathcal{H}^{\kappa(t)} < \mathcal{H}_{\text{stop}}$ **then**
        `break`; ▷ Exit while loop
      end
    end
  end
  $t \leftarrow t + 1$;
end
return Θ

Proposition 3. *Suppose $t_{\max}$ is chosen sufficiently large and $\mathcal{H}_{stop} = 0$. Then algorithm 2 will terminate with a joint policy $\pi^* = \times_{i\in\mathcal{N}}\pi_i^{\theta_i^*}$ which satisfies*

(1) if the game has a PNE, policy π^* is a PNE.

(2) otherwise, the confidence of the terminating policy π^* satisfies

$$\mathcal{H}(\pi_i^{\theta_i^*}) \in [\mathcal{H}^{\kappa(t_{\max}+1)}, \mathcal{H}^{\kappa(t_{\max})}],$$

for all $i \in \mathcal{N}$.

Proof: **(Case 1)**. If the game has a PNE, denoted by $\pi^{\#} = \times_{i\in\mathcal{N}}\pi_i^{\theta_i^{\#}}$, then $\mathcal{H}[\pi^{\#}] = 0$ since it is a pure strategy. The algorithm will not terminate until reaching $\mathcal{H}_{\text{stop}} = 0$ for a large enough $t_{\max}$. **(Case 2)**. If there exist no PNEs, the $\mathcal{H}_{\text{stop}}$ can not be reached, then the algorithm will terminate with $t = t_{\max}$ steps. From the sequential property of the algorithm, the relation

$$\mathcal{H}^{\kappa(t+1)} \leq \mathcal{H}(\pi_j^{\theta_j^t}) \leq \mathcal{H}(\pi_i^{\theta_i^t}) \leq \mathcal{H}^{\kappa(t)},$$

holds for all $j < i$. This implies that when the algorithm exits at $t = t_{\max}$, the confidence

$$\mathcal{H}[\pi_i^{\theta_i^*}] \in [\mathcal{H}^{\kappa(t+1)}, \mathcal{H}^{\kappa(t)}], \quad \forall i.$$

■

Furthermore, as a direct consequence of the previous theorem, the following corollary shows that the hyperparameter $\mathcal{H}_{\text{stop}}$ indicates the expected confidence level of the obtained policy π^*.

Corollary 4. *Given a large enough $t_{\max}$, if $\mathcal{H}_{\text{stop}}$ is set such that there exists a mixed NE policy $\tilde{\pi}^* = \times_{i\in\mathcal{N}}\pi_i^{\theta_i^*}$, and $\mathcal{H}_{\text{stop}} \geq \mathcal{H}[\tilde{\pi}^*]$, then Algorithm 2 terminates with a terminating policy π^* satisfying*

$$\mathcal{H}[\pi^*] \leq \mathcal{H}_{\text{stop}}.$$

In above, we use the *maximum entropy principle* to guarantee that FORL continues until either the last agent's policy meets $\mathcal{H}_{\text{stop}}$ or the maximum steps reached.

It is also possible to set $\mathcal{H}^0$ to $\mathcal{H}_{\max}$. In this case, the entropy of each policy must decay from $\mathcal{H}_{\max}$ to $\mathcal{H}_{\text{stop}}$ without bootstrapping as opposed to decaying from $\mathcal{H}^0$ to $\mathcal{H}_{\text{stop}}$ for all agents except for the first one. By setting $\mathcal{H}^0$ smaller than $\mathcal{H}_{\max}$, only the first agent's policy needs updating until $\mathcal{H}^0$, due to the bootstrapping. In the experiments, we used $\mathcal{H}^0$ between $0.8\mathcal{H}_{\max}$- $0.6\mathcal{H}_{\max}$. Further, due to the sequential-nature of FORL, the experiences of the other agents except for the one who is being trained are discarded; a pretext for poor sample efficiency, especially with large batches. However, compared to IPL, and parameter-sharing, FORL offered better training throughput with medium-sized batches (~ 2500).

C. Ordinal Rank Conditioning

We augment the agent's policy $\pi_i^{\theta_i}$ by incorporating the OR into its observation space. Specifically, instead of conditioning the policy solely on the local observation O_i, the policy is now conditioned on the augmented input $(O_i, \mathcal{I}_i)$. Thus, policy used here reads,

$$\pi_i^{\theta_i}\Big[A_i|O_i, \mathcal{I}_i\Big]. \tag{5}$$

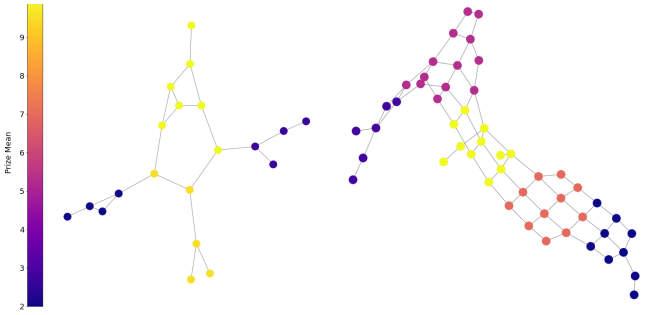
The OR conditioning introduces "state-aliasing" to the learning process thus, preventing the policies overfitting to global ranks: the same global rank may map to different ORs depending on one's neighborhood in the stages increasing the variance, and providing better interpretability for the accumulated experiences. This conditioning provides a particular advantage when the policy parameters are shared between the agents, i.e., $(\theta_1 = \theta_2 = \cdots = \theta_n)$: we observe that the policies generalize much better to large team sizes compared to those conditioned on the global rank. We attribute this behavior to the fact that, even when a policy was deployed on an agent with a global rank that was absent in the training, the policy may still have encountered its OR in the stages, thus choosing the correct ordinal response to its immediate opponents regardless of the global rank.

D. Action Masking and Policy Modeling

The observation space of an agent i used for training contains its local state $\mathcal{S}_i$, the prize state $\mathbf{p}^t$, ordinal rank $\mathcal{I}_i(t)$, and an *action mask* $\mathbf{m}_i^t \in \{1, 0\}^{|\mathcal{V}|}$ where

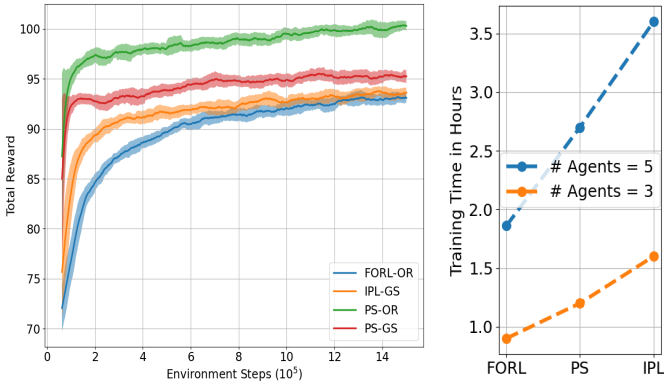
$$\mathbf{m}_i^t[u] = \begin{cases} 0; \text{ if } (x_i^t, u) \notin \mathcal{E}, \\ 1; \text{ otherwise.} \end{cases}$$

Here, $\mathbf{m}_i^t[u]$ denotes the mask corresponding to the u-th node $u \in \mathcal{V}$. Then the policy outputs corresponding to masked nodes are manually set to 0, and renormalized: $\pi_i^{\theta_i}(a_i = u|o_i^t) \leftarrow \frac{\pi_i^{\theta_i}(a_i^t=u|o_i^t)\odot\mathbf{m}_i^t[u]}{\sum_u \pi_i^{\theta_i}(a_i^t=u|o_i^t)}$ for all $u \in \mathcal{V}$, and '$\odot$' is the element-wise product between two vectors. Masking however, can result



(a) (b)

Fig. 3. Stockholm (left) and Manhattan (right) road network graphs. The graphs were obtained by applying the Fruchterman-Reingold algorithm on the original road networks. The colors correspond to the mean prize $(\bar{p}_u, u \in \mathcal{V})$ at the nodes where $\bar{p}_u \propto 1/\|\text{pos}(u) - \text{pos}(\bar{u})\|$, and $\bar{u}$ is the map center, and $\text{pos}(\cdot) : \mathbb{R} \to \mathbb{R}^2$.



(a) (b)

Fig. 4. (a) Convergence plots for a dynamic prize distribution graph with 5 competing agents. In each training method, "OR" and "GS" refers to local observations with Ordinal Rank (OR), and the Global observations (GS). (b) Training time for each method. The training was conducted on a computer equipped with an Nvidia RTX 3090 GPU, and Intel 12700k CPU.

in a lower $\mathcal{H}_{\max}$ compared to the non-masked case which is equivalent to the entropy of the uniform random policy with support $[0, |\mathcal{V}|]$, as the actions corresponding to non-neighboring nodes are explicitly ignored. Therefore, a safer method for setting the $\mathcal{H}^0$ is to use the empirical maximum entropy of the game at the beginning of the training process.

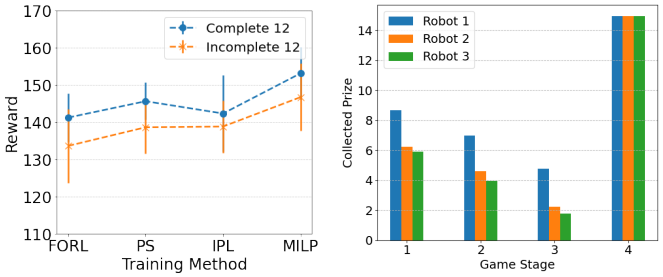
We adopt a Transformer-based neural network architecture, TRXL-I (see [17] for details), to model the policies. Each policy comprise of 6 transformer units that have been shared between the policy, and the value network. Each transformer unit has 6 attention heads of dimensionality of 128, and an attention memory of 10 observations. The output of the transformer sequence is passed to two linear layers in parallel: one for estimating the value-function output, and the second for computing the policy outputs.

VI. Experiments and Results

A. Simulation Environment

We implemented the SPCG on custom multi-agent environments defined on a $500\text{m} \times 500\text{m}$ area of Stockholm, Manhattan road networks, and on random graphs (see Fig. 3). Each road network graph consists of 20 and 50 nodes, where each *dead-end* (border) node served as a terminal node d with equal payoffs p_d. During an episode, the agents were initialized randomly, i.e., each $s_i, \forall i$ were chosen uniformly from non-terminal nodes. The nodes in the random graphs were sampled from the interval $[(-10, -10), (10, 10)]$ on the XY plane with fixed starting and destination nodes.

The prizes of non-terminal nodes were drawn from $\mathbf{p}^t \sim \times_{u \in \mathcal{V}} \texttt{Uniform}_u(0, 10)$ whereas each p_d were set to 15. The policies were updated using the Proximal Policy Optimization (PPO) algorithm with batches of 2500 observations, mini-batches of 200, and 10 stochastic gradient descent iterations per update.



(a) (b)

Fig. 5. (a) The optimality-gap between the SPCG and an equivalent TOP. (b) Stage-specific rewards of the agents in a complete graph under FORL.

B. Training Results

Fig. 4-(a) shows the training results for IPL, FORL, and parameter sharing (PS) methods in the Manhattan environment with dynamic prizes for a team of 5 agents. The policies trained using IPL were provided with the global-state vector (IPL-GS), FORL with the local information conditioned on the OR following Eq. (5) (FORL-OR), PS using both global and local information 5 vectors, denoted by PS-GS, PS-OR, respectively. The OR-conditioned policies trained using FORL converged to almost the same total reward profile as IPL-GS, and PS-GS indicating that OR can indeed successfully encode the global state in prize-collecting games. Interestingly, parameter shared policies conditioned on the OR, PS-OR, significantly outperformed PS-GS, and also the shared policies conditioned on the global rank. While we do not show the training curve of the latter for clarity, we provide a quantitative comparison between the two on their generalizability in Fig. 6-(b). Generally, training methods where the parameters of the policy or the value function are shared among the agents tend to score higher total rewards compared to independent learning methods such as FORL, IPL, due to agents being able to reason about the others' policies symmetrically. In addition, state-aliasing lets the shared policy collect more experiences for the same OR across the agents leading to more exploration, and thus, better performances.

Fig. 4-(b) shows the training time for each method. Noticeably, our method takes much less training time to reach approximately the same total team-reward profile as PS-GS, and IPL, with a speed-up factors 1.56 and 2.2, to train for

the same number of environment steps, and gradient descent iterations.

In Fig. 5, we report total rewards in SPCGs played on two graphs (complete and incomplete random graphs that has 12 nodes) with stationary prizes for each training method, and compared them to an equivalent TOP whose solution was obtained by solving a mixed-integer linear program (MILP) as presented in [18] using Gurobi. In the complete graph, PS-OR reports the highest total reward that is 95% the optimal solution. In the incomplete graph, the worst case equilibria was obtained with FORL with a total reward that is 87% of the optimal solution –*a quantity formally known as the "Price-of-Anarchy" (PoA) in game theory*.

Fig. 5-(b) shows the prizes collected by the agents at each stage in a game 4 stages ($L = 4$), on a complete graph (only the prizes of the first three agents trained using FORL are shown). The PNE of the game states that the prizes collected at each stage obey the agents' ranks. Similarly, we observe that the senior-most agent (robot 1), collecting the highest prize at each stage, while robot 2 and robot 3 picking the second, and the third largest prizes. In the final stage, all the agents reach the terminal node receiving the terminal reward. Nearly zero conflicts were observed during the evaluations.

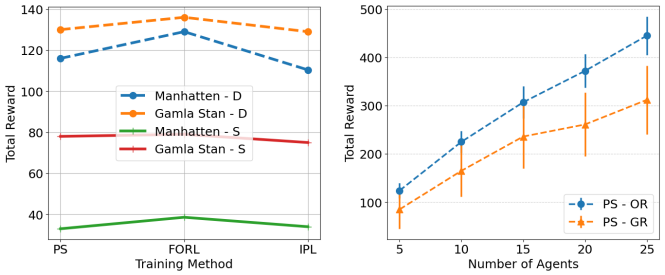
C. Generalizability and Scalability

We evaluated the policies' generalizability by *zero-shot* transferring them onto an imbalanced prize distribution that resembles a demand-driven taxi-fare distribution in a city, i.e.,$p_u^t \sim \text{Normal}(\beta_u, \sigma)$, where β_u is the mean prize at a node that is inversely proportional to the proximity to the map center, whereas σ is the standard deviation. Fig. 3 shows the joint distribution of the prizes in the two city graphs partitioned into four and five price zones. The policies trained using FORL outperformed both PS-OR, and IPL-GS in generalizing to imbalanced prize distributions in both the cities, under both the stationary and dynamic settings (see Fig. 6-a).

Finally, we evaluated the scalability of the shared policies to different team sizes on the Manhattan environment with dynamic prizes. Both of the considered policies were trained using local information, but, conditioned on the ordinal rank (PS-OR), and the global rank (PS-GR). The total reward under PS-OR scaled linearly up to teams of 25 agents with only marginal increment in the variance, whereas PS-GR showed poor, sublinear, total-reward gains with much higher variance for the same team sizes (see Fig. 6-b).

VII. Conclusion

We introduced SPCG, a Markov game formulation of the TOP for planning in self-interested robot (agent) teams operating in stochastic, graph-structured environments. A theoretical analysis conducted on complete graphs shows that there exists a PNE that coincides with the optimal routing solution under a rank-based conflict resolution. We propose two algorithms to efficiently learn best-response policies: 1) ORS algorithm computes the OR of a robot during the stages of the game to induce state-aliasing to prevent policies from overfitting to the global index thus resulting policies that scale linearly to different team sizes, 2) FORL sequentially updates the policies which converges faster and generalizes better to imbalanced prize distributions than IPL, and parameter sharing. Compared to a MILP solution of an equivalent TOP, the learned equilibrium policies in SPCGs show a PoA of 87% 95%.



(a) (b)

Fig. 6. (a) Zero-shot transfer to region-specific, normally-distributed prizes. Here "D", "S" denote (D)ynamic and (S)tationary environments. (b) Scalability of the shared policy with ordinal rank conditioning (PS-OR) compared to the global rank conditioning (PS-GR).

References

[1] 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.

[2] A. Mansfield, S. Manjanna, D. G. Macharet, and M. Ani Hsieh, "Multi-robot scheduling for environmental monitoring as a team orienteering problem," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6398–6404, 2021.

[3] Z. Ma, K. Yin, L. Liu, and G. S. Sukhatme, "A spatio-temporal representation for the orienteering problem with time-varying profits," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6785–6792, 2017.

[4] A. Gunawan, H. C. Lau, and P. Vansteenwegen, "Orienteering Problem: A survey of recent variants, solution approaches and applications," *European Journal of Operational Research*, vol. 255, pp. 315–332, Dec. 2016.

[5] J. Park, C. Kwon, and J. Park, "Learn to solve the min-max multiple traveling salesmen problem with reinforcement learning,"

[6] E. Álvarez Miranda, M. Sinnl, and K. Tanınmış, "Competing for the most profitable tour: The orienteering interdiction game," July 2024. arXiv:2407.02959 [math].

[7] P. Varakantham, H. Mostafa, N. Fu, and H. C. Lau, "DIRECT: A scalable approach for route guidance in Selfish Orienteering Problems,"

[8] T. Murray, J. Garg, and R. Nagi, "Prize collecting multiagent orienteering: Price of anarchy bounds and solution methods," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 1, pp. 531–544, 2020.

[9] Y. Gui, Z. Zhang, D. Tang, H. Zhu, and Y. Zhang, "Collaborative dynamic scheduling in a self-organizing manufacturing system using multi-agent reinforcement learning," *Advanced Engineering Informatics*, vol. 62, p. 102646, 2024.

[10] T. Yu, J. Huang, and Q. Chang, "Optimizing task scheduling in human-robot collaboration with deep multi-agent reinforcement learning," *Journal of manufacturing systems*, vol. 60, pp. 487–499, 2021.

[11] Y. Shen, X. Wang, H. Wang, Y. Guo, X. Chen, and J. Han, "A dynamic task assignment model for aviation emergency rescue based on multi-agent reinforcement learning," *Journal of Safety Science and Resilience*, vol. 4, no. 3, pp. 284–293, 2023.

[12] M. Fernando, R. Senanayake, H. Choi, and M. Swany, "Graph attention multi-agent fleet autonomy for advanced air mobility," *ArXiv*, vol. abs/2302.07337, 2023.

[13] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE transactions on robotics*, vol. 25, no. 4, pp. 912–926, 2009.

[14] V. Gkatzelis, K. Kollias, and T. Roughgarden, "Optimal cost-sharing in general resource selection games," *Operations Research*, vol. 64, no. 6, pp. 1230–1238, 2016.

[15] J. Hu and M. P. Wellman, "Nash q-learning for general-sum stochastic games," *Journal of machine learning research*, vol. 4, no. Nov, pp. 1039–1069, 2003.
[16] Y. Shoham, "Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations," p. 532.
[17] E. Parisotto, H. F. Song, J. W. Rae, R. Pascanu, C. Gulcehre, S. M. Jayakumar, M. Jaderberg, R. L. Kaufman, A. Clark, S. Noury, M. M. Botvinick, N. Heess, and R. Hadsell, "Stabilizing Transformers for Reinforcement Learning," Oct. 2019. arXiv:1910.06764 [cs, stat].
[18] A. Gunawan, H. C. Lau, and P. Vansteenwegen, "Orienteering problem: A survey of recent variants, solution approaches and applications," *European Journal of Operational Research*, vol. 255, no. 2, pp. 315–332, 2016.