Bandit Submodular Maximization for Multi-Robot Coordination in Unpredictable and Partially Observable Environments

Zirui Xu*, Xiaofeng Lin†, Vasileios Tzoumas*
*Department of Aerospace Engineering, †Department of Robotics
University of Michigan
{ziruixu,linxiaof,vtzoumas}@umich.edu

***Abstract*—We study the problem of multi-agent coordination in *unpredictable* and *partially observable* environments, that is, environments whose future evolution is unknown a priori and that can only be partially observed. We are motivated by the future of autonomy that involves multiple robots coordinating actions in dynamic, unstructured, and partially observable environments to complete complex tasks such as target tracking, environmental mapping, and area monitoring. Such tasks are often modeled as submodular maximization coordination problems due to the information overlap among the robots. We introduce the first submodular coordination algorithm with bandit feedback and bounded tracking regret —*bandit feedback* is the robots' ability to compute in hindsight only the effect of their chosen actions, instead of all the alternative actions that they could have chosen instead, due to the partial observability; and *tracking regret* is the algorithm's suboptimality with respect to the optimal time-varying actions that fully know the future a priori. The bound gracefully degrades with the environments' capacity to change adversarially, quantifying how often the robots should re-select actions to learn to coordinate as if they fully knew the future a priori. The algorithm generalizes the seminal Sequential Greedy algorithm by Fisher et al. to the bandit setting, by leveraging submodularity and algorithms for the problem of *tracking the best action*. We validate our algorithm in simulated scenarios of multi-target tracking.**

I. Introduction

In the future, autonomous robots will be collaboratively planning actions in complex tasks such as target tracking [1], environmental mapping [2], and area monitoring [3]. Such multi-robot tasks have been modeled by researchers in robotics and control via maximization problems of the form

$$\max_{a_{i,t}\in\mathcal{V}_i,\forall\, i\in\mathcal{N}} \; f_t(\{a_{i,t}\}_{i\in\mathcal{N}}), \quad t=1,2,\ldots, \tag{1}$$

where $\mathcal{N}$ is the robot set, $a_{i,t}$ is robot i's action at time step t, $\mathcal{V}_i$ is robot i's set of available actions, and $f_t : 2^{\prod_{i\in\mathcal{N}}\mathcal{V}_i} \mapsto \mathbb{R}$ is the objective function that captures the task utility at time step t. Specifically, the objective function f_t is considered computable prior to each time step t given a model about the future evolution of the environment [1]–[11]; *e.g.*, in target tracking, a stochastic model for the targets' future motion is often considered available, and then f_t can be chosen for example as the mutual information between the position of the robots and that of the targets [2].

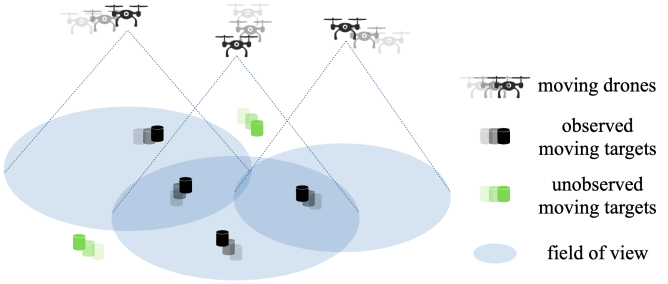


Fig. 1: **Example of Multi-Robot Coordination in Unpredictable and Partially Observable Environments: Target Tracking.** In this paper, we focus on multi-robot coordination tasks where the robots' capacity to select effective actions is compromised by (i) a lack of knowledge about how the environment will evolve, and (ii) a lack of full observability of the environment's evolution. For example, in target tracking, drones are often tasked to coordinate their motion to maximize at each time step the number of tracked targets. But in adversarial scenarios, (i) the robots may be unaware of the targets' intentions and motion model, thus being unable to plan effective actions by simulating the future, and (ii) the robots may carry sensors with a limited field of view, thus being unable to reason even in hindsight whether alternative actions could have been more effective in tracking targets. Notwithstanding the said challenges, in this paper, we aim to provide a general-purpose coordination algorithm achieving bounded suboptimality against the optimal multi-robot actions in hindsight.

The optimization problem in eq. (1) is NP-hard [12] but near-optimal approximation algorithms are possible in polynomial time when f_t has a special structure, especially, when f_t is *submodular* [13] —submodularity is a diminishing returns property, and in multi-robot information gathering tasks it emanates due to the possible information overlap among the information gathered by the robots [14]. One celebrated approximation algorithm for eq. (1) is the *Sequential Greedy* algorithm [13], which achieves a near-optimal $1/2$ approximation bound when f_t is submodular. All the above multi-robot tasks and more, from target tracking and environmental exploration to collaborative mapping and area monitoring, can be modeled as submodular coordination problems, and thus, Sequential Greedy and its variants have been commonly used in robotics [1]–[11], [14]–[18].

But the application of the Sequential Greedy algorithm and its variants can be hindered in challenging environments that are unpredictable and partially observable:

a) Unpredictable Environments: The said complex tasks often evolve in environments that change unpredictably, *i.e.*,

in environments whose future evolution is unknown a priori. For example, during target tracking the targets' actions can be unpredictable when their intentions and maneuvering capacity are unknown [19]. In such challenging environments, the robots that are tasked to track the targets cannot simulate the future to compute f_t prior to time step t, *i.e.*, the robots cannot utilize the Sequential Greedy algorithms and its variants [1]–[11]. Instead, the robots have to coordinate their actions online by relying on past information only, *i.e.*, by relying only on the retrospective reward of their actions upon the observation of the environment's evolution.

Such online coordination algorithms have been recently proposed in [20]. Particularly, [20] provides a submodular coordination algorithm with guaranteed suboptimality against the robots' optimal *time-varying* actions in hindsight —the optimal actions ought to be time-varying to be effective against a changing environment such as an evading target.[1]

b) Partially Observable Environments: But online coordination methods such as [20] become inapplicable when the unpredictable environments are only partially observable: when an environment is partially observable, online learning methods can only compute the utility of their executed actions. Particularly, they cannot compute in hindsight the utility of actions they could have selected alternatively to execute —this partial-information feedback is known as *bandit feedback* [27]. Bandit feedback hence compromises the capacity of online learning methods to learn near-optimal action policies based on past information only. Take the target tracking scenario in Fig. 1 as an example: since the drones have limited fields of view, they can observe only part of the targets (those inside the field of view), being unaware of any unobserved targets (those outside the field of view); consequently, the robots cannot compute even in hindsight how many targets they would have seen instead if they had chosen alternative actions.

Although recent contributions [28]–[33] have focused on multi-robot coordination subject to bandit feedback, they consider partially observable environments where (i) the environments' state evolution is governed by an unknown stochastic i.i.d model, and (ii) the robots' goal is to learn actions that maximize the sum of the robots' individual rewards without accounting for the possible information overlap among the information gathered by the robots; *e.g.*, in the context of Fig. 1, the sum of the robots' individual rewards is 6 since the drones on the left, center, and right observe 2, 3, and 1 targets, respectively.

Goal. In this paper, we focus instead on unpredictable and partially observable environments where: (i) the environments' state is non-stochastic and even adversarial, that is, the environment's behavior is not governed by a probability model and can even be adaptive to the robots' action, *e.g.*, where the robots are tasked to track targets that can adapt their motion to the robots' motion; and (ii) the robots' goal is to learn actions for each time step t that maximize a global objective f_t that is submodular, instead of a mere addition of the individual rewards of the robots. Accounting for the submodularity structure is crucial since it quantifies the possible information overlap among the information gathered by the robots; *e.g.*, in the context of Fig. 1, the number of tracked targets by the drones is 4, instead of 6 as we computed above when we ignored the information overlap.

Contributions. We provide the first bandit submodular optimization algorithm for multi-robot coordination in unpredictable and partially observable environments (Section III). We name the algorithm *Bandit Sequential Greedy* (BSG). BSG generalizes the Sequential Greedy algorithm [13] from the setting where each f_t is fully known a priori to the bandit setting. BSG has the properties:

- *Computational Complexity*: For each agent i, BSG requires only one function evaluation and $O(\log T)$ additions and multiplications per agent per round (Section IV-A).
- *Approximation Performance*: BSG guarantees bounded *tracking regret* (Section IV-B), *i.e.*, bounded suboptimality with respect to optimal time-varying actions that know the future a priori. The bound gracefully degrades with the environments' capacity to change adversarially, quantifying how often the robots should re-select actions to learn to coordinate as if they knew the future a priori. In more detail, the bound guarantees that the agents select actions that asymptotically and in expectation match the near-optimal performance of the Sequential Greedy algorithm [13] in known environments.

To enable BSG, we make the technical contributions:

1) First, we enable each robot to retrospectively estimate the reward of all its available actions despite the bandit feedback. To this end, we use as a subroutine on-board each robot a novel algorithm for the problem *tracking the best action with bandit feedback* [34] —we are inspired to this end by [20], [35], which leverage similar subroutines for online submodular optimization problems in fully observable environments.[2] Particularly, although the current algorithm for the problem of *tracking the best action with bandit feedback*, namely, EXP3-SIX [36], can guarantee a bounded tracking regret for that problem, it requires the a priori knowledge of a parameter capturing how fast the environment is going to change. Satisfying such a requirement is typically infeasible in practice. Therefore, in this paper, we use a "doubling trick" [37] to extend EXP3-SIX to an algorithm that requires no more the a priori knowledge of this parameter (Section III-A); we name the algorithm EXP3*-SIX (Algorithm 1).

2) Then, we leverage (i) EXP3*-SIX's regret guarantee for the problem of tracking the best action with bandit feedback (Theorem 1), (ii) BSG's steps (Algorithm 2), and (iii) f_t's

[1] Additional algorithms have been proposed for the case where in eq. (1) f_t is unknown a priori but these algorithms apply to tasks where the optimal actions are *static* [21]–[26], instead of time-varying, guaranteeing bounded suboptimality with respect to optimal time-invariant actions.

[2] [20] focuses on online submodular coordination in fully observable environments, instead of partially observable environments. Further, [35] focuses on the problem of *cardinality-constrained submodular maximization in fully observable environments*, which has the form $\max_{\mathcal{S} \subseteq \mathcal{V}, |\mathcal{S}| \leq k} f(\mathcal{S})$, given an integer k and an $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$, and is thus distinct from eq. (1).

submodularity to prove BSG's tracking regret guarantee for the coordination problem of this paper (Appendix).

Numerical Evaluations. We evaluate BSG in simulated scenarios of target tracking with multiple robots (Section V), where the robots carry noisy sensors with limited field of view to observe the targets. We consider scenarios where 2 robots pursue 2, 3, or 4 targets. For each scenario, we first consider non-adversarial targets and, then, adversarial targets: the non-adversarial targets traverse predefined trajectories, independently of the robots' motion; whereas, the adversarial targets maneuver in response to the robots' motion. *In both cases, the targets' future motion and maneuvering capacity are unknown to the robots.* Across the simulations, BSG encourages the robots to maximize their tracking capability, also enabling collaborative behaviors such as robots switching targets to improve speed compatibility (fast robot vs. fast target, instead of slow robot vs. fast target).

II. Bandit Submodular Coordination with Bounded Tracking-Regret

We define the problem *Bandit Submodular Coordination*. To this end, we use the notation:

- $\mathcal{V}_{\mathcal{N}} \triangleq \prod_{i\in\mathcal{N}} \mathcal{V}_i$ is the cross product of sets $\{\mathcal{V}_i\}_{i\in\mathcal{N}}$.
- $[T] \triangleq \{1,\dots,T\}$ for any positive integer T;
- $f(\,a\,|\,\mathcal{A}\,) \triangleq f(\,\mathcal{A}\cup\{a\}\,) - f(\,\mathcal{A}\,)$ is the marginal gain of set function $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ for adding $a \in \mathcal{V}$ to $\mathcal{A} \subseteq \mathcal{V}$.
- $|\mathcal{A}|$ is the cardinality of a discrete set $\mathcal{A}$.

The following preliminary framework is also considered.

Agents. $\mathcal{N}$ is the set of all agents —the terms "*agent*" and "*robot*" are used interchangeably in this paper. The agents coordinate actions to complete a task. To this end, they can observe one another's selected actions at each time step.

Actions. $\mathcal{V}_i$ is a *discrete* and *finite* set of actions available and always known to robot i. For example, $\mathcal{V}_i$ may be a set of motion primitives that robot i can execute to move in the environment [6] or robot i's discretized control inputs [2].

Environment. E_t is the state of the environment at time step t. E_t evolves (possibly adversarially) with the agents' past actions up to $t-1$. Also, E_t is unpredictable prior to time t, in particular, the robots are unaware of a model capturing the future evolution of the environment. For example, in the multi-target tracking scenario in Fig. 1, where the robots have no model about the future motion of the targets, at time $t-1$ the robots cannot know where the targets will be at time t.

Observable Environment. $E_t^{obs}(\{a_{i,t}\}_{i\in\mathcal{N}})$ is the part of E_t observed by the robots once the robots have executed their actions $\{a_{i,t}\}_{i\in\mathcal{N}}$ at time step t. For example, in Fig. 1, while E_t includes all targets' positions, $E_t^{obs}(\{a_{i,t}\}_{i\in\mathcal{N}})$ includes only the positions of the targets that are within the collaborative field of view (black-colored targets).

Objective Function. The agents coordinate their actions $\{a_{i,t}\}_{i\in\mathcal{N}}$ to maximize an objective function $f(\{a_{i,t}\}_{i\in\mathcal{N}}, E_t)$ —we explicitly note the dependence of the value f of the actions on the state of the environment. In Fig. 1 for example, f is equal to 4 since four targets are within the field of view of the robots given the robots' and targets' positions at time t. We henceforth consider

$$f_t(\cdot) \triangleq f(\cdot, E_t). \tag{2}$$

Bandit Feedback. $f(\cdot, E_t)$ is unknown prior to the execution of the actions $\{a_{i,t}\}_{i\in\mathcal{N}}$ since E_t is unknown before time t. Upon the execution of the actions $\{a_{i,t}\}_{i\in\mathcal{N}}$, if E_t is fully observable, then the robots can evaluate $f(\mathcal{A}, E_t)$ for all $\mathcal{A} \subseteq \{\mathcal{V}_i\}_{i\in\mathcal{N}}$; *i.e.*, the robots can evaluate in hindsight the performance of any actions that they could have chosen instead for time t. But in this paper, the environment is generally partially observable, hence, upon the execution of the actions $\{a_{i,t}\}_{i\in\mathcal{N}}$, the robots can only evaluate $f(\mathcal{A}, E_t^{obs}(\mathcal{A}))$ for all $\mathcal{A} \subseteq \{a_{i,t}\}_{i\in\mathcal{N}}$. We refer to the two said cases of information feedback as *full feedback* and *bandit feedback*, per similar definitions in the literature of online learning and optimization [27].

Assumption 1 (Exact Function Evaluation Despite Partially Observable Environments)**.** *We assume coordination tasks where for all* $\mathcal{A} \subseteq \{\mathcal{V}_i\}_{i\in\mathcal{N}}$,

$$f(\mathcal{A}, E_t^{obs}(\mathcal{A})) \equiv f(\mathcal{A}, E_t). \tag{3}$$

Coordination tasks that satisfy Assumption 1 include target tracking [1], environmental mapping [2], and area monitoring [11], where, intuitively, the objective function is defined based on observed information only; *e.g.*, in the multi-target tracking scenario in Fig. 1, the robots know exactly how many targets are within their field of view, thus Assumption 1 holds true when f is the number of targets within the robots' field of view. In contrast, Assumption 1 would be violated if the objective in Fig. 1 were to minimize the distance between each target and its nearest drone: then, the drones cannot possibly evaluate their distance to the unobservable targets (colored green in the figure) and, hence, Assumption 1 cannot hold true. In all, satisfying Assumption 1 implies that what the robots perceive onboard exactly equals what they really achieve in a partially observable environment.

Submodular Structure. In information gathering tasks such as target tracking, environmental mapping, and area monitoring, typical objective functions are the covering functions [1]–[3]. These functions capture how much area/information is covered given the actions of all robots. They satisfy the properties defined below (Definition 1).

Definition 1 (Normalized and Non-Decreasing Submodular Set Function [13])**.** *A set function* $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ *is* normalized and non-decreasing submodular *if and only if*

- *(Normalization)* $f(\,\emptyset\,) = 0$*;*
- *(Monotonicity)* $f(\,\mathcal{A}\,) \leq f(\,\mathcal{B}\,)$*,* $\forall\, \mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$*;*
- *(Submodularity)* $f(\,s\,|\,\mathcal{A}\,) \geq f(\,s\,|\,\mathcal{B}\,)$*,* $\forall\, \mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$ *and* $s \in \mathcal{V}$*.*

Normalization holds without loss of generality. In contrast, monotonicity and submodularity are intrinsic to the function. Intuitively, if $f(\,\mathcal{A}\,)$ captures the number of targets tracked by a set $\mathcal{A}$ of drones, then the more drones are deployed, no fewer

targets are covered; this is the monotonically non-decreasing property. Also, the marginal gain of tracked targets caused by deploying a drone s *drops* when *more* drones are already deployed; this is the submodularity property.

Problem Definition. In this paper, we focus on:

Problem 1 (Bandit Submodular Coordination)**.** *Assume a time horizon H of operation discretized to T time steps. At each time step $t \in [T]$, the agents $\mathcal{N}$ select actions $\{a_{i,t}\}_{i\in\mathcal{N}}$* online *such that they solve*

$$\max_{a_{i,t}\in\mathcal{V}_i, \forall\, i\in\mathcal{N}} \; f_t(\{a_{i,t}\}_{i\in\mathcal{N}}), \tag{4}$$

where $f_t : 2^{\prod_{i\in\mathcal{N}}\mathcal{V}_i} \mapsto \mathbb{R}$ is a normalized and non-decreasing submodular set function, and the agents can access the values of $f_t(\mathcal{A})$ only after they have selected $\{a_{i,t}\}_{i\in\mathcal{N}}$, $\forall\, \mathcal{A} \subseteq \{a_{i,t}\}_{i\in\mathcal{N}}$.

Remark 1 (Adversarial Environment and Randomized Algorithm)**.** *Dependent on the agents' past actions, the environment E_t in Problem 1 can be* adversarial*, in that it can decide f_t at each time step t to change for worsening the reward of $\{a_{i,t}\}_{i\in\mathcal{N}}$. E_t here serves as the* adversary *in a bandit problem [38]. In this paper, we provide a randomized algorithm that guarantees in expectation a suboptimality bound that degenerates gracefully as the environment becomes more adversarial. If E_t makes f_t change arbitrarily much between consecutive time steps, then inevitably no algorithm can guarantee a near-optimal performance.*

III. Bandit Sequential Greedy (BSG) Algorithm

We present the Bandit Sequential Greedy (BSG) algorithm for Problem 1. BSG leverages as subroutine an algorithm we introduce for the problem of *tracking the best action with bandit feedback*. Thus, before we present BSG in Section III-B, we first present the algorithm for *tracking the best action with bandit feedback* in Section III-A.

A. The EXP3⋆-SIX Algorithm for Tracking the Best Action with Bandit Feedback

Tracking the best action with bandit feedback is an *adversarial bandit* problem [27]. It involves an agent —instead of the entire team— selecting a sequence of actions to maximize the total reward over a given number of time steps. The challenge is dual: (i) the reward associated with each action is decided by the environment at each time step and unknown to the agent until the action has been executed; and (ii) the agent receives only bandit feedback of the rewards. To solve the problem, the agent needs to leverage past observation of the rewards till the last time step to predict the best action that achieves the highest reward for this time step.

To formally state the problem, we use the notation:

- $\mathcal{V}$ denotes the agents' available action set;
- $a_t \in \mathcal{V}$ denotes the agent's selected action at time t;
- $a_t^\star$ denotes the best action that achieves the highest reward among $\mathcal{V}$ at t;
- $r_{a_t,t} \in [0,1]$ denotes the reward that the agent receives by selecting action a_t at t;

Algorithm 1: EXP3⋆-SIX.

Input: Number of time steps T and action set $\mathcal{V}$.
Output: Probability distribution $p_t \in \{[0,1]^{|\mathcal{V}|} : \|p_t\|_1 = 1\}$ over the actions in $\mathcal{V}$ at each $t \in [T]$.

$J \leftarrow \lceil \log_2 T \rceil$, $\eta \leftarrow \sqrt{\log J \,/\, (2T)}$, $\beta \leftarrow 1 \,/\, (T-1)$;
$\eta^{(j)} \leftarrow \sqrt{\log\left(|\mathcal{V}|\, T\right) / \left(2^{j-1}\, |\mathcal{V}|\right)}$, $\gamma^{(j)} = \eta^{(j)}/2$, for all $j \in [J]$;
$z_1 \leftarrow [z_{1,1}, \ldots, z_{J,1}]^\top$ with $z_{j,1} = 1$, for all $j \in [J]$;
$w_1^{(j)} \leftarrow \left[w_{1,1}^{(j)}, \ldots, w_{|\mathcal{V}|,1}^{(j)}\right]^\top$ with $w_{i,1}^{(j)} = 1$, for all $t \in [T]$ and $i \in \mathcal{V}$;
for each time step $t \in [T]$ **do**
  $q_t \leftarrow z_t \,/\, \|z_t\|_1$, $p_t^{(j)} \leftarrow w_t^{(j)} \,/\, \|w_t^{(j)}\|_1$, for all $j \in [J]$;
  output $p_t \leftarrow \sum_{j=1}^{J} q_{j,t}\, p_t^{(j)}$;
  receive the reward $r_{a_t,t} \in [0,1]$ of selecting the action $a_t \in \mathcal{V}$ at time step t;
  for $j \in [J]$ **do**
    $\tilde{r}_{i,t}^{(j)} \leftarrow 1 - \frac{\mathbf{1}(a_t = i)}{p_{i,t} + \gamma^{(j)}}(1 - r_{a_t,t})$, for all $i \in \mathcal{V}$;
    $\tilde{r}_t^{(j)} \leftarrow \left[\tilde{r}_{1,t}^{(j)}, \ldots, \tilde{r}_{|\mathcal{V}|,t}^{(j)}\right]^\top$;
    $v_{i,t}^{(j)} \leftarrow w_{i,t}^{(j)} \exp\left(\eta^{(j)}\, \tilde{r}_{i,t}^{(j)}\right)$, for all $i \in \mathcal{V}$;
    $W_t^{(j)} \leftarrow v_{1,t}^{(j)} + \cdots + v_{|\mathcal{V}|,t}^{(j)}$;
    $w_{i,t+1}^{(j)} \leftarrow \beta\, \frac{W_t^{(j)}}{|\mathcal{V}|} + (1-\beta)\, v_{i,t}^{(j)}$, for all $i \in \mathcal{V}$;
    $z_{j,t+1} \leftarrow z_{j,t} \exp\left(\eta\, \tilde{r}_t^{(j)\top}\, p_t^{(j)}\right)$;
  end for
end for

- $\tilde{r}_t \in [0,1]^{|\mathcal{V}|}$ denotes the estimation of the rewards of all actions available to the agent at t;
- $\mathbf{1}(\cdot)$ is the indicator function, *i.e.*, $\mathbf{1}(x) = 1$ if the event x is true, otherwise $\mathbf{1}(x) = 0$.
- $P(T) \triangleq \sum_{t=1}^{T-1} \mathbf{1}(a_t^\star \neq a_{t+1}^\star)$ counts how many times the best action changes over T time steps due to the adversary (the environment).

Problem 2 (Tracking the Best Action with Bandit Feedback [36])**.** *Assume a time horizon H of operation discretized to T time steps. The agent selects an action a_t* online *at each time step $t \in [T]$ to solve the optimization problem*

$$\max_{a_t\in\mathcal{V},\, t\in[T]} \; \sum_{t=1}^{T} r_{a_t,t}, \tag{5}$$

where only the reward $r_{a_t,t} \in [0,1]$ becomes known to the agent and only once *a_t has been selected.*

A randomized algorithm is needed for Problem 2, given that the environment can adversarially adapt to the agent's previously selected actions to seek to minimize the agent's total reward. If an algorithm for Problem 2 is deterministic, then the environment can know a priori the action a_t to be selected by the deterministic algorithm for each time step t and accordingly choose the rewards $r_{a_t,t} = 0$ and $r_{a_t',t} = 1$, $\forall a_t' \in \mathcal{V}, a_t' \neq a_t$. This will lead to $\sum_{t=1}^{T} r_{a_t^\star,t} - r_{a_t,t} \geq T(1 - 1/|\mathcal{V}|)$, which means a_t can never converge to $a_t^\star$, $\forall t \in [T]$

[27, Chapter 11.1]. Therefore, at each time step t, we need a randomized algorithm to provide a probability distribution p_t over the action set $\mathcal{V}$, from which the agent can draw the action a_t for time step t.

Moreover, a desired randomized algorithm for Problem 2 should ensure $\mathbb{E}\left[\sum_{t=1}^{T} r_{a_t^\star,t} - r_{a_t,t}\right] = \sum_{t=1}^{T} r_{a_t^\star,t} - r_t^\top p_t$ is sublinear, where the expectation results from the internal randomness of the algorithm, such that as $T \to \infty$, $\frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} r_{a_t^\star,t} - r_{a_t,t}\right] \to 0$, and thus $a_t \to a_t^\star$.

Although EXP3-SIX [36] is an algorithm that achieves a sublinear $\sum_{t=1}^{T} r_{a_t^\star,t} - r_t^\top p_t$, it requires the value of $P(T)$ for picking a "learning rate" that can bound the suboptimality. The learning rate decides how fast the algorithm adapts to the environmental change. But $P(T)$ is unknown a priori. Thus, in this paper we leverage a "doubling trick" technique, common in the literature of online learning [35], [39], and present a new algorithm, EXP3*-SIX, that overcomes EXP3-SIX's said limitation. Specifically, EXP3*-SIX uses the multiplicative weight update (MWU) method [40] to synthesize the results of multiple subroutines of EXP3-SIX with different learning rates (lines 9-16), at least one of which is close enough to the learning rate computed using $P(T)$.

In more detail, Algorithm 1 initializes and maintains J subroutines of EXP3-SIX, each associated with a weight $z_{j,t}$ and a different learning rate $\eta^{(j)}$, $j \in [J]$. For each $j \in [J]$, a weight $w_{i,t}^{(j)}$ is assigned to each available action $i \in \mathcal{V}$ (lines 1-4). At each time step $t \in [T]$, Algorithm 1 first uses MWU to compute the probability distribution p_t based on $\{w_{i,t}^{(j)}\}_{j\in[J]}$ and $\{z_{j,t}\}_{j\in[J]}$ (lines 5-6). Then, after outputting p_t and observing the new reward $r_{a_t,t}$ (lines 7-8), Algorithm 1 computes an estimate $\{\tilde{r}_t^{(j)}\}_{j\in[J]}$ for all available actions' rewards (lines 9-11). $\{\tilde{r}_t^{(j)}\}_{j\in[J]}$ is an optimistically biased estimate of $r_{a_t,t}$, *i.e.*, larger than the unbiased estimate. The smaller is $r_{a_t,t}$, the larger is $(\gamma^{(j)}(1-r_{a_t,t}))/(p_{i,t}(p_{i,t}+\gamma^{(j)}))$, *i.e.*, the amount of the bias of $\tilde{r}_t^{(j)}$. Therefore, the smaller is $r_{a_t,t}$, the more Algorithm 1 is encouraged to explore. Finally, for the next time step, Algorithm 1 updates $\{w_{i,t+1}^{(j)}\}_{j\in[J]}$ with the Fixed Share Forecaster [40] steps (lines 12-14) and obtains $\{z_{j,t+1}\}_{j\in[J]}$ using MWU (line 15). The higher is $\eta^{(j)}$, the more $\{w_{i,t}^{(j)}\}_{j\in[J]}$ depend on the recently observed rewards and, thus, the faster EXP3-SIX j adapts to the environment changes.

Theorem 1 (Performance Guarantee of EXP3*-SIX)**.** *For Problem 2,* EXP3*-SIX *guarantees*

$$\sum_{t=1}^{T} r_{a_t^\star,t} - r_t^\top p_t \tag{6}$$
$$\leq \tilde{O}\left[\sqrt{T|\mathcal{V}|(P(T)+1)}\right] + \log\left(\frac{1}{\delta}\right)\left[\tilde{O}\left(\sqrt{\frac{T|\mathcal{V}|}{P(T)+1}}\right)+1\right]$$

with probability at least $1-\delta$*, where* $\delta \in (0,1)$ *is the confidence level, and* $\tilde{O}(\cdot)$ *hides* log *terms.*

Theorem 1 implies $\frac{1}{T}\sum_{t=1}^{T} r_{a_t^\star,t} - r_t^\top p_t \to 0$ as $T \to \infty$ when $P(T)$ is sublinear in T, that is $P(T)/T \to 0$ for $T \to +\infty$, *i.e.*, when the optimal action $a_t^\star$ does not change too frequently across time steps; *i.e.*, in expectation the agent is able to track the best sequence of actions with high probability as T increases.[3,4]

Algorithm 2: Bandit Sequential Greedy (BSG).

Input: Time steps T and agents' action sets $\{\mathcal{V}_i\}_{i\in\mathcal{N}}$.
Output: Agent actions $\{a_{i,t}^{\mathsf{BSG}}\}_{i\in\mathcal{N}}$ at each $t \in [T]$.

1: Each agent $i \in \mathcal{N}$ initializes an EXP3*-SIX with the value of the parameters T and $\mathcal{V}_i$;
2: Denote the EXP3*-SIX onboard agent i by EXP3*-SIX$|_i$;
3: Order the agents in $\mathcal{N}$ such that $\mathcal{N} = \{1,\ldots,|\mathcal{N}|\}$;
4: **for** each time step $t \in [T]$ **do**
5: **for** $i = 1,\ldots,|\mathcal{N}|$ **do**
6: **get** the output $p_t^{(i)}$ from EXP3*-SIX$|_i$;
7: **draw** an action $a_{i,t}^{\mathsf{BSG}}$ from the distribution $p_t^{(i)}$;
8: **end for**
9: **execute** $\{a_{i,t}^{\mathsf{BSG}}\}_{i\in\mathcal{N}}$;
10: $\mathcal{A}_{0,t}^{\mathsf{BSG}} \leftarrow \emptyset$;
11: **for** $i = 1,\ldots,|\mathcal{N}|$ **do**
12: $\mathcal{A}_{i,t}^{\mathsf{BSG}} \leftarrow \mathcal{A}_{i-1,t}^{\mathsf{BSG}} \cup \{a_{i,t}^{\mathsf{BSG}}\}$;
13: **observe** $f_t(\mathcal{A}_{i,t}^{\mathsf{BSG}})$;
14: $r_{i,t}^{\mathsf{BSG}} \leftarrow f_t(a_{i,t}^{\mathsf{BSG}} \mid \mathcal{A}_{i-1,t}^{\mathsf{BSG}})$;
15: **input** $r_{i,t}^{\mathsf{BSG}}$ to EXP3*-SIX$|_i$ (per line 8 of Algorithm 1);
16: **end for**
17: **end for**

B. The BSG Algorithm

We present BSG (Algorithm 2). BSG generalizes the Sequential Greedy (SG) algorithm [13] to the online setting of Problem 1, leveraging at the agent level EXP3*-SIX. Particularly, when f_t is known a priori, instead of unknown per Problem 1, then SG instructs the agents to sequentially select actions $\{a_{i,t}^{\mathsf{SG}}\}_{i\in\mathcal{N}}$ at each $t-1$ such that

$$a_{i,t}^{\mathsf{SG}} \in \max_{a\in\mathcal{V}_i} f_t(a \mid \{a_{1,t}^{\mathsf{SG}},\ldots,a_{i-1,t}^{\mathsf{SG}}\}), \tag{7}$$

i.e., agent i selects $a_{i,t}^{\mathsf{SG}}$ after agent $i-1$, given the actions of all previous agents $\{1,\ldots,i-1\}$, such that $a_{i,t}^{\mathsf{SG}}$ maximizes the marginal reward given the actions of all previous agents from 1 to $i-1$. But since f_t is unknown and even adversarial per Problem 1, BSG replaces the deterministic action-selection rule of eq. (7) with a *tracking the best action* rule (cf. Remark 1). Thus, BSG is also a sequential algorithm.

BSG starts by instructing each agent $i \in \mathcal{N}$ to initialize an EXP3*-SIX —we denote the EXP3*-SIX onboard for each agent i as EXP3*-SIX$|_i$. Agent i initializes EXP3*-SIX$|_i$ with the number T of total time steps and with its action set $\mathcal{V}_i$ as inputs (line 1). Then, at each time step $t \in [T]$, in sequence:

[3] EXP3*-SIX's suboptimality bound in Theorem 1 is of the same $\tilde{O}(\cdot)$-order as EXP3-SIX's bound, despite EXP3*-SIX not knowing $P(T)$ a priori: in Theorem 1's proof, we show EXP3*-SIX's bound contains only additional log terms with respect to T and $|\mathcal{V}|$ when compared to EXP3-SIX's bound.

[4] The term $\log\left(\frac{1}{\delta}\right)\left[\tilde{O}\left(\sqrt{T|\mathcal{V}|/(P(T)+1)}\right)+1\right]$ in eq. (6) is always sublinear in T since it is bounded by $\log\left(\frac{1}{\delta}\right)\left[\tilde{O}\left(\sqrt{T|\mathcal{V}|}\right)+1\right]$.

- Each agent i draws an action $a_{i,t}^{\mathsf{BSG}}$ given the probability distribution $p_t^{(i)}$ output by EXP3*-SIX$|_i$ (lines 5-8).
- All agents execute their actions $\{a_{i,t}^{\mathsf{BSG}}\}_{i\in\mathcal{N}}$ (line 9).
- Each agent i receives from agent $i-1$ the actions of all agents with a lower index, $\mathcal{A}_{i-1,t}^{\mathsf{BSG}}$, and then observes $f_t(\mathcal{A}_{i,t}^{\mathsf{BSG}})$ (lines 10-13).
- Finally, each agent i computes $r_{i,t}^{\mathsf{BSG}}$, the reward (marginal gain) of $a_{i,t}^{\mathsf{BSG}}$ given $\mathcal{A}_{i-1,t}^{\mathsf{BSG}}$, and inputs $r_{i,t}^{\mathsf{BSG}}$ to EXP3*-SIX$|_i$ per line 8 of Algorithm 1 (lines 14-15). With this input, EXP3*-SIX$|_i$ will compute $p_{t+1}^{(i)}$, *i.e.*, the probability distribution over the agent i's actions for time step $t+1$.

IV. Performance Guarantees of BSG

We present the computational complexity (Section IV-A) and approximation performance (Section IV-B) of BSG.

A. Computational Complexity of BSG

BSG is the first algorithm for Problem 1 with polynomial computational complexity, quantified below.

Proposition 1 (Computational Complexity)**.** BSG *requires each agent $i\in\mathcal{N}$ to perform T function evaluations and $O(T\log T)$ additions and multiplications over T rounds.*

The proposition holds true since at each $t\in[T]$, BSG requires each agent i to perform 1 function evaluation to compute the marginal gain in BSG's line 14 and $O(\log T)$ additions and multiplications to run EXP3*-SIX$|_i$.

Remark 2 (Direct Application of EXP3*-SIX to Problem 1 Requires Exponential Running Time in $|\mathcal{N}|$)**.** EXP3*-SIX *may be directly applied to Problem 1, resulting however an exponential time algorithm since* EXP3*-SIX *would then require $O(T\log T\prod_{i\in\mathcal{N}}|\mathcal{V}_i|)$ additions and multiplications.*

B. Approximation Performance of BSG

We bound BSG's suboptimality with respect to the optimal actions the agents' would select if they knew the $\{f_t\}_{t\in[T]}$ a priori. Particularly, we bound BSG's *tracking regret*, proving that it gracefully degrades with the environment's capacity to select $\{f_t\}_{t\in[T]}$ adversarially (Theorem 2).

To present Theorem 2, we first define tracking regret, particularly, $1/2$-*approximate tracking regret* (Definition 2), and then we quantify the environment's capacity to select $\{f_t\}_{t\in[T]}$ adversarially (Definition 3). We use the notation:

- $\mathcal{A}_t^{\mathsf{OPT}}\in\arg\max_{a_{i,t}\in\mathcal{V}_i,\forall i\in\mathcal{N}} f_t(\{a_{i,t}\}_{i\in\mathcal{N}})$ is the optimal actions the agents would select for time step t if they fully knew f_t a priori;
- $a_{i,t}^{\mathsf{OPT}}$ is agent i's action among the actions in $\mathcal{A}_t^{\mathsf{OPT}}$;
- $\mathcal{A}_t\triangleq\{a_{i,t}\}_{i\in\mathcal{N}}$ is the set of all agents' actions at t.

Definition 2 ($1/2$-Approximate Tracking Regret)**.** *Consider an arbitrary sequence of action sets $\{\mathcal{A}_t\}_{t\in[T]}$. $\{\mathcal{A}_t\}_{t\in[T]}$'s $1/2$-approximate tracking regret is*[5]

$$\textsf{Tracking-Regret}_T^{(1/2)}(\{\mathcal{A}_t\}_{t\in[T]})\triangleq\frac{1}{2}\sum_{t=1}^{T}f(\mathcal{A}_t^{\mathsf{OPT}},E_t)-\sum_{t=1}^{T}f(\mathcal{A}_t,E_t).\tag{8}$$

Equation (8) can be further simplified as follows:

$$\begin{aligned}&\textsf{Tracking-Regret}_T^{(1/2)}(\{\mathcal{A}_t\}_{t\in[T]})\\&=\frac{1}{2}\sum_{t=1}^{T}f(\mathcal{A}_t^{\mathsf{OPT}},E_t^{obs}(\mathcal{A}_t^{\mathsf{OPT}}))-\sum_{t=1}^{T}f(\mathcal{A}_t,E_t^{obs}(\mathcal{A}_t))\\&=\frac{1}{2}\sum_{t=1}^{T}f_t(\mathcal{A}_t^{\mathsf{OPT}})-\sum_{t=1}^{T}f_t(\mathcal{A}_t),\end{aligned}\tag{9}$$

per eqs. (2) and (3). In more detail, eq. (9) evaluates $\{\mathcal{A}_t\}_{t\in[T]}$'s suboptimality against the optimal actions $\{\mathcal{A}_t^{\mathsf{OPT}}\}_{t\in[T]}$ the agents would select if they knew the $\{f_t\}_{t\in[T]}$ a priori. The optimal total value $\sum_{t=1}^{T}f_t(\mathcal{A}_t^{\mathsf{OPT}})$ is discounted by $1/2$ in eq. (8) since solving exactly Problem 1 is NP-hard even when $\{f_t\}_{t\in[T]}$ are known a priori [42]. Specifically, the best possible approximation bound in polynomial time is the $1-1/e$ [42], while the Sequential Greedy algorithm [13], that BSG extends to the bandit online setting, achieves the near-optimal bound $1/2$. In this paper, we prove that BSG can approximate *Sequential Greedy*'s near-optimal performance by bounding eq. (8).

Definition 3 (Environment's Adversarial Effect [20])**.** *The environment's* adversarial effect *on (i) agent i is*

$$\Delta_i(T)\triangleq\sum_{t=1}^{T-1}\mathbf{1}(a_{i,t}^{\mathsf{OPT}}\neq a_{i,t+1}^{\mathsf{OPT}}),\tag{10}$$

and on (ii) all the agents $\mathcal{N}$ is

$$\Delta(T)\triangleq\sum_{i\in\mathcal{N}}\Delta_i(T)=\sum_{t=1}^{T-1}\sum_{i\in\mathcal{N}}\mathbf{1}(a_{i,t}^{\mathsf{OPT}}\neq a_{i,t+1}^{\mathsf{OPT}}).\tag{11}$$

$\Delta(T)$ captures the environment's total effect on selecting $\{f_t\}_{t\in[T]}$ adversarially by counting how many times the optimal actions of the agents must shift across the T steps to adapt to the changing f_t. The larger the environment capacity to adversarially select $\{f_t\}_{t\in[T]}$, the larger the $\Delta(T)$, and the harder for the agents to adapt to near-optimal actions.

Theorem 2 (Approximation Performance)**.** BSG *instructs the agents to select actions $\{a_{i,t}^{\mathsf{BSG}}\}_{i\in\mathcal{N},t\in[T]}$ that guarantee*

$$\begin{aligned}&\mathbb{E}\left[\textsf{Tracking-Regret}_T^{(1/2)}(\{a_{i,t}^{\mathsf{BSG}}\}_{i\in\mathcal{N},t\in[T]})\right]\\&\leq\underbrace{\tilde{O}\left[\sqrt{T|\mathcal{N}||\bar{\mathcal{V}}|(\Delta(T)+|\mathcal{N}|)}\right]}_{\phi_1}\\&\quad+\log\left(\frac{1}{\delta}\right)\underbrace{\tilde{O}\left[\sqrt{T|\mathcal{N}|\sum_{i\in\mathcal{N}}\frac{|\mathcal{V}_i|}{\Delta_i(T)+1}}+|\mathcal{N}|\right]}_{\phi_2}\end{aligned}\tag{12}$$

[5]Definition 2 generalizes existing notions of tracking regret [35], [41] to the online submodular coordination Problem 1.

holds with probability at least $1-\delta$*, for any* $\delta \in (0,1)$*, where the expectation is due to* BSG*'s internal randomness,* $|\bar{\mathcal{V}}| \triangleq \max_{i\in\mathcal{N}}|\mathcal{V}_i|$*, and* $\tilde{O}(\cdot)$ *hides* log *terms.*

The proof of Theorem 2 is presented in Appendix B. Theorem 2 bounds the tracking regret of BSG, as a function of the number of robots, the total time steps T, and the environment's total adversarial effect. If the environment's total adversarial effect grows slow enough with T such that

$$\tilde{O}\left[\sqrt{|\mathcal{N}|\,T\,|\bar{\mathcal{V}}|\,(\,\Delta(T)+|\mathcal{N}|\,)}\right]\Big/T \to 0 \text{ for } T\to+\infty, \quad (13)$$

then eq. (12) implies $f_t(\mathcal{A}_t) \to 1/2 f_t(\mathcal{A}_t^{\mathsf{OPT}})$ in expectation since then both ϕ_1 and ϕ_2 in eq. (12) are sublinear in T.[6] In other words, when eq. (13) holds true, then BSG enables the agents to asymptotically learn (adapt) to coordinate as if they knew $f_1,\ldots,f_T$ a priori, matching the performance of the near-optimal SG. For example, eq. (13) holds true in environments whose evolution is unknown yet predefined, instead of being adaptive to the agents' actions. Then, $\Delta(T)$ is uniformly bounded since increasing the discretization density of time horizon H, *i.e.*, increasing the number of time steps T, does not affect the environment's evolution. Thus, $\Delta(T)/T \to 0$ for $T\to+\infty$, which implies eq. (13). The result agrees with the intuition that the agents should be able to adapt to an unknown but non-adversarial environment when they re-select actions with high enough frequency.

V. Numerical Evaluation in Multi-Target Tracking Tasks with Multiple Robots

We evaluate BSG in simulated scenarios of target tracking with multiple robots, where the robots carry noisy sensors with limited field of view to observe the targets. We consider scenarios where 2 robots pursue 2, 3, or 4 targets. For each scenario, we first consider non-adversarial targets and, then, adversarial targets: the non-adversarial targets traverse predefined trajectories, independently of the robots' motion; whereas, the adversarial targets maneuver in response to the robots' motion. *In both cases, the targets' future motion and maneuvering capacity are unknown to the robots.*

[6]The term ϕ_2 in eq. (12) is always sublinear, *i.e.*, $\phi_2/T\to 0$ for $T\to+\infty$, since ϕ_2 is bounded by $\log\left(\frac{1}{\delta}\right)\tilde{O}\left[|\mathcal{N}|\left(\sqrt{T|\bar{\mathcal{V}}|}+1\right)\right]$.

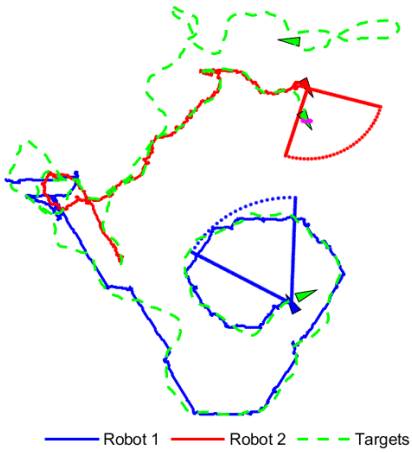


Fig. 2: **Target-Tracking Instance of** 2 **Robots Tracking** 3 **Targets.** Both the robots and the targets move on the same plane. Each robot can collect range and bearing measurements of a target but only if the target is inside the robot's field of view. The measurements are assumed corrupted with zero-mean Gaussian noise, with increasing variance the further away the target is from the robot. The targets' motion model is unknown to the robots, thus the robots coordinate based only on past observations.

Particularly, we first evaluate BSG's effectiveness at different action-selection frequencies (10, 20, 50, and 100Hz). To this end, we consider scenarios of 2 robots pursuing 2 non-adversarial targets in Section V-A, validating the theoretical results in Section IV. Then, we evaluate BSG's effectiveness in enabling the robots to pursue the targets. To this end, we consider scenarios where 2 robots pursue 2, 3, or 4 targets, first focusing on non-adversarial targets (Section V-B) and, then, on adversarial targets (Section V-C). We also compare BSG's performance with a greedy heuristic, showcasing BSG's superiority. We provide video demonstrations for all simulation scenarios at https://bit.ly/3WlxcUy.

Common Simulation Setup across Simulated Scenarios.

a) Targets: The targets move on a 2D plane. We introduce the targets' motion model within each particular scenario considered in Section V-A and Section V-C.

Henceforth, $\mathcal{T}$ denotes the set of targets.

b) Robots: The robots move in the same 2D environment as the targets. To move in the environment, each robot $i\in\mathcal{N}$ can perform one of the actions $\mathcal{V}_i \triangleq$ {"upward", "downward", "left", "right", "upleft", "upright", "downleft", "downright"} at a constant speed.

c) Sensing: We consider that each robot i has a range and bearing sensor to collect measurements about the targets' position inside its field of view. After selecting actions $\{a_{i,t}\}_{i\in\mathcal{N}}$ at time t, the robots share their measurements with one another, enabling each robot i to have an estimate of $d_t(a_{i,t},\,j)$, the distance from robot i to target j, given that j is observed by a robot as a result of actions $\{a_{i,t}\}_{i\in\mathcal{N}}$.

d) Objective Function: The robots coordinate their actions $\{a_{i,t}\}_{i\in\mathcal{N}}$ to maximize at each time step t[7]

$$f_t(\,\{a_{i,t}\}_{i\in\mathcal{N}}\,) = \sum_{j\in\mathcal{T}}\left[-\sum_{i\in\mathcal{N}_j}\frac{1}{d_t(a_{i,t},\,j)}\right]^{-1}, \quad (14)$$

where $\mathcal{N}_j$ is the set of robots that can observe the target j, $d_t(a_{i,t},\,j)$ is the distance between robot i and the estimated location of target j. Therefore, $d_t(a_{i,t},\,j)$ is known only once robot i has executed its action $a_{i,t}$ and the location estimate of target j at time step t has been computed. In particular, it is assumed that the total number of targets in the environment is known to the robots such that Assumption 1 is satisfied.

By maximizing f_t, the robots aim to collaboratively keep the targets inside their field of view. For example, when no robot has target j inside its field of view, *i.e.*, when $\mathcal{N}_j=\emptyset$, which is equivalent to target j being infinitely far away from all robots, it is $\left[-\sum_{i\in\mathcal{N}_j}\,1/d_t(a_{i,t},\,j)\right]^{-1} = -\infty$ —to account

[7]The objective function in eq. (14) is a non-decreasing and submodular function. The proof is presented in the Appendix.

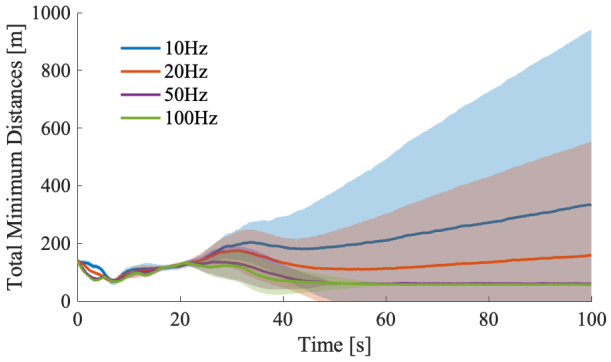


Fig. 3: BSG**'s Performance for Various Action-Selection Frequencies.** Four frequencies are depicted, in a target tracking scenario where 2 robots pursue 3 non-adversarial targets; the targets predefined trajectories are shown in Fig. 4(d). The results are averaged over 50 Monte-Carlo trials.

for the feasibility of our implementation, when $\mathcal{N}_j = \emptyset$, we set $\left[-\sum_{i \in \mathcal{N}_j} 1/d_t(a_{i,t}, j)\right]^{-1} = -4d_{max}$, where d_{max} is the largest sensing range among the robots. On the other end of the spectrum, when a robot i achieves 0 estimated distance from a target j, *i.e.*, when $d_t(a_{i,t}, j) = 0$, then indeed $\left[-\sum_{i \in \mathcal{N}_j} 1/d_t(a_{i,t}, j)\right]^{-1} = 0$.

e) Performance Metric: To measure how closely the robots track the targets, we consider a *total minimum distance* metric. We define the metric as the sum of the distances between each target and its nearest robot, whether this target is observed by any robot or not.

Computer System Specifications. We ran all simulations in MATLAB 2022b on a Windows laptop equipped with the Intel Core i7-10750H CPU @ 2.60 GHz and 16 GB RAM.

Code. Our code is available at: https://github.com/UM-iRaL/bandit-sequential-greedy.

A. Evaluation of BSG at Various Reaction Frequencies

We evaluate the capacity of BSG to improve its performance when the robots' action-selection frequency increases. Particularly, we test BSG when the robots' action-selection frequency increases from 10Hz to 20Hz to 50Hz to 100Hz, in a scenario where 2 robots pursue 3 non-adversarial targets whose predefined trajectories are shown in Fig. 4(d).

Results. The simulation results are presented in Fig. 3, averaged across 50 Monte-Carlo trials. They validate the analysis in Section IV-B, specifically, that the higher is the action-selection frequency the better BSG learns and, thus, the closer the robots pursue the targets. Particularly, although at 10Hz, the robots fail to "learn" the targets' future motion, failing to reduce their distance to them, the situation improves at 20Hz, and even further at 50Hz and 100Hz. In the latter two cases, the robots closely track the targets, maintaining on average a non-increasing distance to them, proportional to the field of view of the robots: the field of view of the robots has a radius of 150m, and the achieved total minimum distance at 50Hz and 100Hz is less than 100m.

B. Evaluation of BSG in Non-Adversarial Target Tracking

We evaluate BSG in simulated target tracking scenarios where the targets are non-adversarial, *i.e.*, they traverse predefined trajectories that are non-adaptive to the robots' locations. To this end, we first describe a heuristic baseline against which we compare BSG and the simulation setup.

Benchmark Algorithm. We compare BSG with a heuristic version of the Sequential Greedy that selects actions at each t based on the previous f_{t-1}. We denote the algorithm by SG-Heuristic. SG-Heuristic selects actions per the rule:

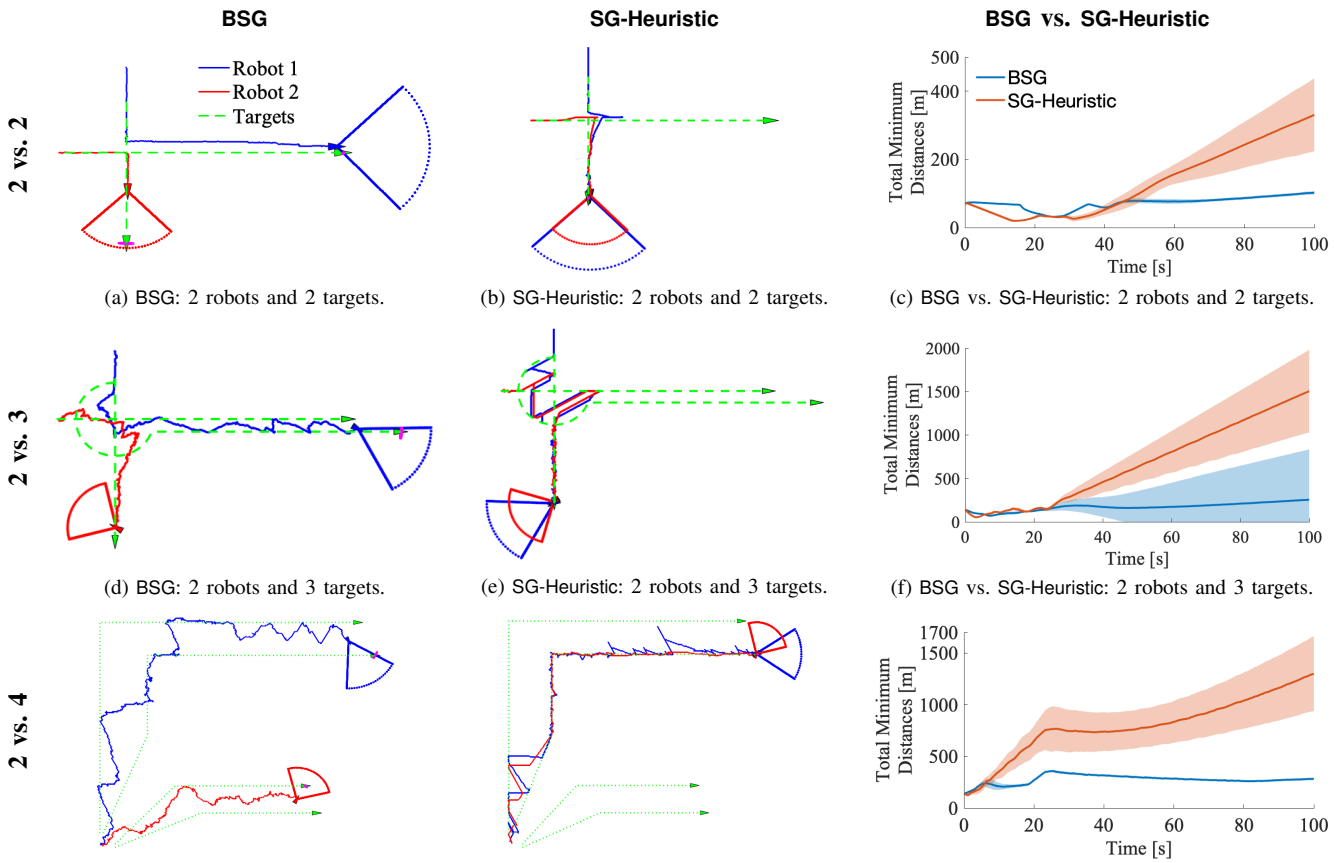
$$a_{i,t}^{\text{SG-Heuristic}} \in \max_{a \in \mathcal{V}_i} f_{t-1}(a \mid \{a_{1,t}^{\text{SG-Heuristic}}, \ldots, a_{i-1,t}^{\text{SG-Heuristic}}\}). \quad (15)$$

Simulation Setup. We consider three scenarios of non-adversarial target tracking: (i) 2 robots vs. 2 targets, where the targets traverse straight lines with a crossing (Fig. 4(a)–(c)); (ii) 2 robots vs. 3 targets, where the targets traverse straight lines and circles with a crossing (Fig. 4(d)–(f)); and (iii) 2 robots vs. 4 targets, where the targets diverse and traverse straight lines with turns (Fig. 4(g)–(i)). Each robot and target have different speeds, but we assume that all targets move with less speed than the robots. In all scenarios, the robots re-select actions with frequency 20Hz. We evaluate the algorithms across 50 Monte-Carlo trials.

Results. The simulation results are presented in Fig. 4. The following observations are due: (i) BSG outperforms SG-Heuristic in all scenarios (Fig. 4(c)(f)(i)). In all the cases of 2 robots vs. 2 targets (Fig. 4(a)–(c)), 2 robots vs. 3 targets (Fig. 4(d)–(f)), and of 2 robots vs. 4 targets (Fig. 4(g)–(i)), BSG maintains near-constant distances to the targets. (ii) BSG enables collaborative behaviors such as robots switching targets to improve speed compatibility. Particularly, in all Fig. 4(a)(d)(g), we observe that the robots eventually switch their corresponding targets to chase. The reason is that the faster robots can match the faster targets. *This desirable switching behavior may emerge even though the robots are unaware of the targets' speed and overall motion model.* (iii) SG-Heuristic instructs the robots to chase only one group of targets once the targets disperse. This is because SG-Heuristic is based *merely* on the outdated detected target locations of the last time step. But looking at only the last time step can be misleading. For example, if a robot loses all targets at a time step, then it will have nothing to feed into the SG-Heuristic in the next time step and, thus, it will then start to randomly scan all of its actions until it tracks some group of targets again. Then, the robot will keep tracking the new targets and may never get back to the past ones. In contrast, BSG uses the reward information of all past selected actions to predict the best actions for the robots, such that a robot can track the same targets again even after it has lost them.

C. Evaluation of BSG in Adversarial Target Tracking

We evaluate BSG in simulated target tracking scenarios where the targets are adversarial, *i.e.*, they traverse predefined trajectories that are adaptive to the robots' locations.



(a) BSG: 2 robots and 2 targets. (b) SG-Heuristic: 2 robots and 2 targets. (c) BSG vs. SG-Heuristic: 2 robots and 2 targets.

(d) BSG: 2 robots and 3 targets. (e) SG-Heuristic: 2 robots and 3 targets. (f) BSG vs. SG-Heuristic: 2 robots and 3 targets.

(g) BSG: 2 robots and 4 targets. (h) SG-Heuristic: 2 robots and 4 targets. (i) BSG vs. SG-Heuristic: 2 robots and 4 targets.

Fig. 4: **Non-Adversarial Target Tracking with Multiple Robots: 2 Robots Pursuing 2, 3, or 4 Targets.** The robots select actions either per BSG, or per the greedy heuristic SG-Heuristic, re-selecting actions with frequency 20Hz. Across the two algorithm cases, the targets traverse the same predefined trajectories, which are non-adaptive to the robots' motion. (a),(d),(g): The robots use BSG against 2, 3, and 4 targets, respectively; (b),(e),(h): the robots use SG-Heuristic against 2, 3, and 4 targets, respectively. (c),(f),(i): Comparison of BSG's and SG-Heuristic's average effectiveness over 50 Monte-Carlo trials.

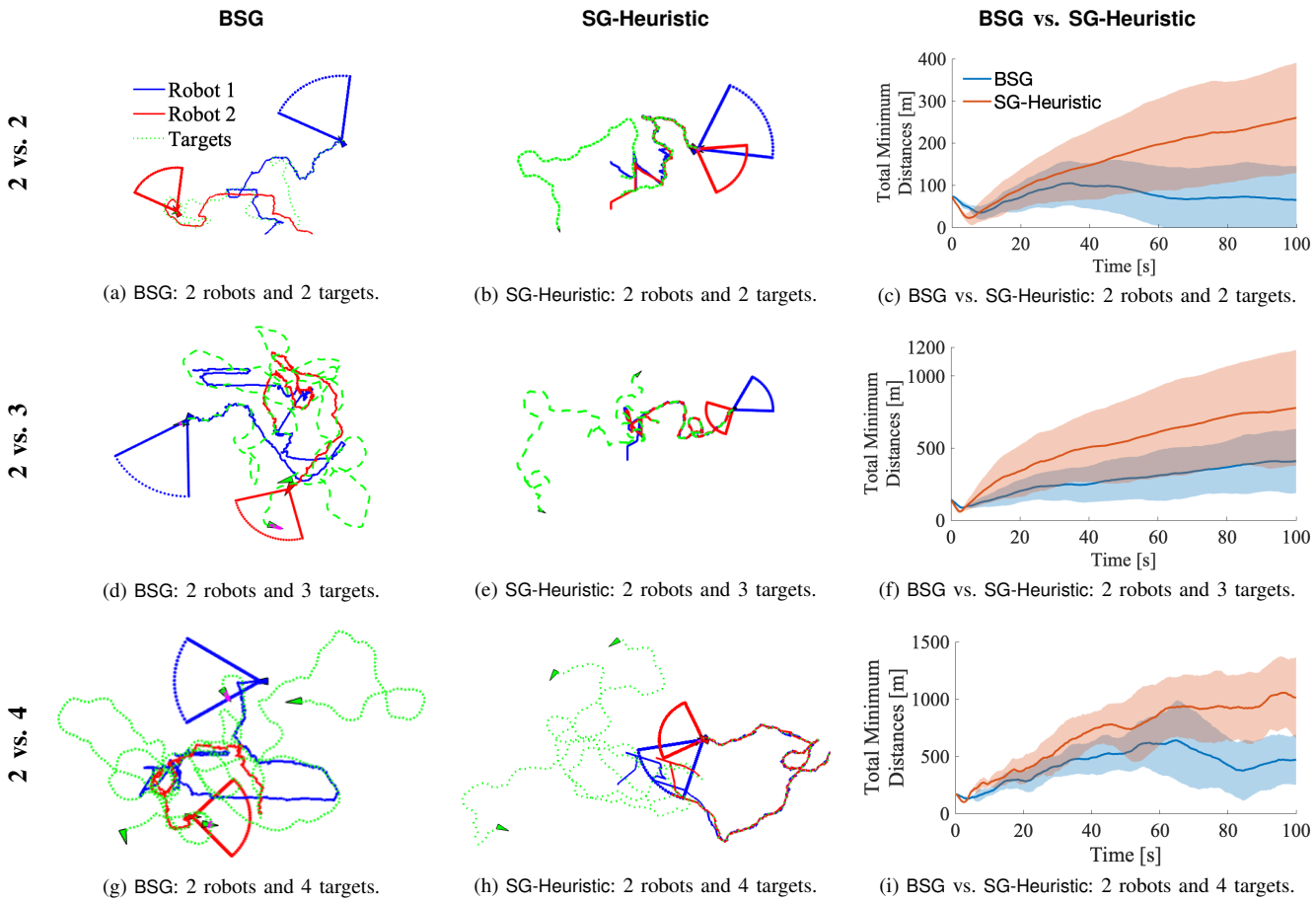
Simulation Setup. The setup is the same as in Section V-B with the exception that here the targets adapt their motion to the robots' motion: as long as all robots are more than 50m away from a target, the target performs a random walk; but if any robot is within 50m from a target, then this target increases its speed by 10m/s for 5s, pointing it to a direction that maximizes the average distance from all robots.

Results. The simulation results are shown in Fig. 5. Similarly to the non-adversarial case, (i) BSG outperforms SG-Heuristic across all scenarios (Fig. 5(c)(f)(i)), and (ii) BSG enables collaborative behaviors among the robots, where fast robots that originally track slow targets eventually switch to faster targets, and slow robots that originally track fast targets switch to slower targets (see, *e.g.*, Fig. 5(a)).

VI. Conclusion

Summary. We introduced the first algorithm for online submodular coordination in unpredictable and partially observable environments with bandit feedback. Particularly, BSG is the first polynomial time algorithm with bounded tracking regret for Problem 1, requiring only one function evaluation and $O(\log T)$ additions and multiplications per agent per time step. The tracking regret bound gracefully degrades with the environments' capacity to change, quantifying how frequently the agents should re-select actions to learn to coordinate as if they fully knew the future a priori. BSG generalizes the seminal Sequential Greedy algorithm [13] to Problem 1's bandit setting. To this end, we first provided the EXP3⋆-SIX algorithm for the problem of *tracking the best action with bandit feedback*. Then, using EXP3⋆-SIX as a subroutine, we proposed the BSG algorithm for Problem 1, leveraging submodularity, inspired by the algorithm in [20]. We validated BSG in simulated scenarios of target tracking with multiple robots, demonstrating how BSG can enable the robots to collaborate and adapt.

Limitations. BSG has the main limitations: (i) BSG is a centralized algorithm where each robot needs to know actions selected by all previous robots to make a decision (Algorithm 2); (ii) BSG requires a fine enough time discretization to achieve a near-optimal performance (Fig. 3); and (iii) BSG can have $O(T)$ tracking regret in the worst case since the environment can arbitrarily evolve such that $\Delta(T)$ is $O(T)$; this is a fundamental limit that emerges even in the single-agent case of the *tracking the best expert* problem [27, Chapter 11], due to the challenging unpredictable environment (Problem 2). To overcome this fundamental limit, we will leverage external advice about the evolution of the environment, managing the risk of erroneous advice, as discussed next.



(a) BSG: 2 robots and 2 targets. (b) SG-Heuristic: 2 robots and 2 targets. (c) BSG vs. SG-Heuristic: 2 robots and 2 targets.

(d) BSG: 2 robots and 3 targets. (e) SG-Heuristic: 2 robots and 3 targets. (f) BSG vs. SG-Heuristic: 2 robots and 3 targets.

(g) BSG: 2 robots and 4 targets. (h) SG-Heuristic: 2 robots and 4 targets. (i) BSG vs. SG-Heuristic: 2 robots and 4 targets.

Fig. 5: **Adversarial Target Tracking with Multiple Robots: 2 Robots Pursuing 2, 3, or 4 Targets.** The robots select actions either per BSG, or per the greedy heuristic SG-Heuristic, re-selecting actions with frequency 20Hz. The targets adapt their motion to the robots' motion: as long as all robots are more than 50m away from a target, the target performs a random walk; but if any robot is within 50m from a target, then this target increases its speed by 10m/s for 5s, pointing it to a direction that maximizes the average distance from all robots. (a),(d),(g): The robots use BSG against 2, 3, and 4 targets, respectively; (b),(e),(h): the robots use SG-Heuristic against 2, 3, and 4 targets, respectively. (c),(f),(i): Comparison of BSG's and SG-Heuristic's average effectiveness over 50 Monte-Carlo trials.

Future Work: Leveraging External Advice. BSG selects actions assuming that the environment may evolve arbitrarily in the future. This assumption is pessimistic when there is side information about the environment's evolution. We will extend BSG such that it can leverage side information in the form of external advice, *e.g.*,in the form of external commands originated by human operators or machine learning algorithms. We will guarantee that the algorithm is *consistent* and *robust*: (i) *consistent*: the algorithm will guarantee enhanced performance when the external advice is better than BSG in hindsight; (ii) *robust*: but when the advice is poor (worse than BSG), the algorithm will still guarantee a comparable performance to the BSG algorithm.

References

[1] M. Corah and N. Michael, "Scalable distributed planning for multi-robot, multi-target tracking," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 437–444.

[2] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Decentralized active information acquisition: Theory and application to multi-robot SLAM," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4775–4782.

[3] Z. Xu and V. Tzoumas, "Resource-aware distributed submodular maximization: A paradigm for multi-robot decision-making," in *IEEE Conference on Decision and Control (CDC)*, 2022, pp. 5959–5966.

[4] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *Jour. of Mach. Learn. Res. (JMLR)*, vol. 9, pp. 235–284, 2008.

[5] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, "Efficient informative sensing using multiple robots," *Journal of Artificial Intelligence Research (JAIR)*, vol. 34, pp. 707–755, 2009.

[6] P. Tokekar, V. Isler, and A. Franchi, "Multi-target visual tracking with aerial robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, pp. 3067–3072.

[7] B. Gharesifard and S. L. Smith, "Distributed submodular maximization with limited information," *IEEE Transactions on Control of Network Systems (TCNS)*, vol. 5, no. 4, pp. 1635–1645, 2017.

[8] D. Grimsman, M. S. Ali, J. P. Hespanha, and J. R. Marden, "The impact of information in distributed submodular maximization," *IEEE Trans. Cont. of Net. Sys. (TCNS)*, vol. 6, no. 4, pp. 1334–1343, 2018.

[9] M. Corah and N. Michael, "Distributed submodular maximization on partition matroids for planning on large sensor networks," in *IEEE Conference on Decision and Control (CDC)*, 2018, pp. 6792–6799.

[10] ——, "Distributed matroid-constrained submodular maximization for multi-robot exploration: Theory and practice," *Autonomous Robots (AURO)*, vol. 43, no. 2, pp. 485–501, 2019.

[11] B. Schlotfeldt, V. Tzoumas, and G. J. Pappas, "Resilient active information acquisition with teams of robots," *IEEE Transactions on Robotics (TRO)*, vol. 38, no. 1, pp. 244–261, 2021.

[12] U. Feige, "A threshold of $ln(n)$ for approximating set cover," *Journal of the ACM (JACM)*, vol. 45, no. 4, pp. 634–652, 1998.

[13] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey, "An analysis of approximations for maximizing submodular set functions–II," in *Polyhedral combinatorics*, 1978, pp. 73–87.

[14] A. Krause and D. Golovin, "Submodular function maximization," *Tractability: Practical Approaches to Hard Problems*, vol. 3, p. 19, 2012.

[15] J. Liu, L. Zhou, P. Tokekar, and R. K. Williams, "Distributed resilient submodular action selection in adversarial environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5832–5839, 2021.

[16] A. Robey, A. Adibi, B. Schlotfeldt, H. Hassani, and G. J. Pappas, "Optimal algorithms for submodular maximization with distributed constraints," in *Learn. for Dyn. & Cont. (L4DC)*, 2021, pp. 150–162.

[17] N. Rezazadeh and S. S. Kia, "Distributed strategy selection: A submodular set function maximization approach," *Automatica*, vol. 153, p. 111000, 2023.

[18] R. Konda, D. Grimsman, and J. R. Marden, "Execution order matters in greedy algorithms with limited information," in *American Control Conference (ACC)*, 2022, pp. 1305–1310.

[19] M. Sun, M. E. Davies, I. Proudler, and J. R. Hopgood, "A gaussian process based method for multiple model tracking," in *Sensor Signal Processing for Defence Conference (SSPD)*, 2020, pp. 1–5.

[20] Z. Xu, H. Zhou, and V. Tzoumas, "Online submodular coordination with bounded tracking regret: Theory, algorithm, and applications to multi-robot coordination," *IEEE Robo. Auto. Lett. (RAL)*, vol. 8, no. 4, pp. 2261–2268, 2023.

[21] M. Streeter and D. Golovin, "An online algorithm for maximizing submodular functions," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 21, 2008.

[22] M. Streeter, D. Golovin, and A. Krause, "Online learning of assignments," *Advances in Neu. Inform. Proc. Sys. (NeurIPS)*, vol. 22, 2009.

[23] D. Suehiro, K. Hatano, S. Kijima, E. Takimoto, and K. Nagano, "Online prediction under submodular constraints," in *International Conf. on Algorithmic Learning Theory (ALT)*, 2012, pp. 260–274.

[24] D. Golovin, A. Krause, and M. Streeter, "Online submodular maximization under a matroid constraint with application to learning assignments," *arXiv preprint:1407.1082*, 2014.

[25] L. Chen, H. Hassani, and A. Karbasi, "Online continuous submodular maximization," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2018, pp. 1896–1905.

[26] M. Zhang, L. Chen, H. Hassani, and A. Karbasi, "Online continuous submodular maximization: From full-information to bandit feedback," *Adv. in Neu. Inform. Proc. Sys. (NeurIPS)*, vol. 32, 2019.

[27] T. Lattimore and C. Szepesvári, *Bandit algorithms*. Cambridge University Press, 2020.

[28] C. Baykal, G. Rosman, S. Claici, and D. Rus, "Persistent surveillance of events with unknown, time-varying statistics," in *IEEE International Conf. on Robotics and Automation (ICRA)*, 2017, pp. 2682–2689.

[29] C. Zhang and S. C. Hoi, "Partially observable multi-sensor sequential change detection: A combinatorial multi-armed bandit approach," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 33, no. 01, 2019, pp. 5733–5740.

[30] K. M. B. Lee, F. Kong, R. Cannizzaro, J. L. Palmer, D. Johnson, C. Yoo, and R. Fitch, "An upper confidence bound for simultaneous exploration and exploitation in heterogeneous multi-robot systems," in *IEEE Inter. Conf. on Robo. and Auto. (ICRA)*, 2021, pp. 8685–8691.

[31] P. Landgren, V. Srivastava, and N. E. Leonard, "Distributed cooperative decision making in multi-agent multi-armed bandits," *Automatica*, vol. 125, p. 109445, 2021.

[32] A. Dahiya, N. Akbarzadeh, A. Mahajan, and S. L. Smith, "Scalable operator allocation for multirobot assistance: A restless bandit approach," *IEEE Transactions on Control of Network Systems (TCNS)*, vol. 9, no. 3, pp. 1397–1408, 2022.

[33] S. Wakayama and N. Ahmed, "Active inference for autonomous decision-making with contextual multi-armed bandits," *arXiv preprint:2209.09185*, 2022.

[34] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The non-stochastic multiarmed bandit problem," *SIAM Journal on Computing*, vol. 32, no. 1, pp. 48–77, 2002.

[35] T. Matsuoka, S. Ito, and N. Ohsaka, "Tracking regret bounds for online submodular optimization," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2021, pp. 3421–3429.

[36] G. Neu, "Explore no more: Improved high-probability regret bounds for non-stochastic bandits," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 28, 2015.

[37] S. Shalev-Shwartz *et al.*, "Online learning and online convex optimization," *Foundations and Trends® in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2012.

[38] A. Slivkins *et al.*, "Introduction to multi-armed bandits," *Foundations and Trends® in Machine Learning*, vol. 12, no. 1-2, pp. 1–286, 2019.

[39] L. Zhang, S. Lu, and Z.-H. Zhou, "Adaptive online learning in dynamic environments," *Adv. in Neu. Info. Proc. Sys. (NeurIPS)*, vol. 31, 2018.

[40] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. Cambridge university press, 2006.

[41] M. Herbster and M. K. Warmuth, "Tracking the best expert," *Machine learning*, vol. 32, no. 2, pp. 151–178, 1998.

[42] M. Sviridenko, J. Vondrák, and J. Ward, "Optimal approximation for submodular and supermodular optimization with bounded curvature," *Math. of Operations Research*, vol. 42, no. 4, pp. 1197–1218, 2017.

APPENDIX A
PROOF OF THEOREM 1

EXP3*-SIX's regret can be decomposed into two parts, as follows:

$$\sum_{t=1}^{T}\left(r_{a^\star,t}-r_t^\top p_t\right)=\sum_{t=1}^{T}\left(r_t^\top p_t^{(j)}-r_t^\top p_t\right)+\sum_{t=1}^{T}\left(r_{a^\star,t}-r_t^\top p_t^{(j)}\right). \tag{16}$$

It suffices to prove that eq. (16) is bounded by

$$4\sqrt{2T\Big(\bar{P}(T)|\mathcal{V}|\log\left(|\mathcal{V}|T\right)+\log\left(1+\log T\right)\Big)}+\left(\sqrt{\frac{|\mathcal{V}|T}{\bar{P}(T)\log\left(|\mathcal{V}|T\right)}}+1\right)\log\left(\frac{1}{\delta}\right),$$

where $\bar{P}(T)\triangleq P(T)+1$.

To this end, we first consider the following special cases:

- if $|\mathcal{V}|=1$, then $\sum_{t=1}^{T}\left(r_{a^\star,t}-r_t^\top p_t^{(j)}\right)=0$ and, thus, Theorem 1 holds true;
- if $T=1$ and $|\mathcal{V}|\geq 2$, then $\sum_{t=1}^{T}\left(r_{a^\star,t}-r_t^\top p_t^{(j)}\right)\leq 1$ since $r_t\in[0,1]^{|\mathcal{V}|}$. Since for $T=1$ it also is $\bar{P}(T)=1$ and, as a result, $4\sqrt{2T(\bar{P}(T)|\mathcal{V}|\log\left(|\mathcal{V}|T\right)+\log\left(1+\log T\right))}\geq 8\sqrt{\log 2}>2$, Theorem 1 again holds true;
- if $T=2$ and $|\mathcal{V}|\geq 2$, then similarly to above $\sum_{t=1}^{T}\left(r_{a^\star,t}-r_t^\top p_t^{(j)}\right)\leq 2$, $\bar{P}(T)$ is equal to either 1 or 2, and

$$4\sqrt{2T(\bar{P}(T)|\mathcal{V}|\log\left(|\mathcal{V}|T\right)+\log\left(1+\log T\right))}\geq 8\sqrt{4\log 2+\log\left(1+\log 2\right)}>2.$$

Thereby, Theorem 1 still holds true.

We now consider the last case where $T\geq 3$ and $|\mathcal{V}|\geq 2$. For this case, we start with bounding the first part of eq. (16). From [40, Theorem 2.2], we have

$$\sum_{t=1}^{T}\left(r_t^\top p_t^{(j)}-r_t^\top p_t\right)\leq 2\eta T+\frac{\log J}{\eta}=2\sqrt{2T\log J}, \tag{17}$$

where we choose $\eta=\sqrt{\log J/(2T)}$.

We next bound the second part of eq. (16). To this end, from [36, Appendix B.2], we have that

$$\begin{aligned}\sum_{t=1}^{T}\left(r_{a^\star,t}-r_t^\top p_t^{(j)}\right)\leq&\frac{\bar{P}(T)\log|\mathcal{V}|}{\eta}+\frac{1}{\eta}\log\frac{1}{\beta^{P(T)}(1-\beta)^{T-P(T)-1}}+\frac{\bar{P}(T)\log|\mathcal{V}|+P(T)\log\left(\frac{eT}{P(T)}\right)+\log\left(\frac{1}{\delta}\right)}{2\gamma}\\&+\left(\frac{\eta}{2}+\gamma\right)|\mathcal{V}|T+\left(\frac{\eta}{2}+\gamma\right)\frac{\log\left(\frac{1}{\delta}\right)}{2\gamma}\end{aligned} \tag{18}$$

holds true with probability at least $1-\delta$, $\delta\in(0,1)$. Choosing now $\beta=\frac{1}{T-1}$ and $\gamma^{(j)}=\frac{1}{2}\eta^{(j)}$, we have

$$\begin{aligned}&\sum_{t=1}^{T}\left(r_{a^\star,t}-r_t^\top p_t^{(j)}\right)\\&\leq\frac{\bar{P}(T)\log|\mathcal{V}|}{\eta}+\frac{P(T)\log T+1}{\eta}+\frac{\bar{P}(T)\log|\mathcal{V}|+P(T)\log\left(\frac{eT}{P(T)}\right)+\log\left(\frac{1}{\delta}\right)}{2\gamma}+\left(\frac{\eta}{2}+\gamma\right)|\mathcal{V}|T+\left(\frac{\eta}{2}+\gamma\right)\frac{\log\left(\frac{1}{\delta}\right)}{2\gamma} &(19)\\&=\frac{2\bar{P}(T)\log|\mathcal{V}|+P(T)\log T+1+P(T)\log\left(\frac{eT}{P(T)}\right)+\log\left(\frac{1}{\delta}\right)}{\eta}+\eta|\mathcal{V}|T+\log\left(\frac{1}{\delta}\right) &(20)\\&=\frac{2\bar{P}(T)\log|\mathcal{V}|+2P(T)\log T+1+P(T)-P(T)\log P(T)}{\eta}+\eta|\mathcal{V}|T+\left(\frac{1}{\eta}+1\right)\log\left(\frac{1}{\delta}\right) &(21)\\&\leq\frac{2\bar{P}(T)\log|\mathcal{V}|+2P(T)\log T+2\log T}{\eta}+\eta|\mathcal{V}|T+\left(\frac{1}{\eta}+1\right)\log\left(\frac{1}{\delta}\right) &(22)\\&=\frac{2\bar{P}(T)\log\left(|\mathcal{V}|T\right)}{\eta}+\eta|\mathcal{V}|T+\left(\frac{1}{\eta}+1\right)\log\left(\frac{1}{\delta}\right) &(23)\end{aligned}$$

holds with probability at least $1-\delta$, where eq. (19) holds from eq. (17) of [35, Appendix B.1], and eq. (22) holds because $1+P(T)-P(T)\log P(T) < 2\log T$ for $T \geq 3$. By the definition of $\{\eta^{(j)}\}_{j\in[J]}$, there always exists a $j \in [J]$ such that

$$\frac{\eta^{(j)}}{2} \leq \sqrt{\frac{\bar{P}(T)\log(|\mathcal{V}|T)}{|\mathcal{V}|T}} \leq \eta^{(j)}. \tag{24}$$

For this j, since $|\mathcal{V}| \geq 2$ and $T \geq 3$, we know $1 < \log(|\mathcal{V}|T)$, and, thus, with probability at least $1-\delta$,

$$\sum_{t=1}^{T}\left(r_{a^\star,t} - r_t^\top p_t^{(j)}\right) \tag{25}$$

$$\leq \frac{2\bar{P}(T)\log(|\mathcal{V}|T)}{\eta} + \eta|\mathcal{V}|T + \left(\frac{1}{\eta}+1\right)\log\left(\frac{1}{\delta}\right) \tag{26}$$

$$\leq 2\bar{P}(T)\log(|\mathcal{V}|T)\sqrt{\frac{|\mathcal{V}|T}{\bar{P}(T)\log(|\mathcal{V}|T)}} + 2\sqrt{\bar{P}(T)|\mathcal{V}|T\log(|\mathcal{V}|T)} + \left(\sqrt{\frac{|\mathcal{V}|T}{\bar{P}(T)\log(|\mathcal{V}|T)}}+1\right)\log\left(\frac{1}{\delta}\right)$$

$$\leq 4\sqrt{\bar{P}(T)|\mathcal{V}|T\log(|\mathcal{V}|T)} + \left(\sqrt{\frac{|\mathcal{V}|T}{\bar{P}(T)\log(|\mathcal{V}|T)}}+1\right)\log\left(\frac{1}{\delta}\right). \tag{27}$$

Hence, eq. (27) holds true for $T \geq 3$ and $|\mathcal{V}| \geq 2$.

In all, combining eqs. (17) and (27), the following steps hold true for $T \geq 3$ and $|\mathcal{V}| \geq 2$ with probability at least $1-\delta$:

$$\sum_{t=1}^{T}\left(r_{a^\star,t} - r_t^\top p_t\right)$$

$$= \sum_{t=1}^{T}\left(r_{a^\star,t} - r_t^\top p_t^{(j)}\right) + \sum_{t=1}^{T}\left(r_t^\top p_t^{(j)} - r_t^\top p_t\right)$$

$$\leq 4\sqrt{\bar{P}(T)|\mathcal{V}|T\log(|\mathcal{V}|T)} + 2\sqrt{2T\log J} + \left(\sqrt{\frac{|\mathcal{V}|T}{\bar{P}(T)\log(|\mathcal{V}|T)}}+1\right)\log\left(\frac{1}{\delta}\right) \tag{28}$$

$$\leq 4\sqrt{\bar{P}(T)|\mathcal{V}|T\log(|\mathcal{V}|T)} + 4\sqrt{T\log(1+\log T)} + \left(\sqrt{\frac{|\mathcal{V}|T}{\bar{P}(T)\log(|\mathcal{V}|T)}}+1\right)\log\left(\frac{1}{\delta}\right) \tag{29}$$

$$\leq 4\sqrt{2T\Big(\bar{P}(T)|\mathcal{V}|\log(|\mathcal{V}|T) + \log(1+\log T)\Big)} + \left(\sqrt{\frac{|\mathcal{V}|T}{\bar{P}(T)\log(|\mathcal{V}|T)}}+1\right)\log\left(\frac{1}{\delta}\right), \tag{30}$$

where eq. (28) holds because $\log J \leq 2\log(1+\log T)$ for $T \geq 3$. □

Appendix B
Proof of Theorem 2

We denote by $\mathcal{A}^{\mathsf{OPT}}_{i-1,t}$ the optimal solution set for the first $i-1$ agents at time step t. Then, we have:

$$\sum_{t=1}^{T} f_t(\mathcal{A}_t^{\mathsf{OPT}})$$

$$\leq \sum_{t=1}^{T} f_t(\mathcal{A}_t^{\mathsf{OPT}} \cup \mathcal{A}_t^{\mathsf{BSG}}) \tag{31}$$

$$= \sum_{t=1}^{T} f_t(\mathcal{A}_t^{\mathsf{BSG}}) + \sum_{t=1}^{T}\sum_{i\in\mathcal{N}} f_t(a_{i,t}^{\mathsf{OPT}} \mid \mathcal{A}_{i-1,t}^{\mathsf{OPT}} \cup \mathcal{A}_t^{\mathsf{BSG}}) \tag{32}$$

$$\leq \sum_{t=1}^{T} f_t(\mathcal{A}_t^{\mathsf{BSG}}) + \sum_{t=1}^{T}\sum_{i\in\mathcal{N}} f_t(a_{i,t}^{\mathsf{OPT}} \mid \mathcal{A}_{i-1,t}^{\mathsf{BSG}}) \tag{33}$$

$$= 2\sum_{t=1}^{T} f_t(\mathcal{A}_t^{\mathsf{BSG}}) + \sum_{t=1}^{T}\sum_{i\in\mathcal{N}} f_t(a_{i,t}^{\mathsf{OPT}} \mid \mathcal{A}_{i-1,t}^{\mathsf{BSG}}) - f_t(a_{i,t}^{\mathsf{BSG}} \mid \mathcal{A}_{i-1,t}^{\mathsf{BSG}}) \tag{34}$$

$$= 2\sum_{t=1}^{T} f_t(\mathcal{A}_t^{\mathsf{BSG}}) + \sum_{t=1}^{T}\sum_{i\in\mathcal{N}} r_{a^{\mathsf{OPT}},t}^{(i)} - r_{a^{\mathsf{BSG}},t}^{(i)}, \tag{35}$$

where eq. (31) holds from the monotonicity of f_t; eqs. (32) and (34) are proved by telescoping the sums; eq. (33) holds from the submodularity of f_t; and eq. (35) holds from the definition of $r_{a^{\mathsf{BSG}},t}^{(i)}$ (Algorithm 2's line 14). Now:

$$\mathbb{E}\Big[\mathsf{Tracking\text{-}Regret}_T^{(1/2)}(\mathcal{A}^{\mathsf{BSG}})\Big]$$

$$= \mathbb{E}\Big[\frac{1}{2}\sum_{t=1}^{T} f_t(\mathcal{A}_t^{\mathsf{OPT}}) - \sum_{t=1}^{T} f_t(\mathcal{A}_t^{\mathsf{BSG}})\Big] \tag{36}$$

$$\leq \frac{1}{2}\sum_{t=1}^{T}\sum_{i\in\mathcal{N}} \mathbb{E}\left(r_{a^{\mathsf{OPT}},t}^{(i)} - r_{a^{\mathsf{BSG}},t}^{(i)}\right) \tag{37}$$

$$= \frac{1}{2}\sum_{t=1}^{T}\sum_{i\in\mathcal{N}} \left(r_{a^{\mathsf{OPT}},t}^{(i)} - r_t^{(i)\top} p_t^{(i)}\right) \tag{38}$$

$$\leq \frac{1}{2}\left[4\sum_{i\in\mathcal{N}}\sqrt{2T\Big(\bar{\Delta}_i(T)|\mathcal{V}_i|\log(|\mathcal{V}_i|T) + \log(1+\log T)\Big)} + \sum_{i\in\mathcal{N}}\left(\sqrt{\frac{|\mathcal{V}_i|T}{\bar{\Delta}_i(T)\log(|\mathcal{V}_i|T)}} + 1\right)\log\left(\frac{1}{\delta}\right)\right] \tag{39}$$

$$\leq 2\sqrt{2|\mathcal{N}|T\sum_{i\in\mathcal{N}}\Big(\bar{\Delta}_i(T)|\mathcal{V}_i|\log(|\mathcal{V}_i|T) + \log(1+\log T)\Big)} + \frac{1}{2}\left(\sqrt{|\mathcal{N}|T\sum_{i\in\mathcal{N}}\frac{|\mathcal{V}_i|}{\bar{\Delta}_i(T)\log(|\mathcal{V}_i|T)}} + |\mathcal{N}|\right)\log\left(\frac{1}{\delta}\right) \tag{40}$$

$$\leq 2\sqrt{2|\mathcal{N}|T\Big((\Delta(T)+|\mathcal{N}|)|\bar{\mathcal{V}}|\log(|\bar{\mathcal{V}}|T) + |\mathcal{N}|\log(1+\log T)\Big)} + \frac{1}{2}\left(\sqrt{|\mathcal{N}|T\sum_{i\in\mathcal{N}}\frac{|\mathcal{V}_i|}{\bar{\Delta}_i(T)\log(|\mathcal{V}_i|T)}} + |\mathcal{N}|\right)\log\left(\frac{1}{\delta}\right), \tag{41}$$

with probability at least $1-\delta$, where eq. (36) holds from eq. (9); eq. (37) holds from eq. (35); eq. (38) holds from the internal randomness of $\mathsf{EXP3}^\star\text{-}\mathsf{SIX}|_i$; eq. (39) holds from [35, Corollary 1]; eq. (40) holds from the Cauchy–Schwartz inequality; and eq. (41) holds true since $\sum_{i\in\mathcal{N}}\Delta_i(T) = \Delta(T)$. □

Appendix C
Proof of Monotonicity and Submodularity of Function (14)

Because the addition of multiple non-decreasing submodular functions results to non-decreasing submodular functions, it suffices to prove that the function $f(\mathcal{S}) = -1/\left(\sum_{s\in\mathcal{S}} 1/s\right)$, $\mathcal{S}\in 2^{\mathbb{R}_+}$ is non-decreasing and submodular, where $f(\emptyset) = -\infty$. We start by proving f's monotonicity: consider $\mathcal{A}\subseteq\mathcal{B}\in 2^{\mathbb{R}_+}$, then we have $-\sum_{a\in\mathcal{A}} 1/a \geq -\sum_{b\in\mathcal{B}} 1/b$, and thus $f(\mathcal{A}) \leq f(\mathcal{B})$. We now prove f's submodularity: consider finite and disjoint $\mathcal{B}_1\in 2^{\mathbb{R}_+}$ and $\mathcal{B}_2\in 2^{\mathbb{R}_+}$, and an arbitrary non-zero real number s. Set $B_1 \triangleq \sum_{b_1\in\mathcal{B}_1} 1/b_1$ and $B_2 \triangleq \sum_{b_2\in\mathcal{B}_2} 1/b_2$; then,

$$\frac{1}{\mathcal{B}_1+\mathcal{B}_2} - \frac{1}{\mathcal{B}_1+\mathcal{B}_2+1/s} \leq \frac{1}{\mathcal{B}_1} - \frac{1}{\mathcal{B}_1+1/s},$$

where the equality is taken when $\mathcal{B}_2 = \emptyset$. Therefore, $f(\mathcal{B}_1\cup\mathcal{B}_2\cup\{s\}) - f(\mathcal{B}_1\cup\mathcal{B}_2) \leq f(\mathcal{B}_1\cup\{s\}) - f(\mathcal{B}_1)$, which proves f's submodularity. □