Online Coalitional Skill Formation

Saar Cohen
Department of Computer Science
Bar Ilan University, Israel
saar30@gmail.com

Noa Agmon
Department of Computer Science
Bar Ilan University, Israel
agmon@cs.biu.ac.il

ABSTRACT

Efficiently allocating heterogeneous tasks to agents that arrive *dynamically* and have *diverse* skills is a central problem in multi-agent systems called *online task allocation*. In many cases, a single agent does not meet the skill levels required by a particular task, which incentivizes the agents to form coalitions for handling it. In this paper, we propose a *new* framework, termed as *online coalitional skill formation* (**OCSF**), for handling online task allocation via coalition formation, where tasks require different skills for being successfully fulfilled, and each agent has different levels at mastering each skill. The goal of the organizer is therefore to assign agents that arrive online to a coalition responsible for performing some task, so as to optimally approach the desired skill levels of all tasks. Focusing on the case in which the set of possible mastering levels for each skill is *discrete*, we suggest different assignment algorithms based on the knowledge the organizer has on the arriving agents. When agents arrive i.i.d. according to some unknown distribution, we propose a *greedy* and adaptive scheme that assigns an agent to a task, proving a tight bound on the system's performance. If the distribution is *known*, we devise a novel correlation to Constrained Markov Decision Processes whose goal is maximizing the rate at which agents are assigned to each task while respecting their requirements. We then construct a *non-adaptive* approach that terminates when all the tasks' requirements are met. Finally, if the distribution is *unknown*, we provide two algorithms that *learn it online*. We have fully implemented the algorithms, showing that in many cases a higher diversity in skills may yield poor assignments.

KEYWORDS

Coalition Formation; Task Allocation; Online Algorithms; Constrained Markov Decision Processes; Reinforcement Learning

ACM Reference Format:
Saar Cohen and Noa Agmon. 2023. Online Coalitional Skill Formation. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 10 pages.

1 INTRODUCTION

In various multi-agent applications such as crowdsourcing [29, 46] and rescue operations [54], agents with diverse skills emerge *dynamically* and are then assigned to tasks with heterogeneous requirements. Such scenarios can be framed as a special case of the notorious multi-agent task allocation problem [43] that attracts extensive attention–*online task allocation* [18, 48]. For instance, crowdsourcing markets allow task requesters to inexpensively access a large manpower of workers with *multiple* diverse skills that sequentially arrive one by one [28]. The workers then solve problems like participatory sensing [56] and human computation [27].

The most desirable goal of online task allocation is to *assign the most suitable agents to tasks* [41, 58]. Namely, an agent with the skill level required by a certain task or at least a sufficient experience should be ideally assigned to that task. It is often the case that a single agent does not have the skill level needed to achieve a particular task, thus it is necessary for the agents to form *real-time* coalitions for completing certain tasks. In coalitions, agents can collaboratively complete tasks more efficiently or accurately by cooperating to meet those requirements [52]. However, adequately *modeling multiple skills* of agents, their correlations, the uncertainty about their conditional dependencies and their contributions to the formed coalitions is difficult, since different agents usually possess different skills and diverse degree of proficiency in the same skill [38]. Further, improperly measuring the coalitions' suitability (i.e., meeting the tasks' requirements as much as possible) may hinder the quality of a task's execution [29].

Against this background, in this paper we develop a novel framework termed as *online coalitional skill formation* (**OCSF**), for handling online task allocation from a standpoint of coalition formation. In our formalization, there is a set of m *skills* and each agent has a *skill vector* that expresses her level at mastering each skill. Additionally, an *organizer* has a fixed set of k *tasks*, each with certain requirements reflecting the desired skill levels essential to complete the task, and the number of agents assigned to each task is limited by some *budget*. Agents arrive online, and must *immediately* and *irrevocably* be assigned to a coalition attending a task upon arrival, if at all. We propose a *new* skill model for online task allocation, where the set of possible mastering levels for each skill is *discrete*, and a coalition is evaluated by the extent each skill level is *covered* by the coalition. We discuss the efficiencies in considering discrete skill domains instead of *continuous* (Section 7), where the latter can be transferred into the former by standard discretization tools.

Accordingly, we suggest different assignment algorithms based on the knowledge the organizer has on the arriving agents. As a first step, we propose a *greedy* algorithm that assigns an agent to a task as long as the task's requirements and budget are not violated, and regardless of the (known or unknown) agent distribution. However, we show that due to its *adaptivity* and the need to simultaneously consider multiple skills and tasks, the expected number of agent screened may be arbitrarily large. Hence, we show that the constraints incurred by the tasks' requirements allows us to formulate the system as constrained MDPs (CMDPs) [3, 20]. When the agents' distribution is *known*, we prove that our goal is maximizing the rate at which agents are assigned to each task while respecting their requirements. Based on the CMDP's optimal and stationary

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

policy, we devise an algorithm that assigns agents to tasks until their budgets are reached. Finally, if the distribution is *unknown*, we provide two algorithms that *learn it online*. We empirically evaluate our algorithms on a synthetic dataset, showing that surprisingly, in many cases that heterogeneity in the agents' skills is a *vulnerability* which yields poor assignments to tasks.

2 RELATED WORK

The online task allocation problem arises in several multi-agent problems that have been studied in the literature (e.g., multi-robot systems [23], crowdsourcing [24, 31]). Several common arrival assumptions on the online agents exist (e.g., adversarial order [4], random arrival order [47, 55]), yet we consider the following two models: agents arrive i.i.d. according to either a *known* [2, 18, 45] or *unknown* distribution [17]. For motivating their consideration, when the distribution is *known*, competitive ratios for certain problems can be improved such as Submodular Welfare Maximization (SWM) [2] and stochastic matching [37]. Additionally, when it is *unknown*, there is a $(1 - 1/e)$-competitive algorithm for SWM [16]. By making the simplifying assumption that the set of agents is known upfront, our problem becomes the well-studied *offline* task allocation problem, which can be solved either optimally via the Hungarian algorithm in a centralized manner [33], or *almost* optimally using a distributed auction-based scheme [9, 36]. In the *online* regime, even the simplest version is NP-hard [23]. Further, greedy algorithms used in certain multi-robot systems have a competitive ratio of 1/3 [22, 40], but they exponentially depend on the number of agents. In contrast, our greedy algorithm only depends on the maximum mastering level among all skills and each budget.

For allocating diverse tasks to the most suitable agents, one needs to reason about the tasks' requirements and the agents' skills. Though *hierarchical* skill models [38, 49] attempt to reflect correlations among multiple skills opposed to others [7, 12, 21], some cannot quantitatively capture multiple skills. For example, Mavridis et al. [38] can only cope with tasks related to a *single* skill and cannot reflect proficiency degree. Despite that previous studies strive to overcome these issues [49], they fail to quantify the *uncertainty* about the conditional dependencies. For instance, a person's ability of *speaking* English increases the possibility that he can also *read* English, but he is not certainly capable of doing so. Moreover, those models assume a *finite* set of agents, which is unrealistic in numerous applications where the number of agents is unbounded and unknown upfront. Such assumption becomes even more problematic in *offline* task allocation [9] as the set of agents is unknown beforehand and may demand contacting a large population. Our proposed model addresses the above challenges, whereas incorporating a Bayesian network whose nodes represent skills for generating the agents' distribution produces probabilistic relations.

Our work is closely related to online learning approaches, which have also been adapted in crowdsourcing [25, 29, 50]. [29] analyze a scenario similar to ours and propose an online primal-dual scheme that is competitive with respect to the offline optimal algorithm in certain settings. In contrast, we capture multiple skills, allowing us to address a wider heterogeneity in both tasks and agents. Different from our work, [25, 50] consider online arrivals of *tasks* instead of *agents*. Further, [25] neglect the agents' skills. Thus, due to the stochastic nature of the arrivals, contextual multi-armed bandits (CMABs) [35] are often ysed as a formulation for capturing the agents' skills as their contexts [26, 39]. Unlike CMABs, we must deal with constraints imposed by both the budgets and the tasks' requirements, whereas traditional CMABs focus on different sorts of constraints (e.g., knapsack constraints [1]). As such, opposed to prior research, we propose to leverage a novel correlation to constrained MDPs (CMDPs) [3]. Specifically, if the distribution is *known*, then the problem boils down to solving only a linear program [3, 57]. Otherwise, following [20], we devise model-based algorithms that can learn the distribution from collected samples.

3 ONLINE COALITIONAL SKILL FORMATION

In this section, we introduce our *online coalitional skill formation* (**OCSF**) framework, for assigning agents to coalitions that attend given tasks, where the agents arrive *online*. Since the set of agents may not be known beforehand, we hereafter assume that the number of agents is ***infinite***. The agents' goal is achieving k tasks $\Gamma = \{\tau_1, \ldots, \tau_k\}$. We refer to m skills within our model, where each task requires the agents to have certain levels at mastering each skill. For instance, in the cooperative object transport problem in multi-robot systems [32, 51], robots cooperatively transport objects from a starting position to a final destination via exerting pushing forces. A task is then described by a physically grounded object, whose weight determines the required *maximal weight* a robot can push and its location yields the *engine power* needed by the robot for traveling to the object and transporting it.

Formally, let $\mathcal{S} = \times_{i=1}^{m} \mathcal{S}_i$ denote the product space of m skill domains. Then for each agent ℓ, $s_\ell^i \in \mathcal{S}_i$ constitutes agent ℓ's level at mastering skill i, and her mastering level at each regarded skill is encompassed by a *skill vector* $\mathbf{s}_\ell \in \mathcal{S}$. We restrict our analysis to OCSF with ***finite*** possible mastering levels for each skill, which model practical applications where there is only a limited access to resources. For instance, in our cooperative object transport example, one may consider robots having certain predefined maximal weight. Concerning the *continuous* skill domains discussed in Section 7, standard discretization tools can be also utilized for transferring them into *finite* ones. For example, instead of allowing for all maximal weights in the interval $[1, 10]$, we may solely consider integer weights within it. This is reflected by $\mathcal{S}_i$ being *finite*, i.e., $\mathcal{S}_i = \{1, \ldots, \alpha_i\}$ for some integer $\alpha_i \in \mathbb{N}$. For brevity, we hereafter denote $[f] := \{1, \ldots, f\}$ for an integer $f > 0$.

The skill levels required for achieving a certain task cannot necessarily be possessed by one agent individually (e.g., pushing an object on its own). Thus, the agents are motivated to form *coalitions* for attending a task, aggregating their skill vectors. Evaluating coalitions directly via agents' skill vectors when considering finite number of mastering levels becomes quite problematic, as their sum might yield mastering levels *outside* the skill domain. Let $\mathcal{G} = \{g^{\tau_q}\}_{\tau_q \in \Gamma}$ be the *goals*, describing the desired skill levels for the tasks. Each task τ_q's individual goal, g^{τ_q}, consists of $g_j^{i,\tau}$ ($j \in [\alpha_i]$) indicating the *fraction* of agents with a level of j at mastering skill i that is required for establishing τ_q. Hence, g^{τ_q} satisfies $g^{i,\tau_q} \in [0,1]^{\alpha_i}$ with $\Sigma_{j=1}^{\alpha_i} g_j^{i,\tau_q} = 1$. For a coalition C, $\mathbf{x}(C)$ denotes C*'s skill coverage* that depicts to what extent each skill level is *covered* by the

coalition, i.e., $\mathbf{x}(C) \in \prod_{i\in[m]}[0,1]^{\alpha_i}$ with $\mathbf{x}^i_j(C) = \frac{|\{\ell\in C: s^i_\ell = j\}|}{|C|}$ being the fraction of agents with a level of j at mastering skill i.

Thereby, the goal of a coalition assigned to a task τ_q is to (jointly) reach a certain level of skills whose distance from the task's goal is as close as possible so as to attain τ_q. For any task τ_q, the distance $d(\mathbf{x}(C), g^{\tau_q})$ is measured in the L^∞-norm, where $d(\mathbf{x}(C), g^{\tau_q}) = \|\mathbf{x}(C) - g^{\tau_q}\|_\infty = \max_{i\in[m], j\in[\alpha_i]} |\mathbf{x}^i_j(C) - g^{i,\tau_q}_j|$. We also denote *the value of a coalition* C as $d(\mathbf{x}(C), \mathcal{G}) = \min_{g\in\mathcal{G}} d(\mathbf{x}(C), g)$, i.e., the distance of $\mathbf{s}(C)$ from $\mathcal{G}$. We measure the distance under the L^∞-norm rather than the L^1- and L^2-norms since the L^∞-norm yields coalition structures that approach the skill levels required by each task *closer* than the other two norms. This property of the L^∞-norm can be attributed to the inequality $\|x\|_\infty \le \|x\|_2 \le \|x\|_1 \le m\|x\|_\infty$ for any $x \in \mathbb{R}^m$, as it will be later confirmed theoretically. Intuitively, our setup induces a *(soft) proportional fairness* constraint [8], i.e., the number of agents having a level of j at mastering skill i should be *fairly* represented in the coalition C^q assigned to task τ_q such that their *proportion* in C^q is as close as possible to g^{i,τ_q}_j. Particularly, coalitions should be neither over- nor under-qualified.

In our *online* model, as the number of agents is unknown and thus modeled as infinite, the time horizon is also ***infinite***. Further, we consider two models for agents' appearance: they arrive i.i.d. according to either a *known* [2] or *unknown* distribution [17]. At each time $t \ge 1$, a central authority (hereafter, the *organizer*) observes a *single* agent that is drawn i.i.d. from a stationary distribution $\mathcal{P}$ over $\mathcal{S}$, i.e., $\mathbf{s}_t \sim \mathcal{P}$. The organizer must *immediately* and *irrevocably* decide to which task τ_q the agent should be assigned, if at all.

The organizer's goal: Selecting a *coalition structure* $\mathcal{CS} = (C^q)_{q=1}^k$ where C^q will be assigned to perform task τ_q, s.t. $C^q \cap C^{q'} = \emptyset$ for all $q \ne q'$ and C^q's skill vector is as close as possible to the goals $\mathcal{G}$ with $q \in \arg\min_{q'} d(\mathbf{s}(C^q), g^{\tau_{q'}})$ being satisfied. Additionally, two properties shall be satisfied due to the infinite horizon: (1) $|C^q| \le B^q$ for a budget $B^q > 0$ that limits the number of agents assigned to each task; and (2) the total number of agents that have arrived until the organizer stops should be minimized. Put differently, let $C^q_t = \{\mathbf{s}_{t'} : t' \le t \wedge a_{t'} = q\}$ be the set of all agents assigned to task τ_q until time $t \ge 0$. Since a single agent arrives at each time t, the first time when each task τ_k is assigned with B^q agents equals to number of agents that have arrived until the organizer stops. It is thus denoted by $\mathcal{T}$. Hence, the organizer follows a possibly randomized algorithm $\mathcal{A}$ for selecting a coalition structure as above while minimizing the *sample complexity* $\mathbb{E}^{\mathcal{A}}_{\mathcal{P}}[\mathcal{T}]$.

3.1 A Simple and Adaptive Greedy Algorithm

In this section, we consider Greedy (Algorithm 1) that outputs a coalition structure that satisfies a variant of *flexible proportionality* known as *upper quota* [13, 34]. If $\alpha_i \ge 2$, we denote $\hat{\alpha}_i = \frac{\epsilon B_q}{\alpha_i - 1}$ for some tolerance $\epsilon > 0$; otherwise, $\hat{\alpha}_i = 0$. Due to the budget constraint on each coalition, once an agent with a skill vector $\mathbf{s}_t$ arrives, Greedy adds her to a coalition C^q_{t-1} assigned to task τ_q if and only if the number of agents $\ell \in C^q_{t-1} \cup \{\mathbf{s}_t\}$ with $s^i_\ell = j$ is at most the upper quota $\lceil g^{i,\tau_q}_j B_q\rceil + \hat{\alpha}_i\ \forall i, j$. Otherwise, the organizer will not assign $\mathbf{s}_t$ to any task. Given the output $\mathcal{CS} = (C^q)_{q=1}^k$ of Greedy, we upper bound the distance $d(\mathbf{x}(C^q), g^{\tau_q})\ \forall q$:

Algorithm 1 Greedy

Input: Budgets $B^q > 0$, Goals g^{τ_q}, Tolerance $\epsilon > 0$

$C^q_0 \leftarrow \emptyset\ \forall q$; $t \leftarrow 0$; If $\alpha_i \ge 2$, let $\hat{\alpha}_i = \frac{\epsilon B_q}{\alpha_i - 1}$; otherwise, $\hat{\alpha}_i = 0$.
while $\exists q \in [k]$ s.t. $|C^q_t| < B_q$ **do**
  Set $n^i_j(C^q_t) \leftarrow |\{\ell \in C^q_t : s^i_\ell = j\}|\ \forall i, j, q$
  Set $t \leftarrow t+1$, $dist \leftarrow 0$, and observe $\mathbf{s}_t \sim \mathcal{P}$
  for each coalition C^q_{t-1} s.t. $|C^q_{t-1}| < B_q$ **do**
    if $n^i_j(C^q_{t-1}) + \mathbb{1}_{s^i_t = j} \le \lceil g^{i,\tau_q}_j B_q\rceil + \hat{\alpha}_i\ \forall i, j$ **then**
      Assign $\mathbf{s}_t$ to τ_q (i.e., $C^q_t \leftarrow C^q_{t-1} \cup \{\mathbf{s}_t\}$) and **BREAK**.
return The coalition structure $\mathcal{CS} = (C^q)_{q=1}^k$

Theorem 3.1. *The L^∞-, L^1- and L^2-norms incurred by Greedy are upper bounded by (resp.):* $\|\mathbf{x}(C^q_T) - g^{\tau_q}\|_\infty \le \frac{\max_{i\in[m]}\alpha_i - 1}{B_q} + \epsilon$, $\|\mathbf{x}(C^q_T) - g^{\tau_q}\|_1 \le \max_{i\in[m]}\alpha_i(\frac{\alpha_i - 1}{B_q} + \epsilon)$ *and* $\|\mathbf{x}(C^q_T) - g^{\tau_q}\|_2 \le (\sum_{i\in[m]}\alpha_i(\frac{\alpha_i-1}{B_q} + \epsilon)^2)^{\frac{1}{2}}$ (See Appendix A for a detailed proof [14]).

Theorem 3.1 indicates that the L^∞-norm is ***harsher*** than the other two since Greedy ensures a smaller upper bound with respect to the L^∞-norm. Thus, the L^∞-norm is a suitable candidate for measuring whether a coalition meets a certain skill level for attaining its assigned task. Further, the upper bounds in Theorem 3.1 for the L^∞- and L^1-norms depend on the maximum mastering level α_i across all skills i divided by the budget B_q. Hence, allowing for *higher* skill levels will negatively affect both norms.

Unfortunately, the expected number of agents contacted by Greedy (i.e., the *sample complexity* $\mathbb{E}^{\text{Greedy}}_{\mathcal{P}}[\mathcal{T}]$) may be arbitrarily large. In Appendix A.2 [14], we consider an instance of our cooperative object transport example that requires at least 1000 agents (on average) to arrive until Greedy terminates. Despite Greedy's simplicity, we infer that a naive approach to our problem may fail. Specifically, the challenge stems from the requirement to simultaneously consider multiple skills and tasks. This difficulty is entangled with Greedy's *adaptivity*, according to which assignments to tasks are made based on the current agent and the already assigned agents. As a first step towards mitigating these drawbacks, in Section 4 we initially refer to a *non-adaptive* approach for the case where the agents' distribution is *known* based on constrained MDPs (CMDPs), which we then adjust to the case where $\mathcal{P}$ is *unknown* in Section 5.

4 THE AGENTS' DISTRIBUTION IS *KNOWN*

In this section, we assume that the distribution $\mathcal{P}$ is *known*. In this scenario, we formalize our online problem as a contextual multi-armed bandit (CMAB) under the constrained MDP (CMDP) framework (Subsection 4.1). This is made possible by considering the case where each budget B_q is unbounded (i.e., $B_q \to \infty$ for each task τ_q). We show that our goal is then maximizing the rate at which agents are assigned to each task, under the constraints imposed by the goals $\mathcal{G}$. Applying the approach to our own context, we present an algorithm that stops when B_q agents have been assigned to each task τ_q (Subsection 4.2). We prove that the skill levels of the resulting coalition structure get closer (with high probability) to those required by each task as the budget *increases*.

Remark 1. *When there is a* single *task (i.e., $k = 1$), our model degenerates to the* online diverse committee selection problem *introduced by Do et al. [19]. Hence, our model also generalizes [19] to selecting* multiple *diverse committees, which thus involves considering* multiple *committees simultaneously. As such, next we design novel reward and cost constraints under our CMDP that are more suitable to our context and differ from those in [19].*

4.1 A Constrained MDP Model for OCSF

Our problem can be viewed as a contextual bandit with a stochastic context $\mathbf{s}_t \sim \mathcal{P}$ at time t and a set $\mathcal{U} = [k] \cup \{0\}$ of $k+1$ actions. An appropriate formulation of the constraints incurred by the system's goals $\mathcal{G}$ is constrained MDPs (CMDPs) [3, 20]. Formally, let $M = (\mathcal{S}, \mathcal{U}, P, r, \eta)$ be a CMDP, where the set of states is the space $\mathcal{S}$ of all m-dimensional skill vectors and the set of actions is $\mathcal{U} = [k] \cup \{0\}$. That is, if the organizer assigns the agent arriving at time t to task τ_q ($q \in [k]$), his action is denoted by $a_t = q$; otherwise, $a_t = 0$. We also consider the deterministic reward $r(\mathbf{s}, a) = \mathbb{1}_{\{a \in [k]\}}$. The transition function $P : \mathcal{S} \times \mathcal{U} \times \mathcal{S} \to [0, 1]$ determines the probability that a skill vector $\mathbf{s}'$ arrives given that the previous one was $\mathbf{s}$ and the organizer took action a. We define P as $P(\mathbf{s}'|a, \mathbf{s}) = \mathcal{P}(\mathbf{s}')$ since agents are drawn i.i.d regardless of the previous agents and actions. As customary under CMDPs, we capture the constraints posed by the goals $\mathcal{G}$ via cost constraints which restrict the set of plausible policies. For every i, j, q, let $r_j^{i,q}(\mathbf{s}, a) = \mathbb{1}_{\{\mathbf{s}^i = j, a = q\}}$, $r^q(\mathbf{s}, a) = \mathbb{1}_{\{a=q\}}$ and define a *cost constraint* by $\eta_j^{i,q} = r_j^{i,q} - g_j^{i,q} r^q$. Similarly, let $M^q = (\mathcal{S}, \mathcal{U}^q, \mathcal{P}, r^q, \eta)$ be the CMDP associated with task τ_q, where the set of actions is $\mathcal{U}^q = \{0, q\}$. Henceforth, we make the following assumption:

Assumption 1. *$\mathcal{P}(\mathbf{s}) > 0$ for all $\mathbf{s} \in \mathcal{S}$ and the MDP is ergodic.*

The organizer aims at deriving a policy $\pi : \mathcal{U} \times \mathcal{S} \to [0, 1]$ with $\pi(a|\mathbf{s})$ specifying the probability of assigning $\mathbf{s}$ to task τ_a. π's performance is measured by its ***long-term average reward*** (*gain*) $\mathcal{J}^{\mathcal{P},\pi}(\mathbf{s}) = \lim_{T\to\infty} \frac{1}{T}\mathbb{E}_\pi[\Sigma_{t=1}^T r(\mathbf{s}_t, a_t)|\mathbf{s}_0 = \mathbf{s}]$, where $\mathbb{E}_\pi[\cdot|\mathbf{s}_0 = \mathbf{s}]$ denotes that the expectation is over $a_t \sim \pi(\cdot|\mathbf{s}_t)$, $s_{t+1} \sim \mathcal{P}(\cdot)$. When the context is clear, we simply write $\mathcal{J}^\pi \triangleq \mathcal{J}^{\mathcal{P},\pi}$. Due to the ergodicity of the MDP (Assumption 1), $\mathcal{J}^{\mathcal{P},\pi}(\mathbf{s})$ is independent of the starting state, i.e., $\mathcal{J}^\pi(\mathbf{s}) \equiv \mathcal{J}^\pi \ \forall \mathbf{s} \in \mathcal{S}$. Equivalently, let $\mathcal{J}_j^{i,q}(\pi) = \lim_{T\to\infty} \frac{1}{T}\mathbb{E}_\pi[\Sigma_{t=1}^T \eta_j^{i,q}(\mathbf{s}_t, a_t)]$ and $\mathcal{J}_q^\pi = \lim_{T\to\infty} \frac{1}{T}\mathbb{E}_\pi[\Sigma_{t=1}^T r^q(\mathbf{s}_t, a_t)]$ be the (constant) long-term average *cost* and *reward of task* τ_q (resp.). For motivating the choice of the long-term average reward as our performance measure, note that the number of agents assigned to task τ_q at time T satisfies $|C_T^q| = \Sigma_{t=1}^T r^q(\mathbf{s}_t, a_t)$. Thus, $\mathcal{J}_q^\pi = \lim_{T\to\infty} \mathbb{E}_\pi[|C_T^q|]/T$, which is the expected assignment rate of agents to task τ_q as $T \to \infty$.

The goal under the CMDP M and each CMDP M^q corresponding to task τ_q is finding policies $\pi_\star$ and $\pi_\star^q : \mathcal{U}^q \times \mathcal{S} \to [0, 1]$ which are the solutions of the optimization problems (resp.):

$$\max_\pi \{\mathcal{J}^\pi | \mathcal{J}_j^{i,q}(\pi) = 0 \forall i, j, q\}; \max_{\pi^q} \{\mathcal{J}_q^{\pi^q} | \mathcal{J}_j^{i,q}(\pi^q) = 0 \forall i, j\} \quad (1)$$

where $\pi^q : \mathcal{U}^q \times \mathcal{S} \to [0, 1]$ is a policy under the CMDP M^q. By Altman [3], the optimal policies $\pi_\star$ and $\pi_\star^q$ are *stationary*. Following standard results pertaining CMDPs, Appendix B.1 [14] depicts that

Algorithm 2 OCSF-CMDP

Input: The stationary optimal policies $\pi_\star^q$ of (1), Budgets $B^q > 0$

$C_0^q \leftarrow \emptyset \ \forall q \in [k]$, $t \leftarrow 0$
for each task τ_q s.t. $|C_t^q| < B_q$ **do**
  $t \leftarrow t + 1$, observe $\mathbf{s}_t^q \sim \mathcal{P}$ and play $a_t^q \sim \pi_\star^q(\cdot|\mathbf{s}_t^q)$
  if $a_t^q = q$ **then** $C_t^q \leftarrow C_t^q \cup \{\mathbf{s}_t^q\}$
return The coalition structure $\mathcal{CS} = (C^q)_{q=1}^k$

(1) for each M^q can be reduced to the following linear program:

$$\begin{aligned} \max_{\pi^q} \quad & \sum_{(\mathbf{s},a)\in\mathcal{S}\times\mathcal{U}^q} \pi^q(a|\mathbf{s})\mathcal{P}(\mathbf{s})r^q(\mathbf{s}, a) \\ \text{s.t.} \quad & \sum_{(\mathbf{s},a)\in\mathcal{S}\times\mathcal{U}^q} \pi^q(a|\mathbf{s})\mathcal{P}(\mathbf{s})\eta_j^{i,q}(\mathbf{s}, a) = 0 \quad \forall i, j \\ & \sum_{a\in\mathcal{U}^q} \pi^q(a|\mathbf{s}) = 1 \quad \forall \mathbf{s} \in \mathcal{S} \end{aligned} \quad (2)$$

Since the CMDP is ergodic, the above linear program (LP) is feasible by [3]. The following lemma provides us with two desirable characteristics of the above CMDP that are induced by the LP (2):

Lemma 4.1. *Let π, π^q be two policies under M, M^q (resp.). Then, $\mathcal{J}^\pi, \mathcal{J}_q^{\pi^q}$ are the rate at which agents are assigned to some task and task τ_q (resp.), i.e., $\mathcal{J}^\pi = Pr^{\mathcal{P},\pi}[a \in [k]]$ and $\mathcal{J}_q^{\pi^q} = Pr^{\mathcal{P},\pi^q}[a = q]$. If π^q is feasible for M^q, then: $g_j^{i,q} = Pr^{\mathcal{P},\pi^q}[\mathbf{s}^i = j|a = q] \ \forall i, j, q$.*

The proof appears in Appendix B.2 [14]. Lemma 4.1 yields that the optimal policies $\pi_\star$ and $\pi_\star^q$ maximize the rate at which agents are assigned to some task and task τ_q (resp.). Further, the first constraints of (2) require that $g_j^{i,q}$ is the proportion of agents that are assigned to τ_q and depicted by a skill vector $\mathbf{s}$ with $\mathbf{s}^i = j \ \forall i, j$.

4.2 Analysis of a CMDP-Based Approach

In this section, we analyze a CMDP-based algorithm to OCSF that is derived from the stationary optimal policies $\pi_\star^q$ (Algorithm 2). As they are stationary, they can be parallelized so as to *simultaneously* assign an agent to each task. Namely, at time t, the organizer observes a skill vector $\mathbf{s}_t^q \sim \mathcal{P}$ for each task τ_q. He then decides whether to assign $\mathbf{s}_t^q$ to task τ_q by playing $a_t \sim \pi_\star^q(\cdot|\mathbf{s}_t)$. Algorithm 2 terminates when B_q agents have been assigned to each task τ_q. Letting $\mathcal{A}$ denote Algorithm 2, we derive a correlation between $\mathcal{J}_q^{\pi^q}$ ($q \in [k]$) and $\mathbb{E}_\mathcal{P}^\mathcal{A}[\mathcal{T}]$ while using Lemma 4.1.

Lemma 4.2. *$\mathbb{E}_\mathcal{P}^\mathcal{A}[\mathcal{T}] \leq \Sigma_{q\in[k]} \frac{B_q}{\mathcal{J}_q^{\pi^q}}$, where π^q is stationary $\forall q$.*

Proof. For each $q \in [k]$, let $\mathcal{T}^q$ be the first time when B^q agents are assigned to task τ_k. Clearly, $\mathcal{T} \leq \Sigma_{q\in[k]} \mathcal{T}^q$. Note that $\mathcal{T}^q + B^q$ follows a negative binomial distribution with B^q successes and a success probability of $\mathcal{J}_q^{\pi^q} = Pr^{\mathcal{P},\pi^q}[a = q]$ (By Lemma 4.1). Thus, $\mathbb{E}_\mathcal{P}^\mathcal{A}[\mathcal{T}^q + B^q] = \frac{B^q(1-\mathcal{J}_q^{\pi^q})}{\mathcal{J}_q^{\pi^q}}$, which yields $\mathbb{E}_\mathcal{P}^\mathcal{A}[\mathcal{T}^q] = \frac{B^q}{\mathcal{J}_q^{\pi^q}}$. By the expectation's linearity and monotonicity, the desired follows. □

Next, we infer an important property derived from Lemma 4.2. Since the event $a \in [k]$ is equivalent to the event $\cup_{q\in[k]} a = q$, using the inclusion–exclusion principle and Lemma 4.1 one can infer that: $\mathcal{J}^\pi = Pr^{\mathcal{P},\pi}[a \in [k]] = \Sigma_{q\in[k]} Pr^{\mathcal{P},\pi}[a = q] = \Sigma_{q\in[k]} \mathcal{J}_q^\pi$.

Algorithm 3 OptOCSF

Input: Confidence $\delta \in (0,1)$, Goals g^{τ_q}, Budgets $B^q > 0$

$C_0^q \leftarrow \emptyset\ \forall q$; $n_0(\mathbf{s}) \leftarrow 0\ \forall \mathbf{s}$; Let $s_0 \sim \mathcal{P}$ and $n_0(\mathbf{s}_0) = 1$; $t \leftarrow 1$
for each episode $e = 1, 2, \ldots$ **do**
  $T_e = t$, Update $\hat{\mathcal{P}}_e(\mathbf{s}) = \frac{n_{T_e-1}(\mathbf{s})}{T_e - 1}\ \forall \mathbf{s} \in \mathcal{S}$
  Compute the solution π_e of (4) by the ELP (5)–(8)
  while $n_t(\mathbf{s}_t) < 2n_{T_e-1}(\mathbf{s}_t)$ **do**
    $t \leftarrow t+1$, observe $\mathbf{s}_t \sim \mathcal{P}$ and set $n_t(\mathbf{s}_t) \leftarrow n_t(\mathbf{s}_t) + 1$
    Play $a_t \sim \pi_e(\cdot|\mathbf{s}_t)$, $a_t \in \{q : |C_t^q| < B_q \wedge \pi_e(q|\mathbf{s}_t) > 0\}$
    if $\exists q \in [k]: a_t = q$ and $|C_t^q| < B_q$ **then** $C_t^q \leftarrow C_t^q \cup \{\mathbf{s}_t\}$
  if $|C_t^q| = B_q\ \forall q$ **then BREAK**
return The coalition structure $\mathcal{CS} = (C^q)_{q=1}^k$

Hence, *maximizing* $\mathcal{J}^\pi$ as done under the CMDP is equivalent to *maximizing* $\Sigma_{q\in[k]} \mathcal{J}_q^\pi$, which in turn translates to *minimizing* the sample complexity $\mathbb{E}_{\mathcal{P}}^{\mathcal{A}}[\mathcal{T}]$ by Lemma 4.2. Given the output $\mathcal{CS} = (C^q)_{q=1}^k$ of Algorithm 2, we now upper bound the distance $d(\mathbf{x}(C^q), g^{\tau_q})$ induced by the L^∞-norm for each τ_q as follows:

Theorem 4.3. *Let $\pi_\star^q$ be the stationary optimal policies under each M^q in* (1)*. Let $\tilde{\alpha} = \Sigma_{i\in[m]}(\alpha_i - 1)$ be the number of pairs $i \in [m]$, $j \in [\alpha_i - 1]$. Let $\delta_q \in (0,1)$, $\mathcal{T}^q \geq B^q$. Then, under $\pi_\star^q$ and $\mathcal{P}$, $\|\mathbf{x}(C_{\mathcal{T}^q}^q) - g^{\tau_q}\|_\infty \leq \sqrt{\log(2\tilde{\alpha}/\delta_q)/(2B^q)}$ with probability $\geq 1 - \delta_q$.*

The proof appears in Appendix B.3 [14]. For each task τ_q, the high probability upper bound of $O(\sqrt{1/B^q})$ in Theorem 4.3 *decreases* with the budget. Hence, Algorithm 2 guarantees that the resulting coalition structure satisfies the skill levels required by each task as much as possible for higher budgets. Further, though $\pi_\star$ acts independently from previously assigned agents, we infer that it performs well for higher budgets. Intuitively, adding an agent to a large coalition has less effect on the coalition's current skill vector.

5 THE AGENTS' DISTRIBUTION IS *UNKNOWN*

In this section, we assume that the distribution $\mathcal{P}$ is *unknown*, and should thus be *learned online*. An (online) learning algorithm with no prior knowledge is required to obtain estimates of $\mathcal{P}$, while obtaining rewards and costs for each state-action pair. Initially, the algorithm does not have good estimates of the model, and thus accumulates a regret and constraint violations as it does not know the optimal policy. Formally, given the stationary optimal policy $\pi_\star$ under the CMDP M in (1), let $\mathcal{J}^\star = \mathcal{J}^{\pi_\star}$ ($\mathcal{J}_q^\star = \mathcal{J}_q^{\pi_\star}$) be the optimal long-term average reward (of task τ_q) under the CMDP when $\mathcal{P}$ is *known*. We define the ***regret*** $R(T)$ as the difference between the expected rewards from running $\pi_\star$ and the cumulative reward obtained for T time steps, i.e., $R(T) = \Sigma_{t=1}^T(\mathcal{J}^\star - r(\mathbf{s}_t, a_t))$. Similarly, let $R^q(T) = \sum_{t=1}^T(\mathcal{J}_q^\star - r^q(\mathbf{s}_t, a_t))$ be the regret of task τ_q. The ***regret of constraint violations*** is defined by $R^c(T) = \max_{i,j,q} R_j^{i,q}(T)$, where $R_j^{i,q}(T) = |\Sigma_{t=1}^T \eta_j^{i,q}(\mathbf{s}_t, a_t)|$ is the constraint violations of task τ_q associated with skill i and level j. Our goal is then *minimizing* both $R(T)$ and $R^c(T)$.

By adapting the OptCMDP algorithm proposed by [20] for *finite-horizon* CMDPs, we first devise Algorithm 3 for *long-term average rewards* in CMDPs (Subsection 5.1). Though a similar adaptation was performed in [19], the challenges discussed in Remark 1 compel us to use a different approach which treats *multiple* coalitions instead of a *single* one. In some applications, the organizer is more sensitive to over-utilizing specific skill levels when considering a certain task compared to the rest. Yet, Algorithm 3 and [19]'s algorithm ignore such scenarios. Unlike [19], we propose an algorithm that enables the organizer to prespecify the desired upper bounds on each component of the regret of constraint violations, and keep them below those bounds (Subsection 5.2).

5.1 Optimism in the Face of Uncertainty

Adapting OptCMDP [20], we herein introduce Algorithm 3. The OptCMDP algorithm builds upon the popular reinforcement learning algorithm UCRL2 [5], that follows the principle of *optimism in the face of uncertainty*. Following OptCMDP, we proceed in episodes, where at each episode we build a set of plausible CMDPs compatible with the observed samples, and then play the optimal policy of the CMDP with the lowest cost (i.e., optimistic CMDP).

Each episode of Algorithm 3 terminates whenever the number of observations for some agent $\mathbf{s}$ doubles. At episode $e \geq 0$ and time $t \geq 0$, agent $\mathbf{s}_t$ is assigned to a task based on a *single* stationary and optimistic policy π_e, which we subsequently depict its construction. Let T_e be the inception time of episode e and $I_e = [T_e, T_{e+1}]$. Let $n_t(\mathbf{s}) = \Sigma_{t'=1}^t \mathbb{1}_{\{\mathbf{s}_{t'}=\mathbf{s}\}}$ be the number of times $\mathbf{s}$ was observed until time t and let $n^q(t-1) = |C_{t-1}^q| = \Sigma_{t'=1}^{t-1} \mathbb{1}_{\{a_{t'}=q\}}$. Let $n_j^i(C_{t-1}^q) = \Sigma_{t'=1}^{t-1} \mathbb{1}_{\{\mathbf{s}_{t'}^i=j, a_{t'}=q\}}$ be the number of agents $\mathbf{s}$ with $\mathbf{s}^i = j$ that were assigned to task τ_q before time t. At each episode e, Algorithm 3 estimates the true distribution $\mathcal{P}$ of the agents via its empirical average $\hat{\mathcal{P}}_e(\mathbf{s}) = \frac{n_{T_e-1}(\mathbf{s})}{T_e-1}$. As in UCRL2, at the beginning of each episode e, we construct confidence intervals $\mathcal{D}_e$ for $\mathcal{P}$. $\mathcal{D}_e$ is built using the L^1 concentration inequality of [53], according to which:

$$\|\hat{\mathcal{P}}_e - \mathcal{P}\|_1 \leq \sqrt{\frac{2|\mathcal{S}|\log(3|\mathcal{S}||\mathcal{U}|T_e(T_e-1)/\delta)}{T_e - 1}} \triangleq \beta_e \quad (3)$$

with probability $\geq 1 - \frac{\delta}{3}$ for any $\delta \in (0,1)$. Hence, let $\mathcal{D}_e = \{\tilde{\mathcal{P}} \in \Delta(\mathcal{S}) : \|\hat{\mathcal{P}}_e - \mathcal{P}\|_1 \leq \beta_e\}$. The set of plausible CMDPs corresponding to $\mathcal{D}_e$ is then $\mathcal{M}_e = \{\tilde{M} = (\mathcal{S}, \mathcal{U}, \tilde{\mathcal{P}}, r, \eta) : \tilde{\mathcal{P}} \in \mathcal{D}_e\}$, using which Algorithm 3 solves the following optimization problem at the inception of each episode e:

$$\max_{\tilde{\mathcal{P}}\in\mathcal{D}_e, \pi} \left\{\mathcal{J}^{\tilde{\mathcal{P}},\pi} \middle| \mathcal{J}_j^{i,q}(\tilde{\mathcal{P}}, \pi) = 0 \quad \forall i \in [m], j \in [\alpha_i], q \in [k]\right\} \quad (4)$$

In Appendix C.3 [14], we prove $(\pi_\star, \mathcal{P})$ to be feasible under (4) and that the policy recovered by solving it is optimistic. Unlike (1), in (4) the transitions are *unknown*, and we thus cannot directly optimize (4). Thus, we rewrite (4) as an ***extended LP*** (ELP). As in [42], we consider the state-action occupation measure $\varphi(\mathbf{s}, a) = \pi(\mathbf{s}, a)\mathcal{P}(\mathbf{s})$ and variables $\beta(\mathbf{s})$ that linearize the L^1 constraint (3) induced by the confidence set $\mathcal{D}_e$ for formulating the ELP:

$$\max_{\varphi,\beta} \quad \sum_{(\mathbf{s},a)\in\mathcal{S}\times\mathcal{U}} \varphi(\mathbf{s}, a) r(\mathbf{s}, a) \quad (5)$$

$$\text{s.t.} \quad \varphi \geq 0, \sum_{(\mathbf{s},a)\in\mathcal{S}\times\mathcal{U}} \varphi(\mathbf{s}, a) = 1 \quad (6)$$

$$\hat{\mathcal{P}}_e(\mathbf{s}) - \beta(\mathbf{s}) \leq \sum_{a\in\mathcal{U}} \varphi(\mathbf{s}, a) \leq \hat{\mathcal{P}}_e(\mathbf{s}) + \beta(\mathbf{s}), \beta(\mathbf{s}) \leq \beta_e \forall \mathbf{s} \in \mathcal{S} \quad (7)$$

$$\sum_{(\mathbf{s},a)\in\mathcal{S}\times\mathcal{U}} \varphi(\mathbf{s},a)\eta_j^{i,q}(\mathbf{s},a) = 0 \quad \forall i,j,q \tag{8}$$

For each task τ_q, (8) enforces that $g_j^{i,q}$ is the proportion of agents that are assigned to τ_q and represented by a skill vector $\mathbf{s}$ with $\mathbf{s}^i = j\ \forall i,j$. The constraints (7) require the compatibility of φ with the L^1 constraint. If the ELP is infeasible, then we set the policy as $\pi_e(a|\mathbf{s}) = \frac{1}{k+1}\ \forall \mathbf{s}, a$. Otherwise, once φ is obtained, we recover the transitions as $\tilde{\mathcal{P}}_e(\mathbf{s}) = \Sigma_a\, \varphi(\mathbf{s},a)$ and the policy as $\pi_e(a|\mathbf{s}) = \frac{\varphi(\mathbf{s},a)}{\tilde{\mathcal{P}}_e(\mathbf{s})}$ if $\tilde{\mathcal{P}}_e(\mathbf{s}) \neq 0$. If $\tilde{\mathcal{P}}_e(\mathbf{s}) = 0$, then $\pi_e(a|\mathbf{s}) = \frac{1}{k+1}\ \forall a$. As the MDP induced by $\tilde{\mathcal{P}}_e$ is still weakly communicating and any policy is unichain (Readers unfamiliar with these notions may refer to [6]), the optimal long-term average reward of our CMDP is unaffected.

Algorithm 3 terminates when B_q agents have been assigned to each task τ_q. Given the output $\mathcal{CS} = (C^q)_{q=1}^k$ of Algorithm 3, Theorem 5.1 that follows provides the regret and constraint violations bounds, as well as a high probability upper bound on the distance $d(\mathbf{x}(C^q), g^{\tau_q})$ induced by the L^∞-norm for each τ_q.

Theorem 5.1. *Let $\delta \in (0,1)$. Then, with probability $\geq 1-\delta$, the regrets $R(T), R^q(T)$ and the regret of constraint violations $R^c(T)$ are upper bounded by $O(\sqrt{|\mathcal{S}|T\log(|\mathcal{S}||\mathcal{U}|T/\delta)} + \sqrt{T})$. Further, after T time steps, the following holds with probability $\geq 1-\delta$:*

$$\|\mathbf{x}(C_T^q) - g^{\tau_q}\|_\infty = O((\sqrt{|\mathcal{S}|\log(|\mathcal{S}||\mathcal{U}|T/\delta)} + 1)\,/\,(\sqrt{T}\mathcal{J}_q^\star)) \tag{9}$$

The proof is deferred to Appendix C.4 [14]. Recalling that $n(C_T^q) = \sum_{t=1}^T \mathbb{1}_{\{a_t=q\}}$ and $n_j^i(C_T^q) = \sum_{t=1}^T \mathbb{1}_{\{\mathbf{s}_t^i=j, a_t=q\}}$, note that $R^q(T)/T = \mathcal{J}_q^\star - n(C_T^q)/T$. Theorem 5.1 thus indicates that, with high probability, the difference between the optimal assignment rate $\mathcal{J}_q^\star$ of agents to task τ_q and the assignment rate $n(C_T^q)/T$ to task τ_q under OptOCSF decreases according to $O(\sqrt{\log(T)/T})$ as T increases. Further, the same applies to the constraint violations and the distance $d(\mathbf{x}(C^q), g^{\tau_q})$. Hence, the organizer shall observe sufficiently many agents so as to assign them to tasks at a low expense of regret, while meeting the skill level requirements as much as possible.

By (9), the distance $d(\mathbf{x}(C^q), g^{\tau_q})$ is *inversely* proportional to the optimal assignment rate $\mathcal{J}_q^\star$ of agents to tasks. Intuitively, if agents are assigned to task τ_q at a *low* rate (i.e., $n(C_T^q)$ is small), then the coalition is too *small* to meet the skill levels required by the task, and thus the distance $\|\mathbf{x}(C_T^q) - g^{\tau_q}\|_\infty = R^q(T)/n(C_T^q)$ may be quite *large*. However, $R_j^{i,q}(T) = |n_j^i(C_T^q) - g_j^{i,\tau_q} n(C_T^q)|$ may be *small* under a *low* assignment rate since OptOCSF controls $n_j^i(C_T^q)$ such that the number of agent with level j at mastering skill i is as close as possible to being a fraction of g_j^{i,τ_q} out of all the $n(C_T^q)$ agents assigned to τ_q. Finally, opposed to Theorem 4.3, not knowing $\mathcal{P}$ hinders the coalitions' compliance with the skill level required by their respective tasks: we suffer an additional factor of $O(\sqrt{|\mathcal{S}|\log(|\mathcal{S}||\mathcal{U}|)})$ due to the estimation (3).

5.2 Tuning Bounds on Constraint Violations

By Theorem 5.1, Algorithm 3 yields the *same* upper bound on both the regret and constraint violations. However, in some scenarios the organizer is more sensitive to over-utilizing specific skill levels when considering a certain task, which translates to a susceptible constraint. Hence, we propose a modification to Algorithm 3 which allows the organizer to *tune* the upper bound on the regret of each constraint violation, and thus referred to as **TuneOptOCSF**.

Though the optimistic approach presented in Subsections 5.1-5.2 incentivizes the algorithm to explore policies that can visit new state-action pairs, it allows exploratory policies that violate the constraints with respect to the true transition dynamics $\mathcal{P}$. That is, it is possible that $\mathcal{J}_j^{i,q}(\tilde{\mathcal{P}}_e, \pi_e) = 0$ for some i,j,q, but $\mathcal{J}_j^{i,q}(\mathcal{P}, \pi_e) \neq 0$. Thus, we consider a tightened version of (4):

$$\max_{\tilde{\mathcal{P}}\in\mathcal{D}_e,\pi} \left\{\mathcal{J}^{\tilde{\mathcal{P}},\pi} \middle| \mathcal{J}_j^{i,q}(\tilde{\mathcal{P}},\pi) \leq \epsilon_j^{i,q}\ \forall i,j,q\right\} \tag{10}$$

where $\varepsilon_j^{i,q} \in (0,1)$ is a constant pessimistic term that restrains the constraint violations. Yet, (10) may not have any feasible solution. Thus, as in [44], we make the following standard assumption:

Assumption 2. *There are a policy π^b and a constant $\theta > 0$ s.t. $\mathcal{J}_j^{i,q}(\mathcal{P},\pi^b) = c_j^{i,q} < \varepsilon_j^{i,q} - \theta\ \forall i,j,q$ (θ is known to the organizer).*

Under Assumption 2, Slater's condition [11, 20] holds and thus (10) is strictly feasible. Accordingly, the TuneOptOCSF algorithm follows Algorithm 3, yet involves solving an extended LP (ELP) different from (5)-(8). Namely, (10) can be rephrased as an ELP similar to (5)-(8) except that (8) is substituted with the constraints $\Sigma_{(\mathbf{s},a)\in\mathcal{S}\times\mathcal{U}}\, \varphi(\mathbf{s},a)\eta_j^{i,q}(\mathbf{s},a) \leq \epsilon_j^{i,q} - \theta_j^{i,q}\ \forall i,j,q$, where $\theta_j^{i,q} = \zeta_j^{i,q}\theta$ with $\zeta_j^{i,q} \in (0,1)$ being parameters chosen by the organizer for controlling the desired upper bounds. Similarly to Subsection 5.1, we can recover a transition dynamics $\tilde{\mathcal{P}}_e$ and a policy π_e from using the solution φ of the resulting ELP. Next, we bounds in the same vein as Theorem 5.1 (The proof appears in Appendix D [14]).

Theorem 5.2. *Let $\delta \in (0,1)$, $\iota = \min_{i,j,q}(\varepsilon_j^{i,q} - \theta - c_j^{i,q})$ and $\omega = \frac{\theta}{\iota}\max_{i,j,q}\zeta_j^{i,q}$. Then, with probability $\geq 1-\delta$, the regret and regret of constraint violations satisfy $R(T) \leq O(\sqrt{|\mathcal{S}|T\log(|\mathcal{S}||\mathcal{U}|T/\delta)} + \sqrt{T}) + \omega T$ and $R^c(T) \leq O(\sqrt{|\mathcal{S}|T\log(|\mathcal{S}||\mathcal{U}|T/\delta)} + \sqrt{T}) + (\epsilon_j^{i,q} - \theta_j^{i,q})T$. After T time steps, the following holds with probability $\geq 1-\delta$:*

$$d(\mathbf{x}(C^q), g^{\tau_q}) = O(\kappa_T/(\mathcal{J}_q^\star - \omega) + (\epsilon_j^{i,q} - \theta_j^{i,q})/(\mathcal{J}_q^\star - \omega - \kappa_T)) \tag{11}$$

where $\kappa_T = (\sqrt{|\mathcal{S}|\log(|\mathcal{S}||\mathcal{U}|T/\delta)} + 1)/\sqrt{T}$.

Remark 2. *Theorem 5.2 dictates that the regret of constraint violations and the distance $d(\mathbf{x}(C^q), g^{\tau_q})$ can be made arbitrarily* small *by the organizer for a proper choice of the controlled parameters $\zeta_j^{i,q}$, at the expense of hindering the regret $R(T)$. Unlike* (9), (11) *comprises of two terms. The first term is analogous to* (9), *yet it is* inversely *proportional to $\mathcal{J}_q^\star - \omega$ instead of $\mathcal{J}_q^\star$. However, the second term is inversely proportional to $\mathcal{J}_q^\star - \omega - \kappa_T$. Namely, even when the assignment rate to task τ_q is* small, *ω allows the organizer to achieve a* smaller *distance by tuning the parameters $\zeta_j^{i,q}$ for reducing both terms. Additionally, the organizer shall observe sufficiently many agents so as to decrease the second term even further.*

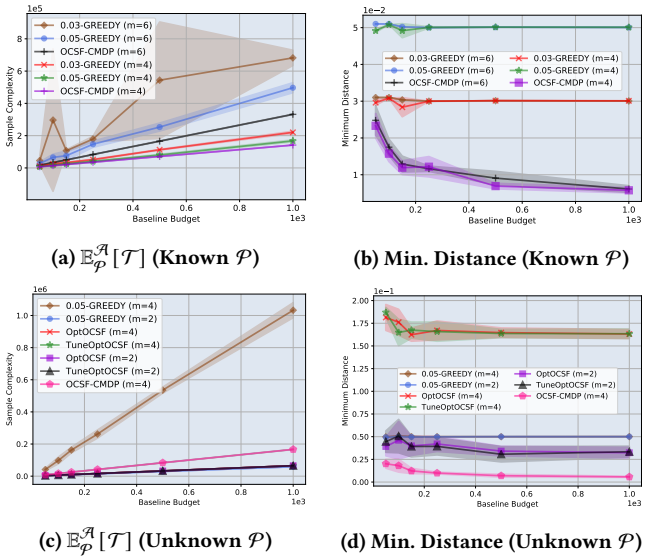
6 EMPIRICAL EVALUATIONS

In this section, we evaluate our algorithms on a synthetic dataset. Our main goal is supplying practical guidelines defining the preferred algorithm to use according to the situation at hand, which depend on the attributes compound by the considered problem instance (e.g., number of skills, budgets). As such, we investigate the impact of such characteristics on our algorithms' performance in terms of both the proximity of the resulting coalition structures' skill coverages from the tasks' goals and the expected number of agents contacted (i.e., the *sample complexity*). Given a coalition structure $\mathcal{CS} = (C^q)_{q=1}^k$, we term the first measure as its *minimum distance*, which is assessed via $\min_q d(\mathbf{s}(C^q), g^{\tau_q})$.

***Experimental Setup* [15].** We generate our novel synthetic dataset as follows. Unless stated otherwise, we refer to a baseline scenario where we consider a set of $k = 10$ tasks and $m = 6$ possible skills. Each of the first four skills allows for $\alpha_i = 2$ ($i \in [4]$) mastering levels, whereas the remaining two skills allow for $\alpha_5 = 3$ and $\alpha_6 = 8$ mastering levels. Since neither the agents' distribution $\mathcal{P}$ nor the tasks' goals $\mathcal{G}$ are controlled by the organizer, both were randomly initialized. Since our desire is also exploring the budgets' effects upon the algorithms, each instance is accompanied with a baseline budget $B \in \{50, 100, 150, 250, 500, 1000\}$ such that the budget of task τ_q is $B \cdot q$. Thus, we model diversity in the budgets in addition to the tasks' requirements and the agents' skills. Further, we consider that Greedy (Algorithm 1) has a tolerance of either $\epsilon = 0.03$ or $\epsilon = 0.05$, whereas both OptOCSF (Algorithm 3) and TuneOptOCSF (Algorithm 1) have a confidence of $\delta = 0.01$. Additionally, in TuneOptOCSF, we set $\varepsilon_j^{i,q} = 5 \cdot 10^{-4}$ $\forall i, j, q$ as the constant pessimistic terms in (10). As the baseline policy π^b and constant $\theta > 0$ in Assumption 2, we select the optimal solution of the given tightened CMDP (10) with $\varepsilon_j^{i,q}/5$ and set $\theta = 2 \cdot 10^{-11}$. Given θ, the organizer selects the controlled parameters $\zeta_j^{i,q} \in (0, 1)$ randomly $\forall i, j, q$. Finally, plots are generated by averaging results over 5 runs of each algorithm so as to reduce noise.

As a side note, we observed that our algorithms are not majorly affected by the number of tasks k, except for our online learning schemes. This is consistent with our theoretical results in Theorems 3.1 and 4.3 regarding both Greedy and OCSF-CMDP (resp.), which dictate that the upper bounds guaranteed by both algorithms do not depend on the number of tasks. In contrast, by Theorems 5.1-5.2, our online learning schemes suffer an additional factor of $O(\sqrt{|\mathcal{S}|\log(|\mathcal{S}||\mathcal{U}|)})$ due to the estimation (3) of $\mathcal{P}$. Thereby, we solely report the effects of the baseline budget B, the number of skills m, the tolerance ϵ and the knowledge of $\mathcal{P}$ on our algorithms.

When the agents' distribution is *known*, we compare OCSF-CMDP and Greedy. For verifying the impact of the diversity in the possible numbers of skills and mastering levels, we regard our baseline scenario as well as another scenario where there are only two binary skills instead of four, and thus $m = 4$ in the second case. Despite OCSF-CMDP's non-adaptivity, Figures 1a-1b depict that it surpasses the other algorithms in terms of both sample complexity and minimum distance, regardless of the scenario at hand. Particularly, as expected by Theorem 4.3, the minimum distance attained by OCSF-CMDP indeed *decreases* with the baseline budget. Further,



(a) $\mathbb{E}_{\mathcal{P}}^{\mathcal{A}}[\mathcal{T}]$ (Known $\mathcal{P}$) (b) Min. Distance (Known $\mathcal{P}$)

(c) $\mathbb{E}_{\mathcal{P}}^{\mathcal{A}}[\mathcal{T}]$ (Unknown $\mathcal{P}$) (d) Min. Distance (Unknown $\mathcal{P}$)

Figure 1: Impact of the baseline budget B, the number of skills m, the tolerance ϵ and the knowledge of $\mathcal{P}$ on our algorithms.

Theorem 4.3 also dictates that the high probability upper bound ensured by OCSF-CMDP should *increase* the number of pairs $i \in [m]$, $j \in [\alpha_i - 1]$, a property that is verified by our experiments as the minimum distance is *higher* when $m = 6$ skills are considered.

Though Theorem 3.1 yields that the upper bound obtained by Greedy *increases* with the maximum possible skill level and *decreases* with the budget, both effects can only be observed for small budgets and Greedy does not improve for larger ones. However, a lower tolerance ($\epsilon = 0.03$) achieves a *lower* distance across all budgets. Concerning the sample complexity, in both scenarios Greedy requires more samples than OCSF-CMDP, as expected. Particularly, note that the better accuracy of Greedy with $\epsilon = 0.03$ comes at the expense of sample complexity, as it requires more samples than Greedy with $\epsilon = 0.05$ for reaching a lower minimum distance.

When the agents' distribution is *unknown*, we consider two different settings: (1) two binary skills ($m = 2$), and (2) two binary skills and two additional skills with 3 and 4 possible mastering levels, respectively ($m = 4$). Initially, we study how the knowledge on the agents' distribution influences our algorithms. Due to the high sample complexity induced by Greedy with $\epsilon = 0.03$ (Figure 1a), we focus on a tolerance of $\epsilon = 0.05$. Figures 1c-1d illustrate that both of our online learning algorithms require *higher* budgets for their incurred minimum distance to *decrease*, compared to OCSF-CMDP which always reaches the lowest minimum distance among all algorithms. Hence, not knowing the agents' distribution $\mathcal{P}$ harms the produced minimum distance as it is required to estimate $\mathcal{P}$. However, in terms of sample complexity, both OptOCSF and TuneOptOCSF contact *less* agents than OCSF-CMDP until the algorithms' termination. Comparing our online learning schemes against Greedy, we observe that for smaller skill domains ($m = 2$) both OptOCSF and TuneOptOCSF reach *lower* values of both minimum distance and sample complexity. Thus, there are

cases where one may prefer *learning* the agents' distribution $\mathcal{P}$ instead of following Greedy with (or without) the knowledge of $\mathcal{P}$.

We also observe that the *larger* the skill vectors' domain $\mathcal{S}$, the *larger* the minimum distance gained by our online learning approaches. This result is aligned with Theorems 5.1-5.2: due to the estimation (3) of $\mathcal{P}$, we suffer an additional factor of $O(\sqrt{|\mathcal{S}|\log(|\mathcal{S}||\mathcal{U}|)})$. Regardless of $|\mathcal{S}|$ and consistently with Remark 2, TuneOptOCSF usually reaches a *lower* minimum distance than OptOCSF.

In summary, a higher diversity in skills may hinder the minimum distance attained. However, when $\mathcal{P}$ is *known*, OCSF-CMDP achieves the *best* performance compared to the other schemes, though Greedy functions sufficiently well for *small* budgets. If $\mathcal{P}$ is *unknown*, then *learning* it online given *less* heterogeneous agents (i.e., $|\mathcal{S}|$ is small) may be preferred over using Greedy. For larger $|\mathcal{S}|$, one should favor Greedy. As Greedy assigns agents to the *first* suitable task instead of the one minimizing the minimum distance, we have simulated the later variant and observed that Greedy performs *better* (Appendix E [14]). Future work thus warrants examining the proper order in which tasks are handled.

7 DISCUSSION: INFINITE SKILL DOMAINS

In this section, we discuss the benefits from considering *finite* skill domains instead of *infinite* ones where $\mathcal{S}_i = \mathbb{R}_{\geq 0}$ for each skill i. Similarly, for each task τ and skill i, τ's required level at mastering skill i is modeled by the level $g^{i,\tau} \in \mathbb{R}$ needed for accomplishing the task τ. Let $g^{\tau} = (g^{i,\tau})_{i=1}^{m}$ be τ*'s goal*. Further, we evaluate a coalition directly via the *summation* of its members' skill vectors, i.e., the *skill vector of a coalition* $C \subseteq \mathcal{A}$ is the sum of skill vectors of agents from C, denoted by $\mathbf{s}(C) = \Sigma_{\ell \in C}\, \mathbf{s}_\ell$. Similarly to Section 3, for any task τ_q, the distance $d(\mathbf{s}(C), g^{\tau_q})$ is measured in the L^∞-norm, where $d(\mathbf{s}(C), g^{\tau_q}) = \|\mathbf{s}(C) - g^{\tau_q}\|_\infty = \max_{i\in[m]} |\mathbf{s}^i(C) - g^{i,\tau_q}|$.

Herein, we consider a more general *online* model for agents' appearance: an agent appears with *multiple* operation modes (i.e., multiple skill vectors, and thus is referred to as such), each one associated with a different task. At time $t \geq 1$, the organizer observes a *single* agent with k operation modes encapsulated by k skill vectors $\mathbf{s}_{t,q} \in \mathcal{S}$ $(q \in [k])$ that appear online. The organizer must *immediately* and *irrevocably* decide to which task the agent should be assigned, if at all. Once the organizer decides to assign the agent to task τ_q, she is equipped with the skill vector $\mathbf{s}_{t,q}$.

7.1 Tight Bounds on the Competitive Ratio

In this section, we prove tight upper and lower bound on the competitive ratio for OCSF under the *multiple skill vectors* model in the L^∞-norm. We follow the standard notions of competitive analysis in online settings [10] (Readers may refer to Appendix F for a brief [14]). In our own context, we assume that (by scaling) the optimal norm for each task is 1. Hence, an online algorithm is β_q*-competitive* if $d(\mathbf{x}(C^q), g^{\tau_q}) \leq \beta_q$ for any task $\tau_q \in \Gamma$. We remark that in the subsequent proofs we first consider that the goal is simultaneously minimizing the L^{p_q}-norms for each task τ_q with $1 \leq p_q \leq \log m$, and then infer the bounds for the L^∞-norm when $p_q = \log k$. Formally, under the L^p-norm $(p \geq 1)$, $d(\mathbf{s}(C), g^{\tau_q}) = \|\mathbf{s}(C) - g^{\tau_q}\|_p = \sqrt[p]{\Sigma_{i\in[m]}(\mathbf{s}^i(C) - g^{i,\tau_q})^p}$.

Theorem 7.1. **(Lower Bound)** *There is a lower bound of* $\Omega(\log m + \log k)$ *on the competitive ratio of deterministic online algorithms for OCSF for each task* τ_q*, where the goal is simultaneously minimizing* $\|\mathbf{s}(C^q) - g^{\tau_q}\|_\infty$ *for every task* τ_q*.*

Proof. See Appendix G.1 for a detailed proof [14]. □

Theorem 7.2. **(Upper Bound)** *There is an online algorithm for OCSF that obtains a competitive ratio of* $O(\log k + \log m)$ *for each task* τ_q *(the goal is simultaneously minimizing* $\|\mathbf{s}(C^q) - g^{\tau_q}\|_\infty$ $\forall q$*).*

Proof. Our algorithm is an adaptation of the algorithm provided in [30, Theorem 4]. In Appendix G.2 [14], we prove that it leads to the asymptotically optimal competitive ratio for every task. □

By Theorems 7.1–7.2, the optimal competitive ratio for OCSF under the multiple skill vectors model in the L^∞-norm is $\Theta(\log m + \log k)$. Though it is independent of the number of agents, the logarithmic dependence on the number of skills (m) and tasks (k) still poses a challenge. When agents arrive i.i.d. following a known distribution, competitive ratios for certain problems can be improved (e.g., Submodular Welfare Maximization [2], stochastic matching [37]), thus motivating our consideration of the i.i.d. stochastic model in the previous sections. Indeed, unlike our results in Theorems 7.1–7.2, the upper bounds in Theorem 3.1 for the L^∞- and L^1-norms do *not* depend on the number of either tasks k or skills m. Namely, increasing the values of either k or m will *not* degrade these bounds. Finally, note that Algorithm 1 can be readily adapted to the infinite skill domains case by the conditional statement substituting in line 4 with $n_j^i(C_{t-1}^q) + \mathbb{1}_{s_t^i=j} \leq g^{i,\tau_q} + \epsilon$ $\forall i, j$. Yet, the rest of the algorithms cannot be extended directly as CMDPs with infinite state spaces will be obtained, a direction that is thus left for future work.

8 CONCLUSIONS AND FUTURE WORK

We introduced the novel framework of *online coalitional skill formation* (OCSF) that takes a coalition formation approach for allocating heterogeneous tasks to agents that arrive *online* and have *diverse* skills. Due to the limitations of existing skill models, we presented a *new* skill model, where the set of possible mastering levels for each skill is *discrete*, and a coalition is evaluated by the extent each skill level is *covered* by the coalition. Based on the knowledge regarding agents' arrivals, we devised different algorithms, involving a greedy scheme, a novel correlation to Constrained Markov Decision Processes and two online learning algorithms. Empirically, we depicted that a higher diversity in skills may yield poor assignments.

Our study is bound to form a windfall of future studies, including cases where agents can strategically modify their skills at a cost for improving their capabilities. Further, an agent's entire skill vector may not be revealed once she arrives. Instead, the organizer might observe only a quantitative value depicting the agent's overall quality. Finally, as typical in online settings, we assumed that decisions are immediate and irrevocable, yet they may be neither of them.

ACKNOWLEDGMENTS

This research was funded in part by ISF grant #1563/22.

REFERENCES

[1] Shipra Agrawal and Nikhil Devanur. 2016. Linear contextual bandits with knapsacks. *Advances in Neural Information Processing Systems* 29 (2016).

[2] Saeed Alaei, MohammadTaghi Hajiaghayi, and Vahid Liaghat. 2012. Online prophet-inequality matching with applications to ad allocation. In *Proceedings of the 13th ACM Conference on Electronic Commerce*. 18–35.

[3] Eitan Altman. 1999. *Constrained Markov Decision Processes*. Vol. 7. CRC Press.

[4] Sepehr Assadi, Justin Hsu, and Shahin Jabbari. 2015. Online assignment of heterogeneous tasks in crowdsourcing markets. In *Third AAAI Conference on Human Computation and Crowdsourcing*.

[5] Peter Auer, Thomas Jaksch, and Ronald Ortner. 2008. Near-optimal regret bounds for reinforcement learning. *Advances in neural information processing systems* 21 (2008).

[6] Peter Bartlett and Ambuj Tewari. 2009. REGAL: a regularization based algorithm for reinforcement learning in weakly communicating MDPs. In *Uncertainty in Artificial Intelligence: Proceedings of the 25th Conference*. AUAI Press, 35–42.

[7] Senjuti Basu Roy, Ioanna Lykourentzou, Saravanan Thirumuruganathan, Sihem Amer-Yahia, and Gautam Das. 2015. Task assignment optimization in knowledge-intensive crowdsourcing. *The VLDB Journal* 24, 4 (2015), 467–491.

[8] Xiaohui Bei, Shengxin Liu, Chung Keung Poon, and Hongao Wang. 2022. Candidate selections with proportional fairness constraints. *Autonomous Agents and Multi-Agent Systems* 36, 1 (2022), 1–32.

[9] Dimitri P Bertsekas. 1988. The auction algorithm: A distributed relaxation method for the assignment problem. *Annals of operations research* 14, 1 (1988), 105–123.

[10] Allan Borodin and Ran El-Yaniv. 1998. *Online computation and competitive analysis*. cambridge university press.

[11] S BoydandL. 2004. Vandenberghe, convex optimization.

[12] Alessandro Bozzon, Marco Brambilla, Stefano Ceri, Matteo Silvestri, and Giuliano Vesci. 2013. Choosing the right crowd: expert finding in social networks. In *Proceedings of the 16th international conference on extending database technology*. 637–648.

[13] Markus Brill, Jean-François Laslier, and Piotr Skowron. 2018. Multiwinner approval rules as apportionment methods. *Journal of Theoretical Politics* 30, 3 (2018), 358–382.

[14] Saar Cohen and Noa Agmon. 2023. Online Coalitional Skill Formation. https://www.cs.biu.ac.il/~agmon/CohenAAMAS23Sup.pdf.

[15] Saar Cohen and Noa Agmon. 2023. Online Coalitional Skill Formation. https://github.com/saarcohen30/ocsf.

[16] Nikhil R Devanur, Kamal Jain, Balasubramanian Sivan, and Christopher A Wilkens. 2011. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. In *Proceedings of the 12th ACM conference on Electronic commerce*. 29–38.

[17] Nikhil R Devanur, Balasubramanian Sivan, and Yossi Azar. 2012. Asymptotically optimal algorithm for stochastic adwords. In *Proceedings of the 13th ACM Conference on Electronic Commerce*. 388–404.

[18] John P Dickerson, Karthik Abinav Sankararaman, Aravind Srinivasan, and Pan Xu. 2018. Assigning Tasks to Workers based on Historical Data: Online Task Assignment with Two-sided Arrivals. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. 318–326.

[19] Virginie Do, Jamal Atif, Jérôme Lang, and Nicolas Usunier. 2021. Online Selection of Diverse Committees. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. International Joint Conferences on Artificial Intelligence Organization, 154–160.

[20] Yonathan Efroni, Shie Mannor, and Matteo Pirotta. 2020. Exploration-exploitation in constrained mdps. *arXiv preprint arXiv:2003.02189* (2020).

[21] Ju Fan, Guoliang Li, Beng Chin Ooi, Kian-lee Tan, and Jianhua Feng. 2015. iCrowd: An adaptive crowdsourcing framework. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*. 1015–1030.

[22] Brian P Gerkey and Maja J Mataric. 2002. Sold!: Auction methods for multirobot coordination. *IEEE transactions on robotics and automation* 18, 5 (2002), 758–768.

[23] Brian P Gerkey and Maja J Matarić. 2004. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International journal of robotics research* 23, 9 (2004), 939–954.

[24] Bin Guo, Yan Liu, Leye Wang, Victor OK Li, Jacqueline CK Lam, and Zhiwen Yu. 2018. Task allocation in spatial crowdsourcing: Current state and future directions. *IEEE Internet of Things Journal* 5, 3 (2018), 1749–1764.

[25] Kai Han, Chi Zhang, and Jun Luo. 2015. Taming the uncertainty: Budget limited robust crowdsensing through online learning. *Ieee/acm transactions on networking* 24, 3 (2015), 1462–1475.

[26] Umair Ul Hassan and Edward Curry. 2014. A multi-armed bandit approach to online spatial task assignment. In *2014 IEEE 11th Intl Conf on Ubiquitous Intelligence and Computing and 2014 IEEE 11th Intl Conf on Autonomic and Trusted Computing and 2014 IEEE 14th Intl Conf on Scalable Computing and Communications and Its Associated Workshops*. IEEE, 212–219.

[27] Umair Ul Hassan, Sean O'Riain, and Edward Curry. 2013. Effects of expertise assessment on the quality of task routing in human computation. In *2nd International Workshop on Social Media for Crowdsourcing and Human Computation (SoHuman 2013) 2*. 1–10.

[28] Danula Hettiachchi, Vassilis Kostakos, and Jorge Goncalves. 2022. A survey on task assignment in crowdsourcing. *ACM Computing Surveys (CSUR)* 55, 3 (2022), 1–35.

[29] Chien-Ju Ho and Jennifer Vaughan. 2012. Online task assignment in crowdsourcing markets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 26. 45–51.

[30] Sungjin Im, Nathaniel Kell, Janardhan Kulkarni, and Debmalya Panigrahi. 2015. Tight bounds for online vector scheduling. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*. IEEE, 525–544.

[31] Jiuchuan Jiang, Bo An, Yichuan Jiang, Donghui Lin, Zhan Bu, Jie Cao, and Zhifeng Hao. 2018. Understanding crowdsourcing systems from a multiagent perspective and approach. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 13, 2 (2018), 1–32.

[32] C Ronald Kube and Hong Zhang. 1993. Collective robotics: From social insects to robots. *Adaptive behavior* 2, 2 (1993), 189–218.

[33] Harold W Kuhn. 2005. The Hungarian method for the assignment problem. *Naval Research Logistics (NRL)* 52, 1 (2005), 7–21.

[34] Jérôme Lang and Piotr Skowron. 2018. Multi-attribute proportional representation. *Artificial Intelligence* 263 (2018), 74–106.

[35] John Langford and Tong Zhang. 2007. The epoch-greedy algorithm for contextual multi-armed bandits. *Advances in neural information processing systems* 20, 1 (2007), 96–1.

[36] Lingzhi Luo, Nilanjan Chakraborty, and Katia Sycara. 2011. Multi-robot assignment algorithm for tasks with set precedence constraints. In *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2526–2533.

[37] Vahideh H Manshadi, Shayan Oveis Gharan, and Amin Saberi. 2012. Online stochastic matching: Online actions based on offline statistics. *Mathematics of Operations Research* 37, 4 (2012), 559–573.

[38] Panagiotis Mavridis, David Gross-Amblard, and Zoltán Miklós. 2016. Using hierarchical skills for optimized task assignment in knowledge-intensive crowdsourcing. In *Proceedings of the 25th International Conference on World Wide Web*. 843–853.

[39] Sabrina Klos née Müller, Cem Tekin, Mihaela van der Schaar, and Anja Klein. 2018. Context-aware hierarchical online learning for performance maximization in mobile crowdsourcing. *IEEE/ACM Transactions on Networking* 26, 3 (2018), 1334–1347.

[40] Lynne E Parker. 1998. ALLIANCE: An architecture for fault tolerant multirobot cooperation. *IEEE transactions on robotics and automation* 14, 2 (1998), 220–240.

[41] Chenxi Qiu, Anna C Squicciarini, Barbara Carminati, James Caverlee, and Dev Rishi Khare. 2016. CrowdSelect: increasing accuracy of crowdsourcing tasks through behavior prediction and user selection. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 539–548.

[42] Aviv Rosenberg and Yishay Mansour. 2019. Online convex optimization in adversarial markov decision processes. In *International Conference on Machine Learning*. PMLR, 5478–5486.

[43] Onn Shehory and Sarit Kraus. 1998. Methods for task allocation via agent coalition formation. *Artificial intelligence* 101, 1-2 (1998), 165–200.

[44] Rahul Singh, Abhishek Gupta, and Ness B Shroff. 2020. Learning in Markov decision processes under constraints. *arXiv preprint arXiv:2002.12435* (2020).

[45] Adish Singla and Andreas Krause. 2013. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In *Proceedings of the 22nd international conference on World Wide Web*. 1167–1178.

[46] Aleksandrs Slivkins and Jennifer Wortman Vaughan. 2014. Online decision making in crowdsourcing markets: Theoretical challenges. *ACM SIGecom Exchanges* 12, 2 (2014), 4–23.

[47] Ashwin Subramanian, G Sai Kanth, Sharayu Moharir, and Rahul Vaze. 2015. Online incentive mechanism design for smartphone crowd-sourcing. In *2015 13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*. IEEE, 403–410.

[48] Hanna Sumita, Shinji Ito, Kei Takemura, Daisuke Hatano, Takuro Fukunaga, Naonori Kakimura, and Ken-ichi Kawarabayashi. 2022. Online Task Assignment Problems with Reusable Resources. 36, 1 (2022), 5199–5207.

[49] Feilong Tang. 2020. Optimal Complex Task Assignment in Service Crowdsourcing. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*. International Joint Conferences on Artificial Intelligence Organization, 1563–1569.

[50] Hien To, Gabriel Ghinita, Liyue Fan, and Cyrus Shahabi. 2016. Differentially private location protection for worker datasets in spatial crowdsourcing. *IEEE Transactions on Mobile Computing* 16, 4 (2016), 934–949.

[51] Elio Tuci, Muhanad HM Alkilabi, and Otar Akanyeti. 2018. Cooperative object transport in multi-robot systems: A review of the state-of-the-art. *Frontiers in Robotics and AI* 5 (2018), 59.

[52] Lovekesh Vig and Julie A Adams. 2006. Market-based multi-robot coalition formation. In *Distributed Autonomous Robotic Systems 7*. Springer, 227–236.

[53] Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdu, and Marcelo J Weinberger. 2003. Inequalities for the L1 deviation of the empirical distribution.

Hewlett-Packard Labs, Tech. Rep (2003).

[54] Andrew K Whitten, Han-Lim Choi, Luke B Johnson, and Jonathan P How. 2011. Decentralized task allocation with coupled constraints in complex missions. In *Proceedings of the 2011 American Control Conference*. IEEE, 1642–1649.

[55] Dong Zhao, Xiang-Yang Li, and Huadong Ma. 2014. How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 1213–1221.

[56] Qingwen Zhao, Yanmin Zhu, Hongzi Zhu, Jian Cao, Guangtao Xue, and Bo Li. 2014. Fair energy-efficient sensing task allocation in participatory sensing with smartphones. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 1366–1374.

[57] Liyuan Zheng and Lillian Ratliff. 2020. Constrained upper confidence reinforcement learning. In *Learning for Dynamics and Control*. PMLR, 620–629.

[58] Yudian Zheng, Jiannan Wang, Guoliang Li, Reynold Cheng, and Jianhua Feng. 2015. QASCA: A quality-aware task assignment system for crowdsourcing applications. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*. 1031–1046.