Decentralized Online Learning by Selfish Agents in Coalition Formation

Saar Cohen and **Noa Agmon**
Department of Computer Science, Bar-Ilan University, Israel
saar30@gmail.com, agmon@cs.biu.ac.il

Abstract

Coalition formation involves self-organized coalitions generated through strategic interactions of autonomous selfish agents. In *online learning* of coalition structures, agents' preferences toward each other are initially unknown *before* agents interact. Coalitions are formed iteratively based on preferences that agents learn *online* from repeated feedback resulting from their interactions. In this paper, we introduce online learning in coalition formation through the lens of *distributed decision-making*, where self-interested agents operate without global coordination or information sharing, and learn only from their own experience. Under our selfish perspective, each agent seeks to maximize her own utility. Thus, we analyze the system in terms of *Nash stability*, where no agent can improve her utility by unilaterally deviating. We devise a sample-efficient decentralized algorithm for selfish agents that minimize their Nash regret, yielding approximately Nash stable solutions. In our algorithm, each agent uses only *one* utility feedback per round to update her strategy, but our algorithm still has Nash regret and sample complexity bounds that are ***optimal*** up to logarithmic factors.

1 Introduction

Freelance developers collaborating on open-source projects pursue individual goals like skill enhancement and portfolio building. Their self-interest affects their collaborative choices of forming project teams, but they cannot make optimal decisions as they initially lack clarity on their preferences about other developers, team size, or project types. After contributing to a certain team, these selfish developers learn about their preferences from feedback only about their team's productivity and satisfaction with the collaborative dynamics. This adaptive process allows them to selfishly optimize their working experiences, making informed decisions to join project teams that best align with their personal goals and working styles. Such scenarios and many other real-life cases exemplify *online learning in coalition formation*, where coalitions are formed iteratively based on preferences that agents learn *online* from repeated feedback about their interactions.

Hedonic games [Dreze and Greenberg, 1980] are a popular framework for studying coalition formation, where the utilities of *selfish* agents only depend on the coalition they are part of, disregarding the structure of other coalitions, i.e., *externalities* are ignored. The outcome of such games is a set of disjoint coalitions (hereafter, *partition*), whose desirability is often assessed in terms of *stability* [Aziz and Savani, 2016; Bullinger and Romen, 2024], reflecting the likelihood of selfish agents maintaining their coalitions. Traditional literature on stability in hedonic games often focuses only on the final outcome of coalition formation, ignoring the process of reaching stable partitions. Particularly, in most existing works it is implicitly assumed that a *central* authority can attain the agents' preferences, find a stable partition and impose it on the agents. Recent works, however, consider a *dynamic* process where, starting from an initial partition, agents deliberately move between coalitions based on their preferences [Boehmer *et al.*, 2023; Brandt *et al.*, 2022]. In many realistic cases as our project teams example, agents' preferences toward others are initially *unknown* prior to interactions (see, e.g., [Cohen and Agmon, 2023a]). Hence, each agent must make decisions individually based on her own preferences, which she learns *online* from repeated feedback by iteratively joining coalitions. To reflect such cases, Cohen and Agmon [2024b] study online learning in hedonic games, aiming to maximize social welfare. Recently, Cohen and Agmon [2025b] explored a *centralized* setting where *selfish* agents seek to maximize their own utility.

In contrast, in this paper we introduce and study a *new* model for online learning in coalition formation, reflecting such realistic situations from the perspective of *decentralized decision-making* by self-interested agents. As agents are usually not aware of other agents' strategies, we assume *bandit feedback*, i.e., agents solely observe their utility from the coalition they joined. We exhibit our findings for *additively separable hedonic games* with *symmetric* preferences [Bogomolnaia and Jackson, 2002], where an agent's utility for a coalition is the sum of her utilities from other coalition members. Essentially, our framework has no global coordination among agents, and thus they learn only from their own history of strategies, coalitions' composition and utility feedbacks.

Agents' selfish behavior can lead to stable partitions, which we evaluate by means of *Nash stability*, where no agent can improve her utility by unilaterally deviating. This stability

notion is well-suited to our distributed context as it does not allow any agent to coordinate with others to determine if she can increase her utility (see, e.g., [Balliu *et al.*, 2019]). In this setting, the partition formed by uncoordinated selfish agents can be inferior to a centrally designed one. Under our online learning setup, a popular metric for such inefficiency of equilibria is *Nash regret* [Ding *et al.*, 2022; Liu *et al.*, 2021], comparing an agent's learnt strategy against her best response strategy at any round. Sublinear Nash regret translates to low sample complexity as it implies best-iterate convergence to an approximate Nash stable solution with fewer samples (by classic online-to-batch conversion due to, e.g., [Jin *et al.*, 2018]), strongly tying these notions to our online learning setting.

Our Contributions. We develop a sample-efficient decentralized algorithms for online learning in coalition formation by selfish agents that minimizes its Nash regret, thus obtaining approximately Nash stable partitions. Such a distributed approach is also often favored over centralized ones due to its simplicity of implementation, versatility, and faster execution resulting from reduced communication overhead. We devise a Frank-Wolfe-based algorithm for *distributed* online learning in coalition formation, where each agent only utilizes *one* sample per round to update her strategy. We prove that our algorithm obtains Nash regret and sample complexity bounds that are ***optimal*** (up to logarithmic factors). **All omitted proofs can be found in the supplementary materials [Cohen and Agmon, 2025a].**

2 Related Work

Hedonic games were introduced by Drèze and Greenberg [1980], and later expanded to the study of various notions of stability, fairness, and optimality (see, e.g., [Aziz and Savani, 2016]). We focus on *additively separable hedonic games* (ASHGs) with *symmetric* preferences [Bogomolnaia and Jackson, 2002], where a large body of work evaluates the system in terms of *stability*. Unlike *cooperative* approaches (e.g., core stability [Bogomolnaia and Jackson, 2002]), we explore a *non-cooperative* perspective, where many works study Nash stability [Balliu *et al.*, 2019; Aloisio *et al.*, 2020; Banerjee *et al.*, 2001; Ballester, 2004]. Particularly, Bogomolnaia and Jackson [2002] proved that Nash stable partitions may not exist in general ASHGs, while Sung and Dimitrov [2010] showed that checking if an instance admits such partition is NP-complete in the strong sense. Yet, for *symmetric* preferences, the existence of a Nash stable outcome is guaranteed by potential function argument [Bogomolnaia and Jackson, 2002], but computing such partitions is PLS-complete [Gairing and Savani, 2019]. However, the above works explore *offline* settings, while we regard *online* ones.

Hence, our work is closely related to *time-dependent* models in hedonic games, including their *online* variant introduced by Flammini *et al.* [2021b], where agents arrive one at a time and should be *immediately* and *irrevocably* assigned to coalitions with the goal of maximizing social welfare. This problem was recently extended to other setups [Cohen and Agmon, 2024a; Bullinger and Romen, 2023], with Bullinger and Romen [2024] also exploring various *stability* concepts. However, the assumption that the agents are partitioned by a *central* authority may be unrealistic, as agents typically make decisions individually. In contrast, we adopt a *decentralized* perspective, where agents make selfishly decide to either form a new coalition or to join an existing one based only on their *local* information from past rounds. Recently, dynamic and distributed approaches to hedonic games have also received increased attention, focusing on deviation dynamics [Bilò *et al.*, 2018; Boehmer *et al.*, 2023; Brandt *et al.*, 2022].

However, existing works on online and dynamic hedonic games unrealistically require that the agents' preferences are fully known. Research on PAC learnability in hedonic games attempts to tackle this issue [Sliwinski and Zick, 2017; Fioravanti *et al.*, 2023]. Unlike our work, they take a *cooperative* approach, aiming to efficiently infer preferences from a limited, fixed number of offline samples. They also assume exact knowledge of preferences *before* making decisions, limiting practicality as agents often need time to learn their own preferences from social interactions, as in our project teams example. In particular, the PAC learning approach is unfit to our *dynamic* setting due to its *static* nature, requiring a fixed set of preferences that is known in advance. We consider situations where each agent *dynamically* learns her own preferences through repeated interactions, enabling her to adapt to changing scenarios so as to learn the coalitions proven most relevant and effective for her selfish desires.

We propose a novel framework that addresses these challenges by examining online learning in coalition formation through the lens of *decentralized decision-making* by selfish agents, unlike the *centralized* approaches presented by Cohen and Agmon [2024b; 2025b]. We also contribute to the growing focus on online learning in combinatorial domains (e.g., online task allocation [Cohen and Agmon, 2023b]). Traditional literature on learning in games often studies how various dynamics asymptotically converge to a Nash equilibrium (e.g., no-regret dynamics [Daskalakis *et al.*, 2021; Chen and Peng, 2020], fictitious play [Daskalakis and Pan, 2014; Leslie and Collins, 2006]). In contrast, we focus on non-asymptotic convergence, as done in recent studies on multi-agent reinforcement learning where *Nash regret* serves as a key performance metric [Ding *et al.*, 2022; Liu *et al.*, 2021], comparing an agent's learnt strategy with her best response strategy at each round. This metric is thus well-suited to the inherently non-stationary nature of our setting. By classic online-to-batch conversion (e.g., [Jin *et al.*, 2018, Section 3.1]), sublinear Nash regret yields low sample complexity, yielding best-iterate convergence to approximate Nash stable solutions with fewer samples.

Matching markets also relate to our work. Unlike prior works [Maheshwari *et al.*, 2022; Zhang *et al.*, 2022; Liu *et al.*, 2020], we address more practical and complex settings with broader utility functions under bandit feedback that go beyond matchings. Our work also ties to potential games, where prior distributed methods often rely on computationally expensive projection-based approaches [Ding *et al.*, 2022; Leonardos *et al.*, 2022], using costly or even intractable projection operations. Conversely, our projection-free Frank-Wolfe method resolves both issues by using a

more efficient linear optimization step. Unlike prior methods for potential games, whose sample complexity and Nash regret scale linearly in the number of actions an agent may take, our algorithm removes this dependency (See Appendix H for details). For congestion games, a subclass of potential games, Cui *et al.* [2022] also provide a projection-free Frank-Wolfe method, yet it suffers from suboptimal sample complexity and Nash regret bounds, relying on large mini-batches of samples and thus not updating optimization parameters frequently enough due to Defazio *et al.* [2019]. Yet, existing one-sample Frank-Wolfe schemes either exhibit suboptimal convergence rates [Mokhtari *et al.*, 2018] or make restrictive assumptions such as knowing each agent's true utility function [Zhang *et al.*, 2020]. In contrast, without knowing actual utilities, our one-sample algorithm uses a single sample per round while having ***optimal*** sample complexity and Nash regret bounds (up to logarithmic factors). Our algorithm also contrasts with that of Dadi *et al.* [2024], which has *suboptimal* sample complexity and generally *exponential* running time, becoming polynomial only for restricted cases, whereas our algorithm runs in *polynomial* time. Finally, unlike prior methods that update policies *independently* [Ding *et al.*, 2022; Leonardos *et al.*, 2022], our algorithms exploit awareness of other coalition members for better guarantees.

3 Preliminaries

We study an *online learning* version of hedonic games, where selfish agents with initially *unknown* preferences partition themselves into disjoint subsets (i.e, *coalitions*) over a known number of rounds T, which holds in many real-life scenarios as our project teams example, where developers may join project teams over a predefined number of project milestones. Formally, our strategic game is given by a finite set $N = \{1, \dots, n\}$ of n selfish agents with *unknown* preferences. Hereafter, we denote $[k] := \{1, \dots, k\}$ for $k \in \mathbb{N}$ and $[0] = \{0\}$. At any time $t \in [T]$, each agent can join one of n *candidate* coalitions since there are n agents (i.e., a partition can contain between 1 to n coalitions). In our project teams example, this can be thought of as if each developer picks which room to enter among n rooms. Thus, each agent i joins a certain coalition among n candidate ones at time t following a ***mixed strategy*** φ_i^t, built based only on *local* information from past rounds. Time $t = 1$ is exceptional, where each agent i arbitrarily initializes her strategy φ_i^1. Formally, $\varphi_i^t \in \mathcal{S}_n$ where $\mathcal{S}_n$ is the probability simplex over $[n]$, i.e., for any $x \in [n]$, agent i picks the xth candidate coalition with probability $\varphi_i^t(x) \in [0, 1]$. Let $\boldsymbol{\varphi}^t = (\varphi_i^t)_{i \in N} \in \mathcal{S}_n^n$ be the agents' ***joint mixed strategy*** at time t. At any time t, each agent i then samples an ***assignment*** $x_i \in [n]$ from φ_i^t independently from other agents, forming a ***joint assignment*** $\mathbf{x} = (x_i)_{i \in N}$. Thus, the agents' iterative process of distributed decision-making at time t unfolds as follows:

1. Each agent i samples $x_i \sim \varphi_i^t$ and joins the x_ith candidate coalition, forming a joint assignment $\mathbf{x} = (x_i)_{i \in N}$.

2. Each agent i observes the other members in the coalition she joined and gets bandit feedback about the utility gained from her own coalition.

3. Based *only* on this obtained information, agent i updates her strategy to be φ_i^{t+1} and moves to the next round $t+1$.

We term the game formed by this learning process as ***decentralized online learning ASHGs*** **(DOL-ASHGs)**. Next, we elaborate on this process in detail. The constructed joint assignment $\mathbf{x}$ induces a partition of the agents $\pi^{\mathbf{x}} = (C_\ell^{\mathbf{x}})_{\ell \in [n]}$, where, for any $\ell \in [n]$, $C_\ell^{\mathbf{x}}$ is the set of agents joining the ℓth candidate coalition, i.e., $C_\ell^{\mathbf{x}} = \{i \in N : x_i = \ell\}$. As the number of candidate coalitions equals to the number of agents, some coalitions may be empty. Thereby, we denote by $|\pi^{\mathbf{x}}|$ the number of non-empty coalitions in $\pi^{\mathbf{x}}$. After her assignment, notice that agent i becomes aware of the other members within the coalition she joined. We thus denote the coalition in $\pi^{\mathbf{x}}$ containing agent i as $\pi^{\mathbf{x}}(i)$.

Afterwards, we can derive each agent's utility from her chosen strategy, determined by aggregating her utilities of other agents. We focus on *additively separable hedonic games* (**ASHGs**) with *symmetric* preferences, where any pair of agents assign the same numerical value toward each other, indicating the intensity by which they prefer each other to another agent. As common in the literature (see, e.g., [Flammini *et al.*, 2021a]), we assume that agents' valuations are within $[-1, 1]$. Recall that preferences are *unknown* and even the agents themselves may not be aware of them. Thus, for any pair of distinct agents i, j, the uncertainty about their mutual valuation is captured by an *unknown* and *fixed* distribution $\mathcal{D}_{i,j}$ over $[-1, 1]$ with mean $d_{i,j}$, which agents i, j aim to learn. At any time t, the ***utility*** $v_{i,j}^t$ of agents i, j for each other is then independently drawn from $\mathcal{D}_{i,j}$. We use the convention that $v_{i,i}^t = d_{i,i} = 0$ for any agent i. For any joint assignment $\mathbf{x}$ sampled at time t, agent i's utility from the induced partition $\pi^{\mathbf{x}}$ is then $v_i^t(\mathbf{x}) = \sum_{i \neq j \in \pi^{\mathbf{x}}(i)} v_{i,j}^t$, whose mean is $d_i(\mathbf{x}) = \sum_{i \neq j \in \pi^{\mathbf{x}}(i)} d_{i,j}$. Agent i's utility from her strategy φ_i^t at time t is thus defined as $V_i(\boldsymbol{\varphi}^t) := \mathbb{E}_{\mathbf{x} \sim \boldsymbol{\varphi}^t}[d_i(\mathbf{x})]$.

Each agent's strategies are evaluated by her *true* unknown utility, but she cannot observe her *true* utility. We thus consider the most general ***decentralized*** setting where each agent should learn her own preferences from repeated *partial* feedbacks. At any time t and for any joint assignment $\mathbf{x}$, we assume that each agent i can only receive ***bandit feedback***, i.e., in practice, agent i only obtains her utility from the entire partition induced by $\mathbf{x}$ (i.e., $v_i^t(\mathbf{x})$), with no information about the utility $v_{i,j}^t$ gained from interacting with any other agent j.

Each agent joins a coalition with the goal of maximizing her own utility. We thus want to study stability under single agents' incentives to deviate between coalitions. The traditional literature on hedonic games focuses on ***pure*** strategies (e.g., [Aziz *et al.*, 2013; Bilò *et al.*, 2018]), where each agent i's strategy is joining a *single* candidate coalition at time t by only selecting some $x_i \in [n]$. Let $\mathbf{x}_{-i} = (x_j)_{j \neq i}$ be the joint strategy of all agents except for agent i. Agent i can then ***deviate*** by moving from her selected coalition to another one with index $y_i \in [n]$, which is a ***Nash deviation*** if it improves her utility, i.e., $V_i^t(\mathbf{x}_{-i}, y_i) > V_i^t(\mathbf{x}_{-i}, x_i)$. Unlike prior work on hedonic games, we also study *mixed* strategies, where we define the other notion of *mixed Nash deviations*. Consider the joint strategy $\boldsymbol{\varphi}^t$ at time t. Letting $\boldsymbol{\varphi}_{-i}^t = (\varphi_j^t)_{j \neq i}$ for any

agent i, agent i may perform a ***(mixed single-agent) deviation*** from her strategy φ_i^t to another strategy $\phi_i \in \mathcal{S}_n$, which is a ***mixed Nash deviation*** only if it immediately makes her better off, i.e., $V_i(\boldsymbol{\varphi}_{-i}^t, \phi_i) > V_i(\boldsymbol{\varphi}^t)$. Hence, a (*pure* or *mixed*) joint strategy for which no Nash deviation is possible is said to be ***Nash stable*** (NS), also called a *Nash equilibrium* (NE). As mentioned earlier, for *symmetric* preferences under *pure* strategies, the existence of a Nash stable outcome is guaranteed by potential function argument [Bogomolnaia and Jackson, 2002], but computing such strategies is PLS-complete [Gairing and Savani, 2019].

We thus further consider an *approximate* notion of Nash stability. At each time t, note that agent i's *best response* to the other agents' strategies is a Nash deviation given by a strategy $\phi_i^{\star,t}$ satisfying $V_i^\star(\boldsymbol{\varphi}_{-i}^t) := V_i(\boldsymbol{\varphi}_{-i}^t, \phi_i^{\star,t}) = \max_{\phi \in \mathcal{S}_n} V_i(\boldsymbol{\varphi}_{-i}^t, \phi)$. Thus, for any $\varepsilon \geq 0$, the agents' joint strategy $\boldsymbol{\varphi}^t$ at time t is ***ε-approximate Nash stable*** (ε-NS) if no agent can improve her gain by more than ε, i.e., $\max_{i \in N}(V_i^\star(\boldsymbol{\varphi}_{-i}^t) - V_i(\boldsymbol{\varphi}^t)) \leq \varepsilon$. Here, the quantity $\max_{i \in N}(V_i^\star(\boldsymbol{\varphi}_{-i}^t) - V_i(\boldsymbol{\varphi}^t))$ measures the ***worst agent's local gap*** between the expected utilities she receives from her best response and her current strategy at time t.

Remark 1. *Consider the following mixed strategy where each agent treats all coalitions as equally desirable, i.e., the mixed strategy φ_i^t of any agent i and each time t satisfies $\varphi_i^t(x) = \frac{1}{n}$ for any $x \in [n]$. Clearly, this is an exact mixed NS strategy. However, it ignores the agents' preferences entirely, which is unrealistic in practical scenarios as agents often act strategically based on their own preferences, not arbitrarily. Instead, our framework aims to learn meaningful mixed NS strategies that align with agents' preferences, reflecting real-life behavior by accounting for agents' incentives.*

Our goal is thus devising a ***decentralized*** algorithm that learns an ε-NS joint strategy for some $\varepsilon \geq 0$, which aligns with agents' preferences. Putting everything together, at any time t, each agent i updates her strategy from time $t-1$ based only on her own *local* history up to time $t-1$, consisting only of her strategies, the composition of the coalitions she joined and the bandit feedbacks from her own utilities until time $t-1$, without any information about other coalitions and utilities. We analyze the performance of such algorithm via ***Nash regret*** [Ding *et al.*, 2022; Liu *et al.*, 2021], comparing each agent's learnt strategy with her best response strategy at any time instant. Formally, given a sequence of joint strategies $\{\varphi^t\}_{t=1}^T$, the Nash regret after T rounds is:

$$\mathcal{R}^T := \sum_{t=1}^T \max_{i \in N}(V_i^\star(\boldsymbol{\varphi}_{-i}^t) - V_i(\boldsymbol{\varphi}^t)) \qquad (1)$$

The intuition behind the Nash regret lies in our definition of approximate NS strategies. As $\max_{i \in N}(V_i^\star(\boldsymbol{\varphi}_{-i}^t) - V_i(\boldsymbol{\varphi}^t))$ is the worst agent's local gap at time t, the Nash regret in (1) is the total sum of the worst agent's local gap at each individual round. That is, at each time t, the Nash regret compares learned strategy φ_i^t of each agent i with the best strategy agent i can take by fixing the other agents' strategies φ_{-i}^t. By our definition of an approximate NS strategy, the Nash regret thus evaluates how far the agents' strategies in each round are from being an (approximate) NS strategy.

For any $\varepsilon \geq 0$, it is well-known that the the above connection between ε-NS strategies and Nash regret can be used to show that an algorithm with a Nash regret bound of ε obtains an ε-NS strategy (see, e.g., [Ding *et al.*, 2022]). Hence, one of our objectives is minimizing the Nash regret, i.e., attaining a ***Nash regret bound*** that is sublinear in the number of rounds T and polynomial in the number of agents n. Another goal is finding an ε-NS strategy using a number of rounds T that is *small* in its dependency on the number of agents n and $1/\varepsilon$, guaranteeing a ***(PAC) sample complexity bound***. An algorithm with low sample complexity requires a fewer rounds for best-iterate convergence to an ε-NS strategy. In fact, any algorithm with sublinear Nash regret can be directly converted to a polynomial-sample algorithm via standard online-to-batch conversion (see, e.g., [Jin *et al.*, 2018]).

3.1 DOL-ASHGs as Series of Potential Games

We begin with a useful structural property satisfied by our model. For a single round (i.e., $T = 1$), a Nash stable strategy is guaranteed for ASHGs with *known* symmetric preferences through a potential function argument [Bogomolnaia and Jackson, 2002]. In general DOL-ASHGs, where the symmetric preferences are *unknown* at any time t, we show that the game associated with each time t is also a potential game:

Lemma 1. *At each time t, the ASHG with symmetric and unknown preferences associated with time t is a* potential game.

Proof. Consider a joint mixed strategy $\boldsymbol{\varphi}^t$ at time t. Given a joint assignment $\mathbf{x} \sim \boldsymbol{\varphi}^t$, note that $\Phi^t(\mathbf{x}) = \sum_{i \in N} v_i^t(\mathbf{x})$ is a potential function for *pure* strategies as $\Phi^t(\mathbf{x}_{-i}, x_i) - \Phi^t(\mathbf{x}_{-i}, x_i') = v_i^t(\mathbf{x}_{-i}, x_i) - v_i^t(\mathbf{x}_{-i}, x_i')$ for any agent i and another assignment $x_i' \in [n]$ of agent i. Therefore, by slight abuse of notation, $\Phi(\boldsymbol{\varphi}^t) = \sum_{i \in N} V_i(\boldsymbol{\varphi}^t)$ is a potential function for *mixed* strategies as $\Phi(\boldsymbol{\varphi}_{-i}^t, \varphi_i^t) - \Phi(\boldsymbol{\varphi}_{-i}^t, \phi_i) = V_i(\boldsymbol{\varphi}_{-i}^t, \varphi_i^t) - V_i(\boldsymbol{\varphi}_{-i}^t, \phi_i)$ for any agent i and another strategy $\phi_i \in \mathcal{S}_n$ of agent i. Note that, for pure strategies, the potential is denoted as Φ^t as it directly depends on t through the utilities. Yet, for mixed strategies, it is expressed by $\Phi(\varphi^t)$ as it does not directly depend on t, while its input φ^t does. □

Thus, online learning algorithms for our context should be based on the principle that any ***Nash stable*** strategy is a stationary point of the potential function from Lemma 1. Other key design factors are computational efficiency and scalability. Note that one main challenge in our setting is that the size of the joint pure strategies' space equals to the number of possible partitions over n agents which grows *exponentially* with the number of agents n [Sandholm *et al.*, 1999]. Thus, an efficient algorithm should exhibit Nash regret and sample complexity polynomial in the number of agents n, without dependence on the size of the joint pure strategies space.

As mentioned in Section 2, existing methods for online learning in potential games are unfit for our context [Cui *et al.*, 2022; Ding *et al.*, 2022; Leonardos *et al.*, 2022]. Hence, we devise a Frank-Wolfe method [Hazan and Kale, 2012], replacing projections with a less expensive linear optimization step, and thus can be implemented efficiently in high-dimensional domains such as ours. The classic Frank-Wolfe

algorithm addresses constrained convex optimization problems by iteratively approximating the objective function with its first-order Taylor expansion around the current solution and solving the resulting linear problem over the feasible set. In each iteration, it identifies a feasible direction by optimizing the linear approximation, and updates the current solution via a convex combination with a properly chosen step size.

Tailored to decentralized online learning in ASHGs, our algorithm also exploits each agent's knowledge about the identities of other members in the coalitions she joins. Namely, each agent makes decisions based only on her own history of strategies, the members of the coalitions she joined and the resulting utility feedbacks, without communicating with other agents to coordinate their strategies. Instead of using large mini-batches as done by Cui *et al.* [2022], in our algorithm each agent only uses *one* sample per round (Section 4), yet it has Nash regret and sample complexity bounds that are ***optimal*** up to logarithmic factors. At each round, agents form coalitions, obtain a *single* utility feedback, update their strategies and move to the next round (i.e., agents consecutively follow the stages (1), (2), (3) from the iterative process specified in Section 3).

4 Distributed One-Sample Algorithm

In this section, we devise a sample-efficient decentralized online learning algorithm for DOL-ASHGs (Algorithm 1), termed as *Decentralized One-Sample Frank-Wolfe* (D1S-FW), that requires only *one* sample per round to update the optimization variables and obtains sublinear Nash regret that is also polynomial in the number of agents n. As input, D1S-FW receives the number of rounds T, a set of n agents and step sizes $\rho_t, \eta_t, \mu_t \in (0,1)$ for each time t whose choices affect the resulting Nash regret and sample complexity bounds, as detailed later. Next, we depict D1S-FW's main components. Note that we do not have access to the *exact* gradient of the potential function from Lemma 1. Thus, at any time t, each agent i first computes an unbiased and variance reduced gradient estimation of the potential function at time t using only *one* utility feedback (Section 4.1). Using this estimation, agent i then follows the classic Frank-Wolfe update rule to pick her next strategy (Section 4.2). Finally, we prove that D1S-FW has Nash regret and sample complexity bounds that are ***optimal*** up to logarithmic factors (Section 4.3).

4.1 The Gradient Estimation Step

Our algorithm first modifies the momentum variance reduction approach in [Mokhtari *et al.*, 2018; Mokhtari *et al.*, 2020; Zhang *et al.*, 2020] to compute an unbiased estimator of the potential's gradient using a *single* utility feedback. Formally, at each time t, note that the potential's gradient w.r.t. any joint strategy φ is given by $\nabla\Phi = (\frac{\partial\Phi}{\partial\varphi_i}(x_i))_{i\in N, x_i\in[n]}$, or equivalently $\nabla\Phi = (\nabla_{\varphi_i}\Phi)_{i\in N}$ where $\nabla_i\Phi := \nabla_{\varphi_i}\Phi = (\frac{\partial\Phi}{\partial\varphi_i}(x_i))_{x_i\in[n]}$ is the potential's gradient w.r.t. agent i's strategy φ_i. A naive approach for agent i is thus using an unbiased *one-sample* estimation $\widehat{\nabla}_i^t\Phi$ at time t of $\nabla_i\Phi(\boldsymbol{\varphi}^t)$. Yet, this method yields a high variance. To reduce the variance, in our algorithm agent i sets her gradient estimation $\mathbf{g}_i^t$ at time t to be a weighted average involving the previous gradient estimate $\mathbf{g}_i^{t-1}$ and the one-sample estimation $\widehat{\nabla}_i^t\Phi$:

$$\mathbf{g}_i^t = (1-\mu_t)(\mathbf{g}_i^{t-1} + \widehat{\boldsymbol{\Delta}}_i^t) + \mu_t\widehat{\nabla}_i^t\Phi \tag{2}$$

where $\mu_t \in (0,1)$ is an averaging parameter and $\widehat{\boldsymbol{\Delta}}_i^t$ is an unbiased estimator of the gradient variation $\boldsymbol{\Delta}_i^t = \nabla_i\Phi(\boldsymbol{\varphi}^t) - \nabla_i\Phi(\boldsymbol{\varphi}^{t-1})$. Intuitively, adding the term $\widehat{\boldsymbol{\Delta}}_i^t$ to $\mathbf{g}_i^{t-1}$ in (2) ensures that $\mathbf{g}_i^t$ is an unbiased estimator of $\nabla_i\Phi$ due to an inductive argument. Indeed, if $\mathbf{g}_i^{t-1}$ is an unbiased estimate of $\nabla_i\Phi(\boldsymbol{\varphi}^{t-1})$ and $\widehat{\boldsymbol{\Delta}}_i^t$ is an unbiased estimate of $\boldsymbol{\Delta}_i^t$, then $\mathbb{E}[\mathbf{g}_i^t] = (1-\mu_t)(\nabla_i\Phi(\boldsymbol{\varphi}^{t-1}) + \nabla_i\Phi(\boldsymbol{\varphi}^t) - \nabla_i\Phi(\boldsymbol{\varphi}^{t-1})) + \mu_t\nabla_i\Phi = \nabla_i\Phi$, where the expectation is over all randomness up to time t. Next, we explain how to compute the one-sample gradient estimator $\widehat{\nabla}_i^t\Phi$ and the gradient variation estimator $\widehat{\boldsymbol{\Delta}}_i^t$, both in an unbiased manner. Particularly, we prove that this can be done by approximating each agent's mean utilities from others using linear regression.

Algorithm 1 D1S-FW

Input: T rounds; n agents; Step sizes $\rho_t, \eta_t, \mu_t \in (0,1)\ \forall t$.

Initialize an arbitrary initial policy φ_i^1 for each agent i.
for each time $t = 1, \dots, T$ **do**
  Each agent i joins a coalition by sampling $x_i^t \sim \varphi_i^t$.
  Each agent i receives a utility v_i^t from her coalition.
  for each agent $i \in N$ individually **do**
    Compute $\widehat{\nabla}_i^t\Phi$ using (5).
    if $t = 1$ **then**
      Set the gradient estimation as $\mathbf{g}_i^1 := \widehat{\nabla}_i^1\Phi$.
    else
      Compute $\widehat{\Delta}_i^t = \widehat{\nabla}_i^t\Phi - \widehat{\nabla}_i^{t-1}\Phi$.
      Compute the gradient estimator $\mathbf{g}_i^t$ using (2).
    Calculate $\hat{\phi}_i^t = \arg\max_{\phi\in\mathcal{S}_n}\langle\phi, \mathbf{g}_i^t\rangle$.
    Compute π_i^t using (6) (See Remark 4).
    Set $\varphi_i^{t+1} = (1-\rho_t)[\varphi_i^t + \eta_t(\hat{\phi}_i^t - \varphi_i^t)] + \rho_t\pi_i^t$.

One-Sample Gradient Estimator. To come up with such estimator, we first need to know the ***closed form of the true gradient***. In Lemma 2, we show that, for each agent i and any selection $x_i \in [n]$, the gradient $(\nabla_i\Phi)_{x_i} := \frac{\partial\Phi}{\partial\varphi_i}(x_i)$ evaluated at $\boldsymbol{\varphi}^t$ can be written as an inner product between the vector $\boldsymbol{\xi}_i = [d_{i,j}]_{j\in N}$ of all agent i's mean utilities from any agent $j \in N$ and the vector $\boldsymbol{\varphi}^t(x_i) \in [0,1]^n$ given by $[\boldsymbol{\varphi}^t(x_i)]_j = \varphi_j^t(x_i)$ for any joint strategy $\boldsymbol{\varphi}^t$ at time t. Intuitively, this will allow us to obtain our one-sample gradient estimator by proving that linear regression can be applied to approximate $\boldsymbol{\xi}_i$ at time t via a least squares estimator.

Lemma 2. *For each agent i, the **gradient evaluated at** $\boldsymbol{\varphi}^t$, which is given by $(\nabla_i\Phi(\boldsymbol{\varphi}^t))_{x_i}$, can be rephrased as follows:*

$$(\nabla_i\Phi(\boldsymbol{\varphi}^t))_{x_i} = \langle\boldsymbol{\varphi}^t(x_i), \boldsymbol{\xi}_i\rangle \tag{3}$$

where $\boldsymbol{\varphi}^t(x_i) \in [0,1]^n$ is defined as $[\boldsymbol{\varphi}^t(x_i)]_j = \varphi_j^t(x_i)$ for any joint strategy $\boldsymbol{\varphi}^t$ at time t, $x_i \in [n]$ and agent $j \in N$.

Proof. See Appendix A for a detailed proof. □

Using Lemma 2, at any time t, each agent i can derive a ***one-sample gradient estimation*** as follows. First, agent i samples her chosen candidate coalition $x_i^t \in [n]$ from her strategy φ_i^t (line 3). After forming the partition $\boldsymbol{\pi}^t := \pi^{\mathbf{x}^t}$, each agent i obtains a utility feedback v_i^t from her coalition $\boldsymbol{\pi}^t(i)$ (line 4), whose expectation given that the agents' joint assignment is $\mathbf{x}^t \in [n]^n$ at time t can be expressed as $\mathbb{E}[v_i^t|\mathbf{x}^t] = d_i(\mathbf{x}^t) = \sum_{i\neq j\in\pi^t(i)} d_{i,j} = \sum_{i\neq j\in N} \mathbb{1}\{x_j^t = x_i^t\} d_{i,j}$, where, for any $x_i, x_j \in [n]$, $\mathbb{1}\{x_j = x_i\}$ equals to 1 if $x_j = x_i$ and 0 otherwise. This can be rephrased as follows:

$$\mathbb{E}[v_i^t|\mathbf{x}^t] = \langle \boldsymbol{\psi}_i(\mathbf{x}^t), \boldsymbol{\xi}_i \rangle \tag{4}$$

where $\boldsymbol{\psi}_i(\mathbf{x}^t) \in \{0,1\}^n$ is given by $[\boldsymbol{\psi}_i(\mathbf{x}^t)]_j = \mathbb{1}\{x_j^t = x_i^t\}$ for any agent $j \in N$, capturing whether agent j joined agent i's selected coalition or not.

Remark 2. *In practice, each agent i can compute $\boldsymbol{\psi}_i(\mathbf{x}^t)$ by* only *using her knowledge about other agents in the coalition she joined via $[\boldsymbol{\psi}_i(\mathbf{x}^t)]_j = \mathbb{1}\{j \in \boldsymbol{\pi}^t(i)\}$ for any agent j.*

Therefore, by Lemma 2, ***we can use linear regression to estimate*** $\boldsymbol{\xi}_i$ at time t via the least squares estimator $\hat{\boldsymbol{\xi}}_i^t = (\hat{\Psi}_i^t)^{-1}\boldsymbol{\psi}_i(\mathbf{x}^t)v_i^t$, where $\hat{\Psi}_i^t = \mathbb{E}_{x_i\sim\varphi_i^t}[\|\boldsymbol{\psi}_i(\mathbf{x}_{-i}^t, x_i)\|_2^2]$ is the covariance matrix and $\|\cdot\|_2$ is the standard Euclidean norm (i.e., $\|\boldsymbol{\psi}_i(\mathbf{x}_{-i}^t, x_i)\|_2^2 = \boldsymbol{\psi}_i(\mathbf{x}_{-i}^t, x_i)\boldsymbol{\psi}_i(\mathbf{x}_{-i}^t, x_i)^\top$). We can obtain an estimator $\widehat{\nabla}_i^t\Phi(x_i) = \langle \boldsymbol{\psi}_i(\mathbf{x}_{-i}^t, x_i), \hat{\boldsymbol{\xi}}_i^t \rangle$ of $(\nabla_i\Phi(\boldsymbol{\varphi}^t))_{x_i}$, which satisfies the following:

$$\widehat{\nabla}_i^t\Phi(x_i) = \boldsymbol{\psi}_i(\mathbf{x}_{-i}^t, x_i)^\top(\hat{\Psi}_i^t)^{-1}\boldsymbol{\psi}_i(\mathbf{x}^t)v_i^t \tag{5}$$

One can easily verify that this is indeed an *unbiased* estimator of the potential's gradient, as we prove in Appendix B.

Remark 3. *From the perspective of agent i, she deduces $\boldsymbol{\psi}_i(\mathbf{x}^t)$* only *from the coalition $\boldsymbol{\pi}^t(i)$ she joined, but we include the implicit dependence on the agents' joint selection $\mathbf{x}^t$ for the sake of the analysis so as to, e.g., reason about the covariance matrix.*

One-Sample Gradient Variation Estimator. For any agent i, our one-sample approach for estimating the gradient variation $\boldsymbol{\Delta}_i^t = \nabla_i\Phi(\boldsymbol{\varphi}^t) - \nabla_i\Phi(\boldsymbol{\varphi}^{t-1})$ at time t is computing the difference between our one-sample gradient estimation for agent i's current selection and that for her previous one, i.e., $\widehat{\Delta}_i^t = \widehat{\nabla}_i^t\Phi - \widehat{\nabla}_i^{t-1}\Phi$. As we saw earlier (Recall Lemma B.1), $\widehat{\nabla}_i^t\Phi$ and $\widehat{\nabla}_i^{t-1}\Phi$ are unbiased estimators of $\nabla_i\Phi$ at times t and $t-1$ (resp.), and thus $\mathbb{E}[\widehat{\Delta}_i^t] = \boldsymbol{\Delta}_i^t$, where the expectation is over all randomness up to time t. That is, $\widehat{\Delta}_i^t$ is an *unbiased* estimator of the gradient variation $\boldsymbol{\Delta}_i^t$.

4.2 The Frank-Wolfe Update Step

At each time t, in line 12 each agent i then optimizes a linear optimization problem to get the feasible ascent strategy $\hat{\phi}_i^t$ as the regular Frank-Wolfe method. Instead of updating her strategy at window $t+1$ to be $\varphi_i^t + \eta_t(\hat{\phi}_i^t - \varphi_i^t)$ as in the standard Frank-Wolfe update, in line 14 agent i mixes this strategy with an exploratory strategy π_i^t which is computed in line 13. Specifically, agent i computes a *G-optimal design* π_i^t for $\{\boldsymbol{\psi}_i(\mathbf{x}_{-i}^t, x_i)\}_{x_i\in[n]}$ (see, e.g., [Lattimore and Szepesvári, 2020, Chapter 21]). The G-optimal design minimizes the maximal mean-squared prediction error in linear regression. In our context, the G-optimal design can be viewed as the strategy allowing a *uniformly well* estimation of $\boldsymbol{\xi}_i$ over all candidate coalitions. That is, sampling a candidate coalition $x_i \in [n]$ in proportion to $\pi_i^t(x_i)$ minimizes the number of samples required for reaching a desired level of accuracy in estimating $\mathbb{E}[v_i^t|\mathbf{x}_{-i}^t, x_i] = \langle \boldsymbol{\psi}_i(\mathbf{x}_{-i}^t, x_i), \boldsymbol{\xi}_i \rangle$. Formally, agent i's ***G-optimal design*** π_i^t at time t satisfies:

$$\pi_i^t \in \arg\min_{\phi\in\mathcal{S}_n} f_i^t(\phi) \tag{6}$$

where $f_i^t(\phi) = \max_{x_i\in[n]} \boldsymbol{\psi}_i(\mathbf{x}_{-i}^t, x_i)^\top(\Psi_{i,\phi}^t)^{-1}\boldsymbol{\psi}_i(\mathbf{x}_{-i}^t, x_i)$. Hence, agent i then mixes π_i^t with the strategy $\varphi_i^t + \eta_t(\hat{\phi}_i^t - \varphi_i^t)$ obtained from the standard Frank-Wolfe update.

Remark 4. *While the original G-optimal design problem is NP-hard [Soare* et al.*, 2014], for our purposes it suffices to compute an* approximate *G-optimal design with minimal impact on performance. For instance, using the Frank-Wolfe algorithm and a proper initialization, agent i can efficiently find an approximate G-optimal design $\hat{\pi}_i^t$ with $f_i^t(\hat{\pi}_i^t) \leq 2n$. For easing the analysis in the main text, we hereafter assume that a G-optimal design can be found efficiently and accurately, which is an assumption common in the literature (see, e.g., [Yang and Tan, 2022]). See Appendix G for a vast discussion on the minor effect of using an approximate G-optimal design in our algorithm, including a time complexity analysis.*

4.3 Nash Regret Analysis of D1S-FW

To obtain the Nash regret bound of D1S-FW (Algorithm 1), the key step is devising an upper bound on our estimators' errors. First, mixing the regular Frank-Wolfe update with the exploration strategy π_i^t allows us to regulate the one-sample gradient estimator's error as follows:

Lemma 3. *At each time t, for all $\delta \in (0,1]$, agent i and $x_i \in [n]$, the following holds with probability at least $1-\delta$:*

$$\begin{aligned}&|\widehat{\nabla}_i^t\Phi(x_i) - (\nabla_i\Phi(\boldsymbol{\varphi}^t))_{x_i}| \leq \\ &\mathcal{O}\Big(\sqrt{\tfrac{n^4\log(nT/\delta)}{\rho_t}} + \tfrac{n^3\log(nT/\delta)}{\rho_t}\Big)\end{aligned} \tag{7}$$

In particular, if $\rho_t \leq \frac{2\log(1/\delta)}{9}$, then the following holds with probability at least $1-\delta$:

$$|\widehat{\nabla}_i^t\Phi(x_i) - (\nabla_i\Phi(\boldsymbol{\varphi}^t))_{x_i}| \leq \mathcal{O}\Big(\tfrac{n^3\log(nT/\delta)}{\rho_t}\Big) \tag{8}$$

Proof. (*Sketch*) At any time t, let $\mathcal{R}_t = \boldsymbol{\psi}_i(\mathbf{x}_{-i}^t, x_i)^\top(\Psi_{i,\varphi_i^t}^t)^{-1}\boldsymbol{\psi}_i(\mathbf{x}^t)v_i^t$. In Appendix D, we first prove that $|\mathcal{R}_t| \leq \frac{n^2}{\rho_t}$ (Lemma D.1) and $\mathbb{E}^t[\mathcal{R}_t^2] \leq \frac{n^3}{\rho_t}$ (Lemma D.2), where $\mathbb{E}^t[\cdot]$ is the expectation over $x_i^t \sim \varphi^t$ for each agent i and all the utilities' randomness at time t. As we proved in Lemma B.1 that $\widehat{\nabla}_i^t\Phi(x_i)$ from (5) is an unbiased estimator of $(\nabla_i\Phi(\boldsymbol{\varphi}^t))_{x_i}$ (i.e., $\mathbb{E}^t[\widehat{\nabla}_i^t\Phi(x_i)] = \mathbb{E}[\mathcal{R}_t] = (\nabla_i\Phi(\boldsymbol{\varphi}^t))_{x_i}$), we can apply Bernstein's inequality to obtain both (7) and (8). □

We next supply the required upper bound on the error $\mathcal{A}_i^t := \|\nabla_i\Phi(\boldsymbol{\varphi}^t) - \mathbf{g}_i^t\|_\infty$ of agent i's gradient estimator in (2), where $\|\cdot\|_\infty$ is the L_∞ norm (i.e., $\|\mathbf{z}\|_\infty = \max_{i\in[n]}|z_i|$ for any $\mathbf{z} \in \mathbb{R}^n$).

Lemma 4. *For any time $t \geq 3$ and any $\delta \in (0, 1]$, the following holds with probability at least $1 - \delta$ for any agent i:*

$$\|\nabla_i \Phi(\boldsymbol{\varphi}^t) - \mathbf{g}_i^t\|_\infty \leq \mathcal{O}\left(\sum_{\tau_1=2}^{t-1} \mathcal{B}^{\tau_1} \prod_{\tau_2=\tau_1+1}^{t} (1-\mu_{\tau_2}) + \mathcal{B}^t\right) \tag{9}$$

where $\mathcal{B}^t := \sqrt{\frac{n^4 \log(nT/\delta)}{\rho_t}} + \frac{n^3 \log(nT/\delta)}{\rho_t}$. If $\rho_t \leq \frac{2\log(1/\delta)}{9}$ for any time t, then an improved bound in (9) *is obtained for $\mathcal{B}^t := \frac{n^3 \log(nT/\delta)}{\rho_t}$. For $t \in \{1, 2\}$, the errors $\mathcal{A}_i^1$ and $\mathcal{A}_i^2$ are bounded as in Lemma 3 since $\mathcal{A}_i^t = \|\nabla_i \Phi(\boldsymbol{\varphi}^t) - \widehat{\nabla}_i^t \Phi\|_\infty$ for $t \in \{1, 2\}$.*

Proof. (*Sketch*) As $g_i^1 = \widehat{\nabla}_i^1 \Phi$, then $\mathcal{A}_i^1$ is bounded as in Lemma 3. For any time $t \geq 2$, we first prove in Appendix E that our estimators satisfy $A_i^t := \|(1-\mu_t)(\nabla_i \Phi(\boldsymbol{\varphi}^{t-1}) - \mathbf{g}_i^{t-1}) + (1-\mu_t)(\widehat{\nabla}_i^{t-1}\Phi - \nabla_i \Phi(\boldsymbol{\varphi}^{t-1})) + (\nabla_i \Phi(\boldsymbol{\varphi}^t) - \widehat{\nabla}_i^t \Phi)\|_\infty$ for each agent i, from which our bound for time 2 can be easily obtained. For each time $t \geq 3$, this equality allows us to prove (9) by induction on t. □

We are now ready to establish our Nash regret bound:

Theorem 1. *For any $\delta \in (0, 1]$, D1S-FW (Algorithm 1) with $\rho_t = \frac{1}{T}$, $\eta_t = \frac{1}{\sqrt{T}}$, $\mu_t = 1 - \frac{1}{T}$ for any time t and $T \geq 2$ obtains the following Nash regret bound with probability at least $1 - \delta$:*

$$\mathcal{R}^T \leq \mathcal{O}\big((n + n^3 + n^4)\sqrt{T} + n^4 \log(nT/\delta)\big) \tag{10}$$

If $\delta \leq e^{-\frac{9}{2T}}$, then, with probability at least $1 - \delta$, the Nash regret bound in (10) *can be reduced to:*

$$\mathcal{R}^T \leq \mathcal{O}((n + n^3 + n^4)\sqrt{T} + n^4 \log(nT/\delta)) \tag{11}$$

Proof. (*Sketch*) The key idea behind our proof is using the fact that the Nash regret is at most the sum of the Frank-Wolfe gaps across all rounds (Appendix C). Namely, as the Frank-Wolfe gap of any joint strategy φ w.r.t. the potential function $\Phi(\boldsymbol{\varphi})$ is given by $G(\boldsymbol{\varphi}) = \max_{\boldsymbol{\varphi}' \in \mathcal{S}_n^n} \langle \boldsymbol{\varphi}' - \boldsymbol{\varphi}, \nabla_\varphi \Phi \rangle$, then $\mathcal{R}^T \leq \sum_{t \in [T]} G(\boldsymbol{\varphi}^t)$, and thus we derive an upper bound on the right-hand side to attain our Nash regret bound. First, consider the joint strategies $\hat{\boldsymbol{\phi}}^t = (\hat{\phi}_i^t)_{i \in N}$ and $\boldsymbol{\pi}^t = (\pi_i^t)_{i \in N}$ obtained from the strategies computed in lines 12-13. As the potential function $\Phi(\boldsymbol{\varphi}) = \sum_{i \in N} V_i(\boldsymbol{\varphi})$ is n^2-smooth w.r.t. the L_1-norm $\|\cdot\|_1$ by Lemma F.1 in Appendix F, then $\Phi(\varphi^{t+1}) \geq \Phi(\boldsymbol{\varphi}^t) + \langle \nabla \Phi(\boldsymbol{\varphi}^t), \boldsymbol{\varphi}^{t+1} - \boldsymbol{\varphi}^t \rangle - \frac{n^2}{2}\|\boldsymbol{\varphi}^{t+1} - \boldsymbol{\varphi}^t\|_1^2$. Combined with line 14, we prove in Appendix F that:

$$\begin{aligned}\Phi(\varphi^{t+1}) \geq \Phi(\boldsymbol{\varphi}^t) + (1-\rho_t)\eta_t \langle \nabla \Phi(\boldsymbol{\varphi}^t), \hat{\boldsymbol{\phi}}^t - \boldsymbol{\varphi}^t \rangle \\ -\rho_t \cdot n^3 - \tfrac{n^4}{2}[(1-\rho_t)^2 \eta_t^2 + \rho_t^2]\end{aligned} \tag{12}$$

Thus, we then devise a lower bound on the right-hand side. In Appendix F, we first show that $\langle \nabla \Phi(\boldsymbol{\varphi}^t), \hat{\boldsymbol{\phi}}^t - \boldsymbol{\varphi}^t \rangle \geq G(\boldsymbol{\varphi}^t) - 2n\|\nabla \Phi(\boldsymbol{\varphi}^t) - \mathbf{g}^t\|_\infty$, where $\mathbf{g}^t = (\mathbf{g}_i^t)_{i \in [n]}$ is the gradient estimator of $\nabla \Phi(\boldsymbol{\varphi}^t)$. After applying this inequality to (12), we rearrange the resulting inequality to obtain:

$$\begin{aligned}G(\boldsymbol{\varphi}^t) \leq \tfrac{\Phi(\boldsymbol{\varphi}^{t+1}) - \Phi(\boldsymbol{\varphi}^t)}{(1-\rho_t)\eta_t} + 2n\|\nabla \Phi(\boldsymbol{\varphi}^t) - \mathbf{g}^t\|_\infty \\ + \tfrac{\rho_t n^3}{(1-\rho_t)\eta_t} + \tfrac{n^4(1-\rho_t)\eta_t}{2} + \tfrac{\rho_t^2 n^4}{2(1-\rho_t)\eta_t}\end{aligned} \tag{13}$$

By summing this inequality over all rounds T while noting that $\sum_{t \in [T]} [\Phi(\boldsymbol{\varphi}^{t+1}) - \Phi(\boldsymbol{\varphi}^t)] = \Phi(\boldsymbol{\varphi}^{t+1}) - \Phi(\boldsymbol{\varphi}^1)$ as a telescoping series and $|\Phi(\boldsymbol{\varphi}^{t+1}) - \Phi(\boldsymbol{\varphi}^1)| \leq n$, we then have that the Nash regret is upper bounded by:

$$\begin{aligned}\mathcal{R}^T \leq 2n \textstyle\sum_{t \in [T]} \|\nabla \Phi(\boldsymbol{\varphi}^t) - \mathbf{g}^t\|_\infty + \tfrac{nT}{(1-\rho_t)\eta_t} + \\ + \tfrac{\rho_t n^3 T}{(1-\rho_t)\eta_t} + \tfrac{n^4 T(1-\rho_t)\eta_t}{2} + \tfrac{\rho_t^2 n^4 T}{2(1-\rho_t)\eta_t}\end{aligned} \tag{14}$$

Recall that this bound also serves as an upper bound on the Nash regret incurred by our algorithm since $\mathcal{R}^T \leq \sum_{t \in [T]} G(\boldsymbol{\varphi}^t)$. Further, our bound in (14) depends on the algorithm's error term, for which we have already supplied an upper bound in Lemma 4. Hence, after plugging the algorithm's inputs $\rho_t = \frac{1}{T}$, $\eta_t = \frac{1}{\sqrt{T}}$, $\mu_t = 1 - \frac{1}{T}$ for any time t and $T \geq 2$ to (14), in Appendix F we then apply Lemma 4 to obtain the Nash regret bounds in (10)–(11). □

Remark 5. *Theorem 1 indicates that D1S-FW has a Nash regret bound of $\tilde{\mathcal{O}}(\sqrt{T})$. By standard online-to-batch conversion (see, e.g., [Jin* et al.*, 2018]), our Nash regret bound suggests a sample complexity bound of $T = \mathcal{O}(n^8/\varepsilon^2)$ for obtaining an ε-NS strategy. Surprisingly, though D1S-FW uses only* one *sample per round, our Nash regret and sample complexity bounds are* **optimal** *up to logarithmic factors [Hassani* et al.*, 2020; Bai and Jin, 2020]. To guarantee those bounds, Theorem 1 dictates that the step sizes should depend on the number of rounds T, which is known in our context. As mentioned earlier, this is practical in many realistic cases as our project teams example from Section 1, where developers may participate in project teams over a predefined number of project milestones. Even if T is not explicitly known in some cases, developers often have a rough estimate based on the nature of open-source projects. For example, they may anticipate a typical project duration based on prior experience or the project's roadmap. Hence, agents can still execute our algorithm in a distributed manner even without knowing T.*

5 Conclusions and Future Work

In this paper, we presented a new model for studying online learning in coalition formation through the lens of *decentralized decision-making* by selfish agents. Our goal was designing sample-efficient distributed algorithms for self-interested agents that minimize their Nash regret, resulting in approximately Nash stable partitions. As such, we devised a Frank-Wolfe methods, where every agent uses *one* sample per round for gradient estimation, yet it attains Nash regret and sample complexity bounds that are ***optimal*** up to logarithmic factors.

Our work opens the way for plenty future studies. Immediate directions include the investigation of other classes of hedonic games, other solution concepts and an empirical analysis. Future studies also warrants exploring other models of partial information (e.g., semi-bandit feedback), and study cases where the number of rounds is not necessarily known. Finally, an interesting future direction is devising distributed algorithms with no-regret guarantees for any agent adopting them, ensuring that each agent has diminishing regret, regardless of how others update their strategies.

Acknowledgments

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

References

[Aloisio *et al.*, 2020] Alessandro Aloisio, Michele Flammini, and Cosimo Vinci. The impact of selfishness in hypergraph hedonic games. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(2):1766–1773, 2020.

[Aziz and Savani, 2016] Haris Aziz and Rahul Savani. Hedonic games. In Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D.Editors Procaccia, editors, *Handbook of Computational Social Choice*, page 356–376. Cambridge University Press, 2016.

[Aziz *et al.*, 2013] Haris Aziz, Felix Brandt, and Hans Georg Seedig. Computing desirable partitions in additively separable hedonic games. *Artificial Intelligence*, 195:316–334, 2013.

[Bai and Jin, 2020] Yu Bai and Chi Jin. Provable self-play algorithms for competitive reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 551–560. PMLR, 2020.

[Ballester, 2004] Coralio Ballester. Np-completeness in hedonic games. *Games and Economic Behavior*, 49(1):1–30, 2004.

[Balliu *et al.*, 2019] Alkida Balliu, Michele Flammini, Giovanna Melideo, and Dennis Olivetti. On non-cooperativeness in social distance games. *Journal of Artificial Intelligence Research*, 66:625–653, 2019.

[Banerjee *et al.*, 2001] Suryapratim Banerjee, Hideo Konishi, and Tayfun Sönmez. Core in a simple coalition formation game. *Social Choice and Welfare*, 18(1):135–153, 2001.

[Bilò *et al.*, 2018] Vittorio Bilò, Angelo Fanelli, Michele Flammini, Gianpiero Monaco, and Luca Moscardelli. Nash stable outcomes in fractional hedonic games: Existence, efficiency and computation. *Journal of Artificial Intelligence Research*, 62:315–371, 2018.

[Boehmer *et al.*, 2023] Niclas Boehmer, Martin Bullinger, and Anna Maria Kerkmann. Causes of stability in dynamic coalition formation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(5):5499–5506, 2023.

[Bogomolnaia and Jackson, 2002] Anna Bogomolnaia and Matthew O Jackson. The stability of hedonic coalition structures. *Games and Economic Behavior*, 38(2):201–230, 2002.

[Brandt *et al.*, 2022] Felix Brandt, Martin Bullinger, and Leo Tappe. Single-agent dynamics in additively separable hedonic games. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(5):4867–4874, 2022.

[Bullinger and Romen, 2023] Martin Bullinger and René Romen. Online Coalition Formation Under Random Arrival or Coalition Dissolution. In *31st Annual European Symposium on Algorithms (ESA 2023)*, volume 274 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 27:1–27:18, 2023.

[Bullinger and Romen, 2024] Martin Bullinger and René Romen. Stability in online coalition formation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(9):9537–9545, 2024.

[Chen and Peng, 2020] Xi Chen and Binghui Peng. Hedging in games: Faster convergence of external and swap regrets. *Advances in Neural Information Processing Systems*, 33:18990–18999, 2020.

[Cohen and Agmon, 2023a] Saar Cohen and Noa Agmon. Complexity of probabilistic inference in random dichotomous hedonic games. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(5):5573–5581, 2023.

[Cohen and Agmon, 2023b] Saar Cohen and Noa Agmon. Online coalitional skill formation. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems, AAMAS*, pages 494–503, 2023.

[Cohen and Agmon, 2024a] Saar Cohen and Noa Agmon. Online friends partitioning under uncertainty. In *ECAI 2024*, pages 3332–3339. IOS Press, 2024.

[Cohen and Agmon, 2024b] Saar Cohen and Noa Agmon. Online learning of partitions in additively separable hedonic games. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 2722–2730, 2024.

[Cohen and Agmon, 2025a] Saar Cohen and Noa Agmon. Decentralized online learning by selfish agents in coalition formation – supplementary materials. https://u.cs.biu.ac.il/~agmon/CohenIJCAI25Sup.pdf, 2025.

[Cohen and Agmon, 2025b] Saar Cohen and Noa Agmon. Online learning of coalition structures by selfish agents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(13):13709–13717, 2025.

[Cui *et al.*, 2022] Qiwen Cui, Zhihan Xiong, Maryam Fazel, and Simon S Du. Learning in congestion games with bandit feedback. *Advances in Neural Information Processing Systems*, 35:11009–11022, 2022.

[Dadi *et al.*, 2024] Leello Dadi, Ioannis Panageas, Stratis Skoulakis, Luca Viano, and Volkan Cevher. Polynomial convergence of bandit no-regret dynamics in congestion games. *arXiv preprint arXiv:2401.09628*, 2024.

[Daskalakis and Pan, 2014] Constantinos Daskalakis and Qinxuan Pan. A counter-example to karlin's strong conjecture for fictitious play. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 11–20. IEEE, 2014.

[Daskalakis *et al.*, 2021] Constantinos Daskalakis, Maxwell Fishelson, and Noah Golowich. Near-optimal no-regret learning in general games. *Advances in Neural Information Processing Systems*, 34:27604–27616, 2021.

[Defazio and Bottou, 2019] Aaron Defazio and Leon Bottou. On the ineffectiveness of variance reduced optimization for deep learning. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 1755–1765, 2019.

[Ding *et al.*, 2022] Dongsheng Ding, Chen-Yu Wei, Kaiqing Zhang, and Mihailo Jovanovic. Independent policy gradient for large-scale Markov potential games: Sharper rates, function approximation, and game-agnostic convergence. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5166–5220, 2022.

[Dreze and Greenberg, 1980] Jacques H Dreze and Joseph Greenberg. Hedonic coalitions: Optimality and stability. *Econometrica: Journal of the Econometric Society*, pages 987–1003, 1980.

[Fioravanti *et al.*, 2023] Simone Fioravanti, Michele Flammini, Bojana Kodric, and Giovanna Varricchio. Pac learning and stabilizing hedonic games: towards a unifying approach. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(5):5641–5648, 2023.

[Flammini *et al.*, 2021a] Michele Flammini, Bojana Kodric, Gianpiero Monaco, and Qiang Zhang. Strategyproof mechanisms for additively separable and fractional hedonic games. *Journal of Artificial Intelligence Research*, 70:1253–1279, 2021.

[Flammini *et al.*, 2021b] Michele Flammini, Gianpiero Monaco, Luca Moscardelli, Mordechai Shalom, and Shmuel Zaks. On the online coalition structure generation problem. *Journal of Artificial Intelligence Research*, 72:1215–1250, 2021.

[Gairing and Savani, 2019] Martin Gairing and Rahul Savani. Computing stable outcomes in symmetric additively separable hedonic games. *Mathematics of Operations Research*, 44(3):1101–1121, 2019.

[Hassani *et al.*, 2020] Hamed Hassani, Amin Karbasi, Aryan Mokhtari, and Zebang Shen. Stochastic conditional gradient++: (non)convex minimization and continuous submodular maximization. *SIAM Journal on Optimization*, 30(4):3315–3344, 2020.

[Hazan and Kale, 2012] Elad Hazan and Satyen Kale. Projection-free online learning. In *29th International Conference on Machine Learning, ICML 2012*, pages 521–528, 2012.

[Jin *et al.*, 2018] Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is q-learning provably efficient? *Advances in Neural Information Processing Systems*, 31:4863–4873, 2018.

[Lattimore and Szepesvári, 2020] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[Leonardos *et al.*, 2022] Stefanos Leonardos, Will Overman, Ioannis Panageas, and Georgios Piliouras. Global convergence of multi-agent policy gradient in markov potential games. In *International Conference on Learning Representations*, 2022.

[Leslie and Collins, 2006] David S Leslie and Edmund J Collins. Generalised weakened fictitious play. *Games and Economic Behavior*, 56(2):285–298, 2006.

[Liu *et al.*, 2020] Lydia T Liu, Horia Mania, and Michael Jordan. Competing bandits in matching markets. In *International Conference on Artificial Intelligence and Statistics*, pages 1618–1628. PMLR, 2020.

[Liu *et al.*, 2021] Qinghua Liu, Tiancheng Yu, Yu Bai, and Chi Jin. A sharp analysis of model-based reinforcement learning with self-play. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7001–7010, 2021.

[Maheshwari *et al.*, 2022] Chinmay Maheshwari, Shankar Sastry, and Eric Mazumdar. Decentralized, communication-and coordination-free learning in structured matching markets. *Advances in Neural Information Processing Systems*, 35:15081–15092, 2022.

[Mokhtari *et al.*, 2018] Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Conditional gradient method for stochastic submodular maximization: Closing the gap. In *International Conference on Artificial Intelligence and Statistics*, pages 1886–1895. PMLR, 2018.

[Mokhtari *et al.*, 2020] Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Stochastic conditional gradient methods: From convex minimization to submodular maximization. *Journal of machine learning research*, 21(105):1–49, 2020.

[Sandholm *et al.*, 1999] Tuomas Sandholm, Kate Larson, Martin Andersson, Onn Shehory, and Fernando Tohmé. Coalition structure generation with worst case guarantees. *Artificial intelligence*, 111(1-2):209–238, 1999.

[Sliwinski and Zick, 2017] Jakub Sliwinski and Yair Zick. Learning hedonic games. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 2730–2736, 2017.

[Soare *et al.*, 2014] Marta Soare, Alessandro Lazaric, and Rémi Munos. Best-arm identification in linear bandits. *Advances in Neural Information Processing Systems*, 27, 2014.

[Sung and Dimitrov, 2010] Shao-Chin Sung and Dinko Dimitrov. Computational complexity in additive hedonic games. *European Journal of Operational Research*, 203(3):635–639, 2010.

[Yang and Tan, 2022] Junwen Yang and Vincent Tan. Minimax optimal fixed-budget best arm identification in linear bandits. *Advances in Neural Information Processing Systems*, 35:12253–12266, 2022.

[Zhang *et al.*, 2020] Mingrui Zhang, Zebang Shen, Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. One sample stochastic frank-wolfe. In *International Conference on Artificial Intelligence and Statistics*, pages 4012–4023. PMLR, 2020.

[Zhang *et al.*, 2022] Yirui Zhang, Siwei Wang, and Zhixuan Fang. Matching in multi-arm bandit with collision. *Advances in Neural Information Processing Systems*, 35:9552–9563, 2022.