Computing Better Approximate Pure Nash Equilibria in Payoff-maximization Potential Games

Angelo Fanelli
CNRS, LAMSADE, Paris Dauphine University – PSL
angelo.fanelli@cnrs.fr

Abstract

Potential games are a fundamental class of games in which pure Nash equilibria are guaranteed to exist, yet computing such equilibria is computationally intractable for several subclasses. This has led to extensive research on computing approximate pure Nash equilibria. In this paper, we study payoff-maximization potential games. For these games, strong approximation guarantees are known only for restricted subclasses, most notably P_d–FLIP games. We show that standard approaches based on unilateral improvement moves can fail to provide any finite approximation guarantee even for simple extensions of P_d–FLIP games. To overcome this limitation, we propose an algorithmic framework based on coordinated moves by small groups of players, whose approximation guarantee and number of moves are controlled by two natural game parameters, the stretch and the spread, which are bounded for broad classes of games of interest. In the special case of P_d–FLIP games, our framework can be configured to recover the existing algorithm, matching its approximation guarantee and the number of moves performed.

1 Introduction

Potential games [Monderer and Shapley, 1996] are a fundamental class of games in which pure Nash equilibria are guaranteed to exist. A defining feature of these games is the existence of a potential function that exactly captures individual incentives: whenever a player unilaterally changes strategy, the resulting change in the potential equals the change in that player's payoff. As a consequence, any sequence of unilateral improvement moves, where a player switches to a strategy with higher payoff, converges to a strategy profile that locally maximizes the potential function, and hence to a pure Nash equilibrium.

Beyond their elegant theoretical properties, potential games play a central role in modern AI. [Babichenko and Tamuz, 2016] establish a tight correspondence between graphical potential games and positive *Markov Random Fields* (MRFs) [Koller and Friedman, 2009]. In this correspondence, the game's potential function coincides exactly with the MRF's log-potential, as implied by the Hammersley–Clifford theorem [Hammersley and Clifford, 1971]. Consequently, pure Nash equilibria of the game coincide with local maxima of the MRF's log-potential, and thus with locally most probable configurations of the MRF. This connection establishes a direct bridge between equilibrium computation and probabilistic inference, enabling techniques from one domain to be systematically applied to the other.

Another important application area of potential games is distributed optimization. In this context, potential games provide a natural framework for modeling systems in which agents optimize local payoffs aligned with a global objective. [Marden *et al.*, 2009] and [Li and Marden, 2013] show that many coordination tasks in multi-agent systems, such as consensus and coverage, can be formulated as potential games, where simple decentralized learning dynamics converge to Nash equilibria that are provably efficient, and in some cases globally optimal.

Given their wide-ranging applications, a fundamental computational question is how to efficiently compute a pure Nash equilibrium in a potential game. Although a pure Nash equilibrium is guaranteed to exist and can be reached through any sequence of improving steps, this simple approach may require exponential time in the worst case. More generally, the problem of computing a pure Nash equilibrium in potential games is PLS-complete even for many natural subclasses (e.g., [Fabrikant *et al.*, 2004; Johnson *et al.*, 1988]). This has led to considerable interest in computing *approximate pure Nash equilibria*. Informally, an α-approximate pure Nash equilibrium is a strategy profile in which no player can improve its payoff by more than a multiplicative factor $\alpha \geq 1$. A pure Nash equilibrium corresponds to the case $\alpha = 1$. The goal of this line of research is to design polynomial-time algorithms that compute equilibria with the smallest possible approximation factor.

Prior work on approximate equilibrium computation has considered both payoff-maximization and cost-minimization potential games. The former have been studied through P_d–FLIP games [Bhalgat *et al.*, 2010; Caragiannis *et al.*, 2017], a well-structured subclass of constraint satisfaction games, in which players seek to satisfy as many Boolean constraints as possible. The latter have been studied through congestion games with polynomial latencies [Caragiannis *et al.*, 2011; Caragiannis *et al.*, 2015; Feldotto *et al.*, 2017; Caragiannis and Fanelli, 2019; Vijayalakshmi and Skopalik, 2020; Giannakopoulos *et al.*, 2022], in which players seek to minimize individual latency costs. Algorithms for both classes share a similar high-level structure, consisting of suitably scheduled sequences of unilateral improvement moves. However, the analysis techniques diverge significantly between the two settings, reflecting the different nature of their objectives. Throughout the paper, the term *potential game* refers to a payoff-maximization potential game. The qualifiers *payoff-maximization* and *cost-minimization* are used only when a distinction between the two settings is required.

Constraint satisfaction games (see Example 1 for a formal definition) form a restricted subclass of potential games, characterized by binary strategies, with each player's payoff given by the number of satisfied Boolean constraints in which it participates. Within this class, the literature has focused on P_d–FLIP games [Bhalgat *et al.*, 2010; Caragiannis *et al.*, 2017], which impose strong structural properties: each constraint has arity at most d, and any unsatisfied constraint can be satisfied by changing the strategy of any participating player. Leveraging these properties, approximate pure Nash equilibria can be computed through suitably scheduled sequences of unilateral improvement moves, with both the approximation guarantee and the number of required moves bounded by a polynomial in the number of players and d. More precisely, the approximation guarantee is quantified by the worst-case ratio between the maximum potential value and the potential value of a sufficiently accurate approximate pure Nash equilibrium, that is, one that is close enough to an exact equilibrium. Beyond the P_d–FLIP setting, however, such a guarantee becomes arbitrarily weak: even in simple potential games (see Example 2), approximate pure Nash equilibria may have very low potential values relative to the maximum potential value, no matter how accurate they are, resulting in an arbitrarily large approximation guarantee.

In this work, we move beyond the restrictive P_d–FLIP setting and develop a general framework for computing approximate pure Nash equilibria in arbitrary potential games. To this end, we adopt the characterization of potential games introduced by [Ui, 2000], which expresses payoffs in terms of a family of interaction functions over subsets of players. Building on this representation, we propose an algorithmic scheme that extends the phase-based improvement approach of [Caragiannis *et al.*, 2017] to the general setting.

To achieve meaningful approximation guarantees beyond the P_d–FLIP setting, we introduce a novel approach: the use of coordinated moves by small groups of players, rather than unilateral moves. Such moves are essential to escape low-quality local optima and to obtain meaningful approximation guarantees in general potential games, where standard unilateral improvement dynamics provably fail (a formal treatment of group moves is deferred to Section 3).

The major contribution of this work is an algorithm (Algorithm 1) that, given as input a potential game $\mathcal{G}$, an initial state, an integer group-size parameter $k \geq 1$, and an accuracy parameter $\gamma \in (0, 1]$, and instantiated with parameters $\beta, \sigma \geq 1$ satisfying $\beta \geq \texttt{Stretch}^k_{1+\gamma}(\mathcal{G})$ and $\sigma \geq \Gamma(\mathcal{G})$, constructs a sequence of moves by groups of at most k players, each improving the "payoff" of the participating group by a factor greater than $1 + \gamma$, leading from the given initial state to a $(\beta + 3\gamma)$-approximate pure Nash equilibrium (Theorem 3). Here, $\texttt{Stretch}^k_{1+\gamma}(\mathcal{G})$ and $\Gamma(\mathcal{G})$ denote two game-dependent parameters, called the $(1 + \gamma, k)$-stretch and the spread of the game, respectively. Informally, the $(1 + \gamma, k)$-stretch captures the worst-case ratio between the maximum potential value of the game and the potential value of any state from which no group of at most k players can improve its "payoff" by a factor greater than $1 + \gamma$. The spread, instead, captures the worst-case ratio between the maximum payoff attainable by a player and the minimum payoff that the player can guarantee for itself regardless of the strategies chosen by the other players. The length of the resulting sequence of moves is polynomial in β, σ, and the logarithm of the maximum payoff attainable by any player.

We further show (Theorem 4) that the last state of the sequence of moves constructed by the algorithm satisfies a stronger stability property: it is stable against coordinated moves by any group of at most k players aiming to improve its "payoff" by a factor greater than $\beta + 3\gamma$. This stronger notion of stability is not guaranteed by the algorithm of [Caragiannis *et al.*, 2017] for P_d–FLIP games, which only ensures stability with respect to unilateral moves.

Finally, we establish (Theorem 5) an explicit upper bound on the $(1 + \gamma, k)$-stretch of every potential game $\mathcal{G}$ of degree $d \geq 2$ (the largest arity among its interaction functions), valid for every $\gamma \geq 0$ and every integer $k \geq d$, so that the algorithm's parameter β can be safely set to this bound. We remark that this latter result is closely related to the quality guarantees for k-optimal solutions in distributed constraint optimization problems (DCOPs) derived by [Pearce and Tambe, 2007].

When applied to P_d–FLIP games with $k = 1$, our algorithm recovers the same approximation guarantee and the same bound on the length of the sequence of moves as the algorithm of [Caragiannis *et al.*, 2017].

2 Preliminaries

Strategic Games. A *strategic game* $\mathcal{G}$ is a tuple $\langle N, (S_i)_{i\in N}, (u_i)_{i\in N}\rangle$, where N is the finite and non-empty set of *players*, and each S_i is the finite and non-empty set of *strategies* of player i. Every player i is associated with a *payoff function* $u_i : S_N \to \mathbb{R}$, where $S_N = S_1 \times S_2 \times \ldots \times S_{|N|}$ represents the set of *strategy profiles* (or *states*) of $\mathcal{G}$. The class of all strategic games is denoted by $\texttt{SG}$. The notation $\texttt{SG}^+ \subset \texttt{SG}$ refers to the class of potential games with non-negative payoffs, while $\texttt{SG}^{++} \subset \texttt{SG}^+$ denotes the subclass of games satisfying the following property: for every player i and every choice of strategies of the other players, there exists a strategy that guarantees player i a strictly positive payoff. Note that this condition is weaker than requiring that every player receives a strictly positive payoff in every state.

Notation for Strategy Profiles. The notion of strategy profiles extends naturally to any subset of players. Consider a strategic game $\langle N, (S_i)_{i\in N}, (u_i)_{i\in N}\rangle \in \texttt{SG}$. For every non-empty subset $P \subseteq N$, $S_P = \prod_{j\in P} S_j$ denotes the set of *strategy profiles* (or simply *strategies*) of P. Recall that, when $P = N$, S_P corresponds to the set of states of $\mathcal{G}$. Moreover,

$S_{\{i\}}$ coincides with S_i. For completeness, $S_\emptyset = \{\perp\}$. Elements of S_P are denoted by bold lowercase letters such as $\mathbf{s}, \mathbf{r}, \mathbf{x}, \mathbf{y}, \mathbf{z}$.

Additional notation is introduced to represent strategy profiles in a more expressive and concise manner. For every subset $Q \subseteq P \subseteq N$ and $\mathbf{s} \in S_P$, let $\mathbf{s}_Q \in S_Q$ be the restriction of $\mathbf{s}$ to Q; that is, $\mathbf{s}_Q$ retains only the components of $\mathbf{s}$ corresponding to the players in Q. Note that if $Q = \emptyset$, then $\mathbf{s}_Q = \perp$. For every $P \subseteq N$, $\mathbf{s} \in S_{N \setminus P}$, and $\mathbf{s}' \in S_P$, let $[\mathbf{s}, \mathbf{s}'] \in S_N$ denote the state defined as follows: if $P \neq \emptyset$ and $P \neq N$, then $[\mathbf{s}, \mathbf{s}'] = \bar{\mathbf{s}} \in S_N$, where $\bar{\mathbf{s}}_{\{j\}} = \mathbf{s}_{\{j\}}$ for every $j \in N \setminus P$, and $\bar{\mathbf{s}}_{\{j\}} = \mathbf{s}'_{\{j\}}$ for every $j \in P$; if $P = \emptyset$, then $[\mathbf{s}, \mathbf{s}'] = [\mathbf{s}, \perp] = \mathbf{s}$; and if $P = N$, then $[\mathbf{s}, \mathbf{s}'] = [\perp, \mathbf{s}'] = \mathbf{s}'$. To simplify the notation for $\mathbf{s}_P$ and S_P, the subscript i is used in place of $\{i\}$, and $\text{-}P$ is used in place of $N \setminus P$. *

Moves and Trajectories. Consider a strategic game $\mathcal{G} = \langle N, (S_i)_{i \in N}, (u_i)_{i \in N} \rangle \in \mathtt{SG}$. For every player $i \in N$ and pair of states $\mathbf{s}, \bar{\mathbf{s}} \in S_N$, with $\mathbf{s} \neq \bar{\mathbf{s}}$, the ordered pair $(\mathbf{s}, \bar{\mathbf{s}})$, where $\bar{\mathbf{s}} = [\mathbf{s}_{\text{-}i}, \mathbf{s}']$ for some $\mathbf{s}' \in S_i$, represents a *move* (i.e., a *strategy change*) by player i from $\mathbf{s}$. The move is said to be α*-improving*, for $\alpha \geq 1$, if $u_i(\bar{\mathbf{s}}) > \alpha u_i(\mathbf{s})$; in this case, $\mathbf{s}'$ is called an α*-improving strategy* for player i at $\mathbf{s}$. Note that an α-improving move (strategy) is also an α'-improving move (strategy), for every $\alpha > \alpha' \geq 1$.

A *trajectory* τ starting at state $\mathbf{s}^0$ and ending at state $\mathbf{s}^h$ is a sequence $\tau = \langle \mathbf{s}^0, \mathbf{s}^1, \ldots, \mathbf{s}^h \rangle$ of $h+1 \geq 1$ states such that, if $h \geq 1$, the pair $(\mathbf{s}^t, \mathbf{s}^{t+1})$ is a move by some player, for every $t \in \{0, 1, \ldots, h-1\}$. The *length* of τ is defined as $h \geq 0$. If $h \geq 1$, the trajectory τ is called α*-improving* if every move in the sequence is α-improving.

Approximate Equilibria. Consider a strategic game $\mathcal{G} = \langle N, (S_i)_{i \in N}, (u_i)_{i \in N} \rangle \in \mathtt{SG}^+$.† For any $\alpha \geq 1$, a state $\mathbf{s} \in S_N$ is called an α*-approximate pure Nash equilibrium* of $\mathcal{G}$ if no α-improving move exists from $\mathbf{s}$, i.e., for every $i \in N$ and every $\mathbf{s}' \in S_i$, it holds that $u_i([\mathbf{s}_{\text{-}i}, \mathbf{s}']) \leq \alpha u_i(\mathbf{s})$. If $\alpha = 1$, $\mathbf{s}$ is simply called a *pure Nash equilibrium*. The set of all α-approximate pure Nash equilibria of $\mathcal{G}$ is denoted by $\mathcal{E}_\alpha(\mathcal{G}) \subseteq S_N$. Observe that the relation $\mathcal{E}_{\alpha'}(\mathcal{G}) \subseteq \mathcal{E}_\alpha(\mathcal{G})$ holds for every $\alpha \geq \alpha' \geq 1$.

Definition 1 ([Monderer and Shapley, 1996]). *A strategic game* $\mathcal{G} = \langle N, (S_i)_{i \in N}, (u_i)_{i \in N} \rangle \in \mathtt{SG}$ *is called a* potential game *if there exists a* potential function $V : S_N \to \mathbb{R}$ *such that*

$$u_i([\mathbf{s}_{\text{-}i}, \mathbf{s}']) - u_i(\mathbf{s}) = V([\mathbf{s}_{\text{-}i}, \mathbf{s}']) - V(\mathbf{s}), \tag{1}$$

for every $i \in N$, $\mathbf{s} \in S_N$, $\mathbf{s}' \in S_i$. *The class of potential games is denoted by* $\mathtt{PG}$.

A key property of a potential game $\mathcal{G}$ with potential function V is that $\mathcal{E}_1(\mathcal{G}) \neq \emptyset$, meaning that at least one pure Nash equilibrium exists. Indeed, from equation (1), we know that V strictly increases along any 1-improving trajectory. Therefore, no state can appear more than once in such a trajectory. Since the set of states S_N is finite, it follows that every improving trajectory visits only finitely many states, and thus has finite length. Consider an improving trajectory of maximal length (i.e., it cannot be extended further). By definition, its final state admits no further 1-improving moves and is therefore a pure Nash equilibrium.

The theorem below characterizes potential games.

Theorem 1 ([Ui, 2000]). $\mathcal{G} = \langle N, (S_i)_{i \in N}, (u_i)_{i \in N} \rangle \in \mathtt{SG}$ *is a potential game if and only if there exist* interaction functions $\{\Psi_P \mid \Psi_P : S_P \to \mathbb{R}, P \in \mathcal{I}\}$*, where* $\mathcal{I} \subseteq 2_+^{N}$‡ *is a collection of* interaction sets*, such that*

$$u_i(\mathbf{s}) = \sum_{T \in \mathcal{I},\, i \in T} \Psi_T(\mathbf{s}_T), \tag{2}$$

for every $i \in N$ *and* $\mathbf{s} \in S_N$. *A potential function of* $\mathcal{G}$ *is given by* $V(\mathbf{s}) = \sum_{T \in \mathcal{I}} \Psi_T(\mathbf{s}_T)$.

Building on the structural characterization provided by Theorem 1, we introduce the representation of potential games adopted throughout the paper, together with the notions of arity and degree.

Our representation of potential games. Motivated by the characterization given in Theorem 1, throughout the paper we consider potential games represented explicitly by a family of interaction functions. Formally, a game in $\mathtt{PG}$ is specified by a tuple

$$\mathcal{G} = \langle N, \mathcal{I}, (\Psi_P)_{P \in \mathcal{I}}, (S_i)_{i \in N}, (u_i)_{i \in N} \rangle,$$

where $\mathcal{I} \subseteq 2_+^N$ is a collection of *interaction sets*, and for each $P \in \mathcal{I}$, the function $\Psi_P : S_P \to \mathbb{R}$ is an *interaction function* as described in Theorem 1. Throughout the paper, only the components of the tuple $\mathcal{G}$ that are relevant to the context are specified. Whenever a component of the tuple is not needed, it will be denoted by a placeholder $\cdot$. For instance, the notation $\mathcal{G} = \langle N, \cdot, \cdot, (S_i)_{i \in N}, (u_i)_{i \in N} \rangle$ is used when the interaction structure and the interaction functions are not explicitly needed. The symbol $\mathtt{PG}^+ \subset \mathtt{PG}$ denotes the class of potential games whose interaction functions are non-negative, while $\mathtt{PG}^{++} \subset \mathtt{PG}^+$ denotes the subclass of games satisfying the following property: for every player i and every choice of strategies of the other players, there exists a strategy that guarantees player i a strictly positive payoff. Note that $\mathtt{PG}^{++} \subset \mathtt{SG}^{++}$.

Arity and game degree. Consider a potential game $\mathcal{G} = \langle \cdot, \mathcal{I}, (\Psi_P)_{P \in \mathcal{I}}, \cdot, \cdot \rangle \in \mathtt{PG}$. The *arity* of an interaction function Ψ_P, with $P \in \mathcal{I}$, is the number of players involved in the interaction, i.e., $|P|$. The *degree* of the game $\mathcal{G}$ is the maximum arity over all interaction functions, i.e., $\max_{P \in \mathcal{I}} |P|$.

The following example introduces constraint satisfaction games and their P_d–FLIP subclass, two canonical examples of potential games, and illustrates how they can be naturally expressed in terms of the interaction functions of Theorem 1.

Example 1 (Constraint Satisfaction Games [Bhalgat *et al.*, 2010; Caragiannis *et al.*, 2017]). *A* constraint satisfaction

*Hence, $S_i = S_{\{i\}}$ and $S_{\text{-}P} = S_{N \setminus P}$. The same simplifications apply to $\mathbf{s}_P$.

†The presence of both negative and non-negative payoffs renders the notion of approximate pure Nash equilibrium meaningless.

‡2_+^N denotes the family of all non-empty subsets of N.

game *is a strategic game* $\mathcal{G} = \langle N, (S_i)_{i \in N}, (u_i)_{i \in N} \rangle \in \mathtt{SG}$ *in which each player* i *has a binary strategy set* $S_i = \{\mathtt{true}, \mathtt{false}\}$*. The game is additionally specified by a family* $\mathcal{C}$ *of Boolean constraints (also called clauses) over the set of players. Each constraint* $C \in \mathcal{C}$ *involves a subset of players* $N(C) \subseteq N$ *and is defined as a Boolean function* $C : S_{N(C)} \to \{\mathtt{true}, \mathtt{false}\}$*. The quantity* $|N(C)|$ *is called the* arity *of* C*. Given a strategy profile* $\mathbf{s} \in S_N$*, a constraint* C *is said to be* satisfied *if* $C(\mathbf{s}_{N(C)}) = \mathtt{true}$*, and* unsatisfied *otherwise. Each constraint* C *is associated with a positive weight* $w_C > 0$*. Its value under* $\mathbf{s}$ *is defined as* $value_C(\mathbf{s}_{N(C)}) = w_C$ *if* C *is satisfied, and* $value_C(\mathbf{s}_{N(C)}) = 0$ *otherwise. The payoff of player* i *under* $\mathbf{s}$ *is then given by* $u_i(\mathbf{s}) = \sum_{C \in \mathcal{C},\, i \in N(C)} value_C(\mathbf{s}_{N(C)})$*.*

Constraint satisfaction games are potential games, which implies that the players' payoffs can be expressed in terms of the interaction functions of Theorem 1. Indeed, it suffices to set $\mathcal{I} = \{N(C) \mid C \in \mathcal{C}\}$ *and* $\Psi_{N(C)}(\mathbf{s}_{N(C)}) = value_C(\mathbf{s}_{N(C)})$ *for every* $C \in \mathcal{C}$*. Thus, each clause corresponds naturally to an interaction set, and its value to the associated interaction function.*

P_d*–*FLIP *games form a subclass of constraint satisfaction games such that (i) every constraint has arity at most* $d \geq 1$*, and (ii) every unsatisfied constraint* C *can be satisfied by changing the strategy of any player in* $N(C)$*. Consequently, every* P_d*–*FLIP *game has degree at most* d *and belongs to* $\mathtt{PG}^{++}$*. Two notable subclasses of* P_d*–*FLIP *games are* cut games *and* party affiliation games*, which have been extensively studied in the algorithmic game theory literature (see, e.g., [Balcan* et al.*, 2013; Bhalgat* et al.*, 2010; Caragiannis and Jiang, 2023; Christodoulou* et al.*, 2012; Fabrikant* et al.*, 2004]).*

3 Approach Overview and Contribution

This section outlines the key ingredients of our algorithm.

Existing algorithms for computing approximate pure Nash equilibria in both payoff-maximization (P_d–FLIP games) and cost-minimization potential games (congestion games with polynomial latencies) largely adhere to an algorithmic paradigm introduced in the seminal work by [Caragiannis *et al.*, 2011; Caragiannis *et al.*, 2015] for congestion games with polynomial latencies. The key idea is to deterministically schedule players' improving moves in order to obtain a polynomial-length trajectory from any initial state to an approximate pure Nash equilibrium.

For both P_d–FLIP games and congestion games with polynomial latencies, the quality of approximation achieved by this approach is captured by the notion of $(1+\gamma)$-*stretch*. For P_d–FLIP games, the $(1+\gamma)$-*stretch* is defined as the maximum ratio between the optimal potential value and the potential at any $(1+\gamma)$-approximate pure Nash equilibrium, for arbitrarily small $\gamma > 0$. The corresponding notion for congestion games with polynomial latencies is obtained by reversing the ratio. We formalize in Definition 4 a generalized notion of stretch for potential games, extending the notion used for P_d–FLIP games to arbitrary potential games and allowing coordinated deviations of groups of up to k players, rather than only unilateral deviations. Notably, the stretch as defined for P_d–FLIP games corresponds exactly to the case $k = 1$ in our framework.

The success of this approach for P_d–FLIP games naturally raises the question of whether the same paradigm can yield meaningful guarantees for arbitrary potential games. Even if the scheduling framework could be extended beyond P_d–FLIP games, its approximation guarantee would still be governed by the $(1+\gamma)$-stretch. Unfortunately, this leads to a severe limitation: in general potential games, the $(1+\gamma)$-stretch can be unbounded. Example 2 exhibits a simple potential game for which the ratio between the optimal potential value and the potential at any pure $(1+\gamma)$-approximate Nash equilibrium can grow arbitrarily large.

Example 2. *For any pair* $M > m > 0$ *and any* $\gamma \geq 0$*, consider a potential game with two players,* $N = \{1, 2\}$*, sharing the same strategy set* $\{0, 1\}$*. There is a unique interaction set* $\{1, 2\}$ *and the corresponding interaction function* $\Psi_{\{1,2\}}$ *is defined as:* $\Psi_{\{1,2\}}(0,1) = \Psi_{\{1,2\}}(1,0) = 0$*,* $\Psi_{\{1,2\}}(0,0) = m$*, and* $\Psi_{\{1,2\}}(1,1) = M$*. Figure 1 illustrates the potential values of this game. Observe that this game does not belong to the class of* P_d*–*FLIP *games, since the interaction function takes two distinct positive values* m *and* M*, whereas in* P_d*–*FLIP *games each constraint contributes a unique positive value when satisfied. The state* $(0,0)$ *is a* $(1+\gamma)$*-approximate pure Nash equilibrium with potential value* m*, while the state* $(1,1)$ *maximizes the potential function with value* M*. Thus, the maximum potential value is* M*, the potential of the worst* $(1+\gamma)$*-approximate pure Nash equilibrium is* m*, and their ratio can be made arbitrarily large.*

		Player 2: 0	Player 2: 1
Player 1	0	m	0
	1	0	M

Figure 1: Potential values of the instance in Example 2.

It is worth noting that the bounds for cut games, initially studied in [Caragiannis *et al.*, 2017], were refined in [Caragiannis and Jiang, 2023]. Similarly, the approximation bounds for congestion games with polynomial latencies, established in [Caragiannis *et al.*, 2011; Caragiannis *et al.*, 2015], were later improved in [Feldotto *et al.*, 2017] and [Giannakopoulos *et al.*, 2022]. All these results build on the framework introduced in [Caragiannis *et al.*, 2011; Caragiannis *et al.*, 2015] and exploit structural properties specific to the underlying games. However, such techniques are often difficult to generalize and differ substantially from the general-purpose approach developed in this work.

Motivated by the observation illustrated in Example 2, this work proposes a new approach to computing approximate equilibria in potential games. *Like previous methods, this approach deterministically schedules sequences of improving moves; however, in contrast to classical techniques that rely on unilateral moves, it allows coordinated moves by groups of* *up to $k \geq 1$ players. That is, in each step, a group of at most k players simultaneously change their strategies in order to increase the "payoff" of the group as a whole, naturally extending the notion of individual improvement to collective improvement.* It is important to emphasize that these *group moves are algorithmically designed and, unlike individual improvement steps, are not necessarily incentive-compatible*, i.e., a move by a group of players that improves the "payoff" of the group may reduce the payoff of some of the players involved.

This approach is motivated by two key insights. First, a strong connection exists between the task of computing approximate equilibria with strong guarantees and that of identifying strategy profiles with near-optimal potential values. Indeed, approximation guarantees obtained through the stretch are fundamentally tied to how close the potential at equilibrium is to the optimal potential value. This connection was originally established in [Caragiannis *et al.*, 2011; Caragiannis *et al.*, 2015] and is further reinforced by our framework. Second, allowing coordinated moves by groups of players enables the algorithm to escape low-potential states and reach higher-potential configurations more effectively. This effect is illustrated in Example 2, where a joint move allows players to bypass a poor equilibrium and reach a globally better state. As we will formally show, when the size of the deviating groups is sufficiently large, the resulting sequence of moves converge to approximate equilibria with provably better approximation guarantees.

To formally capture the notion of an "improving" move by a group of players, we extend the standard definition of payoff to arbitrary subsets of players (Definition 3). Our group payoff function is designed so as to coincide with the individual payoff (as formulated by Theorem 1) when the group consists of a single player, and with the potential function when the group includes all players. Crucially, we prove (Theorem 2) that the proposed group payoff preserves and generalizes the defining property of exact potential games. Specifically, for every move performed by a group Q, the induced variation of the payoff of Q is exactly reflected in the payoff variation of every superset $P \supseteq Q$, including the set of all players, whose payoff coincides with the potential function.

Definition 2. *Let Ω be a finite non-empty set, and let $\mathcal{F}$ be a family of non-empty subsets of Ω. For any subset $A \subseteq \Omega$, the set of* A-hitting sets *of $\mathcal{F}$ is defined as*

$$\mathcal{H}(\mathcal{F}, A) = \{F \in \mathcal{F} \mid F \cap A \neq \emptyset\}.$$

Definition 3. *Let $\mathcal{G} = \langle N, \mathcal{I}, (\Psi_P)_{P\in\mathcal{I}}, (S_i)_{i\in N}, (u_i)_{i\in N}\rangle \in$ $\mathtt{PG}$. For every non-empty group of players $P \subseteq N$ and every state $\mathbf{s} \in S_N$, the* payoff of P in $\mathbf{s}$ *is defined as*

$$u_P(\mathbf{s}) = \sum_{T\in\mathcal{H}(\mathcal{I},P)} \Psi_T(\mathbf{s}_T).$$

Note that $u_{\{i\}}(\mathbf{s}) = u_i(\mathbf{s})$ and $u_N(\mathbf{s})$ is a potential *of $\mathcal{G}$.*

The next subsection generalizes the notions of moves and trajectories (i.e., sequences of moves) to groups of players. The resulting definitions include as special cases the corresponding concepts introduced in Section 2 for individual players.

Moves and Trajectories by Group of Players. Consider a potential game $\mathcal{G} = \langle N, (S_i)_{i\in N}, (u_i)_{i\in N}\rangle \in \mathtt{PG}$. Any group of at most $k \geq 1$ players is called a *k-group*. For every group $P \subseteq N$ and strategy profiles $\mathbf{s}, \bar{\mathbf{s}} \in S_N$, with $\mathbf{s} \neq \bar{\mathbf{s}}$, the ordered pair $(\mathbf{s}, \bar{\mathbf{s}})$, where $\bar{\mathbf{s}} = [\mathbf{s}_{-P}, \mathbf{s}']$ for some $\mathbf{s}' \in S_P$, represents a *move* (i.e., a *strategy change*) by P from $\mathbf{s}$. The move is said to be *α-improving*, for $\alpha \geq 1$, if $u_P(\bar{\mathbf{s}}) > \alpha u_P(\mathbf{s})$; in this case, $\mathbf{s}'$ is called an *α-improving strategy* for P at $\mathbf{s}$. Note that an α-improving move (strategy) is also an α'-improving move (strategy), for every $\alpha > \alpha' \geq 1$.

A *trajectory* τ starting at state $\mathbf{s}^0$ and ending at state $\mathbf{s}^h$ is a sequence $\tau = \langle \mathbf{s}^0, \mathbf{s}^1, \ldots, \mathbf{s}^h\rangle$ of $h+1 \geq 1$ states such that, if $h \geq 1$, the pair $(\mathbf{s}^t, \mathbf{s}^{t+1})$ is a move by some group, for every $t \in \{0, 1, \ldots, h-1\}$. The *length* of τ is defined as $h \geq 0$. If $h \geq 1$, the trajectory τ is called *α-improving* if every move in the sequence is α-improving. If $(\mathbf{s}^h, \mathbf{s}^{h+1})$ is a move by some group from $\mathbf{s}^h$, then $\tau \oplus \mathbf{s}^{h+1}$ denotes the extended trajectory $\langle \mathbf{s}^0, \mathbf{s}^1, \ldots, \mathbf{s}^h, \mathbf{s}^{h+1}\rangle$.

The next theorem generalizes the defining property of potential games (1). Specifically, by setting $Q = \{i\}$ and $P = N$ in the theorem, we obtain (1).

Theorem 2. *Let $\mathcal{G} = \langle N, \mathcal{I}, (\Psi_P)_{P\in\mathcal{I}}, (S_i)_{i\in N}, (u_i)_{i\in N}\rangle \in \mathtt{PG}$. Then, for every $Q \subseteq P \subseteq N$, $\mathbf{s} \in S_N$, and $\mathbf{s}' \in S_Q$, we have*

$$u_Q([\mathbf{s}_{-Q}, \mathbf{s}']) - u_Q(\mathbf{s}) = u_P([\mathbf{s}_{-Q}, \mathbf{s}']) - u_P(\mathbf{s}).$$

Proof. Let $\bar{\mathbf{s}} = [\mathbf{s}_{-Q}, \mathbf{s}']$, then

$$\begin{aligned} u_P(\bar{\mathbf{s}}) - u_P(\mathbf{s}) &= \sum_{T\in\mathcal{H}(\mathcal{I},P)} \Psi_T(\bar{\mathbf{s}}_T) \ - \sum_{T\in\mathcal{H}(\mathcal{I},P)} \Psi_T(\mathbf{s}_T) \\ &= \sum_{T\in\mathcal{H}(\mathcal{I},P)} \big(\Psi_T(\bar{\mathbf{s}}_T) - \Psi_T(\mathbf{s}_T)\big) \\ &= \sum_{T\in\mathcal{H}(\mathcal{I},Q)} \big(\Psi_T(\bar{\mathbf{s}}_T) - \Psi_T(\mathbf{s}_T)\big) \\ &= u_Q(\bar{\mathbf{s}}) - u_Q(\mathbf{s}), \end{aligned}$$

where the third equality follows from the fact that $\Psi_T(\bar{\mathbf{s}}_T) = \Psi_T(\mathbf{s}_T)$ for every $T \subseteq N$ that does not intersect Q. $\square$

The algorithm proposed in this work (Algorithm 1, presented in Section 4) extends the approach of [Caragiannis *et al.*, 2017] for P_d–FLIP games to the broader class $\mathtt{PG}^{++}$. When applied to a game $\mathcal{G}$, the algorithm deterministically schedules moves by groups of at most k players, starting from an initial state provided as input, thereby generating a trajectory that converges to an approximate pure Nash equilibrium.

The algorithm is parameterized by two values, β and σ, which are assumed to be known upper bounds on two key quantities associated with $\mathcal{G}$: the $(1+\gamma, k)$*-stretch*, denoted by $\mathtt{Stretch}^k_{1+\gamma}(\mathcal{G})$, and the *spread*, denoted by $\Gamma(\mathcal{G})$ (these two quantities are formally defined and analyzed later in this section). The approximation guarantee achieved by the algorithm depends on β, whereas both β and σ affect the length of the resulting trajectory. Applying the algorithm to a specific class of games requires selecting suitable values of β and σ, obtained from upper bounds on the corresponding stretch and spread. Some examples of such bounds will be discussed later in this section. The following theorem formalizes the above.

Theorem 3. *For every input consisting of a potential game $\mathcal{G} = \langle N, \mathcal{I}, (\Psi_P)_{P\in\mathcal{I}}, (S_i)_{i\in N}, (u_i)_{i\in N}\rangle \in \texttt{PG}^{++}$, an initial state $\mathbf{s}^I \in S_N$, an integer $k \geq 1$, and an accuracy parameter $\gamma \in (0, 1]$, Algorithm 1, instantiated with parameters $\beta, \sigma \geq 1$ such that $\beta \geq \texttt{Stretch}^k_{1+\gamma}(\mathcal{G})$ and $\sigma \geq \Gamma(\mathcal{G})$, constructs a $(1 + \gamma)$-improving trajectory from $\mathbf{s}^I$ to a $(\beta + 3\gamma)$-approximate pure Nash equilibrium. The length of the trajectory is polynomial in β, σ, and the logarithm of the maximum payoff attainable by any player.*

Theorem 3 guarantees that the final state returned by Algorithm 1 is an approximate pure Nash equilibrium. In fact, the algorithm ensures a stronger form of stability against coordinated moves by groups of at most k players. Formalizing this stronger guarantee requires introducing the notion of *α-approximate k-optimal strategy profile*, which is also central to the notion of stretch introduced later in this section.

α-Approximate k-Optimal Strategy Profiles. Consider a potential game $\mathcal{G} = \langle N, \cdot, \cdot, (S_i)_{i\in N}, (u_i)_{i\in N}\rangle \in \texttt{PG}^{++}$. For every non-empty $P \subseteq N$, $\mathbf{y} \in S_{-P}$, $\alpha \geq 1$, and integer $k \geq 1$, a strategy $\mathbf{x} \in S_P$ for P is called an *α-approximate k-optimal strategy* (given $\mathbf{y}$) if no k-group $Q \subseteq P$ has an α-improving move from the state $[\mathbf{x}, \mathbf{y}]$, i.e., for every $\mathbf{s}' \in S_Q$, $u_Q([[\mathbf{x}, \mathbf{y}]_{-Q}, \mathbf{s}']) \leq \alpha u_Q([\mathbf{x}, \mathbf{y}])$. If $\alpha = 1$ and $k \geq |P|$, then $\mathbf{x}$ is simply called an *optimal strategy* (given $\mathbf{y}$), i.e., $\mathbf{x} \in \arg\max_{\mathbf{z}\in S_P} u_P([\mathbf{z}, \mathbf{y}])$. The set of all α-approximate k-optimal strategies for P given $\mathbf{y} \in S_{-P}$ is denoted by $\mathcal{S}^k_\alpha(\mathcal{G}, P \mid \mathbf{y}) \subseteq S_P$. The set of all optimal strategies for P given $\mathbf{y} \in S_{-P}$ is denoted by $\mathcal{O}(\mathcal{G}, P \mid \mathbf{y}) \subseteq S_P$. To simplify the notation, when $P = N$, $\mathcal{S}^k_\alpha(\mathcal{G})$ is used in place of $\mathcal{S}^k_\alpha(\mathcal{G}, N \mid\perp)$ and denotes the set of α-approximate k-optimal states of $\mathcal{G}$; similarly, $\mathcal{O}(\mathcal{G})$ is used in place of $\mathcal{O}(\mathcal{G}, N \mid\perp)$ and denotes the set of optimal states of $\mathcal{G}$.

We are now ready to formalize the stronger stability guarantee achieved by Algorithm 1.

Theorem 4. *The final state of the trajectory produced by Algorithm 1 is a $(\beta + 3\gamma)$-approximate k-optimal strategy profile of $\mathcal{G}$.*

The remainder of this section is devoted to formally defining the notions of stretch and spread, and to deriving upper bounds on these quantities.

We begin with the notion of (α, k)-stretch. Informally, it captures the worst-case ratio between the maximum potential of the game and the potential of an α-approximate k-optimal strategy profile.

Definition 4. *Let $\mathcal{G} = \langle N, \cdot, \cdot, (S_i)_{i\in N}, (u_i)_{i\in N}\rangle \in \texttt{PG}^{++}$. For every non-empty $P \subseteq N$, $\mathbf{y} \in S_{-P}$, $\alpha \geq 1$, and integer $k \geq 1$, define*

$$\Lambda^k_\alpha(\mathcal{G}, P \mid \mathbf{y}) = \max_{\mathbf{x}\in\mathcal{S}^k_\alpha(\mathcal{G},P\mid\mathbf{y})} \frac{u_P([\mathbf{x}^*, \mathbf{y}])}{u_P([\mathbf{x}, \mathbf{y}])}, \qquad (3)$$

where $\mathbf{x}^*$ is any strategy in $\mathcal{O}(\mathcal{G}, P \mid \mathbf{y})$. The (α, k)-stretch of $\mathcal{G}$ is defined as

$$\texttt{Stretch}^k_\alpha(\mathcal{G}) = \max_{\substack{P\in 2^N_+\\ \mathbf{y}\in S_{-P}}} \Lambda^k_\alpha(\mathcal{G}, P \mid \mathbf{y}). \qquad (4)$$

In Theorem 5, we establish an upper bound on $\texttt{Stretch}^k_{1+\gamma}(\mathcal{G})$ in terms of the number of players, the degree d of the game, and the parameters k and γ, for every $k \geq d \geq 2$ and $\gamma \geq 0$. The assumption $k \geq 2$ is essential, since no finite upper bound exists when $k = 1$ (as shown in Example 2) unless additional restrictions are imposed on the class of games. This explains why moves by groups of size at least two are necessary to obtain meaningful approximation guarantees.

The proof of Theorem 5 closely follows the analysis of [Pearce and Tambe, 2007] for bounding the quality of k-optimal solutions in distributed constraint optimization problems (DCOPs). This connection highlights a strong parallel between approximate equilibria in potential games and locally optimal solutions in DCOPs.

The bound established in the theorem is expressed in terms of the falling factorial, defined as follows. For every integer $t \geq 1$ and every real $x \geq 0$, $x^{\underline{t}} = x(x-1)(x-2)\cdots(x-t+1)$ if $x \geq t$, and $x^{\underline{t}} = 0$ otherwise.

Theorem 5. *Let $\mathcal{G} = \langle N, \cdot, \cdot, \cdot, \cdot\rangle \in \texttt{PG}^+$ be a potential game of degree $d \geq 2$. For every $\gamma \geq 0$ and integer $k \geq d$, we have*

$$\texttt{Stretch}^k_{1+\gamma}(\mathcal{G}) \leq \frac{(1+\gamma)|N|^{\underline{d}} - (|N| - k)^{\underline{d}}}{k^{\underline{d}}}.$$

The upper bound in Theorem 5 is tight, as follows from Proposition 2 of [Pearce and Tambe, 2007], which provides a matching lower bound for k-optimal solutions in DCOPs.

We now specialize Theorem 5 to the case $d = 2$, obtaining the following bound.

Corollary 1. *Let $\mathcal{G} = \langle N, \cdot, \cdot, \cdot, \cdot\rangle \in \texttt{PG}^+$ be a potential game of degree 2. For every $\gamma \geq 0$ and integer $k \geq 2$, we have*

$$\texttt{Stretch}^k_{1+\gamma}(\mathcal{G}) \leq 2|N| - 3 + \gamma|N|(|N| - 1).$$

For completeness, we also recall that, for every P_d–FLIP game $\mathcal{G}$, the quantity $\texttt{Stretch}^1_{1+\gamma}(\mathcal{G})$ is at most $d + 1 + \gamma d$, as shown in [Caragiannis *et al.*, 2017].

Now we define the notion of spread. Its definition relies on two additional quantities, namely the *maximum value* and the *minmax value*. While applied to potential games, these notions are more general: the maximum and minmax values are defined for all strategic games, while the spread applies to all games in $\texttt{SG}^{++}$.

Maximum Value, Minmax Value, and Spread. Consider a strategic game $\mathcal{G} = \langle N, (S_i)_{i\in N}, (u_i)_{i\in N}\rangle \in \texttt{SG}$. For every player $i \in N$, the *maximum value* of i is defined as $u^{\max}_i = \max_{\mathbf{s}\in S_N} u_i(\mathbf{s})$, i.e., the maximum payoff achievable by player i over all strategy profiles.

The *minmax value* of player i, denoted by $\underline{u}_i$, captures the best payoff player i can ensure in the worst-case scenario, where the other players behave adversarially. Formally, $\underline{u}_i = \min_{\mathbf{y}\in S_{-i}} \max_{\mathbf{x}\in S_i} u_i([\mathbf{x}, \mathbf{y}])$.

Finally, assume that $\mathcal{G} \in \texttt{SG}^{++}$. The *spread* of player i is defined as $\frac{u^{\max}_i}{\underline{u}_i}$. Since $\mathcal{G} \in \texttt{SG}^{++}$, the quantity $\underline{u}_i$ is strictly positive, and the ratio is therefore well defined. Moreover, the maximum value can also be written as $u^{\max}_i = \max_{\mathbf{y}\in S_{-i}} \max_{\mathbf{x}\in S_i} u_i([\mathbf{x}, \mathbf{y}])$, which shows that the spread measures how much larger the best-case payoff of player i can be compared to the worst-case guaranteed payoff. The *spread of the game* is then defined as $\Gamma(\mathcal{G}) = \max_{i\in N} \frac{u_i^{\max}}{\underline{u}_i}$.

We conclude the section with two simple examples of upper bounds on the spread. First, consider the game introduced in Example 2. If player 2 selects strategy 0, the maximum payoff achievable by player 1 is m, whereas if player 2 selects 1, the maximum achievable payoff becomes M. Since $M > m$, the spread of player 1 is exactly M/m. By symmetry, the same holds for player 2, and therefore the spread of the game is M/m.

Next, consider an instance $\mathcal{G}$ of the P_d–FLIP games introduced in Example 1. A key property of these games is that, for every player i and every strategy profile $\mathbf{s}_{-i} \in S_{-i}$, $u_i^{\max} \leq u_i([\mathbf{s}_{-i}, \mathtt{true}]) + u_i([\mathbf{s}_{-i}, \mathtt{false}])$. Therefore, for every fixed choice of the other players, at least one of the two strategies guarantees player i a payoff of at least $u_i^{\max}/2$. It follows that every P_d–FLIP game has spread at most 2.

4 Description of Algorithm 1

Algorithm 1 takes as input a potential game $\mathcal{G} \in \mathtt{PG}^{++}$, an initial state $\mathbf{s}^I \in S_N$, an integer $k \geq 1$, and an accuracy parameter $\gamma \in (0, 1]$. It is instantiated with parameters $\beta, \sigma \geq 1$ satisfying $\beta \geq \mathtt{Stretch}^k_{1+\gamma}(\mathcal{G})$ and $\sigma \geq \Gamma(\mathcal{G})$, and returns a $(1 + \gamma)$-improving trajectory τ from $\mathbf{s}^I$ to a $(\beta + 3\gamma)$-approximate pure Nash equilibrium. The length of τ is polynomial in β, σ, and the logarithm of the maximum payoff attainable by any player.

The first part of the algorithm (lines 2–14) partitions the set of players N into m disjoint blocks $B_1, B_2, \ldots, B_m$ according to the values $\hat{u}_i = \max_{\mathbf{x}\in S_i} u_i([\mathbf{s}^I_{-i}, \mathbf{x}])$, computed for every player i in lines 4–6. The quantity $\hat{u}_i$ represents the maximum payoff that player i can obtain by performing a unilateral move from the initial state $\mathbf{s}^I$.

The blocks are defined through a geometrically decreasing sequence of boundaries $b_0, b_1, \ldots, b_m$. Specifically, b_0 is initialized to $\max_{i\in N} \hat{u}_i$ (line 7), and each subsequent boundary is defined by $b_r := \Delta^{-2} b_{r-1}$ (lines 9–11), while b_m is set to 0 (line 12). The parameter Δ is chosen as a function of $|N|$, β, σ, and γ^{-1} (lines 2–3), and the value of m is selected so that the entire range of possible values of $\hat{u}_i$ is covered (line 8).

Each player i is then assigned to the unique block whose interval contains its value $\hat{u}_i$: block B_1 contains the players with $\hat{u}_i \in [b_1, b_0]$ (line 13), while, for every $r \in \{2, \ldots, m\}$, block B_r contains the players with $\hat{u}_i \in [b_r, b_{r-1})$ (line 14).

The second part of the algorithm (lines 15–28) consists of a sequence of m phases. Before the execution of the phases, the current state $\mathbf{s}$ is initialized to $\mathbf{s}^I$ and the trajectory τ to $\langle \mathbf{s}^I \rangle$ (lines 15–16). The phases are defined as follows.

Phase 0 *(lines 17–21).* The algorithm repeatedly searches for a k-group $T \subseteq B_1$ having a $(1 + \gamma)$-improving move. Whenever such a group exists, the strategy of T is updated to an optimal strategy. The phase terminates when no such group remains, in which case the resulting strategy profile is a $(1 + \gamma)$-approximate k-optimal strategy for B_1.

Algorithm 1

Input: $\mathcal{G} = \langle N, \mathcal{I}, (\Psi_P)_{P\in\mathcal{I}}, (S_i)_{i\in N}, (u_i)_{i\in N}\rangle \in \mathtt{PG}^{++}$, $\mathbf{s}^I \in S_N$, an integer $k \geq 1$, a parameter $\gamma \in (0, 1]$;
Output: A trajectory τ of $\mathcal{G}$;

1: Let β and σ be two parameters such that $\beta \geq \mathtt{Stretch}^k_{1+\gamma}(\mathcal{G})$ and $\sigma \geq \Gamma(\mathcal{G})$;
2: $\delta := (\beta + \gamma)(\beta + 2)|N|\gamma^{-1}$;
3: $\Delta := |N|\delta\sigma^2$;
4: **for** $i \leftarrow 1$ **to** $|N|$ **do**
5: $\quad \hat{u}_i := \max_{\mathbf{x}\in S_i} u_i([\mathbf{s}^I_{-i}, \mathbf{x}])$;
6: **end for**
7: $b_0 := \max_{i\in N} \hat{u}_i$;
8: $m := \lceil \log_\Delta b_0 \rceil + 1$;
9: **for** $r \leftarrow 1$ **to** $m - 1$ **do**
10: $\quad b_r := \Delta^{-2} b_{r-1}$;
11: **end for**
12: $b_m := 0$;
13: $B_1 := \{i \in N \mid \hat{u}_i \in [b_1, b_0]\}$;
14: $B_r := \{i \in N \mid \hat{u}_i \in [b_r, b_{r-1})\}$, for every $r \in \{2, \ldots, m\}$;
15: $\mathbf{s} \leftarrow \mathbf{s}^I$;
16: $\tau \leftarrow \langle \mathbf{s}^I \rangle$;
/* *Phase* 0 */
17: **while** $\exists$ a k-group $T \subseteq B_1$ that has a $(1 + \gamma)$-improving move **do**
18: $\quad$ Let $\mathbf{r}^* \in \arg\max_{\mathbf{r}\in S_T} u_T([\mathbf{s}_{-T}, \mathbf{r}])$;
19: $\quad \mathbf{s} \leftarrow [\mathbf{s}_{-T}, \mathbf{r}^*]$;
20: $\quad \tau \leftarrow \tau \oplus \mathbf{s}$;
21: **end while**
/* *Phases* $1, 2, \ldots, m - 1$ */
22: **for** $r \leftarrow 1$ **to** $m - 1$ **do**
23: $\quad$ **while** $\exists$ a k-group T such that **either** $T \subseteq B_{r+1}$ and T has a $(1 + \gamma)$-improving move **or** $T \subseteq B_{r+1} \cup B_r$ and T has a $(\beta + \gamma)$-improving move **do**
24: $\quad\quad$ Let $\mathbf{r}^* \in \arg\max_{\mathbf{r}\in S_T} u_T([\mathbf{s}_{-T}, \mathbf{r}])$;
25: $\quad\quad \mathbf{s} \leftarrow [\mathbf{s}_{-T}, \mathbf{r}^*]$;
26: $\quad\quad \tau \leftarrow \tau \oplus \mathbf{s}$;
27: $\quad$ **end while**
28: **end for**
29: **return** τ;

Phase $r \in \{1, \ldots, m-1\}$ *(lines 23–27).* The algorithm considers the two consecutive blocks B_r and B_{r+1} and repeatedly performs the following updates. If there exists a k-group $T \subseteq B_{r+1}$ having a $(1 + \gamma)$-improving move, or a k-group $T \subseteq B_r \cup B_{r+1}$ having a $(\beta + \gamma)$-improving move, then the strategy of T is updated to an optimal strategy. At the end of phase r, the resulting strategy profile is a $(\beta+\gamma)$-approximate k-optimal strategy for $B_r \cup B_{r+1}$ and a $(1 + \gamma)$-approximate k-optimal strategy for B_{r+1}.

References

[Babichenko and Tamuz, 2016] Yakov Babichenko and Omer Tamuz. Graphical potential games. *Journal Economic Theory*, 163:889–899, 2016.

[Balcan *et al.*, 2013] Maria-Florina Balcan, Avrim Blum, and Yishay Mansour. Circumventing the price of anarchy: Leading dynamics to good behavior. *SIAM Journal of Computing*, 42(1):230–264, 2013.

[Bhalgat *et al.*, 2010] Anand Bhalgat, Tanmoy Chakraborty, and Sanjeev Khanna. Approximating pure Nash equilibrium in cut, party affiliation, and satisfiability games. In *Proceedings of the 11th ACM Conference on Electronic Commerce (EC)*, pages 73–82, 2010.

[Caragiannis and Fanelli, 2019] Ioannis Caragiannis and Angelo Fanelli. On approximate pure Nash equilibria in weighted congestion games with polynomial latencies. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 133:1–133:12, 2019.

[Caragiannis and Jiang, 2023] Ioannis Caragiannis and Zhile Jiang. Computing better approximate pure Nash equilibria in cut games via semidefinite programming. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC)*, pages 710–722, 2023.

[Caragiannis *et al.*, 2011] Ioannis Caragiannis, Angelo Fanelli, Nick Gravin, and Alexander Skopalik. Efficient computation of approximate pure Nash equilibria in congestion games. In *Proceedings of the 52th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 532–541, 2011.

[Caragiannis *et al.*, 2015] Ioannis Caragiannis, Angelo Fanelli, Nick Gravin, and Alexander Skopalik. Approximate pure Nash equilibria in weighted congestion games: Existence, efficient computation, and structure. *ACM Transactions on Economics and Computation*, 3(1):2:1–2:32, 2015.

[Caragiannis *et al.*, 2017] Ioannis Caragiannis, Angelo Fanelli, and Nick Gravin. Short sequences of improvement moves lead to approximate equilibria in constraint satisfaction games. *Algorithmica*, 77(4):1143–1158, 2017.

[Christodoulou *et al.*, 2012] George Christodoulou, Vahab S. Mirrokni, and Anastasios Sidiropoulos. Convergence and approximation in potential games. *Theoretical Computer Science*, 438:13–27, 2012.

[Fabrikant *et al.*, 2004] Alex Fabrikant, Christos H. Papadimitriou, and Kunal Talwar. The complexity of pure nash equilibria. In László Babai, editor, *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 604–612, 2004.

[Feldotto *et al.*, 2017] Matthias Feldotto, Martin Gairing, Grammateia Kotsialou, and Alexander Skopalik. Computing approximate pure Nash equilibria in Shapley value weighted congestion games. In *Proceedings of the 11th International Conference on Web and Internet Economics (WINE)*, pages 191–204, 2017.

[Giannakopoulos *et al.*, 2022] Yiannis Giannakopoulos, Georgy Noarov, and Andreas S. Schulz. Computing approximate equilibria in weighted congestion games via best-responses. *Mathematics of Operations Research*, 47(1):643–664, 2022.

[Hammersley and Clifford, 1971] John Michael Hammersley and Peter Clifford. Markov fields on finite graphs and lattices. 1971.

[Johnson *et al.*, 1988] David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37(1):79–100, 1988.

[Koller and Friedman, 2009] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models - Principles and Techniques*. MIT Press, 2009.

[Li and Marden, 2013] Na Li and Jason R. Marden. Designing games for distributed optimization. *IEEE Journal of Selected Topics in Signal Processing*, 7(2):230–242, 2013.

[Marden *et al.*, 2009] Jason R. Marden, Gürdal Arslan, and Jeff S. Shamma. Cooperative control and potential games. *IEEE Transactions on Systems, Man, and Cybernetics*, 39(6):1393–1407, 2009.

[Monderer and Shapley, 1996] Dov Monderer and Lloyd S. Shapley. Potential games. *Games and Economic Behavior*, 14(1):124–143, 1996.

[Pearce and Tambe, 2007] Jonathan P. Pearce and Milind Tambe. Quality guarantees on k-optimal solutions for distributed constraint optimization problems. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1446–1451, 2007.

[Ui, 2000] Takashi Ui. A Shapley value representation of potential games. *Games and Economic Behavior*, 31(1):121–135, 2000.

[Vijayalakshmi and Skopalik, 2020] Vipin Ravindran Vijayalakshmi and Alexander Skopalik. Improving approximate pure nash equilibria in congestion games. In *Proceedings of the 16th International Conference on Web and Internet Economics (WINE)*, pages 280–294, 2020.