Efficient Stackelberg Strategies for Finitely Repeated Games

Natalie Collina
University of Pennsylvania
Philadelphia, United States
ncollina@seas.upenn.edu

Eshwar Ram Arunachaleswaran
University of Pennsylvania
Philadelphia, United States
eshwar@seas.upenn.edu

Michael Kearns
University of Pennsylvania
Philadelphia, United States
mkearns@seas.upenn.edu

ABSTRACT

We study Stackelberg equilibria in finitely repeated games, where the leader commits to a strategy that picks actions in each round and can be adaptive to the history of play (i.e. they commit to an algorithm). In particular, we study static repeated games with no discounting. We give efficient algorithms for finding approximate Stackelberg equilibria in this setting, along with rates of convergence depending on the time horizon T. In many cases, these algorithms allow the leader to do much better on average than they can in the single-round Stackelberg. We give two algorithms, one computing strategies with an optimal $\frac{1}{T}$ rate at the expense of an exponential dependence on the number of actions, and another (randomized) approach computing strategies with no dependence on the number of actions but a worse dependence on T of $\frac{1}{T^{0.25}}$. Both algorithms build upon a linear program to produce simple automata leader strategies and induce corresponding automata best-responses for the follower. We complement these results by showing that approximating the Stackelberg value in three-player finite-horizon repeated games is a computationally hard problem via a reduction from balanced vertex cover.

KEYWORDS

Equilibrium Computation, Stackelberg Games, Repeated Games

ACM Reference Format:
Natalie Collina, Eshwar Ram Arunachaleswaran, and Michael Kearns. 2023. Efficient Stackelberg Strategies for Finitely Repeated Games. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 9 pages.

1 INTRODUCTION

The central solution concept in games is the Nash Equilibria (NE), where each agent is playing optimally against the mixed strategy of the other agents. NE are known to always exist in finite games [22], and the problem of computing them has been widely studied, with efficient algorithms developed for many special classes (See [23]), and hardness results for general two player games (See [10], [12], [24]).

There are many natural variants of this problem. A solution concept of particular interest is the Stackelberg Equilibrium (SE) [16], in which one of the players, called the leader, is allowed to commit to a strategy first, and communicates this strategy to the other players. This problem was first studied through a computational lens by Conitzer and Sandholm [11], who showed efficient algorithms in the case of two players, and hardness results for three player games, and has since been the subject of many other works, including for settings such as security games (See [1]).

In another parallel line of work, repeated games have been studied as a natural generalization of the single-shot, simultaneous game — agents repeatedly play the same game over multiple rounds, where the outcome in each round depends only upon the actions played in that round. A critical point of interest about these games is that the actions employed in each round may depend arbitrarily upon the history of play and the internal state of each player.

Littman and Stone [20] gave an efficient algorithm for computing finite automata Nash equilibrium strategies achieving any given feasible payoff profile in infinitely repeated games. In more recent work of Zhou and Tang [28], an analogous result was given for infinitely repeated Stackleberg games, again providing an efficient algorithm for computing finite automata strategies (and a generalization to automata with infinite states) for both the leader and follower.

Both of these works focus solely on *infinitely* repeated games. Reasoning about equilibria in finite-horizon games introduces many complexities not present in the infinite-horizon case. In infinite-horizon equilibria, any finite prefix of play can be ignored, and players can have threats of unbounded length. This is not true in finite-horizon games, and furthermore finite-horizon equilibria are sensitive to order of play and “final-round” complexities.

Benoit and Krishna [3] tackle the finite-horizon setting by providing a folk theorem for Subgame-Perfect Nash equilibria (SPNE) for finitely repeated games. While the authors do not discuss Stackelberg games in their work, their results have an implicit connection to SE computation. [3] shows that, for many static, finitely repeated games, any individually rational and feasible payoff pair in the game matrix can be approximated by an SPNE in the repeated game. Given any such payoff pair, they give a method for constructing a SPNE that converges to this payoff value. As we show in our work, the payoff pair of any SE is indeed an individually rational and feasible in the game matrix. Thus, for games which satisfy [3]’s specifications, one could optimize over the convex set of feasible and individually rational payoff pairs (for the payoff of the leader). One could then use their method to construct a pair of algorithms according to this value pair. The leader could then commit to their side of the SPNE, which will converge to the Stackelberg value.

While this observation is a promising sign that SE computation in finitely repeated games may be tractable, the above approach leaves a few things open. First of all, it will only work for a subset of repeated games; [3]’s construction requires that all players have an equilibrium with a value above their minimax value. This excludes a large class of games with interesting finitely-repeated

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

SE, such as Prisoner's Dilemma [1]. Second, [3]'s result focuses on characterizing the set of SPNE and does not provide any guarantee on the rate of convergence. In particular, they prove that there *exists* a T_0 for which their construction will converge to the desired approximation, but they provide no upper bound on how large this T_0 needs to be. Thus, there are no guarantees for the approximation rate of their construction for any fixed horizon T, which makes it infeasible to reason about any particular finite-horizon game.

In our work, we provide the first algorithm that computes SE for *all* finitely repeated games, including games with nontrivial SE which do not fulfill Benoit and Krishna's specifications, such as Prisoner's Dilemma. In addition, our algorithm constructing commitments has an approximation rate for every fixed T. Our results specifically search beyond just the set of SPNE, and we explicitly use the power of the leader to commit to playing sub-optimally in later rounds, something which is not possible in a SPNE (See Section 3.1 for an example).

Like previous papers on repeated games, the players in our game are allowed to employ a broad class of algorithms (essentially any finite time algorithm) to decide their strategies in each round based upon this history of play and their internal state. This allows for a large space of commitment schemes; for example, the leader can employ "threats" such as the grim-trigger strategy in the repeated Prisoner's Dilemma. We summarize our main results below:

- We give two efficient algorithms that find additively approximate Stackelberg strategies for the leader that trade off different approximation factors. First, we show a $\left(\frac{2^{\text{poly}(n)}}{T}\right)$- additive approximate Stackelberg strategy where n is the number of actions. This is constructed using an algorithm that runs in polynomial time in the game size and $\log T$. We also present an approach using randomized sampling which gives a $O(\frac{1}{T^{0.25}})$-additive approximation that runs in time polynomial in the game size and T [2]. This second approach provides a slightly weaker approximation factor in T in exchange for entirely avoiding dependence on n. Both constructions use simple automata (including states that play distributions over actions) for the leader and also for a corresponding follower best-response. We leave it as an open question whether there is an algorithm combining the best of both rates.
- Building upon a reduction of Conitzer and Sandholm [11] that showed that finding the Stackleberg value of a three player game is NP-hard, we show that even approximating the Stackelberg value of a three player repeated game upto an additive factor of $\frac{1}{T^{1/k}}$ is NP-hard (Theorem 3) for any integer k, ruling out any extension of our results for two player repeated games. We also use the same reduction to observe that approximating the Stackelberg value of three-player infinitely repeated games is NP-hard. These results can be found in Section 5.

We also show a sequence of auxiliary results, that we discuss briefly here to help the reader to navigate through the paper.

- We show that finding a follower's best-response even to a poly-time leader strategy is in general a computationally hard problem However, our approximate Stackelberg leader algorithm (Theorem 2) has an efficiently computable best-response We discuss these results in the full version.
- We show that the optimal Stackelberg leader strategy may be different from simply playing the single-shot game's Stackelberg strategy in each round of play, through the example of the well known Prisoner's Dilemma (Section 3.1).
- No-Regret algorithms are widely studied in the context of repeated games, due to their strong performance and strategic properties (See [6], [13]). A natural question is whether no-regret algorithms are in fact optimal leader algorithms in the Stackelberg setting. We show that, somewhat unsurprisingly, arbitrary no-regret algorithms are in fact Stackelberg leader algorithms in two player zero-sum games. In contrast, we show via a counterexample that these properties do not hold when the zero-sum condition is removed, necessitating the broadening of the search to a more general class of algorithms. These results can be found in the full version.
- Further, we generalize this result by showing that the class of learning algorithms (i.e. those algorithms that do not assume initial knowledge of the matrix, but learn the payoffs of each action after each round of play) does not always contain the optimal Stackelberg strategy for the leader. In the full version, we show a simple pricing game where learning algorithms are strictly sub-optimal for the leader to commit to.

Our Techniques. In proving our main result (Theorem 2) about efficiently computing a near optimal Stackelberg leader algorithm, we use a sequence of ideas. Our main workhorse is a linear program (similar in spirit to that of [28]) that is used to upper bound the average (per round) payoff that the leader can guarantee themselves against a rational follower. This linear program additionally gives us an "ideal" empirical distribution over the action pairs that the transcript of play must follow to achieve the aforementioned value. We then convert this ideal empirical distribution into a concrete transcript of play for both the leader and follower. Akin to [20], [28] and [3] we use the notion of "threats" to stabilize such a transcript so as to force the follower to play along with it. Our construction carefully choreographs the order of action pairs in this transcript to ensure that the follower's best response is to follow the prescribed transcript.

One might ask why we explicitly prescribe these ordered pairs, instead of allowing the leader to draw pairs from the LP distribution i.i.d. each round. The problem is that even if the leader commits to drawing from this distribution and playing their side of the strategy, they have no way to signal to the follower what they should play at any given round before play occurs. Our approach, in spirit, involves the leader pre-committing to these signals at the start of the game. However, this means that the follower can see the payoffs of the specific move pairs they will be asked to play each round (as opposed to an expected payoff over an i.i.d. draw), which necessitates the ordering of pairs by increasing follower payoff. The transcript we construct forces through the worst action pairs for the follower first, and adds "treats" at the end to ensure that the follower is incentivized to follow the prescription at each round

[1] The unique SPNE for Prisoner's dilemma is both players defecting in every round of play, via backward induction. Interestingly, we show in Section 3.1 that the Stackelberg value of the finitely repeated Prisoner's Dilemma is strictly larger than the value from the unique SPNE

[2] The approximation factors also depend upon the maximum number of bits needed to represent a single payoff.

of the game. The approximation to the upper bound comes from "fitting" a probability distribution to a finite number of rounds T, and improves as T grows larger. While the average values of the leader and follower in the transcript will approximate their value in the LP distribution for large T, the ordering means that the actual transcript will not look like a typical sequence of i.i.d. draws from the distribution.

While the above approach gives us a rate of convergence to optimality that depends, possibly exponentially badly, on the number of actions for each player in the game, we show in the full version how to eliminate the dependence upon the number of actions. The main idea involved is a randomized sampling scheme that allows us to simplify the ideal empirical distribution found by the linear program. This process can also be viewed as randomized rounding of the linear program's solution to a more succinctly representable solution. This rounding also introduces randomization in the leader's algorithm, and converts the absolute guarantee of near-optimality into a high probability guarantee.

In Theorem 3, where we show the hardness of approximation of computing the Stackelberg value of repeated three player games, we demonstrate a gap-producing reduction from the balanced Vertex Cover problem. This reduction builds upon [11], which showed that finding the Stackelberg value of three-player games is NP-hard also via reduction from balanced Vertex Cover. Their hard instance does not directly extend to repeated games because of the richness of new strategies available to the leader(s) in the repeated setting. Most of the work in our reduction goes into proving a stronger result about the single-shot case, which we do in the full version. We can then lift this stronger result to the game with T repeated rounds and prove hardness of approximation in Theorem 3. We also observe that the same lemma can be used to show that finding the approximate Stackelberg value of three-player infinitely repeated games is NP-hard. In the interest of spaces, many proofs have been deferred to the full version of the paper.

2 RELATED WORK

The notion of Stackelberg Equilibria (SE) was first introduced by von Stackelberg in [16], who observed the advantage of strategic commitment in Cournot's duopoly model. Some followup work in the next decades did focus on applying the concept to repeated games, but these works focused on proving necessary and sufficient conditions for SE existence in various settings, rather than on constructive algorithms. [26] explores SE in repeated games where the payoff matrix is either static or dynamic. They consider games where players have have an infinite number of moves (a setting where even Nash Equilibria may not exist) and derive conditions for SE existence. [8] consider *sequential* Stackelberg games, repeated games where the state of the game matrix is affected by the moves that the players play each round. This setting is distinct from our own in that the leader makes a commitment each round instead of committing to an algorithm at the start of the game.

[3] study Subgame Perfect Nash Equilibria (SPEs) in finitely repeated games and show that for a subset of these games, any feasible, individually rational pair of payoffs can be approximated to arbitrary precision for large enough time horizons. As discussed in the introduction, this work implies an algorithm for finding SE in finitely repeated games, but only for a subclass of games and without convergence bounds.

In [11], Conitzer and Sandholm gave a polynomial algorithm for computing exact Stackelberg mixed strategies in single-shot, two-player games, bringing new algorithmic interest to the study of computing SE. Their method involves solving n linear programs, with the ith program representing the best payoff that the leader can get given that they are incentivizing the follower to play pure strategy i. Picking the highest payoffs over all of these LPs gives the leader their Stackelberg strategy for a single-shot game. They go on to prove that computing an exact Stackelberg strategy in a three-player game is NP-hard, via a reduction from Vertex Cover. A closely related work is [27], which proves various properties about single-shot SE games with convex strategy sets. These works led to further research considering the complexity of computing SE in various settings. The most closely related followup work involves NP-hard results for stochastic and extensive-form games, which are superclasses of finite-horizon repeated games, and a poly-time algorithm for *infinite*-horizon repeated games. These results do not imply our results; more discussion on this can be found in the full version.

One recent and closely related line of work investigates *learning* Stackelberg strategies in static repeated games. In the learning setting, the Stackelberg leader begins with no knowledge of the payoff matrix, and only gains information via seeing their own payoff (and the opponent's move) after each round. Most works in this space assume that the opponent is responding myopically each round [2], [21], [18], and thus the leader's goal is to learn and begin to play their single-round Stackelberg strategy. More similarly to our work, [14] instead studies a non-myopic (but discounting) follower and a learning leader, and develops a leader algorithm via a reduction to bandit optimization in the presence of myopic agents. As with previous papers in the learning setting, their algorithms use the single-round Stackelberg value as a benchmark.

Like [14], our paper assumes a non-myopic follower, though without discounting. In sharp contrast to their work, the leader in our model has full access to the entire payoff matrix at the start of the game. Stackelberg strategies in the full information setting are substantially different from those in the learning setting. To emphasize this point, we show a simple repeated pricing game (with no discounting) in which the Stackelberg value of the leader in the full information setting is strictly larger than the Stackelberg value of a leader in the learning setting. This can be found in the full version.

There is also work which reasons directly about the repeated-game setting, but considers the substantially distinct case of infinite-horizon repeated games. Zuo and Tang give a poly-time algorithm for computing SE in the space of machine strategies, which are automata that may have infinite states [28], and generalize their results to probability distributions over subclasses of these machine strategies. Similar to our work, [28] solve an LP which optimizes the leader's payoff when constrained to offer the follower better than their safety value (called a security level in their paper). However, their proof heavily relies upon the infinite nature of the game. They present a Folk Theorem for infinite-horizon repeated games using machine strategies and use this to construct optimal commitments in the form of state machines. These state machines may have an infinite number of states, and therefore they do not explicitly represent them but rather show the existence of a Turing Machine which can do so. They also show additional results that examine how limiting the memory level of the leader and follower's machine strategies can affect game outcomes. Their work can be seen as a Stackelberg analogue of Littman and Stone's paper, [20]. Both works examine infinite-horizon repeated games, and both utilize constructive versions of Folk Theorems in order to construct strategies (in the space of algorithms) that are in (the respective type of) equilibrium.

Work in finding equilibria in infinite-horizon repeated games cannot be directly applied to finite-horizon repeated games, as the algorithms for these two settings are structurally incomparable. While algorithms for infinite-horizon games have no sense of game length or game ending, algorithms for finite-horizon games can be constructed with the game length in mind. An algorithm for a finite-horizon repeated game can therefore behave differently, for example, in the final move of a game, or operate entirely differently depending on the initial game length.

From a hardness perspective, Conitzer and Sandholm [11] showed that finding the Stackelberg value of a three player game is NP-hard. Conitzer and Letchford [19] showed that finding the Stackelberg value of a three player extensive-form game is NP-Hard. We extend the former and generalize the latter result by proving hardness of approximation for finite-horizon repeated three-player games. Borgs et al. [7] showed that no efficient algorithms exist for finding Nash Equilibrium in three player infinite-horizon repeated games unless PPAD is in P when players use algorithms that depend only upon the history of play (and not upon internal state). Halpern et. al. [15] later showed that this hardness result can be surmounted if the players are allowed to use (probabilistic polynomial time) algorithms that depend both upon the history of play as well as internal state (by demonstrating an efficient algorithm that finds a ε-approximate equilibrium). Their model is an infinite-round analogue of our model (and that of [19] for EFGs), which also allows the player's algorithms to generate actions using randomness, history of play and internal state.

In the context of playing repeated games using algorithms, we also mention no-regret algorithms, which were developed for decision making in online environments with experts (See [9]). Multiple papers (See [13], [4]) observed that these algorithms can be used by self interested agents to play repeated games due to the guarantees associated with them. In particular, in two player zero-sum games, the transcript of play converges to a Nash Equilibrium if all players use a no-regret algorithm, and to coarse correlated equilibria in general games. For special classes of games, such as atomic routing games/ congestion games ([5], [17]) these are particularly compelling algorithms to use, since they guarantee convergence to Nash Equilibria and stability. However, as we show in the full version, no-regret algorithms are not in general good algorithms for a leader to commit to in the Stackelberg setting.

We summarize the main results in the field of finding equilibria in single-shot games, finite-horizon repeated games, and infinite-horizon repeated games in Table 1. Our new contributions are **bolded and in blue**.

3 NOTATION AND PRELIMINARIES

Definition 1 (Bimatrix Games). *A bimatrix game G is defined by two $n \times n$ matrices M_1 and M_2 that respectively denote the payoffs for Player 1 and Player 2 for each combination of actions chosen by the two players. We assume, without loss of generality, that both players have the same number of actions, and that their respective action sets are both indexed by the set $\{1, 2, \cdots n\}$ denoted as $[n]$. If Player 1 plays action i and Player 2 plays action j, then they get payoffs $M_1(i, j)$ and $M_2(i, j)$. In the asymmetric Stackelberg setting, Player 1, or the row player, is assumed to go first and is referred to as the leader while Player 2, or the column player, is called the follower. The strategies for Players 1 and 2 are points x, y in $\Delta^n := \{x \in \mathbb{R}_+^n : \sum_i x_i = 1\}$ denoting the weights they place on the actions in their respective action sets. A pure strategy $i \in [n]$ is represented by the standard basis vector e_i. The expected payoff of Player 1 (respectively 2) is $x^\top M_1 y$ (respectively $x^\top M_2 y$). We assume that each entry in the payoff matrix is in $[-1, 1]$ and is additionally an integer multiple of $\frac{1}{A}$, where A is a sufficiently large integer. A can be seen as a measure of the granularity of the payoff values and will become relevant when defining our approximation guarantees.*

Definition 2 (Nash Equilibria in Single-Shot Games). *A pair of strategies $x, y \in \Delta^n$ form a Nash Equilibrium for the game G if there is no incentive for either player to deviate unilaterally, i.e. for all pure strategies e_i (e_j) of Player 1 (2), we have $x^\top M_1 y \geq e_i^\top M_1 y$ ($x^\top M_2 y \geq x^\top M_2 e_j$).*

Definition 3 (Stackelberg Equilibria in Single-Shot Games). *A strategy $x \in \Delta^n$ is said to be a Stackelberg strategy for the leader if it maximizes $x^\top M_1 y$ where $y \in \Delta^n$ is a best response for the follower to x. Note that it suffices for the follower to play a pure strategy e_j as a best response. The induced play (x, y) is called the Stackelberg Equilibria (SE) of the game. For the purposes of this paper, we assume that the follower selects a best possible best response for the leader if there are multiple best response strategies, a common assumption in Stackelberg literature. SE with this sort of tiebreaking assumption are known as* Strong Stackelberg equilibria, *but for simplicity we will simply refer to* SE *throughout this paper.*

Definition 4 (Repeated Bimatrix Game). *A repeated bimatrix game is defined by a bimatrix game G and a time horizon T, both of which are known to all players. Players will play the game G for exactly T rounds, and their payoffs will be the sum of their payoffs over all rounds. The payoff matrix will remain static throughout all rounds. However, players may play different mixed strategies on different rounds and dynamically update their strategies based upon previous rounds of play. We refer to the pairs of actions $(i_1, j_2); (i_2, j_2), \cdots (i_T, j_T)$ played in the T rounds as the transcript of the game and use $\mathcal{T}$ to represent this transcript. The values of interest to us are the* **expected average** *per-round payoffs of the two players where the expectation is over the random bits used by the players. If $T = \infty$, then we call this an* infinite-horizon *game. For much of this paper, we will consider finite T, in which case we call this a* finite-horizon *game.*

It is obvious that players in a finite-horizon repeated game have a much richer action space than in a single-shot game, since their actions can depend upon the outcomes of the previous rounds, and even upon their internal states (possibly influenced by the random

Table 1: Summary of known results

	Single-shot	Finite Horizon	Infinite Horizon
2-player NE	Hardness of $o(1)$ - approximation [24]	Poly-time algorithm for Convergence to every SPNE [3], Hardness of $o\left(\frac{1}{T}\right)$- approximation[1]	Poly-time algorithm for exact solution [20]
3-player NE	Hardness of $o(1)$-approximation [24]	Hardness of $o\left(\frac{1}{T}\right)$-approximation[1]	Hardness of exact solution [7], hardness of $\frac{1}{poly(n)}$-approximation with no internal states [7]
2-player SE	Poly-time algorithm for exact solution [11]	**Poly-time** $\min\left(\left(\frac{2^{\mathbf{poly}(n)}}{T}\right), \frac{1}{T^{0.25}}\right)$ - **approximation algorithm** (Thm 2)	Poly-time algorithm for exact solution [28]
3-player SE	Hardness of $o(1)$- approximation [11][2]	**Hardness of** $\left(\frac{1}{T^{\frac{1}{c}}}\right)$- **approximation** (Thm 3)	**Hardness of** $\left(\frac{1}{poly(n)}\right)$- **approximation** (Full Version) [3]

[1] These results come from the fact that, in finite-horizon repeated games, any pair of NE strategies must involve always playing a single-shot NE on the final round. The single-shot hardness result can therefore be used to implies that getting a $o(\frac{1}{T})$ approximation over T rounds is PPAD-hard.
[2] While their paper does not explicitly claim such a result in a lemma, their proof for hardness of finding an exact SE is in fact a hardness result for finding any approximate SE.
[3] We do not provide a complete proof of this result, as the model we developed for this paper only makes sense for the finite-horizon case, but we provide a proof sketch and intuition.

coins flipped in previous rounds). Therefore, we need to formalize exactly what this action spaces consists of – in our paper, the players are assumed to delegate their play to algorithms, possibly randomized, that make their choices for them. We offer a general definition below.

Definition 5 (Game Playing Algorithm). *A Game Playing Algorithm or GPA for a finite-horizon repeated game (G, T) is defined as a randomized algorithm that, in each round, takes as input the previous round's action pair and outputs an action for the current round. We allow this algorithm to have memory; in particular, it remembers what it needs to of the previous history of play as well as the random bits used to come up with actions in previous rounds.* [3]. *Implicitly, it computes a distribution in Δ^n over the n actions to be played in the t-th round, and then uses the randomness to select and play a particular action.*

We say that a Game Playing Algorithm is deterministic if it does not use any random bits. It is worth noting that each deterministic GPA can be rewritten as/ shown to be equivalent to a look-up table , implying that operationally, the set of deterministic GPAs is finite.

To give further intuition for the richness of GPAs, consider some arbitrary game where the leader player has two moves, i_1 and i_2, and the follower player has two moves, j_1 and j_2. We provide some mechanics that GPAs support, along with (informal) descriptions of example leader GPAs exhibiting these mechanics:

- History-dependency: play i_1 as long as the follower plays j_1; if the follower ever plays j_2, play i_2 for all remaining rounds.
- Time-dependency: play i_2 on the final round, and i_1 on all other rounds.
- Randomness: flip a new coin each round; play i_1 if heads, and i_2 if tails.
- *Correlated* randomness: flip a coin at the start of the game. If heads, in all T rounds of the game, flip a new coin, and play i_1 if heads and i_2 if tails. Otherwise, in all T rounds of the game, play i_1.

Observation 1. *The set of GPAs can equivalently be seen to be probability distributions over deterministic look-up tables. However, we keep this definition, since it offers a framework for succinct representations for these algorithms (for example - no-regret algorithms).*

We are almost ready to define a notion of SE for repeated bimatrix games. However, it is not a priori obvious that there exists a well defined best-response Game Playing Algorithm for the follower given a fixed GPA for the leader. We show that a best-response GPA is in fact well defined, and that, in fact, it can be found among the set of deterministic (though potentially adaptive) GPAs. This result is analogous to the result in single-shot games that there always exists a best response pure strategy, and in fact uses this connection to work backward from the last round of play. Another way of seeing this result is to observe that finite-horizon repeated games can be rewritten as a large normal form game, implying the result. The proof can be found in the full version.

[3] The fact that the algorithm can store previously used random bits allows for correlations in the probabilistic decisions made across rounds.

Lemma 1. *For any leader GPA by Player 1, there exists a best response Game Playing Algorithm by Player 2 within the set of deterministic lookup table GPAs, which is a finite set. Therefore, a best response is well-defined in the GPA space.*

It is even less apparent that an optimal leader algorithm to commit to is well defined – we provide a constructive proof of existence.

Theorem 1. *There exists an optimal GPA $\mathcal{P}$ for the leader such that no other GPA, when paired up with its corresponding follower best response, gives the leader a better payoff than $\mathcal{P}$ does against the corresponding follower best response.*

The key idea behind this proof is to observe that finite-horizon repeated games can be rewritten as normal form games, albeit at the cost of an exponential blow-up in the number of actions. This is not a new idea; [19] makes the same observation in the context of Extensive-form games, which generalize finite-horizon repeated games. For sake of completeness, we sketch out the proof of this result in the full version.

Definition 6 (Approximate Stackelberg GPA). *Let $P_{\max}$ be the payoff that an optimal leader GPA obtains against a best-responding follower. Then a leader GPA $\mathcal{P}$ is a c-approximate Stackelberg GPA for any $c > 0$ if it obtains payoff at least $P_{\max} + c$. Note that the approximation definition only allows slack in the leader's payoff–we still require that the follower is* exactly *best-responding.*

The precise computational question that we answer affirmatively in this paper is thus as follows: given a two player game G (parameterized by the number of actions n and the granularity A), and a time horizon T, does there exist an efficient algorithm to find an approximate Stackelberg GPA for the leader?

3.1 Separation of Repeated SE from Single-Round SE

Even in a single-shot game, the Stackelberg leader can often expect value higher than that of their best Nash equilibrium. As being a leader in a single-shot game is so powerful, a natural question to ask is if repeated rounds can really strengthen the leader's hand. In particular, one might observe that simply committing to play the single-shot Stackelberg strategy in each round will guarantee the leader the Stackelberg value of the single-shot game on average. We address this question by showing an example where a leader in a finite-horizon repeated game can do strictly better on average than getting their single-shot Stackelberg value each round. For our example, we focus on the simple and well-studied Prisoner's Dilemma game $\mathcal{PD}$. We will set the row player to be the leader and the column player to be the follower. The first move for both players represents cooperating, while the second move represents defecting.

$$\begin{bmatrix} (3,3) & (0,5) \\ (5,0) & (1,1) \end{bmatrix}$$

Lemma 2. *The game $\mathcal{PD}$ repeated for T rounds with $T \geq 3$ exhibits a constant separation between the Stackelberg value and the leader value obtained by playing the single-shot game's Stackelberg strategy in each round.*

The intuition is as follows: in the single-round case, a Stackelberg leader can do no better than get payoff 1, as the follower has a dominant strategy of defecting. However, in a multi-round game, the leader can incentivize the follower to cooperate a majority of the time by promising to occasionally cooperate themselves, and threatening to defect if the follower does not play along. We include a full proof of the above lemma, along with further discussion, in the full version.We also include an explicit construction of an approximate Stackelberg GPA for $\mathcal{PD}$ using our techniques in the full version.

4 ALGORITHMS FOR APPROXIMATE STACKELBERG EQUILIBRIUM

We state the main result of our paper, which gives efficient algorithms for computing approximate Stackelberg GPAs.

Theorem 2. *We give two different efficient algorithms to compute approximate Stackelberg GPAs for a given bimatrix game repeated for T rounds:*

- *An algorithm with running time polynomial in n, $\log A$, $\log T$ that finds a $\left(\frac{2^{poly(n)} \cdot poly(A)}{T}\right)$-approximate Stackelberg GPA.*
- *A randomized algorithm with running time polynomial in n, $\log A$, T that finds a $O\left(\frac{\sqrt{A}}{T^{0.25}}\right)$-approximate Stackelberg GPA (with high probability).*

The proof begins by describing an LP for any bimatrix game which upper bounds the value guaranteed by any leader GPA for any T. Then, we describe a way to construct a GPA using the LP solution such that a best-response by the follower results in a leader payoff closely approximating the LP value. We observe that, while the setup of our LP is different than that in [28], the optimal value is always the same (for a certain regime in their paper). The value of the LP solution in [28] is equal to the value of the optimal SE strategy in infinite-horizon repeated games. Thus, in finding an approximation to the LP upper bound here, we are not only approximating the best Stackelberg strategy for any finite T but also the best Stackelberg straetgy for an infinite-horizon repeated game.

An LP Upper Bound. We describe the construction of the linear program for any bimatrix game. First, we precompute the "threat" value V of the column player along with the associated strategy for the row player x^*, which we call the threat strategy. This is defined as the minimum payoff that the column player can receive by best responding to the row player's strategy in the static setting i.e. $V := \min_{x \in \Delta^n} \max_j x^\top M_2 e_j$ (recall that M_2 is the payoff matrix of the follower/ column player). Note that this is equal to the value of a hypothetical zero-sum game between the two players where the row player's payoff is changed to be the negative of the payoff of the column player, and can hence be computed in polynomial time, as can the associated strategy x^* of the row player. The reason we call this the threat value is that the row player could find a strategy that ensures the column player can do no better than V against this strategy in any given round of play.

Next, we compute the solution of a linear program whose variables are weights $\{\alpha_{i,j}\}_{i,j}$ attached to each action pair in $\{i, j\}^2$.

Intuitively, the point of the linear program is to find a prescribed probability distribution over the action pairs that maximizes the payoff of the leader subject to the follower receiving at least their threat value.

$$\begin{aligned} \max \sum_{i,j} M_1(i,j)\alpha_{i,j} & \qquad \alpha \in \mathbb{R}^n \\ \text{subject to } \sum_{i,j} M_2(i,j)\alpha_{i,j} &\geq V \\ \sum_{i,j} \alpha_{i,j} &= 1 \\ \alpha_{i,j} \geq 0 & \qquad \text{for } i,j \in [n]^2 \end{aligned}$$

We observe that the above LP is feasible. In particular, the distribution over action pairs induced by the threat strategy x^* of the leader and the best response of the follower is by construction a feasible point. Additionally, since the objective describes the payoff generated for the leader by some distribution over the action pairs, the LP is bounded above by the maximum payoff of the leader in the game. Consequently, the LP has a well defined optimum solution, which we also call α with value OPT_{LP}.

Lemma 3. *The optimum value OPT_{LP} of the LP upper bounds the expected per-round payoff of a Stackleberg GPA for any T.*

The key intuition for this proof is that, no matter how intricate GPA behavior may be thanks to randomness or adaptability, every pair of leader and follower GPAs induces *some* distribution over sequences of move pairs. This distribution, which is analogous to the frequencies we solve for in our LP, determines the expected payoff for both players. If the expected payoff for the follower from this move pair distribution is less than their safety value, they are certainly not best responding; thus, for this distribution to be feasible, it must adhere to our LP constraint. Given this constraint, our LP maximizes the payoff for the leader. Therefore, any induced distribution which corresponds to expected payoff for the leader cannot get value above our LP. A full proof is given in Appendix ??.

Construction of a Stackelberg GPA from the LP. A short description of our candidate Stackelberg GPA is as follows — the leader prescribes a sequence $\mathcal{A} = a_1, a_2, \cdots a_T$ of action pairs for both leader and follower to adhere to. In case the follower ever deviates from this prescribed sequence, the leader then plays the threat strategy x^* for the remaining rounds. This sequence is built based upon the LP solution α and has two key properties — first, that the empirical distribution over action pairs in this sequence approximates the distribution α and second, that at any point in the sequence, the follower never gains more by deviating from the sequence than following it.

The LP solution gives us the optimal distribution over action pairs that the follower would be incentivized to play. If the leader and follower had a shared source of randomness, they could draw from this joint distribution each round and preserve both optimality for the leader and incentive compatibility for the follower. However, in our setting, there is no shared source of randomness or any mechanism for the leader to pre-signal to the follower what to play before each round. Therefore in our GPA the leader will prescribe a T-round sequence inspired by the LP solution, which preserves follower rationality while approximating optimality. We call this GPA $\mathcal{P}^*(\alpha)$.

To aid with explicitly describing this sequence, we introduce some notation. We know that there is a polynomial time algorithm for solving a linear program with rational coefficients that outputs a rational solution([25]). All coefficients in our linear program are rational — in addition, the threat value V is also a rational number, since it is itself the optimum of a linear program with rational coefficients. Therefore, we can assume that each $\alpha_{i,j}$ can be written in canonical form as $\frac{p_{i,j}}{q_{i,j}}$ where $p_{i,j}, q_{i,j} \in \mathbb{N}$ and $gcd(p_{i,j}, q_{i,j}) = 1$. For the next step, we re-index the action pairs from 1 to n^2 such that for any two action pair $k_1, k_2 \in [n^2]$ with $k_1 < k_2$, we have $M_2(k_1) \leq M_2(k_2)$. We use this indexing for the action pairs henceforth. Let $N = LCM(q_1, q_2 \cdots q_{n^2})$ and let $T = c \cdot N + r$ where c is a natural number and $r \in \{1, 2, \cdots N-1, N\}$. We assume T is large enough to ensure $c \geq 1$. Additionally, we refer to N as the cycle length, for reasons that will be clear from the full description of the GPA $\mathcal{P}^*(\alpha)$.

For each action pair k, we calculate the number of times n_k to prescribe playing this action pair in the first $c \cdot N$ rounds, $n_k := \alpha_k \cdot cN = c\frac{p_k}{q_k}N$. By definition, N is divisible by q_k and hence n_k is a natural number. Further, note that $\sum_k n_k = \sum_k \alpha_k \cdot cN = (\sum_k \alpha_k) \cdot cN = cN$. The prescribed sequence for the first cN moves is to play action pair 1 for the first n_1 rounds, action pair 2 for the next n_2 rounds and so on until the end of cN rounds. For all the remaining rounds, the GPA prescribes the action pair that maximizes the follower's payoff.

The intuition behind this ordering is that the GPA forces through the more 'painful' action pairs for the follower at the beginning while promising rewards for cooperation and threats for defection. As we have required $r \geq 1$, the final round will always allow the follower to get their optimal possible payoff if they have obeyed the sequence up to this point, which resolves any "last-round" incentive concerns.

This completes the description of the candidate Stackelberg GPA's prescribed sequence. The logic of the GPA itself will be to play according to the prescribed sequence, as long as the follower has played according the the prescribed sequence in all previous rounds. If the follower has ever deviated from the sequence, instead play x^*, the threat strategy, for all remaining rounds. This logic is enforced by the functions the GPA is comprised of. A detailed example construction, which includes explicitly constructing the functions for the GPA, is shown in the full version.

The following lemma uses backward induction and the ordering of the action pairs in the prescribed transcript to show that it is in the follower's best interest to follow the prescription. The proof of this lemma can be found in the full version.

Lemma 4. *Following the prescribed sequence for all T rounds of play is a best-response GPA (for the follower) to the candidate leader GPA described above.*

Next, we show a lower bound on the payoff of the leader if the follower follows the prescribed transcript.

Lemma 5. *If the follower obeys the prescribed sequence, the resulting payoff for the leader will get a $2N/T$ approximation to the optimum value OPT_{LP} of the LP.*

Proof. In the first cN rounds, the empirical frequencies of the action pair in the resulting transcript (when the follower obeys the prescribed sequence) exactly equals the probability distribution α. Therefore, the average payoff of the leader in the first cN rounds equals $\sum_{i,j} M_1(i,j)\alpha_{i,j}$, which is the objective of the LP for the optimal solution. In the remaining rounds, the difference in the average payoff and the LP's optimum objective is at most 2 — therefore, the total payoff is at least $OPT_{LP}.(cN+r) - 2r$. Thus, we have a $2r/T \leq 2N/T$- approximation to the LP optimum, where the last inequality comes from $r \leq N$. □

Putting together Lemmas 3 and 5, our candidate GPA is shown to guarantee a payoff to the leader that is a $\frac{2N}{T}$-approximation to the value generated by any GPA against a rational follower [4]. More generally, our approximation factor can be rewritten as $\frac{f(n,b)}{T}$ where $f(n,b) = 2^{\text{poly}(n,b)} = 2^{\text{poly}(n)} \cdot \text{poly}(A)$ since the number of bits used to represent N is polynomial in n and b (recall that b is the number of bits used to represent each payoff value). Additionally, the algorithm to construct the GPA solves a linear program that is polynomial in the size of the game, and only does counting operations on T, and is therefore a polynomial in $n, b, \log T$ algorithm [5].

We complement this result by showing in the full versionthat it is impossible to exactly achieve the LP upper bound. Specifically, we show that a $\frac{1}{T}$-approximation to the LP upper bound is inevitable, showing that we have reached the limits of the LP based approach with respect to the dependence upon the time horizon T. We complete the proof of the theorem by showing that the judicious use of randomization can guarantee convergence to optimality which is independent of the number of actions n. This part is deferred to the full version.

5 HARDNESS OF COMPUTING AN APPROXIMATE STACKELBERG GPA IN 3-PLAYER GAMES

While we have thus far considered only two-player Stackelberg games, the notion of an n-player Stackelberg game is also well-defined. In an n-player Stackelberg game, there is a strict ordering among all players. The first player in order makes a commitment, which all players get to see. Then the second player in order makes a commitment, and so on until the final player. In a single-shot game, this commitment is a (possibly mixed) strategy, while in the repeated games setting this commitment is a GPA. The SE is achieved when the first player in order commits to something which maximizes their expected value, given that all the remaining players best-respond in turn. The added complexity with more than two players is that, when making commitments, players must now reason about not just how a single follower might respond to them but how the follower might try to affect players further down the line. When we discuss additive approximations, we are still referring only to the approximation to the payoff of the very first player in line. All followers are assumed to be perfectly best-responding.

We must be careful when discussing hardness in algorithm space; we cannot simply say that constructing an optimal GPA is hard just because finding the optimal move commitments in a GPA is hard. This is because GPAs are composed of algorithms, so the hard computational work could be delegated to the GPA itself. Thus, our hardness result for three-player approximation tells us two things: 1) finding the approximate *value* of a Stackleberg equilibrium in the three-player setting is hard, and 2) finding an approximately optimal *GPA* for Player 1 in the three-player setting is hard, as long as all GPAs are restricted to run in poly-time. The proof of this result has been deferred to the full version.

Theorem 3. *In three-player finite-horizon repeated games over T rounds, there exists no polynomial in n, A, T time algorithm computing a* $\left(\frac{A^k}{T^{\frac{1}{k}}}\right)$*-additive approximationto the Stackelberg value of the game unless* $P = NP$*. Here,* k *is any natural number.*

ACKNOWLEDGMENTS

The second author was supported by the grant NSF CCF 1926872.

REFERENCES

[1] Bo An, Milind Tambe, and Arunesh Sinha. 2017. Stackelberg security games (ssg) basics and application overview. *Improving Homeland Security Decisions* (2017), 485.

[2] Maria-Florina Balcan, Avrim Blum, Nika Haghtalab, and Ariel D. Procaccia. 2015. Commitment Without Regrets: Online Learning in Stackelberg Security Games. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation* (Portland, Oregon, USA) *(EC '15)*. Association for Computing Machinery, New York, NY, USA, 61–78. https://doi.org/10.1145/2764468.2764478

[3] Jean-Pierre Benoit, Vijay Krishna, et al. 1984. Finitely repeated games. (1984).

[4] Avrim Blum. 1998. On-line algorithms in machine learning. *Online algorithms* (1998), 306–325.

[5] Avrim Blum, Eyal Even-Dar, and Katrina Ligett. 2006. Routing without regret: On convergence to Nash equilibria of regret-minimizing algorithms in routing games. In *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*. 45–52.

[6] Avrim Blum and Yishay Monsour. 2007. Learning, regret minimization, and equilibria. (2007).

[7] Christian Borgs, Jennifer Chayes, Nicole Immorlica, Adam Tauman Kalai, Vahab Mirrokni, and Christos Papadimitriou. 2010. The myth of the folk theorem. *Games and Economic Behavior* 70, 1 (2010), 34–43.

[8] Michel Le Breton, A. Alj, and Alain Haurie. 1985. Sequential Stackelberg equilibria in two-person games. *Journal of Optimization Theory and Applications* 59 (1985), 71–97.

[9] Nicolo Cesa-Bianchi and Gábor Lugosi. 2006. *Prediction, learning, and games*. Cambridge university press.

[10] Xi Chen and Xiaotie Deng. 2006. Settling the complexity of two-player Nash equilibrium. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*. IEEE, 261–272.

[11] Vincent Conitzer and Tuomas Sandholm. 2006. Computing the optimal strategy to commit to. In *Proceedings of the 7th ACM conference on Electronic commerce*. 82–90.

[12] Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. 2009. The complexity of computing a Nash equilibrium. *SIAM J. Comput.* 39, 1 (2009), 195–259.

[13] Yoav Freund and Robert E Schapire. 1999. Adaptive game playing using multiplicative weights. *Games and Economic Behavior* 29, 1-2 (1999), 79–103.

[14] Nika Haghtalab, Thodoris Lykouris, Sloan Nietert, and Alexander Wei. 2022. Learning in Stackelberg Games with Non-myopic Agents. In *Proceedings of the 23rd ACM Conference on Economics and Computation*. 917–918.

[15] Joseph Y Halpern, Rafael Pass, and Lior Seeman. 2014. The truth behind the myth of the folk theorem. In *Proceedings of the 5th conference on Innovations in theoretical computer science*. 543–554.

[16] J. R. Hicks. 1935. *The Economic Journal* 45, 178 (1935), 334–336. http://www.jstor.org/stable/2224643

[4] The follower may choose some other equally good best response GPA – however in the definition of Stackleberg equilbria/ GPA, the follower always picks the best possible option from the leader's perspective while breaking ties, so the leader's candidate GPA is in fact guaranteed this approximation factor

[5] Finding an implicit representation of GPA can be done in time polynomial in $\log T$, by, say, using a for loop. However, explicitly writing the GPA as T-functions, one for each round, would still take time linear in T

[17] Walid Krichene, Benjamin Drighes, and Alexandre Bayen. 2014. On the convergence of no-regret learning in selfish routing. In *International Conference on Machine Learning*. PMLR, 163–171.

[18] Niklas Lauffer, Mahsa Ghasemi, Abolfazl Hashemi, Yagiz Savas, and Ufuk Topcu. 2022. No-Regret Learning in Dynamic Stackelberg Games. https://doi.org/10.48550/ARXIV.2202.04786

[19] Joshua Letchford and Vincent Conitzer. 2010. Computing Optimal Strategies to Commit to in Extensive-Form Games. In *Proceedings of the 11th ACM Conference on Electronic Commerce* (Cambridge, Massachusetts, USA) *(EC '10)*. Association for Computing Machinery, New York, NY, USA, 83–92. https://doi.org/10.1145/1807342.1807354

[20] Michael L Littman and Peter Stone. 2005. A polynomial-time Nash equilibrium algorithm for repeated games. *Decision Support Systems* 39, 1 (2005), 55–66.

[21] Janusz Marecki, Gerry Tesauro, and Richard Segal. 2012. Playing Repeated Stackelberg Games with Unknown Opponents. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 2* (Valencia, Spain) *(AAMAS '12)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 821–828.

[22] John Nash. 1951. Non-cooperative games. *Annals of mathematics* (1951), 286–295.

[23] Tim Roughgarden. 2010. Algorithmic game theory. *Commun. ACM* 53, 7 (2010), 78–86.

[24] Aviad Rubinstein. 2016. Settling the complexity of computing approximate two-player Nash equilibria. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 258–265.

[25] Alexander Schrijver et al. 2003. *Combinatorial optimization: polyhedra and efficiency*. Vol. 24. Springer.

[26] M. Simaan and J. B. Cruz. 1973. On the Stackelberg Strategy in Nonzero-Sum Games. *J. Optim. Theory Appl.* 11, 5 (may 1973), 533–555. https://doi.org/10.1007/BF00935665

[27] Bernhard Von Stengel and Shmuel Zamir. 2010. Leadership games with convex strategy sets. *Games and Economic Behavior* 69, 2 (2010), 446–457.

[28] Song Zuo and Pingzhong Tang. 2015. Optimal Machine Strategies to Commit to in Two-Person Repeated Games. In *AAAI*.