Efficiently Solving Turn-Taking Stochastic Games with Extensive-Form Correlation

HANRUI ZHANG, Carnegie Mellon University, USA
YU CHENG, Brown University, USA
VINCENT CONITZER, Carnegie Mellon University, USA

We study equilibrium computation with extensive-form correlation in two-player turn-taking stochastic games. Our main results are two-fold: (1) We give an algorithm for computing a Stackelberg extensive-form correlated equilibrium (SEFCE), which runs in time polynomial in the size of the game, as well as the number of bits required to encode each input number. (2) We give an efficient algorithm for approximately computing an optimal extensive-form correlated equilibrium (EFCE) up to machine precision, i.e., the algorithm achieves approximation error ε in time polynomial in the size of the game, as well as $\log(1/\varepsilon)$.

Our algorithm for SEFCE is the first polynomial-time algorithm for equilibrium computation with commitment in such a general class of stochastic games. Existing algorithms for SEFCE typically make stronger assumptions such as no chance moves, and are designed for extensive-form games in the less succinct tree form. Our algorithm for approximately optimal EFCE is, to our knowledge, the first algorithm that achieves 3 desiderata simultaneously: approximate optimality, polylogarithmic dependency on the approximation error, and compatibility with stochastic games in the more succinct graph form. Existing algorithms achieve at most 2 of these desiderata, often also relying on additional technical assumptions.

CCS Concepts: • **Theory of computation → Exact and approximate computation of equilibria**.

Additional Key Words and Phrases: equilibrium computation, stochastic games, extensive-form games, extensive-form correlation

ACM Reference Format:
Hanrui Zhang, Yu Cheng, and Vincent Conitzer. 2023. Efficiently Solving Turn-Taking Stochastic Games with Extensive-Form Correlation. In *Proceedings of the 24th ACM Conference on Economics and Computation (EC '23), July 9–12, 2023, London, United Kingdom.* ACM, New York, NY, USA, 26 pages. https://doi.org/10.1145/3580507.3597665

1 INTRODUCTION

Equilibrium computation is one of the most important topics in algorithmic game theory. Decades of effort has painted a fairly complete landscape for the computational complexity of various equilibrium concepts in normal-form games: Roughly speaking, computing an (optimal) equilibrium is computationally tractable, if either correlation is allowed between both players' actions, or one of the two players has the power to commit to a strategy. In other words, in normal-form games, there are polynomial-time algorithms for computing an optimal (i.e., maximizing a convex combination of the two players' utilities) correlated equilibrium, and for computing a Stackelberg equilibrium (see, e.g., [Papadimitriou, 2007]).

The situation is subtler for games in dynamic environments, where the two players iteratively take actions, each affecting the state of the world, and together determining the overall payoff of each player. Such games are conventionally modeled as stochastic games or extensive-form

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
EC '23, July 9–12, 2023, London, United Kingdom
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0104-7/23/07.
https://doi.org/10.1145/3580507.3597665

games (we will discuss the differences between the two formulations momentarily). In these games, neither correlation nor commitment power guarantees tractability. In fact, it is NP-hard to compute a Stackelberg equilibrium in these games, even if normal-form correlation — meaning a mediator recommends a whole strategy, consisting of an action to be played in each possible information set of the game, to each player at the beginning of the play — is allowed [Letchford and Conitzer, 2010]. In light of such hardness results, it has been argued that the right notion of correlation in extensive-form games is extensive-form correlation, where the mediator reveals a recommended action to be played in an information set only when a player has reached that information set.

The notion of extensive-form correlation leads to a number of natural solution concepts, which generalize correlated equilibria in normal-form games and are computationally more tractable. Among them, the most important and well-studied ones are extensive-form correlated equilibria (henceforth EFCE) and Stackelberg extensive-form correlated equilibria (henceforth SEFCE) [von Stengel and Forges, 2008]. While significant effort has been made on designing efficient algorithms for computing (optimal) EFCE and SEFCE, most existing algorithms are designed for extensive-form games in the tree form (for exceptions, see Section 1.2): The input to such an algorithm is by default a tree capturing all possibilities in a game, where each leaf corresponds to a possible way for the game to play out, and the time complexity of the algorithm is polynomial in the size of this game tree. Such a game tree is often not the most succinct representation of a game. For example, consider the following adapted version of the game Nim [Bouton, 1901]: Initially there are k matches on the table. Alice and Bob take turns removing matches, where in each turn, the acting player can remove either 1 or 2 matches. The player who removes the last match wins. The natural state space of the game is quite succinct: The state is fully determined by the number of matches left and the identity of the player to act next, so the state space is of size $O(k)$. However, the tree form of the game must encode the entire history through which a state is reached (e.g., "Alice removes 1 match; Bob removes 2 matches; Alice removes 2 matches; Bob removes 1 match; ..."), which means the game tree has $2^{\Omega(k)}$ nodes. In such cases, an algorithm that runs in time polynomial in the size of the game tree would not appear particularly efficient. In this paper, we address the above issue by designing efficient algorithms for optimal EFCE and SEFCE that work with stochastic games, which are by default represented in the graph form that succinctly encodes a game.

1.1 Our Results

Throughout the paper, we focus on two-player, finite-horizon, turn-taking stochastic games. Put in different words, we focus on two-player, perfect-information extensive-form games in the graph form. Our main results are twofold:

- We give an algorithm for computing an SEFCE, which runs in time polynomial in the size of the game, as well as the number of bits required to encode each input number.
- We give an efficient algorithm for computing an approximately optimal approximate EFCE up to machine precision, i.e., the algorithm achieves approximation error ε in time polynomial in the size of the game, as well as $\log(1/\varepsilon)$.

Our algorithm for SEFCE is, to our knowledge, the first polynomial-time algorithm for equilibrium computation with commitment in such a general class of stochastic games (the main assumption being that the game is turn-taking). As discussed in Section 1.2, existing algorithms for SEFCE typically make stronger assumptions such as no chance moves, and are designed for extensive-form games in the less succinct tree form. Our algorithm for approximately optimal EFCE is, to our knowledge, the first algorithm that achieves 3 desiderata simultaneously: approximate optimality, polylogarithmic dependency on the approximation error, and compatibility with stochastic games

in the graph form. As discussed in Section 1.2, existing algorithms typically achieve at most 2 of these desiderata, often also relying on additional technical assumptions.

Technically, our algorithms are built on ideas fundamentally different from the most commonly seen techniques in equilibrium computation, i.e., linear programming and no-regret dynamics. We take a semi-combinatorial approach centered around the notion of Pareto frontier curves. Roughly speaking, the Pareto frontier curve for a state-action pair captures the optimal tradeoff between the two players' onward utilities subject to equilibirum conditions, in the subgame induced by the state-action pair. These curves can be viewed as a multidimensional generalization of the Q-function commonly used in planning and reinforcement learning. Given the right equilibrium conditions, computing an SEFCE or an optimal EFCE reduces to evaluating Pareto frontier curves, which at first sight appears to be a numerical problem in nature. In order to perform the necessary evaluations efficiently, we establish combinatorial properties of the Pareto frontier curves, including recursive relations between curves for different state-action pairs, as well as lower bounds on the numerical "resolution" of the curves (i.e., how close two turning points can be on a curve). The curves for SEFCE exhibit particularly nice properties, based on which we are able to design an essentially combinatorial procedure for evaluating the Pareto frontier curves recursively. This involves binary searching over "directions of evaluation" up to a carefully chosen precision, as well as a memorization technique that avoids redundant recursive calls. For EFCE, the Pareto frontier curves are less structured, and in particular, the curves can be very (i.e., doubly exponentially) fine in terms of their numerical resolution. It thus becomes infeasible to exactly evaluate the curves, and our algorithm instead performs evaluations up to any desired precision in polylogarithmic time. For a more detailed overview of our algorithms, see Section 3.1.

1.2 Further Related Work

Equilibrium computation in normal-form games has been extremely well-studied. For example, without commitment, Daskalakis et al. [2009] and Chen et al. [2009] show that computing a Nash equilibrium is PPAD-complete in two-player normal-form games, and computing optimal Nash equilibria is generally NP-hard [Conitzer and Sandholm, 2008, Gilboa and Zemel, 1989]. In contrast, when correlation is allowed, one can compute an optimal correlated equilibrium efficiently (see, e.g., [Papadimitriou, 2007]). With commitment, Conitzer and Sandholm [2006] give an efficient algorithm for computing a Stackelberg equilibrium in two-player normal-form games (see also von Stengel and Zamir [2010]), and that this becomes hard with 3 players; however, if the committing player can also send signals to the other players, thereby effectively taking over the role of the mediator in correlated equilibrium, then the problem is again efficiently solvable with any number of players [Conitzer and Korzhyk, 2011]. So in short, efficient equilibrium computation is possible if either correlation or commitment is allowed.

Equilibrium computation becomes more difficult in dynamic environments, such as extensive-form games and stochastic games. There, commitment does not imply efficient computation anymore: Letchford and Conitzer [2010] show that it is NP-hard to compute a Stackelberg equilibrium in two-player extensive-form games, even with perfect information. Moreover, their hardness result holds even if normal-form correlation (as opposed to extensive-form correlation to be discussed momentarily) is allowed. Similar hardness results hold for various structured families of stochastic games [Letchford et al., 2012]. To circumvent such hardness results, von Stengel and Forges [2008] introduce the notion of extensive-form correlation, where conceptually, recommended actions are revealed on the fly. They give an efficient algorithm for computing an optimal extensive-form correlated equilibrium (EFCE) when the game has no chance moves, and show that with chance moves, the same problem is NP-hard. Notably, their hardness result assumes imperfect information,

which turn-taking stochastic games do not have. In short, extensive-form correlation is generally necessary for efficient computation in dynamic environments.

Subsequently, there has been a long line of research on the computation of optimal EFCE, as well as its Stackelberg version, Stackelberg EFCE (henceforth SEFCE). However, as far as we know, all existing positive results, i.e., efficient algorithms, are for extensive-form games in the tree form. To name a few examples, Cermak et al. [2016] give a polynomial-time algorithm for computing an SEFCE in extensive-form games without chance moves. Relatedly, Bošanskỳ et al. [2017] show, among other results, that it is possible to compute an SEFCE in perfect-information extensive-form games in polynomial time. Farina et al. [2019] give a saddle-point formulation for optimal EFCE and design gradient-based algorithms that scale better in practice. Farina and Sandholm [2020] give a polynomial-time algorithm for computing optimal EFCE when chance moves are public. Zhang et al. [2022c] give fixed-parameter algorithms for computing optimal EFCE, as well as related solution concepts. Zhang and Sandholm [2022] give a polynomial-time algorithm for a general class of equilibrium computation problems that involve a mediator, which in particular generalize the results by Zhang et al. [2022c]. The type of computational problem considered in these results is "easier" than ours, in the sense that the graph form that we consider can be much more succinct (and never less succinct) than the tree form of the same game, and in such cases, an algorithm that runs in time that is polynomial in the size of the graph form is much more efficient. Conversely, an algorithm that runs in time that is polynomial in the size of the tree form is not necessarily polynomial-time in the graph form.

Another line of research studies no-regret dynamics that converge to an arbitrary (i.e., not necessarily optimal) EFCE. This translates to polynomial-time algorithms in $1/\varepsilon$ for ε-EFCE. Compared to these algorithms, our algorithm for approximately optimal EFCE (1) guarantees approximate optimality, (2) runs in polynomial time in $\log(1/\varepsilon)$ instead of $1/\varepsilon$, and (3) works with the more succinct graph form of the game, instead of the tree form. Huang and von Stengel [2008] show how to compute an arbitrary EFCE exactly in polynomial time. However, their algorithm does not guarantee optimality, nor is it compatible with the graph form.

Technically, computing optimal or Stackelberg equilibria generalizes the problem of planning in Markov decision processes under constraints. Particularly related to our results is "planning with participation constraints" [Zhang et al., 2022a,b]: Roughly speaking, in that problem, the principal (corresponding to the leader in a Stackelberg game) chooses which action to take in each state, subject to the constraint that the agent (corresponding to the follower) is always willing to participate, i.e., the agent's onward utility in each state is always nonnegative. This can be viewed as a highly restricted class of turn-taking stochastic games, where the agent's (follower's) only actions in each state are to stay and to quit. Zhang et al. [2022a,b] show that even in these restricted environments, an optimal policy (i.e., a Stackelberg equilibrium) may have to be history-dependent, and give a polynomial-time algorithm for planning with participation constraints. However, their algorithm is tailored to the essentially non-strategic setting where the agent's power is extremely limited. In contrast, we consider general game-theoretic settings where the two players are generally equally powered, except that in the Stackelberg setting, one player in addition has commitment power.

Pareto frontier curves have also been (implicitly) considered in prior work in equilibrium computation, e.g., [Bošanskỳ et al., 2017, Letchford and Conitzer, 2010]. In particular, Letchford and Conitzer [2010] and Bošanskỳ et al. [2017] present algorithms that essentially keep track of the entire Pareto frontier curves, by computing and storing all turning points on each curve. In the settings that we study, doing so would generally require exponential computation — in fact, the key idea behind our algorithms is to avoid computing and storing all turning points, and only focus on the important ones.

2 PRELIMINARIES

Stochastic games. We focus on two-player finite-horizon turn-taking stochastic games in this paper. There is a finite set of states $\mathcal{S} = [n]$, and a finite set of actions $\mathcal{A} = [m]$. $s_{\text{init}} = 1$ and $s_{\text{term}} = n$ are the initial and terminal states, respectively. For each state $s \in \mathcal{S}$, there is an acting player $\text{ap}(s) \in \{1, 2\}$, who unilaterally decides which action to play in state s. For each player $i \in \{1, 2\}$, there is a reward function $r_i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}_+$, which specifies the immediate reward $r_i(s, a)$ that player i receives when action a is played in state s. We assume rewards are normalized, i.e., $r_i(s, a) \in [0, 1]$ for each $i \in \{1, 2\}$, $s \in \mathcal{S}$ and $a \in \mathcal{A}$. A transition operator $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ specifies the distribution $P(s, a)$ of the next state when action a is played in state s, where for each s', $P(s, a, s')$ is the probability that the next state is s'.

Unless otherwise specified, we assume the transition operator is acyclic, i.e., $P(s, a, s') > 0$ only if $s' > s$ or $s = s' = n$. For the terminal state $s_{\text{term}} = n$ in particular, we assume $r_i(n, a) = 0$ and $P(n, a, n) = 1$ for each action $a \in \mathcal{A}$. In other words, there is no meaningful action in the terminal state $s_{\text{term}} = n$. These assumptions essentially mean the game is finite-horizon. In particular, note that in the finite-horizon case, the acyclicity assumption is without loss of generality, as the state could include the index of the current period (with a blowup proportional to the time horizon T).

Histories, strategies, and utilities. Fix a stochastic game $(\mathcal{S}, \mathcal{A}, \text{ap}, r_1, r_2, P)$. A history h of length t is a sequence of t states and t actions $h = (s_1, a_1, s_2, a_2, \ldots, s_t, a_t)$ which fully describes t consecutive steps of a play. Let $\mathcal{H}$ be the collection of all histories of all lengths not exceeding n (so $\mathcal{H}$ is finite). For brevity, we also let $|h|$ denote the length of h, and $h + (s, a)$ be the history obtained by appending (s, a) to the end of h.

A deterministic (history-dependent) strategy $\pi : \mathcal{H} \times \mathcal{S} \to \mathcal{A}$ maps each history-state pair (h, s) to the action to be played in s given history h. Note that we do not explicitly partition a strategy into two parts corresponding to the two players, since such a partition is induced by the mapping ap from each state to the corresponding acting player. A randomized strategy Π is a distribution over deterministic strategies. For any deterministic strategy π, we say a history $h = (s_1, a_1, \ldots, s_t, a_t)$ is admissible if the action played in each step is the one specified by π, i.e., for each $t' \in [t]$, $\pi((s_1, a_1, \ldots, s_{t'-1}, a_{t'-1}), s_{t'}) = a_{t'}$. For any randomized strategy Π, we say a history h is admissible under Π if h is admissible under some π in the support of Π. Let $\mathcal{H}^\pi$ (resp. $\mathcal{H}^\Pi$) be the set of admissible histories under π (resp. Π). For a randomized strategy Π and an admissible history $h \in \mathcal{H}^\Pi$ under Π, let $\Pi \mid h$ denote the conditional version of Π given that the states reached and actions played in the first $|h|$ steps are h.

For each $i \in \{1, 2\}$, the onward utility $u_i^\pi(h, s)$ of a player i, under a deterministic strategy π, in state s, given history h, is

$$u_i^\pi(h, s) = \mathop{\mathbb{E}}_{\{s_t\}_t} \left[\sum_{t \in [n]} r_i(s_t, a_t) \right],$$

where $s_1 = s$, $h_1 = h$, $a_t = \pi(h_t, s_t)$ for each $t \in [n]$, $h_t = h_{t-1} + (s_{t-1}, a_{t-1})$ for each $t \in \{2, \ldots, n\}$, and $s_t \sim P(s_{t-1}, a_{t-1})$ for each $t \in \{2, \ldots, n\}$. Here we only need to consider n steps, since the maximum meaningful length of a play is n — after n steps, we must have reached the state n, and any extra steps would give reward 0 to both players. The unconditional onward utility $v_i^\Pi(h, s)$ of player i, under a randomized strategy Π, in state s, given history h, is

$$v_i^\Pi(h, s) = \mathop{\mathbb{E}}_{\pi \sim \Pi} [u_i^\pi(h, s)].$$

Note that this unconditional onward utility is not the actual expected onward utility; we need it mostly for notational simplicity. Conceptually, it is the utility that player i expects to receive in the future if they believe the posterior strategy is Π. Given the unconditional onward utility, for each

$i \in \{1, 2\}$, the actual onward utility $u_i^{\Pi}(h, s)$ of player i, under strategy Π, in state s, given history h, is simply

$$u_i^{\Pi}(h, s) = v_i^{\Pi|h}(h, s).$$

Extensive-form correlated equilibria. Extensive-form correlated equilibria (EFCE) and their Stackelberg version (Stackelberg EFCE, or SEFCE) are the natural generalizations of correlated equilibria and Stackelberg equilibria to dynamic settings such as stochastic games and extensive-form games. In the original definition of EFCE for extensive-form games by von Stengel and Forges [2008], a mediator specifies a distribution over deterministic strategies (i.e., a randomized strategy according to our definition above), where each deterministic strategy specifies a recommended action in each node of the game tree (corresponding to a history-state pair in our formulation). A deterministic strategy is drawn and fixed at the beginning of the play, but the recommended action in each node given by this strategy is revealed to the acting player only when the node is actually reached. If a player decides to not follow a recommended action, that player will not receive recommended actions in the rest of the play.

For any $\varepsilon \geq 0$, we say a player is ε-best responding under a randomized strategy if that player cannot increase their onward utility by more than ε by deviating from the recommended action at any point of a recommended path of play, i.e., an admissible history. A randomized strategy is an ε-EFCE if both players are ε-best responding. Moreover, consider a Stackelberg setting where player 1 is the leader and player 2 is the follower. Then, a randomized strategy is an SEFCE if player 1's utility is maximized subject to the constraint that player 2 is 0-best responding (or simply best responding).

Put in our language, a player $i \in \{1, 2\}$ is ε-best responding under a randomized strategy Π iff for any admissible history $h \in \mathcal{H}^{\Pi}$, state s where $\mathrm{ap}(s) = i$, action a where $h + (s, a) \in \mathcal{H}^{\Pi}$, and deterministic strategy π' where $\pi'(h, s) \neq a$:

$$v_i^{\Pi|(h+(s,a))}(h, s) \geq \varepsilon + \mathop{\mathbb{E}}_{\pi \sim \Pi|(h+(s,a))} \left[u_i^{(i:\pi', 3-i:\pi)}(h, s) \right].$$

Here, $(i : \pi', 3 - i : \pi)$ denotes a strategy obtained by combining π' restricted to player i's actions and π restricted to $(3 - i)$'s actions (note that $3 - i = 2$ when $i = 1$, and vice versa; $3 - i$ simply means the other player than i). That is,

$$(i : \pi', 3 - i : \pi)(h, s) = \begin{cases} \pi'(h, s), & \text{if } \mathrm{ap}(s) = i \\ \pi(h, s), & \text{otherwise.} \end{cases}$$

We will use this notation repeatedly in the rest of the paper. The left hand side is the utility i expects to receive if both players keep following the recommendations, where in particular, i's belief for the strategy is $\Pi \mid (h + (s, a))$ because i has already received the recommended action a. The right hand side is the utility i expects to receive if i unilaterally deviates to π' and the other player keeps following the recommendations, which should never be larger than the left hand side. Recall that a randomized strategy Π is an ε-EFCE iff both players are ε-best responding under Π. One can check this is in fact equivalent to the definition by von Stengel and Forges [2008] when $\varepsilon = 0$. A randomized strategy Π is an SEFCE (with player 1 being the leader) iff

$$\Pi \in \mathop{\mathrm{argmax}}_{\text{player 2 is best responding under } \Pi} u_1^{\Pi}(\emptyset, s_{\text{init}}).$$

3 STACKELBERG EXTENSIVE-FORM CORRELATED EQUILIBRIA

3.1 Overview of Our Approach

Maximum punishment without loss of generality. Our algorithm is based on the standard observation that in an SEFCE, it is without loss of generality to maximally punish the follower when they deviate from the prescribed path of play, regardless of how that would affect the leader's utility at that point. In fact, for any SEFCE, there exists an effectively equivalent SEFCE where deviation always immediately triggers maximum punishment, so once the follower deviates, the game immediately becomes effectively zero-sum. This is because intuitively, the sole purpose of the leader's equilibrium strategy in parts of the game where the follower has deviated is to threaten the follower and cancel out any potential incentive to deviate. In particular, such a threat would never be actually executed, because in equilibrium no player would deviate in the first place. As such, it never hurts to threaten with the worst punishment possible. This greatly simplifies the problem from a computational perspective, since computing a strategy for maximum punishment is no harder than solving turn-taking zero-sum stochastic games, which can be done by simple backward induction.

Reducing to constrained planning. Once the punishment strategy is fixed, we only need to optimize over strategies where the follower never faces worse utility than what they would face after deviating in the optimal way and being maximally punished thereafter. One key observation here is that in any state, regardless of the recommended action, the optimal way to deviate is always the same. So, to prevent the follower from deviating, we only need to guarantee that conditioned on the recommended action, the onward utility of the follower is at least the utility resulting from deviating optimally. In particular, the latter utility depends only on the state (which is only true in turn-taking stochastic games). Given this observation, the problem becomes a constrained planning problem, where we want to find an optimal strategy subject to the constraint that in each state where the follower is the acting player, the onward utility of the follower (conditioned on the recommended action) is at least some state-dependent quantity that can be efficiently pre-computed. This is very similar to planning in constrained MDPs, except for one key difference: In constrained MDPs, typically there are a constant number of feasibility constraints over the cumulative reward vector, whereas in our constrained planning problem, there are separate feasibility constraints on the follower's onward utility in each state of the game where the follower is the acting player.

Pareto frontier curves and pivotal points. The way we approach the constrained planning problem is by considering the Pareto frontier curves for each state-action pair. Roughly speaking, the Pareto frontier curve $f_{s,a}$ for a state-action pair (s, a) captures the Pareto-optimal way to trade off between the two players' onward utilities after the acting player takes action a in state s, subject to feasibility constraints *in the future*. This can be viewed as a generalization of the Q-function that is commonly considered in reinforcement learning that captures the tradeoff between the two players' utilities. For technical reasons, we intentionally disregard the constraint (if there is one) in the (current) state s. With this definition, the problem of constrained planning (or at least, the problem of computing the maximum objective value therein) becomes the problem of evaluating the Pareto frontier curves subject to feasibility constraints. In particular, as illustrated in Figure 1, given an objective direction (which can be any combination of the two player's utilities), the maximum objective value onward for each state-action pair is simply the farthest point on (the feasible part of) the Pareto frontier curve along that direction.

Observe that under our definition, Pareto frontier curves are always concave, since from each state-action pair onward, the convex combination of two feasible strategies is also feasible — this is true because we consider extensive-form correlation, so when we take a convex combination

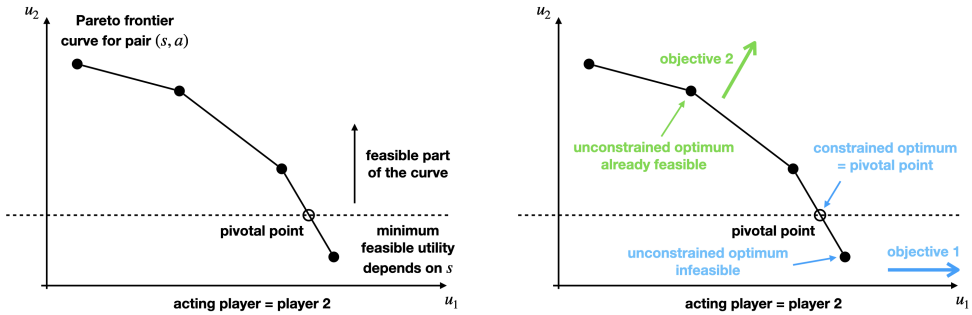

Fig. 1. Illustration of Pareto frontier curves.

through randomization, players are not allowed to know the realization before reaching the state where the randomization happens. Given this observation, finding this point involves two conceptual steps: One first finds the unconstrained optimum on the curve. If that point turns out to be infeasible, then one "rounds" that point to the nearest feasible point, which is always the pivotal point (hollow points in Figure 1), i.e., the unique point on the curve where the feasibility constraint is binding. This highlights the importance of pivotal points, which play a central role in our algorithm.

Evaluating Pareto frontier curves. Now the problem becomes efficiently evaluating Pareto frontier curves subject to feasibility constraints. At a high level, this can be done in a recursive fashion: Suppose we want to evaluate the curve $f_{s,a}$ for (s, a) in a certain direction $\alpha \in \mathbb{R}^2$, subject to the feasibility constraint in state s. Moreover, suppose we can efficiently evaluate the constrained curves for all later state-action pairs. Then we can perform the evaluation using the following procedure:

(1) For each later state $s' > s$:
  (a) For each action a', evaluate the constrained curve $f_{s',a'}$ along direction α.
  (b) Let the farthest point along α found in the above evaluation be the partial result for state s'.
(2) Aggregate the partial results for all later states s' according to the transition probabilities $P(s, a, s')$, and shift the aggregated result by the immediate rewards $(r_1(s, a), r_2(s, a))$ induced by (s, a).
(3) If the shifted result above is feasible, return it; otherwise, return the pivotal point on $f_{s,a}$.

There is one gap in the above procedure: In step 3, when we "round" an infeasible shifted result, it is assumed that we already know the pivotal point on the curve $f_{s,a}$ — in fact, without this rounding step, we would be evaluating the curve $f_{s,a}$ without the feasibility constraint in s. In reality we need to compute this pivotal point efficiently. In what follows we discuss how this can be done up to machine precision.

Again assume we can evaluate the curves for all later state-action pairs. To approximate the pivotal point, we only need to find two points on $f_{s,a}$ that are close enough, to the left and the right of the pivotal point respectively. Then a particular convex combination of the two points will be a good approximation of the pivotal point. Suppose we want to find a point to the left of the pivotal point that is close enough, and for concreteness, suppose the acting player is player 2, as in the left subfigure in Figure 1. There, a point is to the left of the pivotal point if and only if it is feasible. Conceptually, the pivotal point can be found using a "moving direction" procedure: We start with direction $(0, 1)$, and perform an evaluation of $f_{s,a}$ in that direction without the feasibility

constraint in s. Such an unconstrained evaluation can be done recursively without knowing the pivotal point (using the procedure above without step 3). If the evaluation returns a point to the left of the pivotal point (i.e., a feasible point), we rotate the direction of evaluation to the right, and evaluate the unconstrained curve again. The rotation stops as soon as the point found makes the feasibility constraint binding, which means we have found the pivotal point. To make this conceptual procedure practical, we replace the rotation with a binary search, which finds a point to the left of the pivotal point that is at most ε away (in terms of the polar angle) in $O(\log(1/\varepsilon))$ iterations. When the constrained planning problem corresponds to the computation of SEFCE, we show that any two turning points on a Pareto frontier curve must be well separated, by some quantity that is at most exponentially small in the size of the game. So, if we make ε exponentially small (which means there are polynomially many iterations in the binary search), the binary search is guaranteed to find the closest turning point to the left of the pivotal point. We can then compute the pivotal point by finding the closest point to the right in the same way, and taking a convex combination of the two points found.

Bounding the number of evaluations. The above gives a recursive algorithm for evaluating Pareto frontier curves, but executed in the naïve way, the procedure may take exponential time in the size of the game. We need one final observation to make the algorithm polynomial time in these parameters: For each (s, a) pair, the pivotal point on the curve $f_{s,a}$ only needs to be evaluated once. Given this, the total number of recursive evaluations triggered must be polynomial in the size of the game. This is because new directions of evaluation emerge only when we binary search for a pivotal point. Each binary search may create polynomially many new directions, and since we binary search for each pivotal point only once, there are $O(mn)$ binary searches in total, which means the total number of relevant directions is polynomial. Moreover, each direction can only appear in $O(mn)$ evaluations (one for each state-action pair), since we never need to perform the same evaluation twice. This means the total number of evaluations for all relevant directions is polynomial.

3.2 Reduction to Constrained Planning

We first provide a formal reduction from computing an SEFCE to the constrained planning problem.

The punishment amplifier. Fixing a stochastic game $(\mathcal{S}, \mathcal{A}, \mathrm{ap}, r_1, r_2, P)$, our reduction involves the punishment amplifier pa, which maps each deterministic strategy to its maximally punishing version against a subset of players — for SEFCE this subset is $\{2\}$, and as we will see later, for EFCE this subset is $\{1, 2\}$. For each player $i \in \{1, 2\}$, consider the zero-sum stochastic game $(\mathcal{S}, \mathcal{A}, \mathrm{ap}, r'_1, r'_2, P)$, where $r'_i = r_i$ and $r'_{3-i} = -r_i$. Let π_i be a deterministic subgame-perfect equilibrium strategy in this zero-sum game — here, i is the player being punished, but note that π_i comprises both players' actions. Such a strategy can be found by backward induction. Note that without loss of generality, π_i is history-independent, so we write $\pi_i(s)$ for simplicity. If there are multiple candidates for π_i, we pick an arbitrary one among them (the choice does not affect our results).

Given a deterministic strategy π and a subset of players $S \subseteq \{1, 2\}$, the punishment-amplified version $\pi' = \mathrm{pa}(\pi, S)$ of π is given by: For each $h = (s_1, a_1, \ldots, s_t, a_t) \in \mathcal{H}$ and $s \in \mathcal{S}$,

- If $h \in \mathcal{H}^\pi$ (i.e., h is feasible under π), then $\pi'(h, s) = \pi(h, s)$.
- If $\mathrm{ap}(s) \notin S$, then $\pi'(h, s) = \pi(h, s)$.
- Otherwise, $\pi'(h, s) = \pi_i(s)$, where $h' = (s_1, a_1, \ldots, s_{t'})$ is the longest feasible prefix of h, and $i = \mathrm{ap}(s_{t'})$ (i.e., i is the first player who deviated, in state $s_{t'}$).

The punishment amplifier can be naturally extended to randomized strategies: For a randomized strategy Π and a subset of players S, $\mathrm{pa}(\Pi, S)$ is obtained by mapping every deterministic strategy

π in the support of Π to $\mathrm{pa}(\pi, S)$, and assign the latter the same probability mass in Π' as π has in Π.

We first prove maximum punishment is without loss of generality, which is formally captured by the following lemma:

Lemma 1. *Fix a stochastic game $(\mathcal{S}, \mathcal{A}, \mathrm{ap}, r_1, r_2, P)$. For any Π under which the follower is best responding, the follower is also best responding under $\mathrm{pa}(\Pi, \{2\})$, and moreover,*

$$u_1^{\Pi}(\emptyset, s_{\mathrm{init}}) = u_1^{\mathrm{pa}(\Pi,\{2\})}(\emptyset, s_{\mathrm{init}}).$$

We defer the proof of Lemma 1, as well as all other missing proofs, to Appendix A. The lemma suggests that when optimizing over strategies where the follower is best responding, we can focus on those with the maximum punishment structure as described above. Next we show this gives us a reduction to the constrained planning problem.

Lemma 2. *Fix a stochastic game $(\mathcal{S}, \mathcal{A}, \mathrm{ap}, r_1, r_2, P)$. For any randomized strategy Π, the follower is best responding under $\mathrm{pa}(\Pi, \{2\})$ if the following condition holds: For each admissible history $h \in \mathcal{H}^{\Pi}$, state $s \in \mathcal{S}$ where $\mathrm{ap}(s) = 2$, and action $a \in \mathcal{A}$ such that $h + (s, a) \in \mathcal{H}^{\Pi}$,*

$$v_2^{\Pi|(h+(s,a))}(h, s) \geq \max_{a' \in \mathcal{A}} \left(r_2(s, a') + \mathop{\mathbb{E}}_{s' \sim P(s,a')} [u_2^{\pi_2}(h + (s, a'), s')] \right),$$

where π_2 is the subgame perfect equilibrium when the leader tries to minimize the follower's utility, as defined above.

Observe that in the above lemma, the right-hand side of the inequality does not depend on Π. Moreover, since π_i is history-independent, it does not depend on h either. So the right hand-side is a constant that depends only on the state s. From now on, we call this quantity the utility under punishment in state s, defined as

$$u^p(s) = \max_{a' \in \mathcal{A}} \left(r_i(s, a') + \mathop{\mathbb{E}}_{s' \sim P(s,a')} [u_i^{\pi_i}(h + (s, a'), s')] \right),$$

where $i = \mathrm{ap}(s)$ (note that although we define the utility under punishment for both players, for SEFCE we only need it for the follower, i.e., player 2). The utility under punishment $u^p(s)$ can be efficiently computed in all states.

Lemmas 1 and 2 together imply the following claim, which states that finding an SEFCE is equivalent to constrained planning.

Theorem 1. *Fix a stochastic game $(\mathcal{S}, \mathcal{A}, \mathrm{ap}, r_1, r_2, P)$. For any $x \geq 0$, there exists a strategy Π under which the follower is best responding such that $u_1^{\Pi}(\emptyset, s_{\mathrm{init}}) \geq x$, if and only if there exists a strategy Π' such that $u_1^{\Pi'}(\emptyset, s_{\mathrm{init}}) \geq x$, and for each admissible history $h \in \mathcal{H}^{\Pi'}$, state $s \in \mathcal{S}$ where $\mathrm{ap}(s) = 2$, and action $a \in \mathcal{A}$ such that $h + (s, a) \in \mathcal{H}^{\Pi'}$,*

$$v_2^{\Pi'|(h+(s,a))}(h, s) \geq u^p(s).$$

Feasible strategies. We say a strategy Π is feasible if it satisfies the condition in the theorem which involves the utility under punishment, i.e., for each $h \in \mathcal{H}^{\Pi}$, $s \in \mathcal{S}$ where $\mathrm{ap}(s) = 2$, and $a \in \mathcal{A}$ such that $h + (s, a) \in \mathcal{H}^{\Pi}$,

$$v_2^{\Pi|(h+(s,a))}(h, s) \geq u^p(s).$$

We say a strategy Π is feasible after state s, if for each $h \in \mathcal{H}^{\Pi}$, $s' > s$ where $\mathrm{ap}(s') = 2$, and $a \in \mathcal{A}$ such that $h + (s', a) \in \mathcal{H}^{\Pi}$,

$$v_2^{\Pi|(h+(s',a))}(h, s') \geq u^p(s').$$

Our problem now becomes finding a feasible strategy that maximizes player 1's utility.

3.3 Pareto Frontier Curves

Before proceeding to the full description of our algorithm, we first quickly (and somewhat informally) define Pareto frontier curves and discuss some useful properties. Intuitively, the Pareto frontier curve $f_{s,a}$ for a state-action pair (s, a) is the curve capturing all Pareto-optimal pairs of onward utilities (assuming an empty history) for both players after playing action a in state s, induced by strategies that are feasible after s. Another way to view $f_{s,a}$ is it is the top right boundary of the region of pairs of onward utilities (say $F_{s,a}$) induced by strategies that are feasible after s. We call $F_{s,a}$ the feasible region for (s, a), which can be defined in the following way:

$$F_{s,a} = \left\{ (v_1^{\Pi|(s,a)}(\emptyset, s), v_2^{\Pi|(s,a)}(\emptyset, s)) \,\middle|\, \Pi \text{ is feasible after } s,\ (s, a) \in \mathcal{H}^{\Pi} \right\}.$$

We first argue that both $F_{s,a}$ and $f_{s,a}$ are well behaved:

Lemma 3. *Fix a stochastic game $(\mathcal{S}, \mathcal{A}, \mathrm{ap}, r_1, r_2, P)$. For any state $s \in \mathcal{S}$ and action $a \in \mathcal{A}$, $F_{s,a}$ is always a convex region and $f_{s,a}$ is always a concave curve.*

Given Lemma 3, $f_{s,a}$ in fact specifies a bijection between the two players' utilities under feasible strategies, corresponding to the two coordinates of a point. In the rest of the paper, we will abuse notation, and use $f_{s,a}$ in 4 ways:

- For $x \in \mathbb{R}_+$, $f_{s,a}(x) \in \mathbb{R}_+$ denotes the y-coordinate of the point on $f_{s,a}$ whose x-coordinate is x. When used in this way, the argument to $f_{s,a}$ will always be x, possibly with subscripts or superscripts.
- For $y \in \mathbb{R}_+$, $f_{s,a}(y) \in \mathbb{R}_+$ denotes the x-coordinate of the point on $f_{s,a}$ whose y-coordinate is y. When used in this way, the argument to $f_{s,a}$ will always be y, possibly with subscripts or superscripts.
- For $p \in \mathbb{R}_+^2$, we say $p \in f_{s,a}$ if p is in $f_{s,a}$ as a set of points (i.e., the graph of $f_{s,a}$ as a mapping).
- For $\alpha \in \mathbb{R}_+^2$, $f_{s,a}(\alpha) \in \mathbb{R}_+^2$ denotes the farthest point on $f_{s,a}$ along direction α. That is,

$$f_{s,a}(\alpha) = \operatorname*{argmax}_{p \in f_{s,a}} \alpha \cdot p.$$

3.4 Evaluating the Pareto Frontier Curves

Observe that if we can evaluate $f_{s_{\text{init}},a}$ for each $a \in \mathcal{A}$, then it is not hard to find the optimal utilities of the two players induced by a feasible strategy, given a particular objective direction $\alpha \in \mathbb{R}_+^2$ (for an SEFCE in particular, we want $\alpha = (1, 0)$): Without loss of generality, an optimal strategy picks a deterministic action in the initial state s_{init}, so we only need to try every one of the actions. For each action $a \in \mathcal{A}$, the optimal utilities induced by a strategy that is feasible after s_{init} is $f_{s,a}(\alpha)$. If $\mathrm{ap}(s_{\text{init}}) = 1$ or $f_{s,a}(\alpha)_{\mathrm{ap}(s_{\text{init}})} \geq u^p(s_{\text{init}})$, then this strategy is a feasible strategy, and $f_{s,a}(\alpha)$ gives the optimal utilities if the first action is a. Otherwise, since $f_{s,a}$ is concave, the optimal utilities induced by a feasible strategy must correspond to the point on $f_{s,a}$ where the constraint in s_{init} is binding. More specifically, suppose $\mathrm{ap}(s_{\text{init}}) = 2$, and let $y_{s_{\text{init}}} = u^p(s_{\text{init}})$. Then the optimal utilities when the first action is a must be $(f_{s,a}(y_{s_{\text{init}}}), y_{s_{\text{init}}})$.

Pivotal points. The above discussion suggests that the point $(f_{s,a}(y_{s_{\text{init}}}), y_{s_{\text{init}}})$ plays a particularly important role in finding the optimal utilities. More generally, we define the pivotal point $\mathrm{pp}(s, a)$ on $f_{s,a}$ for each (s, a) where $\mathrm{ap}(s) = 2$ to be the rightmost point (if there is one) on $f_{s,a}$ such that $\mathrm{pp}(s, a)_{\mathrm{ap}(s)} \geq u^p(s)$. If such a point does not exist, then we let $\mathrm{pp}(s, a) = (-n, -n)$ (here, $-n$ is without loss of generality — any quantity that is small enough would be consistent with our results). For notational convenience, if $\mathrm{ap}(s) = 1$, we let $\mathrm{pp}(s, a) = f_{s,a}((1, 0))$. Below we demonstrate how to evaluate the Pareto frontier curves recursively with the help of the pivotal points.

Lemma 4. *Fix a stochastic game* $(\mathcal{S}, \mathcal{A}, \mathrm{ap}, r_1, r_2, P)$. *For any* $s \in \mathcal{S}$, $a \in \mathcal{A}$, *and* $\alpha \in \mathbb{R}_+^2$,

$$f_{s,a}(\alpha) = (r_1(s,a), r_2(s,a)) + \mathop{\mathbb{E}}_{s' \sim P(s,a)} [p_{s'}],$$

where for each $s' > s$ *and* $a' \in \mathcal{A}$,

$$p_{s',a'} = \begin{cases} f_{s',a'}(\alpha), & \text{if } \mathrm{ap}(s') = 1 \text{ or } f_{s',a'}(\alpha)_2 \geq u^p(s') \\ \mathrm{pp}(s',a'), & \text{otherwise,} \end{cases}$$

and for each $s' > s$,

$$p_{s'} = \mathop{\mathrm{argmax}}_{p \in \{p_{s',a'}\}_{a' \in \mathcal{A}}} p \cdot \alpha.$$

In words, the lemma says that once the pivotal points for all later state-action pairs have been computed, evaluating $f_{s,a}$ can be reduced to at most mn evaluations of curves for later state-action pairs. This reduction plays a central role in our algorithm. Moreover, it also provides a way for bounding the numerical resolution of the Pareto frontier curves, which is captured by the following lemma.

Lemma 5. *Fix a stochastic game* $(\mathcal{S}, \mathcal{A}, \mathrm{ap}, r_1, r_2, P)$. *Suppose all parameters of the game can be encoded using* L *bits, i.e., for each* $s, s' \in \mathcal{S}$ *and* $a \in \mathcal{A}$, $r_1(s,a)$, $r_2(s,a)$ *and* $P(s,a,s')$ *are all multiples of* 2^{-L}. *Then for each* $s \in \mathcal{S} \setminus \{s_{\mathrm{term}}\} = [n-1]$, *there exists some integer* $C_s \leq 2^{(n-s)(n+1)L}$, *such that for each* $a \in \mathcal{A}$:

- *For any* $\alpha \in \mathbb{R}_+^2$, *the* y*-coordinate of* $f_{s,a}(\alpha)$ *is a multiple of* $2^{-(n-s)L}$, *and the* x*-coordinate is a multiple of* $1/C_s$.
- *If* $\mathrm{ap}(s) = 2$, *then the* y*-coordinate of* $\mathrm{pp}(s,a)$ *is a multiple of* $2^{-(n-s)L}$, *and the* x*-coordinate is a multiple of* $1/C_s$.

Moreover, for each $s \in [n-2]$, C_s *is a multiple of* C_{s+1}.

We will use the following direct corollary of Lemma 5:

Corollary 1. *Fix a stochastic game* $(\mathcal{S}, \mathcal{A}, \mathrm{ap}, r_1, r_2, P)$. *Suppose all parameters of the game can be encoded using* L *bits. Then there exists an integer* $C \leq 2^{n^2 L}$ *such that for each* $s \in \mathcal{S}$ *and* $a \in \mathcal{A}$, *both coordinates of* $\mathrm{pp}(s,a)$ *are multiples of* $1/C$, *and for each* $\alpha \in \mathbb{R}_+^2$, *both coordinates of* $f_{s,a}(\alpha)$ *are multiples of* $1/C$.

3.5 Algorithm and Analysis

Now we are ready to formally describe and analyze our full algorithm, Algorithm 1, which calls Algorithm 2 as a subroutine.

Below we analyze our algorithm. First we show that the binary search in Algorithm 1 is exact, in the sense that in line 18, q_ℓ and q_r are adjacent turning points to each other on $f_{s,a}$.

Lemma 6. *Fix a stochastic game* $(\mathcal{S}, \mathcal{A}, \mathrm{ap}, r_1, r_2, P)$, *where all parameters of the game can be encoded using* L *bits. In every execution of line 18 of Algorithm 1, assuming* $q_\ell = f_{s,a}(\ell)$ *and* $q_r = f_{s,a}(r)$ *(we will prove this later),* q_ℓ *and* q_r *are adjacent turning points to each other on* $f_{s,a}$.

Now we are ready to prove the key property of the algorithm, which is captured by the following lemma.

Lemma 7. *Fix a stochastic game* $(\mathcal{S}, \mathcal{A}, \mathrm{ap}, r_1, r_2, P)$, *where all parameters of the game can be encoded using* L *bits. The following statements regarding Algorithms 1 and 2 hold:*

- *For each* $s \in \mathcal{S}$, $a \in \mathcal{A}$ *and* $\alpha \in \mathbb{R}_+^2$, *if* $(s,a,\alpha) \in \mathcal{D}$ *then* $\mathcal{D}(s,a,\alpha) = f_{s,a}(\alpha)$.
- *For each* $s \in \mathcal{S}$ *and* $a \in \mathcal{A}$, $p_{s,a}$ *computed in Algorithm 1 is the same as* $\mathrm{pp}(s,a)$.

ALGORITHM 1: A polynomial-time algorithm for computing an SEFCE in turn-taking stochastic games.

Input: a turn-taking stochastic game $(\mathcal{S} = [n], \mathcal{A}, \mathrm{ap}, r_1, r_2, P)$.
Output: the leader's utility under an SEFCE, together with an implicit representation of an SEFCE in the input game.

1 create a data structure $\mathcal{D}$ that stores the results of all evaluations (used by `eval`);
2 **for** *each state* $s = n-1, n-2, \ldots, 1$ **do**
3 **if** $\mathrm{ap}(s) = 1$ **then**
4 **for** *each action* $a \in \mathcal{A}$ **do**
5 let $p_{s,a} \leftarrow \mathtt{eval}(s, a, (1, 0))$;
6 **end**
7 **end**
8 **else**
9 **for** *each action* $a \in \mathcal{A}$ **do**
10 let $\ell \leftarrow (0, 1)$, $r \leftarrow (1, 0)$, $q_\ell \leftarrow \mathtt{eval}(s, a, \ell)$, $q_r \leftarrow \mathtt{eval}(s, a, r)$;
11 if $(q_\ell)_2 < u^p(s)$, let $p_{s,a} \leftarrow (-n, -n)$;
12 if $(q_r)_2 \geq u^p(s)$, let $p_{s,a} \leftarrow q_r$;
13 **if** $(q_\ell)_2 \geq u^p(s)$ *and* $(q_r)_2 < u^p(s)$ **then**
14 **while** $\|\ell - r\|_1 \geq \frac{1}{3n \cdot 2^{2n^2 L}}$ **do**
15 let $q \leftarrow \mathtt{eval}(s, a, (\ell + r)/2)$ (see Algorithm 2);
16 let $\ell \leftarrow (\ell + r)/2$ if $q_2 \geq u^p(s)$, and $r \leftarrow (\ell + r)/2$ otherwise;
17 **end**
18 let $q_\ell \leftarrow \mathtt{eval}(s, a, \ell)$, $q_r \leftarrow \mathtt{eval}(s, a, r)$, $\ell_{s,a} \leftarrow \ell$, $r_{s,a} \leftarrow r$;
19 let $p_{s,a} \leftarrow \left(\frac{(q_\ell)_2 - u^p(s)}{(q_\ell)_2 - (q_r)_2} \cdot (q_r)_1 + \frac{u^p(s) - (q_r)_2}{(q_\ell)_2 - (q_r)_2} \cdot (q_\ell)_1, u^p(s) \right)$;
20 **end**
21 **end**
22 **end**
23 **end**
24 let opt $\leftarrow \max_{a \in \mathcal{A}} (p_{s_{\mathrm{init}}, a})_1$;
25 **return** opt, $\{p_{s,a}\}_{s,a}$ $\{\ell_{s,a}\}_{s,a}$, $\{r_{s,a}\}_{s,a}$, and $\mathcal{D}$;

Proof. Apply induction on s. When $s = n - 1$, it is easy to check $p_{s,a} = \mathrm{pp}(s, a)$ and $\mathcal{D}(s, a, \alpha) = f_{s,a}(\alpha)$. Now suppose the statements hold for all $s' > s$. Consider the first bullet point. For each $a \in \mathcal{A}$ and $\alpha \in \mathbb{R}^2_+$, observe that if $(s, a, \alpha) \in \mathcal{D}$, then it is computed precisely in the way given in Lemma 4. Given the induction hypothesis, this implies $\mathcal{D}(s, a, \alpha) = f_{s,a}(\alpha)$.

As for the second bullet point, consider 4 cases:

- $\mathrm{ap}(s) = 1$. The first bullet point immediately implies $p_{s,a} = \mathrm{pp}(s, a)$.
- $\mathrm{ap}(s) = 2$ and $(q_\ell)_2 < u^p(s)$ in line 11. Given the first bullet point, this means there is no point on $f_{s,a}$ whose y-coordinate is at least $u^p(s)$, and by definition, $\mathrm{pp}(s, a) = (-n, -n) = p_{s,a}$.
- $\mathrm{ap}(s) = 2$ and $(q_r)_2 \geq u^p(s)$ in line 12. Given the first bullet point, this means the entire $f_{s,a}$ is above $y = u^p(s)$, and by definition, $\mathrm{pp}(s, a) = f_{s,a}(0, 1) = p_{s,a}$.
- $\mathrm{ap}(s) = 2$, $(q_\ell)_2 \geq u^p(s)$, and $(q_r)_2 < u^p(s)$. This is the case where the binary search is executed. By Lemma 6 (and also given the first bullet point), q_ℓ and q_r are adjacent turning points on $f_{s,a}$. Moreover, $(q_\ell)_2 \geq u^p(s)$ and $(q_r)_2 < u^p(s)$. Then $\mathrm{pp}(s)$ must be the unique convex combination of q_ℓ and q_r whose y-coordinate is precisely $u^p(s)$, which is $p_{s,a}$ computed in line 19.

□

ALGORITHM 2: `eval`: A subroutine of Algorithm 1 that performs recursive evaluations as needed.

Input: a state s, an action a, a direction of evaluation α, all variables in Algorthm 1.
Output: $f_{s,a}(\alpha)$.

```
if s = s_term = n then return (0, 0);
if (s, a, α) ∉ D (i.e., if D(s, a, α) does not exist) then
    for s' = s + 1, ..., n do
        for a' ∈ A do
            let q_{s',a'} ← eval(s', a', α);
            if ap(s') = 2 and (q_{s',a'})_2 < u^p(s') then
                let q_{s',a'} ← p_{s',a'};
            end
        end
        let q_{s'} ← argmax_{q∈{q_{s',a'}}_{a'∈A}} α · q;
    end
    let D(s, a, α) ← (r_1(s, a), r_2(s, a)) + E_{s'~P(s,a)}[q_{s'}];
end
return D(s, a, α);
```

Now we can put everything together and prove the correctness and efficiency of Algorithm 1 (we will discuss how to decode the output of Algorithm 1 momentarily).

Theorem 2. *Algorithm 1 computes the leader's (player* 1*'s) utility in an SEFCE in time polynomial in n, m, and L.*

Proof. For correctness: By Lemma 7, for each $a \in \mathcal{A}$, $p_{s_{\text{init}},a} = \text{pp}(s_{\text{init}}, a)$, so $\text{opt} = \max_a (p_{s_{\text{init}},a})_1 = \max_a \text{pp}(s_{\text{init}}, a)_1$ is player 1's optimal utility induced by a feasible strategy (where player 2 is best responding), which is player 1's utility in an SEFCE.

For efficiency: We only need to bound the number of times that `eval` is called. Observe that the number of times that `eval` is called in Algorithm 1 is $O(nm \log(n2^{n^2L})) = O(n^2 mL \log n) = \text{poly}(n, m, L)$. As for recursive calls, observe that `eval` makes $O(nm)$ recursive calls only when (s, a, α) is not in $\mathcal{D}$ yet. So each tuple (s, a, α) may trigger $O(nm)$ calls in $\texttt{eval}(s, a, \alpha)$. Let $A = \{\alpha \mid (s, a, \alpha) \in \mathcal{D}\}$. Then the total number of recursive calls is at most $|\{(s, a, \alpha) \mid s \in \mathcal{S}, a \in \mathcal{A}, \alpha \in A\}| = O(nm|A|)$, so we only need to bound $|A|$. To this end, observe that for each $\alpha \in A$, there must be some $s \in \mathcal{S}$ and $a \in \mathcal{A}$ such that the binary search for (s, a) involves α. Each binary search involves $O(\log(n2^{n^2L})) = \text{poly}(n, L)$ directions, so the total number of directions involved in these binary searches is $\text{poly}(n, m, L)$. The latter is an upper bound of $|A|$. □

3.6 Decoding the Output Strategy

Now we discuss the final missing piece of our algorithm: extracting the strategy encoded in the output of Algorithm 1. We present a procedure, Algorithm 3, which, given the output of Algorithm 1, computes a random action for any given history-state pair. We will prove that the strategy implicitly given by Algorithm 3 is the one encoded in the output of Algorithm 1. In particular, it is feasible, and achieves the leader's utility in an SEFCE computed by Algorithm 1.

Theorem 3. *Algorithm 3 outputs a feasible strategy* Π *(restricted to admissible histories), which satisfies* $u_1^{\Pi}(\emptyset, s_{\text{init}}) = \text{opt}$*, where* opt *is the leader's utility in an SEFCE computed by Algorithm 1.*

Proof. First observe that Algorithm 3 does output a strategy restricted to admissible histories. In fact, it specifies a random action for each history-state pair, which can be viewed as a Bayesian

ALGORITHM 3: A procedure that decodes the output of Algorithm 1.

Input: A turn-taking stochastic game $(\mathcal{S}, \mathcal{A}, \mathrm{ap}, r_1, r_2, P)$, the output of Algorithm 1, a history $h = (s_1, a_1, \ldots, s_t, a_t)$, and a state s.

Output: $\pi(h, s)$, where $\pi \sim \Pi \mid h$, and Π is the strategy encoded in the output of Algorithm 1; -1 if h is not an admissible history under Π.

let $a \leftarrow \mathrm{argmax}_{a' \in \mathcal{A}} (p_{s,a'})_1$, $\alpha \leftarrow (1, 0)$, $q \leftarrow p_{s,a}$;
if $|h| = 0$ **return** a;
if $a_1 \neq a$, **return** -1;
for $i = 1, 2, \ldots, t-1$ **do**
  if $\mathrm{ap}(s_i) = 2$ *and* $q = p_{s_i,a_i}$ **then**
    let $\ell \leftarrow \ell_{s_i,a_i}$, $r \leftarrow r_{s_i,a_i}$, $a_\ell \leftarrow \mathrm{argmax}_{a' \in \mathcal{A}} \ell \cdot \max^2\{p_{s_{i+1},a'}, \mathcal{D}(s_{i+1}, a', \ell)\}$,
      $a_r \leftarrow \mathrm{argmax}_{a' \in \mathcal{A}} r \cdot \max^2\{p_{s_{i+1},a'}, \mathcal{D}(s_{i+1}, a', \ell)\}$;
    /* for two points q_1 and q_2, $\max^k\{q_1, q_2\}$ denotes the point with the larger k-th coordinate between the two */
    let $\alpha \leftarrow \ell$ if $a_{i+1} = a_\ell$;
    let $\alpha \leftarrow r$ if $a_{i+1} = a_r$;
    if $a_{i+1} \notin \{a_\ell, a_r\}$, **return** -1;
  end
  else
    let $a \leftarrow \mathrm{argmax}_{a' \in \mathcal{A}} \alpha \cdot \max^2\{p_{s_{i+1},a'}, \mathcal{D}(s_{i+1}, a', \alpha)\}$;
    if $a_{i+1} \neq a$ **return** -1;
  end
  let $q \leftarrow \max^2\{p_{s_{i+1},a_{i+1}}, \mathcal{D}(s_{i+1}, a_{i+1}, \alpha)\}$;
end
if $\mathrm{ap}(s_t) = 2$ *and* $q = p_{s_t,a_t}$ **then**
  let $\ell \leftarrow \ell_{s_i,a_i}$, $a_\ell \leftarrow \mathrm{argmax}_{a' \in \mathcal{A}} \ell \cdot \max^2\{p_{s,a'}, \mathcal{D}(s, a', \ell)\}$, $q_\ell \leftarrow \max^2\{p_{s,a_\ell}, \mathcal{D}(s, a_\ell, \ell)\}$; let
    $r \leftarrow r_{s_i,a_i}$, $a_r \leftarrow \mathrm{argmax}_{a' \in \mathcal{A}} r \cdot \max^2\{p_{s,a'}, \mathcal{D}(s, a', r)\}$, $q_r \leftarrow \max^2\{p_{s,a_r}, \mathcal{D}(s, a_r, r)\}$;
  let $a \leftarrow a_\ell$ with probability $\frac{q_r - q}{q_r - q_\ell}$, $a \leftarrow a_r$ with probability $\frac{q - q_\ell}{q_r - q_\ell}$;
end
else
  let $a \leftarrow \mathrm{argmax}_{a' \in \mathcal{A}} \alpha \cdot \max^2\{p_{s,a'}, \mathcal{D}(s, a', \alpha)\}$;
end
return a;

description of a randomized strategy Π. We need to show that $u_1^{\Pi}(\emptyset, s_{\mathrm{init}}) = \mathrm{opt}$, and for each $h \in \mathcal{H}^{\Pi}$, $s \in \mathcal{S}$ where $\mathrm{ap}(s) = 2$, and a such that $h + (s, a) \in \mathcal{H}^{\Pi}$,

$$v_2^{\Pi|h+(s,a)}(h, s) \geq u^p(s).$$

To this end, observe that the output strategy faithfully implements the corresponding point on the corresponding Pareto frontier curve. That is, for each $i \in \{1, 2\}$, $h \in \mathcal{H}^{\Pi}$, $s \in \mathcal{S}$ where $\mathrm{ap}(s) = 2$, and a such that $h + (s, a) \in \mathcal{H}^{\Pi}$,

$$v_i^{\Pi|h+(s,a)}(h, s) = \max^2\{p_{s,a}, \mathcal{D}(s, a, \alpha)\} = \max^2\{\mathrm{pp}(s, a), f_{s,a}(\alpha)\}.$$

(Recall that for two points q_1 and q_2, $\max^2\{q_1, q_2\}$ denotes the point with the larger y-coordinate between the two.) This can be proved inductively, and we omit the details (which are already quite repetitive at this point). Given the above correspondence, for the first condition, $u_1^{\Pi}(\emptyset, s_{\mathrm{init}}) = \max_{a' \in \mathcal{A}} (p_{s,a'})_1$, which is equal to opt. For the second condition, observe that $\max^2\{\mathrm{pp}(s, a), f_{s,a}(\alpha)\}$

is always a feasible point whose y-coordinate is at least $u^p(s)$, whenever $\mathrm{ap}(s) = 2$. This completes the proof. □

Finally, note that Algorithm 3 only specifies actions for admissible histories. For inadmissible histories, both players should follow the equilibrium π_2 that maximally punishes player 2 defined earlier.

4 APPROXIMATELY OPTIMAL EXTENSIVE-FORM CORRELATED EQUILIBRIA

Now we proceed to the computation of approximately optimal EFCE. We present a bi-criteria algorithm that, given an objective direction (i.e., a combination of the two players' utilities), computes an ε-EFCE whose objective value is at least that of the optimal EFCE minus ε, in time $\log(1/\varepsilon)$. The idea and structure of our algorithm for approximately optimal EFCE is overall quite similar to that for SEFCE. There are two key differences:

- Recall that for SEFCE, we optimize over strategies where player 2 is best responding. This reduces to optimizing over feasible strategies, where feasibility means that when player 2 is the acting player, their onward utility must be at least the utility under punishment. For ε-EFCE, both players need to be ε-best responding, which leads to a different definition for feasible strategies. The definition and structural properties of Pareto frontier curves also need to be modified accordingly. Such modifications lead to minor changes in the proofs of the structural properties and the algorithm.
- A more substantial difference is in the numerical resolution of the Pareto frontier curves. For SEFCE, the feasibility constraints are all in the same direction, i.e., parallel to the x-axis. This is no longer true for ε-EFCE, where the direction of the feasibility constraint in a state depends on the acting player. Such alternating constraints break the asymmetry between the two axes, which was crucial in the analysis of the numerical resolution of the Pareto frontier curves. As a result, a binary search with polynomially many iterations is no longer guaranteed to find the pivotal point exactly. Instead, the guarantee we have is that the error diminishes exponentially fast as the number of iterations grows. Importantly, this means inaccuracy in terms of both the objective value and the feasibility constraints. A careful analysis shows that the inaccuracy does not blow up too much as we approximately evaluate the Pareto frontier curves recursively.

4.1 Useful Facts

Before stating the full algorithm, we quickly state the new reduction from ε-EFCE to constrained planning, as well as modified definitions and properties of Pareto frontier curves. The proofs of these properties are similar to those of their counterparts for SEFCE.

Reduction to constrained planning.

Lemma 8. *Fix a stochastic game $(\mathcal{S}, \mathcal{A}, \mathrm{ap}, r_1, r_2, P)$. For any $\varepsilon \geq 0$ and Π under which both players are ε-best responding, both players are also ε-best responding under $\mathrm{pa}(\Pi, \{1,2\})$, and moreover, for each $i \in \{1,2\}$,*

$$u_i^{\Pi}(\emptyset, s_{\mathrm{init}}) = u_i^{\mathrm{pa}(\Pi,\{1,2\})}(\emptyset, s_{\mathrm{init}}).$$

Lemma 9. *Fix a stochastic game $(\mathcal{S}, \mathcal{A}, \mathrm{ap}, r_1, r_2, P)$. For any $\varepsilon \geq 0$ and randomized strategy Π, both players are ε-best responding under $\mathrm{pa}(\Pi, \{1,2\})$ if the following condition holds: For each admissible history $h \in \mathcal{H}^{\Pi}$, state $s \in \mathcal{S}$, and action $a \in \mathcal{A}$ such that $h + (s,a) \in \mathcal{H}^{\Pi}$,*

$$v_{\mathrm{ap}(s)}^{\Pi|(h+(s,a))}(h,s) \geq \max_{a' \in \mathcal{A}} \left(r_{\mathrm{ap}(s)}(s,a') + \mathop{\mathbb{E}}_{s' \sim P(s,a')} [u_{\mathrm{ap}(s)}^{\pi_{\mathrm{ap}(s)}}(h + (s,a'), s')] \right) - \varepsilon,$$

where $\pi_{\mathrm{ap}(s)}$ is the subgame perfect equilibrium when player $3-\mathrm{ap}(s)$ tries to minimize player $\mathrm{ap}(s)$'s utility, as defined above.

Theorem 4. *Fix a stochastic game $(\mathcal{S}, \mathcal{A}, \mathrm{ap}, r_1, r_2, P)$. For any $\varepsilon \geq 0$ and $(x, y) \in \mathbb{R}^2_+$, there exists a strategy Π under which both players are ε-best responding such that $u_1^{\Pi}(\emptyset, s_{\mathrm{init}}) \geq x$ and $u_2^{\Pi}(\emptyset, s_{\mathrm{init}}) \geq y$, if and only if there exists a strategy Π' such that $u_1^{\Pi'}(\emptyset, s_{\mathrm{init}}) \geq x$, $u_2^{\Pi'}(\emptyset, s_{\mathrm{init}}) \geq y$, and for each admissible history $h \in \mathcal{H}^{\Pi'}$, state $s \in \mathcal{S}$, and action $a \in \mathcal{A}$ such that $h + (s, a) \in \mathcal{H}^{\Pi'}$,*

$$v_{\mathrm{ap}(s)}^{\Pi'|(h+(s,a))}(h, s) \geq u^p(s) - \varepsilon.$$

Feasible strategies. We say a strategy Π is ε-feasible (we will omit ε when it is clear from the context) if it satisfies the condition in the corollary which involves the utility under punishment, i.e., for each $h \in \mathcal{H}^{\Pi}$, $s \in \mathcal{S}$, and $a \in \mathcal{A}$ such that $h + (s, a) \in \mathcal{H}^{\Pi}$,

$$v_{\mathrm{ap}(s)}^{\Pi|(h+(s,a))}(h, s) \geq u^p(s) - \varepsilon.$$

We say a strategy Π is ε-feasible after state s, if for each $h \in \mathcal{H}^{\Pi}$, $s' > s$, and $a \in \mathcal{A}$ such that $h + (s', a) \in \mathcal{H}^{\Pi}$,

$$v_{\mathrm{ap}(s')}^{\Pi|(h+(s',a))}(h, s') \geq u^p(s') - \varepsilon.$$

Pareto frontier curves. Again, we define the Pareto frontier curve $f_{s,a}$ (dependence on ε omitted) for a state-action pair (s, a) to be the curve capturing all Pareto-optimal pairs of onward utilities (assuming an empty history) for both players after playing action a in state s, induced by strategies that are feasible after s.

Lemma 10. *Fix a stochastic game $(\mathcal{S}, \mathcal{A}, \mathrm{ap}, r_1, r_2, P)$. For any state $s \in \mathcal{S}$ and action $a \in \mathcal{A}$, $f_{s,a}$ is always a concave curve.*

Pivotal points. Given an objective direction α_{obj}, we define the pivotal point $\mathrm{pp}(s, a)$ on $f_{s,a}$ for each (s, a) to be the farthest point (if there is one) along α_{obj} on $f_{s,a}$ such that $\mathrm{pp}(s, a)_{\mathrm{ap}(s)} \geq u^p(s)$. If such a point does not exist, then we let $\mathrm{pp}(s, a) = (-n, -n)$ (again, $-n$ is without loss of generality, and any quantity that is small enough would be consistent with our results).

Lemma 11. *Fix a stochastic game $(\mathcal{S}, \mathcal{A}, \mathrm{ap}, r_1, r_2, P)$. For any $s \in \mathcal{S}$, $a \in \mathcal{A}$, and $\alpha \in \mathbb{R}^2_+$,*

$$f_{s,a}(\alpha) = (r_1(s, a), r_2(s, a)) + \mathop{\mathbb{E}}_{s' \sim P(s,a)} [p_{s'}],$$

where for each $s' > s$ and $a' \in \mathcal{A}$,

$$p_{s',a'} = \begin{cases} f_{s',a'}(\alpha), & \text{if } f_{s',a'}(\alpha)_{\mathrm{ap}(s')} \geq u^p(s') \\ \mathrm{pp}(s', a'), & \text{otherwise,} \end{cases}$$

and for each $s' > s$,

$$p_{s'} = \mathop{\mathrm{argmax}}_{p \in \{p_{s',a'}\}_{a' \in \mathcal{A}}} p \cdot \alpha.$$

For EFCE, we need an approximate version of Lemma 11, which roughly says if we can approximately compute the pivotal points for all later state-action pairs, then approximately evaluating $f_{s,a}$ can be reduced to at most mn evaluations of curves for later state-action pairs. This is captured by the following claim, which is a direct corollary of Lemma 11.

Corollary 2. *Fix a stochastic game $(\mathcal{S}, \mathcal{A}, \mathrm{ap}, r_1, r_2, P)$. For any $s \in \mathcal{S}$, $a \in \mathcal{A}$, and $\alpha \in \mathbb{R}^2_+$,*

$$\alpha \cdot f_{s,a}(\alpha) \leq \alpha \cdot \left((r_1(s, a), r_2(s, a)) + \mathop{\mathbb{E}}_{s' \sim P(s,a)} [q_{s'}] \right) + \varepsilon,$$

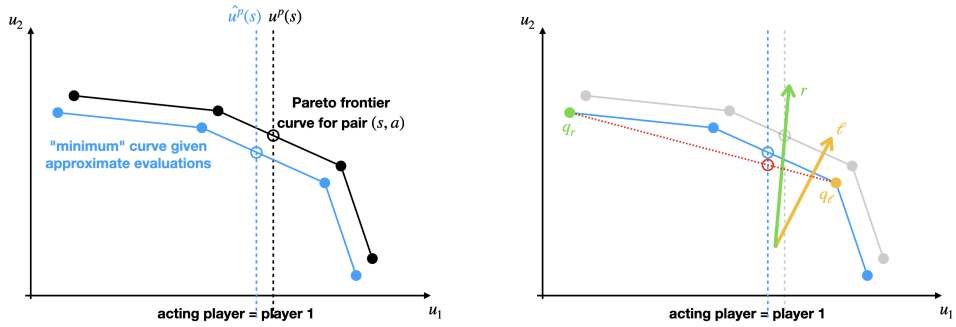

Fig. 2. Illustration of the proof of Lemma 12.

where for each $s' > s$, $q_{s'}$ satisfies

$$\alpha \cdot p_{s'} \leq \alpha \cdot q_{s'} + \varepsilon.$$

Here, $p_{s'}$ is defined in the same way as in Lemma 11.

4.2 Algorithm and Analysis

Now we are ready to present and analyze our algorithm for approximately optimal EFCE, Algorithm 4, which uses Algorithm 5 as a subroutine. We defer both these algorithms, as well as Algorithm 6 to be mentioned later, to Appendix B, since these algorithms are similar to their counterparts in Section 3.

The key differences between Algorithms 4 and 5, and Algorithms 1 and 2, are:

- Now we need to satisfy feasibility constraints in all states, whereas for SEFCE constraints exist only in states where the acting player is the follower.
- The binary search stops when the two directions are ε/n-close to each other, and in general, it only finds an approximate pivotal point as opposed to an exact one. Accordingly, we also allow inaccuracy in the feasibility constraints.

In order to analyze the algorithm, we first show that the binary search finds a point that is close to the actual pivotal point. We defer the proof of Lemma 12, as well as that of Lemma 13 below, to Appendix B. For an illustration of the proof of Lemma 12, see Figure 2.

Lemma 12. *Fix a stochastic game $(\mathcal{S}, \mathcal{A}, \mathrm{ap}, r_1, r_2, P)$. In Algorithm 4, assuming for each $s \in \mathcal{S}$, $a \in \mathcal{A}$ and $\alpha \in \mathbb{R}^2_+$ where $\|\alpha\|_1 \leq 1$, $\mathcal{D}(s, a, \alpha)$ satisfies $\alpha \cdot \mathcal{D}(s, a, \alpha) \geq \alpha \cdot f_{s,a}(\alpha) - \frac{n-s-1}{n} \cdot \varepsilon$, $p_{s,a}$ computed in line 15 satisfies $(p_{s,a})_{3-\mathrm{ap}(s)} \geq \mathrm{pp}(s, a)_{3-\mathrm{ap}(s)} - \frac{n-s}{n} \cdot \varepsilon$.*

We then establish the key properties of Algorithm 4 — essentially an approximate version of Lemma 7.

Lemma 13. *Fix a stochastic game $(\mathcal{S}, \mathcal{A}, \mathrm{ap}, r_1, r_2, P)$. The following statements regarding Algorithms 4 and 5 hold: For each $s \in \mathcal{S}$, $a \in \mathcal{A}$ and $\alpha \in \mathbb{R}^2_+$ where $\|\alpha\|_1 = 1$,*

- *if $(s, a, \alpha) \in \mathcal{D}$, then $\alpha \cdot \mathcal{D}(s, a, \alpha) \geq \alpha \cdot f_{s,a}(\alpha) - \frac{n-s-1}{n} \cdot \varepsilon$;*
- *$\alpha \cdot (p_{s,a}) \geq \alpha \cdot \mathrm{pp}(s, a) - \frac{n-s}{n} \cdot \varepsilon$.*

Given Lemma 13, it is not hard to prove the correctness and efficiency of Algorithm 4.

Theorem 5. *Algorithm 4 runs in time polynomial in n, m, and $\log(1/\varepsilon)$ (assuming all arithmetic operations take constant time), and the output* opt *satisfies:*

- opt *is smaller than the optimal objective value of any EFCE by at most* ε*, i.e.,*

$$\text{opt} \geq \max_{\Pi \text{ is an EFCE}} \alpha_{\text{obj}} \cdot (u_1^{\Pi}(\emptyset, s_{\text{init}}), u_2^{\Pi}(\emptyset, s_{\text{init}})) - \varepsilon.$$

- *There exists an* ε*-EFCE whose objective value is* opt.

The proof of the above theorem is similar to that of Theorem 2, with one exception: We prove the second bullet point by giving an algorithm that constructs an ε-EFCE whose objective value is opt. The algorithm (Algorithm 6 in Appendix B) is overall quite similar to Algorithm 3. As such, Algorithm 6 only specifies actions for admissible histories. For inadmissible histories, both players should follow the equilibrium π_i that maximally punishes player i defined earlier, where i is the player who first deviates. The proof of the following claim is similar to that of Theorem 3.

Theorem 6. *Algorithm 6 outputs a feasible strategy* Π *(restricted to admissible histories), which satisfies* $\alpha_{\text{obj}} \cdot (u_1^{\Pi}(\emptyset, s_{\text{init}}), u_2^{\Pi}(\emptyset, s_{\text{init}})) = \text{opt}$*, where* opt *is the approximately optimal objective value computed by Algorithm 4.*

ACKNOWLEDGMENTS

Zhang and Conitzer thank the Cooperative AI Foundation, Polaris Ventures (formerly the Center for Emerging Risk Research) and Jaan Tallinn's donor-advised fund at Founders Pledge for financial support. Cheng is supported in part by NSF Award CCF-2307106.

REFERENCES

Branislav Bošanskỳ, Simina Brânzei, Kristoffer Arnsfelt Hansen, Troels Bjerre Lund, and Peter Bro Miltersen. 2017. Computation of Stackelberg equilibria of finite sequential games. *ACM Transactions on Economics and Computation (TEAC)* 5, 4 (2017), 1–24.

Charles L Bouton. 1901. Nim, a game with a complete mathematical theory. *The Annals of Mathematics* 3, 1/4 (1901), 35–39.

Jiri Cermak, Branislav Bosansky, Karel Durkota, Viliam Lisy, and Christopher Kiekintveld. 2016. Using correlated strategies for computing stackelberg equilibria in extensive-form games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 30.

Xi Chen, Xiaotie Deng, and Shang-Hua Teng. 2009. Settling the complexity of computing two-player Nash equilibria. *Journal of the ACM (JACM)* 56, 3 (2009), 1–57.

Vincent Conitzer and Dmytro Korzhyk. 2011. Commitment to Correlated Strategies. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*. San Francisco, CA, USA, 632–637.

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.

Vincent Conitzer and Tuomas Sandholm. 2008. New Complexity Results about Nash Equilibria. *Games and Economic Behavior* 63, 2 (2008), 621–641.

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.

Gabriele Farina, Chun Kai Ling, Fei Fang, and Tuomas Sandholm. 2019. Correlation in extensive-form games: Saddle-point formulation and benchmarks. *Advances in Neural Information Processing Systems* 32 (2019).

Gabriele Farina and Tuomas Sandholm. 2020. Polynomial-time computation of optimal correlated equilibria in two-player extensive-form games with public chance moves and beyond. *Advances in Neural Information Processing Systems* 33 (2020), 19609–19619.

Itzhak Gilboa and Eitan Zemel. 1989. Nash and correlated equilibria: Some complexity considerations. *Games and Economic Behavior* 1 (1989), 80–93.

Wan Huang and Bernhard von Stengel. 2008. Computing an extensive-form correlated equilibrium in polynomial time. In *International Workshop on Internet and Network Economics*. Springer, 506–513.

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*. 83–92.

Joshua Letchford, Liam MacDermed, Vincent Conitzer, Ronald Parr, and Charles L Isbell. 2012. Computing optimal strategies to commit to in stochastic games. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Christos H Papadimitriou. 2007. The complexity of finding Nash equilibria. *Algorithmic game theory* 2 (2007), 30.

Bernhard von Stengel and Françoise Forges. 2008. Extensive-form correlated equilibrium: Definition and computational complexity. *Mathematics of Operations Research* 33, 4 (2008), 1002–1022.

Bernhard von Stengel and Shmuel Zamir. 2010. Leadership Games with Convex Strategy Sets. *Games and Economic Behavior* 69 (2010), 446–457.

Brian Hu Zhang, Gabriele Farina, Andrea Celli, and Tuomas Sandholm. 2022c. Optimal correlated equilibria in general-sum extensive-form games: Fixed-parameter algorithms, hardness, and two-sided column-generation. In *Proceedings of the 23rd ACM Conference on Economics and Computation.* 1119–1120.

Brian Hu Zhang and Tuomas Sandholm. 2022. Polynomial-Time Optimal Equilibria with a Mediator in Extensive-Form Games. In *Advances in Neural Information Processing Systems.*

Hanrui Zhang, Yu Cheng, and Vincent Conitzer. 2022a. Efficient Algorithms for Planning with Participation Constraints. In *Proceedings of the 23rd ACM Conference on Economics and Computation.* 1121–1140.

Hanrui Zhang, Yu Cheng, and Vincent Conitzer. 2022b. Planning with Participation Constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence.*

A OMITTED PROOFS IN SECTION 3

Proof of Lemma 1. For brevity let $\Pi' = \mathrm{pa}(\Pi, \{2\})$. We first prove Π and Π' induce the same utility for the leader. Observe that Π and Π' share the same set of admissible histories, i.e., $\mathcal{H}^{\Pi} = \mathcal{H}^{\Pi'}$. Moreover, for any $h \in \mathcal{H}^{\Pi}$, $\mathrm{pa}(\Pi \mid h, \{2\}) = \Pi' \mid h$. Given the above, we have a stronger claim: For each $i \in \{1, 2\}$, h and $h' \in \mathcal{H}^{\Pi}$, and $s \in \mathcal{S}$,

$$v_i^{\Pi|h}(h', s) = v_i^{\Pi'|h}(h', s).$$

This can be verified by expanding both sides using the definition of v_i, coupling $\pi \sim \Pi \mid h$ with $\pi' = \mathrm{pa}(\pi, \{2\}) \sim \Pi' \mid h$, and checking π and π' always induce the same play given h' and s since h' is admissible. Setting $h = h' = \emptyset$ and $s = s_{\text{init}}$, this immediately implies that both players have the same utilities under Π and Π'.

We now prove that the follower is best responding under Π'. Consider any admissible history $h \in \mathcal{H}^{\Pi}$, state s where $\mathrm{ap}(s) = 2$, action a where $h + (s, a) \in \mathcal{H}^{\Pi}$, and deterministic strategy π'' where $\pi''(h, s) \neq a$ (we reserve π' for later use). Since player 2 is best responding under Π, by definition we have

$$v_2^{\Pi|(h+(s,a))}(h, s) \geq \mathop{\mathbb{E}}_{\pi \sim \Pi|(h+(s,a))} \left[u_2^{(2:\pi'', 1:\pi)}(h, s) \right].$$

We already know that

$$v_2^{\Pi|(h+(s,a))}(h, s) = v_2^{\Pi'|(h+(s,a))}(h, s).$$

So we only need to prove that

$$\mathop{\mathbb{E}}_{\pi \sim \Pi|(h+(s,a))} \left[u_2^{(2:\pi'', 1:\pi)}(h, s) \right] \geq \mathop{\mathbb{E}}_{\pi' \sim \Pi'|(h+(s,a))} \left[u_2^{(2:\pi'', 1:\pi')}(h, s) \right].$$

Again we couple $\pi \sim \Pi \mid (h + (s, a))$ with $\pi' = \mathrm{pa}(\pi) \sim \Pi' \mid (h + (s, a))$, so we only need to compare $u_2^{(2:\pi'', 1:\pi)}(h, s)$ and $u_2^{(2:\pi'', :\pi')}(h, s)$. Both quantities involve summing over the rewards in up to n steps and taking expectations over random transitions. To this end, observe that the first steps in both quantities are always the same (player 2 playing $\pi''(h, s)$), so we further couple them. Now we only need to prove in the subgame induced by $h + (s, \pi''(h, s))$ and $s' \sim P(s, \pi''(h, s))$, player 2's utility when the other player follows π is at least player 2's utility when the other player follows π'. This follows almost directly from the definition of π': Restricted to this subgame and player 1, π' behaves identically as π_2, which is a subgame-perfect equilibrium when the other player tries to minimize player 2's utility. In other words, fixing player 2's strategy (which is π''), player 2's utility in this subgame against π is no smaller than player 2's utility against π'. Now taking the expectations over π, π', and s' gives the desired inequality. □

Proof of Lemma 2. Let $\Pi' = \mathrm{ap}(\Pi, \{2\})$. We only need to verify that if Π satisfies the condition in the lemma, then for any admissible history $h \in \mathcal{H}^{\Pi'}$, state s where $\mathrm{ap}(s) = 2$, action a where

$h + (s, a) \in \mathcal{H}^{\Pi'}$, and deterministic strategy π'' where $\pi''(h, s) \neq a$,

$$v_2^{\Pi|(h+(s,a))}(h, s) \geq \mathop{\mathbb{E}}_{\pi' \sim \Pi'|(h+(s,a))} \left[u_2^{(2:\pi'',1:\pi')}(h, s) \right].$$

In particular, we only need to show that the right hand side of the above inequality is upper bounded by

$$\max_{a' \in \mathcal{A}} \left(r_2(s, a') + \mathop{\mathbb{E}}_{s' \sim P(s,a')} [u_2^{\pi_2}(h + (s, a'), s')] \right).$$

Again, this follows almost directly from the definition of π': Restricted to this subgame and player 1, π' behaves identically as π_2, which is a subgame-perfect equilibrium when the other player tries to minimize player 2's utility. In other words, player 2's utility in this subgame against π' is at most

$$r_2(s, a'') + \mathop{\mathbb{E}}_{s'' \sim P(s,a'')} [u_2^{\pi_2}(h + (s, a''), s'')],$$

where $a'' = \pi''(h, s)$. This is clearly upper bounded by

$$\max_{a' \in \mathcal{A}} \left(r_2(s, a') + \mathop{\mathbb{E}}_{s' \sim P(s,a')} [v_2^{\pi_2}(h + (s, a'), s')] \right),$$

since the latter is obtained by taking the maximum over a'. This finishes the proof. □

Proof of Lemma 3. We only need to prove $F_{s,a}$ is convex. Consider any two points (x_1, y_1) and (x_2, y_2) in the feasible region $F_{s,a}$, and feasible-after-s strategies Π_1 and Π_2 that induced these points. Without loss of generality, suppose $\Pi_1 = \Pi_1 \mid (s, a)$ (otherwise let $\Pi_1 \leftarrow \Pi_1 \mid (s, a)$) and $\Pi_2 = \Pi_2 \mid (s, a)$. For any $\alpha \in (0, 1)$, the strategy $\Pi = \alpha \cdot \Pi_1 + (1 - \alpha) \cdot \Pi_2$ obtained by running Π_1 with probability α and Π_2 with probability $1 - \alpha$ gives utilities $\alpha \cdot (x_1, y_1) + (1 - \alpha) \cdot (x_2, y_2)$ in state s after playing action a. We only need to argue that $\Pi = \alpha \cdot \Pi_1 + (1 - \alpha) \cdot \Pi_2$ is feasible after s.

Consider any $h \in \mathcal{H}^{\Pi} = \mathcal{H}^{\Pi_1} \cup \mathcal{H}^{\Pi_2}$, $s' > s$ where $\mathrm{ap}(s') = 2$, and $a' \in \mathcal{A}$ such that $h + (s', a') \in \mathcal{H}^{\Pi} = \mathcal{H}^{\Pi_1} \cup \mathcal{H}^{\Pi_2}$. We only need to show

$$v_2^{\Pi|(h+(s',a'))}(h, s') \geq u^p(s').$$

Observe that there is some $\beta \in [0, 1]$ such that

$$\Pi \mid (h + (s', a')) = \beta \cdot (\Pi_1 \mid (h + (s', a'))) + (1 - \beta) \cdot (\Pi_2 \mid (h + (s', a'))),$$

where β is not necessarily equal to α due to conditioning. This means

$$v_2^{\Pi|(h+(s',a'))}(h, s') = \beta \cdot v_2^{\Pi_1|(h+(s',a'))}(h, s') + (1 - \beta) \cdot v_2^{\Pi_2|(h+(s',a'))}(h, s').$$

Since Π_1 and Π_2 are both feasible after s, we have

$$\beta \cdot v_2^{\Pi_1|(h+(s',a'))}(h, s') + (1 - \beta) \cdot v_2^{\Pi_1|(h+(s',a'))}(h, s') \geq \beta \cdot u^p(s') + (1 - \beta) \cdot u^p(s') = u^p(s').$$

This finishes the proof. □

Proof of Lemma 4. We first prove $\mathbb{E}_{s' \sim P(s,a)} [p_{s'}] \in f_{s,a}$, and

$$\alpha \cdot f_{s,a}(\alpha) \geq \alpha \cdot \left((r_1(s, a), r_2(s, a)) + \mathop{\mathbb{E}}_{s' \sim P(s,a)} [p_{s'}] \right).$$

Let $\Pi_{s'}$ and $a_{s'}$ be the strategy and action corresponding to $p_{s'}$ for each $s' > s$, respectively. Clearly $\Pi_{s'}$ is feasible after s', and by construction the constraint in s' (if there is one) is also satisfied by $\Pi_{s'}$. Moreover, for each $i \in \{1, 2\}$,

$$u_i^{\Pi_{s'}|(s',a_{s'})}(\emptyset, s') = (p_{s'})_i.$$

Now we only need to construct a strategy Π satisfying: (1) Π is feasible after s, and (2) for each $i \in \{1, 2\}$,

$$u_i^{\Pi|(s,a)}(\emptyset, s) = \mathop{\mathbb{E}}_{s' \sim P(s,a)} [(p_{s'})_i] .$$

This is achieved by the following construction: Draw $\pi_{s'} \sim \Pi_{s'} \mid (s', a_{s'})$ for each $s' > s$. Let $\pi(\emptyset, s) = a$. For each (h, s'') where $h = (s_1, a_1, \ldots, s_t, a_t)$, let

$$\pi(h, s) = \pi_{s_1}((s_2, a_2, \ldots, s_t, a_t), s'').$$

Let Π be the distribution of π. One can check Π satisfies the two desired conditions.

Now we prove

$$\alpha \cdot f_{s,a}(\alpha) \leq \alpha \cdot \left((r_1(s, a), r_2(s, a)) + \mathop{\mathbb{E}}_{s' \sim P(s,a)} [p_{s'}] \right).$$

Suppose otherwise. Let Π be a strategy that is feasible after s that implements $f_{s,a}(\alpha)$. There must be some $s' > s$ such that

$$\alpha \cdot (v_1^{\Pi|(s,a)}(\emptyset, s'), v_2^{\Pi|(s,a)}(\emptyset, s')) > \alpha \cdot p_{s'}.$$

This means there exists a strategy $\Pi_{s'} = \Pi \mid (s, a)$ such that

$$\alpha \cdot \mathop{\mathbb{E}}_{\pi_{s'} \sim \Pi_{s'}} \left[(v_1^{\Pi_{s'}|(s', \pi_{s'}(\emptyset, s'))}(\emptyset, s'), v_2^{\Pi_{s'}|(s', \pi_{s'}(\emptyset, s'))}(\emptyset, s')) \right] > \alpha \cdot p_{s'},$$

which means there exists some a' such that

$$\alpha \cdot (v_1^{\Pi_{s'}|(s', a')}(\emptyset, s'), v_2^{\Pi_{s'}|(s', a')}(\emptyset, s')) > \alpha \cdot p_{s', a'}.$$

This contradicts the definition of $p_{s', a'}$, since $\Pi_{s'}$ is feasible after s. □

Proof of Lemma 5. We prove the claim inductively. Consider $s = n - 1$ first. For each $a \in \mathcal{A}$, $f_{s,a}$ consists of a single point $(r_1(s, a), r_2(s, a)$, so the claim holds trivially.

Now fix some $s \in [n - 1]$ and suppose the claim holds for all $s' > s$. Fix an action $a \in \mathcal{A}$ and consider the first bullet point. By Lemma 4, for any $\alpha \in \mathbb{R}_+^2$,

$$f_{s,a}(\alpha) = (r_1(s, a) + r_2(s, a)) + \mathop{\mathbb{E}}_{s' \sim P(s,a)} [p_{s'}] = (r_1(s, a) + r_2(s, a)) + \sum_{s' > s} P(s, a, s') \cdot p_{s'}.$$

Here, for each $s' > s$, $(p_{s'})_2$ is a multiple of $2^{-(n-s-1)L}$, and $(p_{s'})_1$ is a multiple of C_{s+1} because of the induction hypothesis. Since $r_1(s, a)$, $r_2(s, a)$ and $P(s, a, s')$ are multiples of 2^{-L}, $(f_{s,a}(\alpha))_2$ must be a multiple of $2^{-(n-s)L}$, and $(f_{s,a}(\alpha))_1$ must be a multiple of $1/(2^L C_{s+1})$.

Now if $\mathrm{ap}(s) = 2$, we need to further consider the second bullet point. When there is no point on $f_{s,a}$ whose y-axis is precisely $u^p(s)$, we know $\mathrm{pp}(s, a)$ is either $(-n, -n)$ or some turning point on $f_{s,a}$. In both cases, $(\mathrm{pp}(s, a))_2$ is a multiple of $2^{-(n-s)L}$, and $(\mathrm{pp}(s, a))_1$ is a multiple of $1/(2^L C_{s+1})$. Alternatively, when there is a point on $f_{s,a}$ whose y-axis is precisely $u^p(s)$, this point must be $\mathrm{pp}(s, a)$. Moreover, there exist two turning points p_1 and p_2 on $f_{s,a}$ such that $\mathrm{pp}(s, a)$ is the unique convex combination of p_1 and p_2 whose y-axis is $u^p(s)$. That is,

$$\mathrm{pp}(s, a) = \left(\frac{(p_1)_2 - u^p(s)}{(p_1)_2 - (p_2)_2} \cdot (p_2)_1 + \frac{u^p(s) - (p_2)_2}{(p_1)_2 - (p_2)_2} \cdot (p_1)_1, u^p(s) \right).$$

Here, $u^p(s)$ is a multiple of $2^{-(n-s)L}$ (we will prove this later), so $(\mathrm{pp}(s, a))_2)$ is a multiple of $2^{-(n-s)L}$. $(p_1)_1$ and $(p_2)_1$ are multiples of $1/(2^L C_{s+1})$. As for the coefficients of $(p_1)_1$ and $(p_2)_1$, observe that $(p_1)_2$ and $(p_2)_2$ are multiples of $2^{-(n-s)L}$, and they are between 0 and $n - s$. So there must be some integer $k \leq 2^{(n-s)L} \cdot (n - s)$ such that both coefficients are multiples of $1/k$. As a result, $(\mathrm{pp}(s, a))_1$ is a multiple of $1/(2^L C_{s+1} k)$, and we can let $C_s = 2^L C_{s+1} k$, which satisfies

$$C_s \leq 2^L \cdot 2^{(n-s-1)(n+1)L} \cdot 2^{(n-s)L} \cdot (n - s) \leq 2^{(n-s-1)(n+1)L} \cdot 2^{(n+1)L} \leq 2^{(n-s)(n+1)L}.$$

Finally we quickly argue that when $\mathrm{ap}(s) = 2$, $u^p(s)$ is a multiple of $2^{-(n-s)L}$. Recall that $u^p(s)$ is player 2's maximum onward utility in the subgame induced by s when the other player tries to minimize player 2's utility. Without loss of generality, the equilibrium strategy is deterministic and history-independent (such a strategy can be computed by backward induction, for example). Again we can bound player 2's onward utility inductively. In state $n-1$, player 2's onward utility must be $r_2(n-1, a)$ for some action a, which means it is a multiple of 2^{-L}. In state $n-2$, player 2's onward utility can be written as the sum of $r_2(n-2, a)$ for some action a, and the product of $P(n-2, a, n-1)$ and player 2's utility in state $n-1$. So this utility must be a multiple of 2^{-2L}. Repeating this argument for each s, we can show that player 2's utility in each state s is a multiple of $2^{(n-s)L}$. This concludes the proof. □

Proof of Lemma 6. Without loss of generality suppose $\ell_1 \le \frac{1}{2}$ (otherwise we can flip the two axes and apply the same argument). Since $\|\ell - r\|_1 < 1/(3n \cdot 2^{2n^2L})$, we also have $r_1 \le \frac{2}{3}$. Suppose otherwise, i.e., there is another point $q \in f_{s,a}$ between q_ℓ and q_r, such that q is strictly above the line defined by q_ℓ and q_r, i.e.,

$$\frac{q_2 - (q_\ell)_2}{q_1 - (q_\ell)_1} > \frac{(q_r)_2 - q_2}{(q_r)_1 - q_1}.$$

Recall that $\max\{|q_\ell|, |q_r|, |q|\} \le n$. Moreover, by Corollary 1, all quantities in the above inequality are multiples of $1/C$, where $C \le 2^{n^2L}$. Observe that

$$\frac{q_2 - (q_\ell)_2}{q_1 - (q_\ell)_1} - \frac{(q_r)_2 - q_2}{(q_r)_1 - q_1} = \frac{C^2((q_2 - (q_\ell)_2)((q_r)_1 - q_1) - ((q_r)_2 - q_2)(q_1 - (q_\ell)_1))}{C^2(q_1 - (q_\ell)_1)((q_r)_1 - q_1)}.$$

Here, both the numerator and the denominator are integers, and the denominator is no larger than $C^2 \cdot n \le n \cdot 2^{2n^2L}$. Since the fraction is strictly positive, we must have

$$\frac{q_2 - (q_\ell)_2}{q_1 - (q_\ell)_1} - \frac{(q_r)_2 - q_2}{(q_r)_1 - q_1} \ge \frac{1}{n \cdot 2^{2n^2L}}.$$

On the other hand, since $q_\ell = f_{s,a}(\ell)$ and $q_r = f_{s,a}(r)$, we have

$$-\frac{\ell_1}{\ell_2} \ge \frac{q_2 - (q_\ell)_2}{q_1 - (q_\ell)_1} \quad \text{and} \quad -\frac{r_1}{r_2} \le \frac{(q_r)_2 - q_2}{(q_r)_1 - q_1}.$$

This implies

$$\frac{r_1}{r_2} - \frac{\ell_1}{\ell_2} \ge \frac{1}{n \cdot 2^{2n^2L}}.$$

So we have

$$\begin{aligned}
\|\ell - r\|_1 = 2(r_1 - \ell_1) &\ge 2r_2 \cdot \frac{r_1 - \ell_1}{r_2} \ge 2r_2 \cdot \frac{r_1}{r_2} - 2\ell_2 \cdot \frac{\ell_1}{\ell_2} \\
&= 2\ell_2 \cdot \frac{r_1}{r_2} + 2(r_2 - \ell_2) \cdot \frac{r_1}{r_2} - 2\ell_2 \cdot \frac{\ell_1}{\ell_2} \\
&\ge \frac{2\ell_2}{n \cdot 2^{2n^2L}} - \|\ell - r\|_1 \cdot \frac{r_1}{r_2}.
\end{aligned}$$

Recall that without loss of generality, $\ell_1 \le \frac{1}{2}$ (so $\ell_2 \ge \frac{1}{2}$) and $r_1 \le \frac{2}{3}$ (so $r_2 \ge \frac{1}{3}$). Plugging these in and rearranging terms, the above inequality implies

$$3\|\ell - r\|_1 \ge \|\ell - r\|_1 + \frac{r_1}{r_2} \cdot \|\ell - r\|_1 \ge \frac{2\ell_2}{n \cdot 2^{2n^2L}} \ge \frac{1}{n \cdot 2^{2n^2L}} \implies \|\ell - r\|_1 \ge \frac{1}{3n \cdot 2^{2n^2L}},$$

a contradiction. □

B OMITTED ALGORITHMS AND PROOFS IN SECTION 4

Proof of Lemma 12. Without loss of generality suppose $\mathrm{ap}(s) = 1$. We show this in two steps. First imagine the "minimum" curve possible given approximate evaluations satisfying the condition stated in the lemma. This is the curve that the binary search actually operates on in the worst case. As illustrated in the left subfigure of Figure 2, this minimum curve is the blue one, which is lower than the actual (black) curve at most by $\frac{n-s-1}{n} \cdot \varepsilon$ in every direction. Also recall that the blue dashed line is obtained by shifting the black dashed line to the left by $\frac{n-s-1}{n} \cdot \varepsilon$. These facts imply that the x-coordinate of the blue hollow point is smaller than that of the pivotal point (the black hollow point) by at most $\frac{n-s-1}{n} \cdot \varepsilon$.

Now consider how well the binary search approximates the blue hollow point. Suppose ℓ, r, q_ℓ and q_r are as illustrated in the right subfigure of Figure 2. We need to bound the distance between the blue hollow point and the red one. Recall that $\|\ell - r\|_1 < \frac{\varepsilon}{10n^2}$, which means the angle θ between ℓ and r is smaller than $\frac{\varepsilon}{2n^2}$. Moreover, observe that θ upper bounds the sum of the two acute angles in the triangle containing the red segment, so the angle at q_ℓ is at most $\theta \le \frac{\varepsilon}{2n^2}$. This implies that the distance between the two points we care about is at most the length of the segment to the left of q_ℓ, times $\sin\theta$. The length of the segment is at most $\sqrt{2} \cdot n$, and $\sin\theta \le \theta \le \frac{\varepsilon}{2n^2}$, so the distance is at most ε/n. Putting the two parts together, we conclude that $(p_{s,a})_{3-\mathrm{ap}(s)} \ge \mathsf{pp}(s,a)_{3-\mathrm{ap}(s)} - \frac{n-s}{n}(s)$. □

Proof of Lemma 13. Apply induction on $s = n-1, \ldots, 1$. When $s = n-1$, the first bullet point holds because $\mathtt{eval}(s, a, \alpha)$ is alwaus exact. Lemma 12 then implies the second bullet point. Now fix some s and suppose the two bullet points hold for all $s' > s$. Consider lines 3-12 in Algorithm 5, where $\mathcal{D}(s, a, \alpha)$ is recursively computed. Let $q^*_{s',a'} = \mathrm{argmax}_{q \in f_{s',a'} : q_{\mathrm{ap}(s')} \ge u^p(s')} \alpha \cdot q$, and $q^*_{s'} = \mathrm{argmax}_{q \in \{q^*_{s',a'}\}_{a'}} \alpha \cdot q$. By the induction hypothesis, we have

$$\alpha \cdot q_{s',a'} \ge \alpha \cdot q^*_{s',a'} - \frac{n-s'}{n} \cdot \varepsilon \ge \alpha \cdot q^*_{s',a'} - \frac{n-s-1}{n} \cdot \varepsilon.$$

As a result, we have $\alpha \cdot q_{s'} \ge \alpha \cdot q^*_{s'} - \frac{n-s-1}{n} \cdot \varepsilon$, and therefore $\alpha \cdot \mathcal{D}(s, a, \alpha) \ge \alpha \cdot f_{s,a}(\alpha) - \frac{n-s-1}{n} \cdot \varepsilon$. Lemma 12 then implies the second bullet point. □

ALGORITHM 4: An algorithm for computing an approximately optimal ε-EFCE in turn-taking stochastic games, in time $\text{poly}(n, m, \log(1/\varepsilon))$.

Input: a turn-taking stochastic game $(\mathcal{S} = [n], \mathcal{A}, \text{ap}, r_1, r_2, P)$, an objective direction α_{obj} where $\|\alpha_{\text{obj}}\|_1 \leq 1$, a desired accuracy ε.

Output: an approximately optimal objective value under EFCE, together with an implicit representation of an ε-EFCE achieving the approximate objective value in the input game.

create a data structure $\mathcal{D}$ that stores the results of all evaluations (used by `eval`);
for each state $s \in \mathcal{S}$, let $\hat{u^p}(s) \leftarrow u^p(s) - \frac{n-s-1}{n} \cdot \varepsilon$;
for *each state* $s = n-1, n-2, \ldots, 1$ **do**
  for *each action* $a \in \mathcal{A}$ **do**
    let $(\ell, r) \leftarrow ((1,0), \alpha_{\text{obj}})$ if $\text{ap}(s) = 1$, and $(\ell, r) \leftarrow ((0,1), \alpha_{\text{obj}})$ if $\text{ap}(s) = 2$;
    let $q_\ell \leftarrow \texttt{eval}(s, a, \ell)$, $q_r \leftarrow \texttt{eval}(s, a, r)$;
    if $(q_\ell)_{\text{ap}(s)} < \hat{u^p}(s)$, let $p_{s,a} \leftarrow (-n, -n)$;
    if $(q_r)_{\text{ap}(s)} \geq \hat{u^p}(s)$, let $p_{s,a} \leftarrow q_r$;
    if $(q_\ell)_{\text{ap}(s)} \geq \hat{u^p}(s)$ *and* $(q_r)_{\text{ap}(s)} < \hat{u^p}(s)$ **then**
      while $\|\ell - r\|_1 \geq \frac{\varepsilon}{10n^2}$ **do**
        let $q \leftarrow \texttt{eval}(s, a, (\ell + r)/2)$ (see Algorithm 5);
        let $\ell \leftarrow (\ell + r)/2$ if $q_{\text{ap}(s)} \geq \hat{u^p}(s)$, and $r \leftarrow (\ell + r)/2$ otherwise;
      end
      let $q_\ell \leftarrow \texttt{eval}(s, a, \ell)$, $q_r \leftarrow \texttt{eval}(s, a, r)$, $\ell_{s,a} \leftarrow \ell$, $r_{s,a} \leftarrow r$;
      let $p_{s,a} \leftarrow \frac{(q_\ell)_{\text{ap}(s)} - \hat{u^p}(s)}{(q_\ell)_{\text{ap}(s)} - (q_r)_{\text{ap}(s)}} \cdot q_r + \frac{\hat{u^p}(s) - (q_r)_{\text{ap}(s)}}{(q_\ell)_{\text{ap}(s)} - (q_r)_{\text{ap}(s)}} \cdot q_\ell$;
    end
  end
end
let opt $\leftarrow \max_{a \in \mathcal{A}} (p_{s_{\text{init}}, a})_1$;
return opt, $\{p_{s,a}\}_{s,a}$ $\{\ell_{s,a}\}_{s,a}$, $\{r_{s,a}\}_{s,a}$, and $\mathcal{D}$;

ALGORITHM 5: `eval`: A subroutine of Algorithm 4 that performs approximate recursive evaluations as needed.

Input: a state s, an action a, a direction of evaluation α, all variables in Algorthm 4.

Output: an approximation of $f_{s,a}(\alpha)$.

if $s = s_{\text{term}} = n$ then **return** $(0, 0)$;
if $(s, a, \alpha) \notin \mathcal{D}$ *(i.e., if* $\mathcal{D}(s, a, \alpha)$ *does not exist)* **then**
  for $s' = s+1, \ldots, n$ **do**
    for $a' \in \mathcal{A}$ **do**
      let $q_{s',a'} \leftarrow \texttt{eval}(s', a', \alpha)$;
      if $(q_{s',a'})_{\text{ap}(s')} < (p_{s',a'})_{\text{ap}(s')}$ **then**
        let $q_{s',a'} \leftarrow p_{s',a'}$;
      end
    end
    let $q_{s'} \leftarrow \text{argmax}_{q \in \{q_{s',a'}\}_{a' \in \mathcal{A}}} \alpha \cdot q$;
  end
  let $\mathcal{D}(s, a, \alpha) \leftarrow (r_1(s, a), r_2(s, a)) + \mathbb{E}_{s' \sim P(s,a)}[q_{s'}]$;
end
return $\mathcal{D}(s, a, \alpha)$;

ALGORITHM 6: A procedure that decodes the output of Algorithm 4.

Input: A turn-taking stochastic game $(\mathcal{S}, \mathcal{A}, \mathrm{ap}, r_1, r_2, P)$, an objective direction α_{obj}, the output of Algorithm 4, a history $h = (s_1, a_1, \ldots, s_t, a_t)$, and a state s.

Output: $\pi(h, s)$, where $\pi \sim \Pi \mid h$, and Π is the strategy encoded in the output of Algorithm 4; -1 if h is not an admissible history under Π.

let $\alpha \leftarrow \alpha_{\mathrm{obj}}$, $a \leftarrow \mathrm{argmax}_{a' \in \mathcal{A}}\, \alpha \cdot p_{s,a'}$, $q \leftarrow p_{s,a}$;
if $|h| = 0$ **return** a;
if $a_1 \neq a$, **return** -1;
for $i = 1, 2, \ldots, t-1$ **do**
  if $q = p_{s_i,a_i}$ **then**
    let $\ell \leftarrow \ell_{s_i,a_i}$, $r \leftarrow r_{s_i,a_i}$, $a_\ell \leftarrow \mathrm{argmax}_{a' \in \mathcal{A}}\, \ell \cdot \max^{\mathrm{ap}(s_{i+1})}\{p_{s_{i+1},a'}, \mathcal{D}(s_{i+1}, a', \ell)\}$,
      $a_r \leftarrow \mathrm{argmax}_{a' \in \mathcal{A}}\, r \cdot \max^{\mathrm{ap}(s_{i+1})}\{p_{s_{i+1},a'}, \mathcal{D}(s_{i+1}, a', \ell)\}$;
    /* for two points q_1 and q_2, $\max^k\{q_1, q_2\}$ denotes the point with the larger k-th coordinate between the two */
    let $\alpha \leftarrow \ell$ if $a_{i+1} = a_\ell$;
    let $\alpha \leftarrow r$ if $a_{i+1} = a_r$;
    if $a_{i+1} \notin \{a_\ell, a_r\}$, **return** -1;
  end
  else
    let $a \leftarrow \mathrm{argmax}_{a' \in \mathcal{A}}\, \alpha \cdot \max^{\mathrm{ap}(s_{i+1})}\{p_{s_{i+1},a'}, \mathcal{D}(s_{i+1}, a', \alpha)\}$;
    if $a_{i+1} \neq a$ **return** -1;
  end
  let $q \leftarrow \max^{\mathrm{ap}(s_{i+1})}\{p_{s_{i+1},a_{i+1}}, \mathcal{D}(s_{i+1}, a_{i+1}, \alpha)\}$;
end
if $q = p_{s_t,a_t}$ **then**
  let $\ell \leftarrow \ell_{s_i,a_i}$, $a_\ell \leftarrow \mathrm{argmax}_{a' \in \mathcal{A}}\, \ell \cdot \max^{\mathrm{ap}(s)}\{p_{s,a'}, \mathcal{D}(s, a', \ell)\}$, $q_\ell \leftarrow \max^{\mathrm{ap}(s)}\{p_{s,a_\ell}, \mathcal{D}(s, a_\ell, \ell)\}$;
    let $r \leftarrow r_{s_i,a_i}$, $a_r \leftarrow \mathrm{argmax}_{a' \in \mathcal{A}}\, r \cdot \max^{\mathrm{ap}(s)}\{p_{s,a'}, \mathcal{D}(s, a', r)\}$, $q_r \leftarrow \max^{\mathrm{ap}(s)}\{p_{s,a_r}, \mathcal{D}(s, a_r, r)\}$;
  let $a \leftarrow a_\ell$ with probability $\frac{q_r - q}{q_r - q_\ell}$, $a \leftarrow a_r$ with probability $\frac{q - q_\ell}{q_r - q_\ell}$;
end
else
  let $a \leftarrow \mathrm{argmax}_{a' \in \mathcal{A}}\, \alpha \cdot \max^{\mathrm{ap}(s)}\{p_{s,a'}, \mathcal{D}(s, a', \alpha)\}$;
end
return a;