

# The Complexity of Equilibrium Refinements in Potential Games\*

Ioannis Anagnostides<sup>1</sup>, Maria-Florina Balcan<sup>1</sup>, Kiriaki Fragkia<sup>1</sup>, Tuomas Sandholm<sup>1,3</sup>,  
Emanuel Tewolde<sup>1</sup>, and Brian Hu Zhang<sup>2</sup>

<sup>1</sup>Carnegie Mellon University

<sup>2</sup>Massachusetts Institute of Technology

<sup>3</sup>Additional affiliations: Strategy Robot, Inc., Strategic Machine, Inc., Optimized Markets, Inc.

{ianagnos,ninamf,kiriakif,sandholm,etewolde}@cs.cmu.edu, zhangbh@csail.mit.edu

November 7, 2025

## Abstract

Refinements of the Nash equilibrium—most notably Selten’s trembling-hand perfect equilibrium and Myerson’s proper equilibrium—are at the heart of microeconomic theory, addressing some of the deficiencies of Nash’s original concept. While the complexity of computing equilibrium refinements has been at the forefront of algorithmic game theory research, it has remained open in the seminal class of potential games; we close this fundamental gap in this paper.

We first establish that computing a pure-strategy *perfect* equilibrium is PLS-complete under different game representations—including extensive-form games and general polytope games, thereby being polynomial-time equivalent to pure Nash equilibria. For *normal-form proper equilibria*, our main result is that a perturbed (proper) best response can be computed efficiently in extensive-form games; this is despite the fact that, as we show, computing a best response using the classic perturbation of Kohlberg and Mertens based on the permutahedron is #P-hard even in Bayesian games. As a byproduct, we establish FIXP<sub>a</sub>-completeness of normal-form proper equilibria in extensive-form games, resolving a long-standing open problem. In stark contrast, we show that computing a normal-form proper equilibrium in polytope potential games is both NP-hard and coNP-hard. This marks the first natural class in which the complexity of computing equilibrium refinements does not collapse to that of Nash equilibria, and the first problem in which equilibrium computation in polytope games is strictly harder—unless there is a collapse in the complexity hierarchy—relative to extensive-form games.

We next turn to more structured classes of games, namely symmetric network congestion and symmetric matroid congestion games. For both classes, we show that a perfect pure-strategy equilibrium can be computed in polynomial time, strengthening the existing results for pure Nash equilibria. More broadly, we make a connection between strongly polynomial-time algorithms and efficient perturbed optimization using fractional interpolation. On the other hand, we establish that, for a certain class of potential games, there is an exponential separation in the length of the best-response path between perfect and Nash equilibria.

Finally, for mixed strategies, we prove that computing a point geometrically near a perfect equilibrium requires a doubly exponentially small perturbation even in 3-player potential games in normal form. As a byproduct, this significantly strengthens and simplifies a seminal result of Etessami and Yannakakis (FOCS ’07). On the flip side, in the special case of polymatrix potential games, we show that equilibrium refinements are amenable to perturbed gradient descent dynamics, thereby belonging to the complexity class CLS. This provides a principled and practical way of refining the landscape of gradient descent in constrained optimization.

---

\*Authors ordered alphabetically.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Equilibrium refinements . . . . .	1
1.2	Our results . . . . .	3
1.3	Further related work . . . . .	11
<b>2</b>	<b>Preliminaries</b>	<b>13</b>
<b>3</b>	<b>Search problems and modeling assumptions</b>	<b>16</b>
3.1	Search problems . . . . .	17
3.2	Evaluating circuits on polynomials . . . . .	18
<b>4</b>	<b>Lower bound on verification of perfect equilibrium</b>	<b>19</b>
<b>5</b>	<b>Computing pure perfect equilibria</b>	<b>20</b>
5.1	Constant number of players . . . . .	20
5.2	Algorithmic approach and complexity implications . . . . .	21
5.3	Perfect equilibria of polytope games . . . . .	22
<b>6</b>	<b>Mixed perfect equilibria</b>	<b>23</b>
6.1	Almost implies near: a refinement in polymatrix games . . . . .	23
6.2	Doubly exponentially small $\epsilon$ is necessary . . . . .	27
<b>7</b>	<b>Exponential path lengths for pure perfect equilibria</b>	<b>29</b>
7.1	Proof of Theorem 7.1 . . . . .	32
7.2	Proof of Theorem 7.2 and its subsequent corollaries . . . . .	33
<b>8</b>	<b>Positive results for structured games</b>	<b>35</b>
8.1	Symmetric matroid congestion games . . . . .	35
8.2	Symmetric network congestion games . . . . .	38
8.3	Strongly polynomial algorithms and perturbed optimization . . . . .	39
<b>9</b>	<b>Computing proper equilibria</b>	<b>40</b>
9.1	Normal-form games . . . . .	40
9.2	Hardness of proper equilibria in polytope games . . . . .	40
9.3	Optimal spanning sets . . . . .	43
9.4	Efficient normal-form proper equilibria in extensive-form games . . . . .	46
<b>10</b>	<b>Price of anarchy of perfect and proper equilibria</b>	<b>48</b>
<b>11</b>	<b>Conclusions and future research</b>	<b>50</b>
<b>A</b>	<b>Equilibrium refinements for games in extensive form</b>	<b>56</b>
A.1	Notation and background . . . . .	56
A.2	Computing extensive-form perfect equilibria . . . . .	56
<b>B</b>	<b>Quasi-perfect versus normal-form proper equilibria</b>	<b>57</b>
<b>C</b>	<b>Relations between different equilibrium refinements</b>	<b>58</b>

# 1 Introduction

A forceful critique of the Nash equilibrium [56]—the predominant solution concept in game theory—is centered on its non-uniqueness: a game may admit multiple Nash equilibria, with some being more sensible than others. A pathbreaking line of work in economics has put forward *equilibrium refinements* as a compelling antidote to this conundrum. By now, an entire hierarchy of equilibrium concepts has emerged that differ depending on the game representation and the refinement criterion [73]. Perhaps the most popular refinement of the Nash equilibrium is Selten’s (*trembling-hand*) *perfect equilibrium* [69]. The impact of this concept is witnessed by the fact that Selten, together with Nash and Harsanyi, went on to win the Nobel prize in economics for this work.

From an algorithmic standpoint, a flurry of results has established that, for the most part, identifying such equilibrium refinements is no harder than computing Nash equilibria (highlighted at greater length in [Section 1.3](#)). In particular, the rough complexity landscape that has come to light after decades of research is that equilibrium computation—under any of the usual refinements and game representations—in two-player zero-sum games is in P, in two-player general-sum games is PPAD-complete, whereas in  $n$ -player games is FIXP-complete.

However, surprisingly, the complexity of equilibrium refinements has remained unexplored in the seminal class of *potential games* [54, 63]. Such games encompass a diverse panoply of strategic interactions including *tacit coordination* among agents with aligned interests, a central problem in the social sciences with ubiquitous applications [68]; routing problems in large communication networks in the absence of a centralized authority, such as the Internet [65]; and more broadly, resource allocation problems [63]. Characterizing and indeed enhancing the quality of Nash equilibria—primarily through the lens of *price of anarchy* [46]—in potential games has been at the forefront of the algorithmic game theory agenda, as discussed further in [Section 1.3](#). We take an orthogonal perspective on this problem revolving around equilibrium refinements.

As it turns out, the complexity of computing a Nash equilibrium in potential games differs drastically from that in general games. First, unlike general games, potential games always admit a *pure*—a strategy profile in which no player randomizes—Nash equilibrium. Fabrikant, Papadimitriou, and Talwar [26] famously showed that computing a pure Nash equilibrium is complete for the class PLS, which stands for *polynomial local search* [44]. Second, computing an approximate *mixed* Nash equilibrium is complete for  $PLS \cap PPAD = CLS$  [5, 29].

In this context, our main contribution here is to fill this gap and completely characterize the complexity of various equilibrium refinements in potential games.

## 1.1 Equilibrium refinements

Before we dive into our results, we give a brief overview of the basic equilibrium refinements that we consider together with some motivating examples specifically in potential games. We begin with the usual normal-form representation of games, and then extend our scope to other more involved representations.

	C1	C2
R1	1, 1	0, 0
R2	0, 0	0, 0

Figure 1: A  $2 \times 2$  identical-interest game in normal form. It has exactly two Nash equilibria: (R1, C1) and (R2, C2). Of the two, only (R1, C1) is a perfect equilibrium.

**Perfect equilibrium** The (trembling-hand) perfect equilibrium was put forward by Selten [69] so as to exclude certain unreasonable Nash equilibria, such as ones supported on strictly dominated actions away from the equilibrium path. At a high level, perfect equilibria account for players’ mistakes—hence the term “trembling-hand.” In particular, a strategy profile qualifies as a perfect equilibrium if it arises as the limit point of a sequence of Nash equilibria in a perturbed game wherein each action is played with some positive probability (Definition 2.5 formalizes this).

Figure 1 portrays a simple illustrative example. Of the two equilibria of the game, (R1, C1) is the only sensible one—R2 is weakly dominated by R1 and C2 is weakly dominated by C1. As soon as one accepts that all actions are played with some positive probability (for example, through a mistake or a “tremble”), (R1, C1) emerges unequivocally as the unique perfect equilibrium—in the limit as the trembles vanish, that is. This is a rather prosaic and brittle example, but we shall analyze more interesting examples in the sequel.

**Proper equilibrium** Moving on, *proper equilibria*, introduced by another Nobel laureate, namely Roger Myerson [55], were in turn shown to refine perfect equilibria. Like perfect equilibria, proper equilibria are guaranteed to exist [55]. The high-level idea of the definition is this: although players can make mistakes, as before, they now do so in a somewhat rational way. In particular, some mistakes are more costly than others, and so should be made with a smaller probability. As observed by Kohlberg and Mertens [45], a proper equilibrium can be obtained as a limit point of Nash equilibria in a sequence of  $\epsilon$ -perturbed games in which players select strategies from a certain *permutahedron*—namely, the convex hull of permutations of  $(1, \epsilon, \epsilon^2, \dots, \epsilon^{m-1})$ , up to some normalization factor; this means that the action resulting in the highest utility is to be allotted probability roughly 1, the second highest roughly  $\epsilon$ , and so on, so that progressively worse actions end up being played with gradually smaller probability (Definition 2.8).

A motivating example, due to Myerson [55], is given in Figure 2. As in Figure 1, it seems evident that the unique outcome of the game should be (R1, C1); after all, only strictly dominated actions were inserted. And yet, this is not in accordance with the set of perfect equilibria of that game: (R2, C2) is now in fact a perfect equilibrium. (To see this, one can take  $\mathbf{x}_1^{(\epsilon)} = (\epsilon, 1 - 2\epsilon, \epsilon)$  and  $\mathbf{x}_2^{(\epsilon)} = (\epsilon, 1 - 2\epsilon, \epsilon)$ . It then follows that  $(\mathbf{x}_1^{(\epsilon)}, \mathbf{x}_2^{(\epsilon)})$  is an equilibrium in the  $\epsilon$ -perturbed game *à la* Selten.) On the other hand, Myerson [55] observed that the only proper equilibrium is (R1, C1). Peter Bro Miltersen has anecdotally referred to proper equilibria as “the mother of all refinements” in normal-form games [23]; it will indeed be the most refined concept that we examine.

	C1	C2	C3
R1	1, 1	0, 0	-9, -9
R2	0, 0	0, 0	-7, -7
R3	-9, -9	-7, -7	-7, -7

Figure 2: A  $3 \times 3$  identical-interest game in normal form devised by Myerson [55].

**Extensive-form games** We now turn our attention to extensive-form games. These are common representations of sequential interactions featuring imperfect information [70]. Now, one can employ the previous notion of perfection to the induced normal-form game, giving rise to what is known as *normal-form perfect equilibrium* (NFPE). But NFPE is not the most attractive refinement in extensive-form games [23]. We instead first examine the more refined notions of *extensive-form perfect equilibria* (EFPEs) and *quasi-perfect equilibria* (QPEs); these are incomparable with each other, in that an EFPE need not be a QPE and a QPE need not be an EFPE [50]. We refrain from

formally introducing those concepts at this point. Suffice it to say that, for QPEs, we make use of a characterization due to Gatti, Gilli, and Marchesi [33] in multi-player extensive-form games, showing that they can be obtained as limit points of a sequence of Nash equilibria of a certain class of perturbed games in *sequence form* (cf. Miltersen and Sørensen [53]). Similarly, for EFPEs, there is again a characterization as limit points of Nash equilibria of a certain class of perturbed games in sequence form [11, 27]; this was used by Farina and Gatti [27] to place EFPEs in PPAD. The precise definitions of the perturbed games are deferred to a later section.

To make a connection with the foregoing definitions in normal-form games, we point out that one can compute a quasi-perfect equilibrium of an extensive-form game [53] by identifying a proper equilibrium of the induced normal-form game [72], since the former are a superset of the latter. The solution concept that arises is called *normal-form proper equilibrium*. (A pictorial illustration of the basic equilibrium refinements in extensive-form games appears in Section C.) It is a powerful solution concept that has been extensively motivated, for example, by Miltersen and Sørensen [52]; indeed, there are games in which only normal-form proper equilibria prescribe sensible play (Section B). But characterizing its complexity beyond zero-sum games has so far remained elusive.

**Polytope games** Taking a step further, we finally consider *polytope games*. These are games in which each player selects a strategy from a (convex) polytope described explicitly through a set of linear inequalities; the number of vertices is generally exponential—as is the case for the sequence-form polytope—so our approach for the normal-form case needs to be refined. Canonical examples are (network) congestion games [65] and, indeed, extensive-form games. For polytope games, we again examine the complexity of normal-form perfect and proper equilibria.

## 1.2 Our results

Table 1: Overview of our main results.

Game class / Problem	Perfect	Normal-form proper
<i>Pure strategies in potential games</i>		
Normal form	PLS-c (Theorem 1.2)	PLS-c (Theorem 1.2)
Extensive form	PLS-c (Theorems A.2 and A.3)	PLS-c (Theorem 1.4)
Polytope games		#P-hard via KM [45] (Theorem 9.4) NP-h and coNP-h (Theorem 1.6)
<i>Mixed strategies in potential games</i>		
Polymatrix	CLS (Corollary 1.11)	CLS (Corollary 1.11)
3-player identical-interest	Distance $\epsilon$ -PE to exact PE $\Omega(1)$ even for $\epsilon = 1/2^{2^n}$ (Theorems 1.12 and 6.11)	
<i>Tractable subclasses</i>		
Symmetric matroid congestion	P (Theorem 1.7)	—
Symmetric network congestion	P (Theorem 1.8)	—
<i>Verification complexity</i>		
3-player identical-interest	NP-hard (Proposition 1.1)	—
<i>Convergence separations for best-response dynamics</i>		
Nash poly. / Refinement exponential	Theorem 1.9	Corollary 7.5
Refinement poly. / Nash exponential	Theorem 1.10	Corollary 7.5
<i>Price of anarchy</i>		
Polynomial degree- $d$ congestion games	$d^{\Omega(d)}$ (Theorem 10.3)	$d^{\Omega(d)}$ (Theorem 10.3)

We provide a thorough characterization of the complexity of different equilibrium refinements in potential games under different game representations. We first outline our basic model.

**Computational model** We focus on *concise* potential games, which are represented through what we refer to as *multilinear arithmetic circuits*; multilinearity here is enforced structurally by insisting that different inputs to any multiplication gate must involve disjoint sets of players. Multilinear arithmetic circuits can be evaluated efficiently on rational inputs ([Lemma 3.1](#)); thus we circumvent the issue of repeated squaring present when using general, unfettered arithmetic circuits [29]. We rely on conciseness because the explicit representation of a game—given as the entire payoff tensor—grows exponentially in the number of players. Also, for explicitly represented identical-interest games, there is a straightforward algorithm for computing a pareto-optimal perfect equilibrium (and other equilibrium refinements): identify a strategy profile corresponding to a maximum entry of the payoff tensor *in the perturbed game* ([Proposition 5.1](#))—this algorithm quickly becomes inefficient as the number of players grows.

**Hardness of verification** Despite this simple fact, and somewhat paradoxically, we show that *verifying* whether a pure strategy profile is a perfect equilibrium even in 3-player identical-interest games is NP-hard.

**Proposition 1.1.** *Verifying whether a pure strategy is a perfect equilibrium in a 3-player identical-interest game is NP-hard.*

This strengthens the hardness result of Hansen, Miltersen, and Sørensen [38] pertaining to 3-player general-sum games. Similar to their proof, the starting point of our reduction is the problem of computing the team minimax equilibrium (TME) value in adversarial team games (defined in [Section 4](#)). We show that a gap-amplified hard TME instance allows a reduction to go through even when the players in the resulting game have identical interests.

Prompted by [Proposition 1.1](#), we take the opportunity to point out a subtle issue in the definition of FNP concerning the ability to verify purported solutions in polynomial time. Following the original treatment of Johnson et al. [44], we take FNP to contain all relations that can be accepted by a nondeterministic polynomial-time Turing machine (formalized in [Definition 3.5](#)). This definition does *not* imply the usual efficient verification property, as we discuss further in [Section 3.1](#). In particular, [Proposition 1.1](#) does not preclude placing perfect equilibria—and refinements thereof—in FNP.

**Complexity results** Indeed, we begin by showing the following characterizations.

**Theorem 1.2.** *Computing a pure perfect equilibrium in concise potential games is PLS-complete. The same holds for proper equilibria.*

PLS-hardness is readily inherited from existing, well-known results concerning pure Nash equilibria [26], so we discuss the proof of PLS membership. It is established by executing *symbolic* best-response dynamics. Specifically, for a symbolic parameter  $\epsilon > 0$ , we run best-response dynamics on a perturbed game parameterized by  $\epsilon$ , which differs depending on the underlying equilibrium refinement notion. First, we show that by having access to a multilinear arithmetic circuit, one can still evaluate utilities symbolically in terms of  $\epsilon$  through the use of polynomial interpolation ([Lemma 3.10](#)). Furthermore, when  $\epsilon$  is taken arbitrarily small, it is possible to compare polynomials by contrasting their coefficients in lexicographic order. We are thus able to efficiently implement each iteration of the symbolic best-response dynamics when  $\epsilon$  is small enough. Finally, the *potential*

that arises from these dynamics can itself be thought of as a polynomial in  $\epsilon$ , so that the key local improvement property attached to PLS can be established lexicographically with respect to the coefficients of the potential.

As a non-trivial corollary of [Theorem 1.2](#), it follows that pure perfect and pure proper equilibria always exist in potential games; an earlier work in the economics literature by Carbonell-Nicolau and McLean [\[15\]](#) had already shown the existence of a pure perfect equilibrium in potential games.

Taking a step further, we establish PLS membership beyond normal-form games. We first show that computing a pure EFPE or pure QPE in extensive-form games is PLS-complete ([Theorems A.2](#) and [A.3](#)). The high-level argument is similar to that of [Theorem 1.2](#), but with an added complication concerning symbolic best responses in the perturbed game—be it the one arising from EFPEs or QPEs. This step is straightforward in normal-form games: for perfect equilibria it boils down to computing a maximum entry of the utility vector, while for proper equilibria—where the perturbed strategy set is a permutahedron—it reduces to sorting. In extensive-form games, we show that symbolic best responses for both EFPEs and QPEs can be computed in polynomial time through a bottom-up traversal of the tree.

**Polytope games and normal-form proper equilibria** Turning to the more general setting of polytope games, we first show that inclusion in PLS for perfect equilibria holds more broadly by merely assuming access to a linear optimization oracle for each strategy set—that is, a best response oracle ([Theorem 5.6](#)).

On the other hand, the complexity of normal-form proper equilibria turns out to be more nuanced. To put this question into perspective, this is a long-standing open problem, beyond potential games, even in extensive-form games—a special case of polytope games—first mentioned by Miltersen and Sørensen [\[52\]](#) and more recently highlighted by Sørensen [\[71\]](#) for the case of two-player games. There is a daunting obstacle here: using the classic perturbation of Kohlberg and Mertens [\[45\]](#) based on the permutahedron, one would naively have to sort exponentially many vertices, which is clearly hopeless. We formalize this barrier, showing  $\#P$ -hardness for computing a perturbed best response *à la* Kohlberg and Mertens ([Theorem 9.4](#)).

Nevertheless, we find that, surprisingly, there is an alternative perturbation that circumvents our aforementioned hardness result. The key concept that drives our approach is what we refer to as an *optimal spanning set*, defined with respect to a utility vector  $\mathbf{u}$ . A subset  $B$  of the vertices of a polytope is an optimal spanning set if every vertex  $\mathbf{x}$  belongs to the affine hull of  $\{\mathbf{x}' \in B : \langle \mathbf{u}, \mathbf{x}' - \mathbf{x} \rangle \geq 0\}$ . The crucial role of an optimal spanning set is that, informally, it contains enough information to construct a proper best response ([Theorem 9.15](#)). At a high level, this holds for the following reason: consider any vertex  $\mathbf{x}$ . The probability that  $\mathbf{x}$  “should” be assigned is much less than the probability of all elements in  $B_{\succeq \mathbf{x}}$ , because those strategies are better than  $\mathbf{x}$ . But, if we have a fully mixed strategy over the spanning set, we can pretend that  $\mathbf{x}$  was actually part of the mixture by 1) writing  $\mathbf{x}$  as an affine combination of the elements in  $B_{\succeq \mathbf{x}}$ , and then 2) modifying the mixture by subtracting off the correct amounts of each element in  $B_{\succeq \mathbf{x}}$  (in accordance with the coefficients in the affine combination) and adding in a small amount of  $\mathbf{x}$ , to maintain that the new mixture has the same expectation.

The upshot now is that, even when the underlying polytope has exponentially many vertices, one can often determine efficiently a compact (that is, polynomially sized) optimal spanning set. A simple example of this is the hypercube  $[0, 1]^d$ : one can construct an optimal spanning tree by determining first a best response to  $\mathbf{u}$ , namely  $\mathbf{x}^* = \text{sign}(\mathbf{u})$ , and then gathering all vertices that are a single bit-flip away from  $\mathbf{x}^*$ . More broadly, it turns out that one can efficiently construct a polynomially sized optimal spanning set even in general extensive-form games ([Theorem 9.18](#)),



which in turn enables computing a symbolic proper best response.

**Theorem 1.3.** *There is a polynomial-time algorithm for computing a proper best response in extensive-form games.*

As a result, we are able to show that computing a (normal-form) proper equilibrium is in PLS even in extensive-form potential games.

**Theorem 1.4.** *Computing a normal-form proper equilibrium in potential games represented in extensive form is PLS-complete.*

This is the first positive result for proper equilibria not obtained through the perturbation of Kohlberg and Mertens, with the exception of a result by Miltersen and Sørensen [52] pertaining to zero-sum games. A byproduct of Theorem 1.3 beyond potential games is an exact characterization of the complexity of normal-form proper equilibria in general-sum extensive-form games.

**Corollary 1.5.** *Computing a strong approximation (that is, a point close in geometric distance) to a normal-form proper equilibrium in extensive-form games is  $\text{FIXP}_a$ -complete.*

Beyond extensive-form games, we show that the complexity landscape changes dramatically.

**Theorem 1.6.** *It is both NP-hard and coNP-hard to compute a normal-form proper equilibrium in two-player identical-interest polytope games.*

The argument proceeds by reducing from PARTITION, and appears in Section 9.2. From a broader standpoint, Theorem 1.6 gives rise to the first natural class in which the complexity of equilibrium refinements does not collapse to that of Nash equilibria, and the first problem in which equilibrium computation in polytope games is strictly harder—subject to standard complexity assumptions—relative to extensive-form games.

Taken together, our results resolve the complexity of normal-form proper equilibria in multi-player extensive-form games.

**Polynomial-time convergence in structured classes of games** Notwithstanding the PLS-hardness of computing a pure-strategy Nash equilibrium, the question of carving out classes of games wherein best-response dynamics converge in polynomial time has received ample interest in the literature (e.g., we point to Ackermann, Röglin, and Vöcking [1], Caragiannis, Fanelli, Gravin, and Skopalik [13], Caragiannis, Fanelli, and Gravin [14], and references therein). This begs the question: do these positive results carry over to perfect equilibria?

To begin with, we provide an affirmative answer in *symmetric matroid congestion* games. Matroid congestion games were introduced by Ackermann et al. [1] and have been studied extensively since then [7, 21, 39, 40]. The key premise is that each player’s strategy set comprises the bases of an underlying matroid (Definitions 8.1 and 8.2); the special case where the rank of the matroid is 1—meaning that the strategy set of each player comprises single resources—is known as *singleton games* [42]. On top of that, we posit symmetric strategy sets among the players; the reason we need this symmetry assumption will become clear shortly. Under these preconditions, we give a polynomial-time algorithm for computing perfect equilibria.

**Theorem 1.7.** *In symmetric matroid congestion games, symbolic best-response dynamics converge to a perfect equilibrium after a polynomial number of steps.*



In proof, it suffices to show that the perturbed game is itself a matroid congestion game—since one can then appeal to the result of Ackermann et al. [1], which is agnostic to the utility structure of the game. To do so, the challenge is that the delay functions in the perturbed game are considerably more convoluted, so much so that, in fact, the perturbed game may no longer be a congestion game! The basic reason for this is that, in the perturbed game, the delay function can depend not just on the number of players using it, but also on their identities. Yet, we observe that this cannot happen when players have the same strategy set, paving the way to [Theorem 1.7](#).

The second class of games we consider is *symmetric network congestion* games. Here, players' strategies correspond to paths linking a designated source to its destination. For this class of games, Fabrikant et al. [26] gave a polynomial-time algorithm for finding a global optimum of the potential function—and thereby a pure Nash equilibrium—based on min-cost flow. We provide an analogous reduction mapping perfect equilibria to solving a *symbolic* min-cost flow problem, which in turn can be solved using known combinatorial algorithms or linear programming.

**Theorem 1.8.** *In symmetric network congestion games, there is a polynomial-time algorithm for finding a perfect equilibrium.*

**Strongly polynomial-time algorithms and perturbed optimization** A recurring question that arises throughout the line of work on equilibrium refinements—and our paper in particular—is whether an algorithm can be made to run efficiently under a symbolic class of instances. What we have been implicitly using so far is that certain basic operations—such as comparisons and additions—can still be performed symbolically when  $\epsilon$  is small enough. We make a significantly more general connection: any *strongly polynomial-time algorithm*—which comprises basic arithmetic operations, namely  $\{+, *, /, <\}$ —can be made to run *symbolically* ([Theorem 8.6](#)). Most of our positive results—with the one exception of linear programming—can be seen as instances of this more general connection. The basic idea is to execute the underlying algorithm on multiple inputs in the neighborhood of  $\epsilon \approx 0$ , and then use fractional interpolation to derive the exact form of the symbolic output. On the other hand, weakly polynomial-time algorithms do not share this property; the simplest example where this becomes evident is binary search over the real numbers.

**Best-response paths: exponential separations** The landscape that has emerged so far is that, in potential games, computing pure perfect equilibria is polynomial-time equivalent to computing pure Nash equilibria. On the other hand, our next result shows that, for certain potential games, the length of the symbolic perfect best-response path in the perturbed game can be exponentially larger than the length of the best-response path in the original game.

**Theorem 1.9.** *There is a class of identical-interest games with the following property:*

- *From any starting point, best-response dynamics converge to a Nash equilibrium in polynomially many steps, whereas*
- *there exist starting points such that symbolic perfect best-response dynamics can take exponentially many steps.*

This class of games is predicated on the usual reduction from the local version of MAXCUT with respect to the FLIP neighborhood, but with a crucial twist: every node is to be represented by a triplet of players, each of whom has a say on which side of the cut the corresponding node should belong to. In particular, the assignment of that node is decided by the majority. In addition, we create a small incentive for the triplets to be unanimous; since the underlying game needs to be

identical-interest, this added term in the utility accounts for the cumulative number of unanimous triplets. A subtle issue in our construction, whose role will become clear shortly, is that the penalty for non-unanimous triplets is only present when there are at least 2 non-unanimous triplets. The upshot is that best-response dynamics quickly get trapped into spurious equilibria—in the sense of failing to account for the local structure of MAXCUT—in which all but at most 1 triplet are unanimous. On the other hand, *symbolic* perfect best-response dynamics can only converge to local optima of MAXCUT.

In more detail, when we execute best-response dynamics and there are at least 2 non-unanimous triplets, at some point we will update a player whose action differs from that of the majority in that triplet. That player’s best response will be to switch to the side of the majority, for there is an incentive to maximize the number of unanimous triplets. From then onward, players in that newly unanimous triplet will retain their action under best-response dynamics given that the side of the cut is decided by the majority. As a result, we will end up with at most 1 non-unanimous triplet, and such a triplet can perform a best-response update only once.

We now turn to the second part of [Theorem 1.9](#). Following the above reasoning, we can assume that there is at most 1 non-unanimous triplet, whereupon we claim that after the players in that triplet perform an update under symbolic perfect best-response dynamics, there will be unanimity, and what is more, the elected node will be on the side of the cut that locally maximizes the weight. In fact, the key claim is that those two invariances will be maintained throughout, which means that symbolic perfect best-response dynamics can only converge to local optima of MAXCUT under the FLIP neighborhood. The proof proceeds as follows. Let us consider a unanimous triplet in which the elected node is not on the optimal side of the partition. Under symbolic perfect best-response dynamics, where the player accounts for other players’ trembles, there are two conflicting forces *neither of which is present under best-response dynamics*: choosing the other side of the partition will incur a penalty since it breaks unanimity, but will also result in switching the side of the node. The crucial point here is that both of these forces are *first-order*, in the sense that they both manifest in the coefficient of the degree-1 term in the polynomial that measures progress; this is where we make use of the fact that the penalty for non-unanimity is only introduced in the presence of at least 2 non-unanimous triplets, so that it only introduces a lower order effect. The claim then follows by selecting the penalty term to be sufficiently small, so that the local structure of MAXCUT outweighs the cost of breaking unanimity.

Our next result turns [Theorem 1.9](#) on its head: it shows that, for certain potential games, symbolic perfect best-response dynamics can be fast while best-response dynamics can be slow.

**Theorem 1.10.** *There is a class of identical-interest games with the following property:*

- *From any starting point, symbolic perfect best-response dynamics converge to a perfect equilibrium in polynomially many steps, whereas*
- *there exist starting points such that best-response dynamics can take exponentially many steps.*

As before, we start from the usual reduction from MAXCUT, but we now introduce two additional players, each with two actions—say **d** and **e**. When either of the two additional players selects **d**, the game proceeds as usual. But when **(e, e)** is selected, every player gets to collect a large reward  $M \gg 1$  plus a bonus proportional to the number of players that belong to the left partition. Now, if the two additional players start from **(d, d)**, it follows that best-response dynamics can only converge to local optima of MAXCUT. This is not so under symbolic perfect best-response dynamics: the possibility of a tremble immediately entices both additional players to action **e**, whereupon all players will end up entering the left partition—this is so because of the bonus given under **(e, e)**.

**Theorem 1.10** is not a computational result; to be sure, finding a Nash equilibrium is no harder than finding a perfect equilibrium, but it does present a counter-intuitive exponential separation under a natural class of algorithms.

**Mixed strategies** The foregoing results fully characterize the complexity of computing a *pure* equilibrium refinements in potential games under different game representations. We now turn our attention to mixed strategies.

We begin with the class of polymatrix (multi-player) games (**Definition 2.3**). Combining **Theorem 1.2** with the known PPAD inclusion due to Hansen and Lund [36],<sup>1</sup> together with the fact that  $\text{CLS} = \text{PPAD} \cap \text{PLS}$  [29], we arrive at the following consequence.

**Corollary 1.11.** *Computing a perfect or proper equilibrium of a potential, (multi-player) polymatrix game in normal form is in CLS.*

It is generally believed that even computing a Nash equilibrium of an identical-interest polymatrix game is CLS-hard; for example, Hollender, Maystre, and Nagarajan [41] recently showed CLS-hardness when the team is facing multiple independent adversaries. If that conjecture turns out to hold, **Corollary 1.11** would imply CLS-completeness for both perfect and proper equilibria. Be that as it may, there is a precise sense in which perfect and proper equilibria in potential polymatrix games are polynomial-time equivalent to Nash equilibria: it can be shown that for a sufficiently small value of  $\epsilon$ , representable in polynomially many bits, computing a perfect or proper equilibrium reduces to finding a Nash equilibrium of a perturbed potential game (**Proposition 6.8**); this reduction does not go through beyond polymatrix games [25]. The proof of **Proposition 6.8** makes use of the general theory of Farina and Gatti [27] pertaining to perturbed LCPs.

**Proposition 6.8** implies that a perfect equilibrium can be computed by running a perturbed variant of gradient descent, thereby furnishing a more direct CLS membership proof. Interestingly, we observe that the *symbolic* version of gradient descent can get stuck (**Proposition 6.9**), so the previous claim only holds by setting  $\epsilon$  to a sufficiently small numerical value, representable with polynomially many bits. At this point, it is worth revisiting the example of **Figure 1**. An unsatisfactory feature of that example is that while a nonsensible Nash equilibrium exists, it is unlikely to be reached by natural learning algorithms. In particular, in that example, if one initializes gradient descent at random—a common practice in optimization with strong theoretical guarantees in the unconstrained setting (**Section 1.3**)—the dynamics converge to the optimal equilibrium *almost surely*. But we observe that this is not always the case even in  $2 \times 2$  (potential) games.

An example game is given in **Figure 3**. The corresponding gradient descent dynamics are illustrated in **Figure 4**. We see that vanilla gradient descent often converges to suboptimal Nash equilibria. On the other hand, executing the dynamics on the perturbed game converges to the welfare-optimal point. From a price of anarchy perspective, the example of **Figure 1** already shows that perfect equilibria can be arbitrarily better than Nash equilibria. **Figure 4** takes a step further: it reveals that the *average* price of anarchy of gradient descent—in the sense of Sakos, Leonardos, Stavroulakis, Overman, Panageas, and Piliouras [66]—can still be far from the welfare attained by perfect equilibria. On the flip side, we will later discuss how to use **Theorem 1.2** in conjunction with the framework of Roughgarden [64] to obtain non-trivial lower bounds for the price of anarchy of perfect equilibria.

---

<sup>1</sup>Hansen and Lund [36] showed inclusion for computing  $\epsilon$ -symbolic proper equilibria per the perturbation of Kohlberg and Mertens [45]; one can verify that any such purported solution is indeed an  $\epsilon$ -symbolic proper equilibrium in polynomial time.

	C1	C2
R1	12, 2	2, 2
R2	11, 1	0, 0

Figure 3: A  $2 \times 2$  identical-interest game in normal form. Any strategy profile in which R1 is played with probability 1 is a Nash equilibrium. The only perfect equilibrium is (R1, C1).

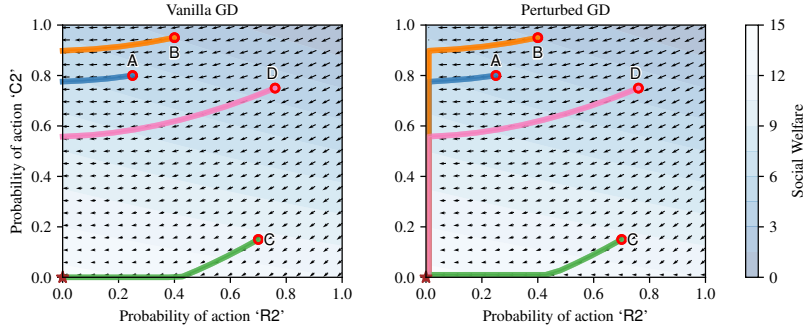


Figure 4: Vanilla gradient descent (left) on the game of Figure 3 versus gradient descent on the perturbed game with perturbation magnitude  $\epsilon := 0.01$  (right) under 4 different initializations. The welfare-optimal point and unique perfect equilibrium (R1, C1) appears in the bottom-left corner. We see that gradient descent in the unperturbed game often converges to Nash equilibria with much lower welfare than the optimal one.

**Beyond polymatrix games** The landscape changes dramatically when moving to general potential games. Indeed, it is not even known whether computing an exact mixed Nash equilibrium in potential games—even though one supported on rational numbers always exists—lies in PPS, let alone any of its refinements. Instead, based on existing results, we can only place mixed Nash equilibria in  $\text{PLS} \cap \text{FIXP}$ , but characterizing the exact complexity of this problem is an important open problem in this area; to our knowledge, even the relation between FIXP and PLS is unexplored. So what can we hope to prove for the complexity of equilibrium refinements?

Etessami et al. [25] have shown that, in general-sum games, as long as one takes the perturbation parameter  $\epsilon$  to be *doubly exponentially small*, one immediately recovers a point that is close geometrically to an exact perfect equilibrium. Now, if one posits general arithmetic circuits, one can produce doubly exponentially small values through repeated squaring, thereby reducing perfect equilibria to Nash equilibria. But, of course, this is not possible in the standard model of computation. The main question here is whether one needs to take  $\epsilon$  doubly exponentially small in potential games; after all, such games are very structured. Our next main result shows that it is indeed necessary.

**Theorem 1.12.** *For every positive integer  $n$  there exists a normal-form potential game  $\Gamma_n$  with  $4n + 1$  players and two actions per player such that, for all  $\epsilon \in [1/2^{2^n}, 1/2]$ , the perturbed game  $\Gamma_n^{(\epsilon)}$  admits a Nash equilibrium that is distance  $1/2$  away in  $\ell_\infty$ -norm from any Nash equilibrium of  $\Gamma_n$ .*

As an immediate byproduct of independent interest, this theorem shows that, in potential games, even when  $\epsilon$  is doubly exponentially small, an  $\epsilon$ -approximate Nash can still be far from an exact Nash in geometric distance. (This holds because an exact Nash equilibrium in the perturbed game induces an approximate Nash in the original game with approximation proportional to the perturbation.) This was hitherto only known in three-player general-sum games due to the tour

de force of Etessami and Yannakakis [24]. Besides holding for a significantly more structured class of games, [Theorem 1.12](#) is established based on a simple and succinct argument; this is unlike the original proof of Etessami and Yannakakis [24] which is especially intricate.

We now sketch the main idea of our argument. It is instructive to begin with general polynomial optimization problems—without restricting to multilinear polynomials. The starting observation is that one can perform repeating squaring by considering the function  $f : [0, 1]^n \rightarrow \mathbb{R}$  given by

$$f(\mathbf{x}) = (2x_1 - 1)^2 + (x_2 - x_1^2)^2 + \cdots + (x_{n-1} - x_{n-2}^2)^2 - x_n x_{n-1},$$

which is to be minimized. It is not hard to show that the point  $\mathbf{x} = (1/2, 1/4, 1/16, \dots, 1/2^{2^{n-2}}, 0)$  is an  $\epsilon$ -KKT point with  $\epsilon$  being proportional to  $1/2^{2^{n-2}}$ . However, every exact KKT point has  $x_{n-1} > 0$  and therefore  $x_n = 1$ . The name of the game now is to make this argument go through while using only multilinear polynomials. To do so, we construct the potential function

$$\Phi(\mathbf{x}, \mathbf{x}', \mathbf{c}, \mathbf{d}, t) = \sum_{i=1}^n \left[ (t - c_i)(x_i - x_{i-1}x'_{i-1}) + \left( d_i - \frac{1}{2} \right) (x_i - x'_i) \right] - 2x_n - 2x'_n - 2n \cdot t,$$

where  $\mathbf{x}, \mathbf{x}', \mathbf{c}, \mathbf{d} \in [0, 1]^n$  and  $t \in [0, 1]$ ; for simplicity of notation, we set  $x_0 = x'_0 := 1/2$ . The basic idea is that  $\mathbf{x}'$  is intended to be a copy of  $\mathbf{x}$ , and it is essential to make sure that, in equilibrium,  $\mathbf{x} \approx \mathbf{x}'$ . The argument boils down to proving the following two claims.

- In every (exact) Nash equilibrium of  $\Gamma_n$ , there is some  $i \in [n]$  such that  $d_i \in \{0, 1\}$  or  $c_i = 1$ .
- For  $\epsilon \in [1/2^{2^n}, 1/2]$ , the perturbed game  $\Gamma_n^{(\epsilon)}$  has an equilibrium in which  $\mathbf{d} = \frac{1}{2}\mathbf{1}$  and  $\mathbf{c} = \epsilon\mathbf{1}$ .

We provide the details of the argument in [Section 6.2](#). Also, we show how to embed this construction even in a 3-player potential game in normal form ([Theorem 6.11](#)).

**Price of anarchy of perfect equilibria** We have seen through some simple motivating examples that perfect equilibria can lead to significantly higher welfare *vis-à-vis* Nash equilibria. On the flip side, we also provide lower bounds for the price of anarchy with respect to perfect and proper equilibria by leveraging the previously established PLS membership ([Theorem 1.2](#)). This makes use of the elegant framework of Roughgarden [64] that relies on hardness of approximation for the underlying optimization problem. We provide a concrete application in polynomial congestion games ([Theorem 10.3](#)), perhaps the most well-studied class from a price of anarchy perspective.

### 1.3 Further related work

We conclude this section by expanding on additional related work. We begin by covering many of the advances in the complexity of equilibrium refinements. We then connect our work with prior literature on improving the quality of equilibria in potential games and constrained optimization problems more broadly.

**Complexity of equilibrium refinements** Hansen et al. [38] proved that verifying whether a strategy profile is a perfect equilibrium even in 3-player games is NP-hard (and SQRTSUM-hard); this of course stands in stark contrast to (exact) Nash equilibria that trivially admit polynomial-time verifiers. (For two-player games, it is known [73] that a strategy profile is a perfect equilibrium if and only if it is undominated, which can be in turn ascertained in polynomial time via linear programming.)

Etessami et al. [25] showed that approximating a perfect equilibrium—in the sense of being close in  $\ell_\infty$  distance to an exact one; this is a “strong approximation” guarantee in the parlance of Etessami and Yannakakis [24]—in games with more than three players is polynomial-time equivalent to approximating Nash equilibria, thereby being  $\text{FIXP}_a$ -complete; this is a complexity class introduced by Etessami and Yannakakis [24] that contains search problems reducible to (strongly) approximating a Brouwer fixed point of a function given by an algebraic circuit with gates  $+$ ,  $-$ ,  $*$ ,  $/$ ,  $\max$ ,  $\min$ . On the other hand, computing an exact perfect equilibrium in two-player games is  $\text{PPAD}$ -complete; this follows by carefully analyzing exact pivoting algorithms [76].

Etessami [23] characterized the complexity of various refinements in multi-player extensive-form games of perfect recall. Namely, *sequential equilibria* (SEs) [47],<sup>2</sup> which refine both Nash equilibria and *subgame-perfect equilibria* (SPEs); extensive-form perfect equilibria (EFPEs), which refine SEs; normal-form perfect equilibria (NFPEs); and quasi-perfect equilibria (QPEs) [72]. The most refined of those notions are QPEs and EFPEs.

Hansen and Lund [36] examined the complexity of proper equilibria. They showed that even in two-player games in normal form, the task of verifying the proper equilibrium conditions is  $\text{NP}$ -complete; this is in contrast to perfect equilibria [38]. For multi-player games, they showed that strongly approximating a proper equilibrium is  $\text{FIXP}_a$ -complete, while computing a *symoblic* proper equilibrium in polymatrix games is  $\text{PPAD}$ -complete; the latter strengthens an earlier result due to Sørensen [71]. Relatedly, Hansen and Lund [37] studied the complexity of *quasi-proper equilibria* in extensive-form games, a refinement of quasi-perfect equilibria [72].

A general technique for proving membership in  $\text{FIXP}$  was recently developed by Filos-Ratsikas, Hansen, Høgh, and Hollender [31]; among others, they provided a simple proof handling proper equilibria. Even more recently, Filos-Ratsikas, Hansen, Høgh, and Hollender [32] introduced a simple framework for proving  $\text{PPAD}$  membership, and gave several new results on the complexity of equilibrium refinements.

It should be noted that the complexity landscape is drastically different in (two-player) zero-sum games. For example, Miltersen and Sørensen [51] showed that an exact proper equilibrium can be computed in polynomial time through linear programming, and similar positive results have been established for other equilibrium refinements as well. Notably, Miltersen and Sørensen [52] showed that a (normal-form) proper equilibrium can be computed in polynomial time. A more practical algorithm for zero-sum games was developed by Farina, Gatti, and Sandholm [28]. Bernasconi, Marchesi, and Trovò [10] developed learning algorithms for EFPEs in zero-sum games.

**Improving the quality of equilibria in potential games** Our work can also be viewed as part of the research agenda endeavoring to identify better equilibria in potential games [4, 6, 34, 66], thereby circumventing the *price of anarchy*—the ratio between the welfare-optimal state and the worst-case welfare attained at a Nash equilibrium. Yet the viewpoint we take in this paper is quite different. One related work by Balcan, Blum, and Chen [8] shows that *diversified* strategies—ones that do not put too much probability mass on a single action—can lead to higher-welfare equilibria; the idea of diversification closely ties to the perfect equilibrium perturbation. Another related concept is “price of uncertainty,” introduced by Balcan et al. [7]. It deals with the impact of small fluctuations in the behavior of best response dynamics. Such fluctuations can be due to to players’ mistakes, not unlike perfect equilibria.

As we have discussed already, one straightforward observation—which follows readily even from the simple example of Figure 1—is that the price of anarchy defined with respect to perfect equilibria

<sup>2</sup>The definition of a sequential equilibrium is somewhat cumbersome, being predicated on a set of beliefs; we do not expand on that definition since it is not relevant for our purposes.



can be arbitrarily smaller than that with respect to Nash equilibria. These type of considerations are not without precedent: Leme, Syrgkanis, and Tardos [49] introduced the “sequential price of anarchy.” Informally, it relates the outcome of the worst possible *subgame perfect equilibrium* of all sequential versions of the game. They showed that the sequential price of anarchy can be significantly more favorable in certain classes of games.

**Optimization perspective** On a similar note, there has been tremendous interest on understanding how to avoid certain undesirable stationary points in constrained optimization problems. In unconstrained problems, it is known that a perturbed version of gradient descent converges to second-order stationary points, thereby avoiding *strict* saddle points [43, 48]. The situation is more complex in the constrained setting. Nouiehed, Lee, and Razaviyayn [57] showed that gradient descent with random initialization can often fail to converge to second-order stationary points; in fact, they showed that even checking the conditions of second-order stationarity is NP-hard. Our work contributes to this line of work by identifying a strict subset of stationary points that can be reached through a perturbed variant of gradient descent. It is worth referring to van der Laan, Talman, and Yang [74] for a notion of “perfect stationary point.” In particular, Dang, Meng, and Talman [17] came up with an interior-point-type algorithm for finding such refinements.

## 2 Preliminaries

In this section, we provide some notation and basic background in game theory that is most relevant to our work.

Consider a *normal-form* game  $\Gamma$  that consists of a finite set of players  $[n] = \{1, \dots, n\}$ . Each player has a finite set of actions  $\mathcal{A}_i$  that consists of pure strategies. We denote  $m_i = |\mathcal{A}_i|$  and  $m = \max_{i \in [n]} |\mathcal{A}_i|$ . The utility function of player  $i$ , denoted as  $u_i : \prod_{j=1}^n \mathcal{A}_j \rightarrow \mathbb{R}$  maps a strategy profile,  $(a_1, \dots, a_n) \in \prod_{j=1}^n \mathcal{A}_j := \mathcal{A}$ , to player  $i$ ’s payoff  $u_i(a_1, \dots, a_n)$ . So, we represent a game  $\Gamma$  as the tuple  $\Gamma := (\mathcal{A}_1, \dots, \mathcal{A}_n, u_1, \dots, u_n)$ .

Players can also mix their strategies by playing a probability distribution  $\mathbf{x}_i = (\mathbf{x}_i(a_i))_{a_i \in \mathcal{A}_i}$  over their action set. We will use the generic notation  $\Delta(S)$  to denote the set of probability distributions on set  $S$ , and abbreviate  $\Delta([n]) = \Delta(n)$ . We denote with  $\mathcal{X}_i = \Delta(\mathcal{A}_i)$  the set of such mixed strategies for player  $i$ , which we will call the mixed strategy space for player  $i$ . By  $\mathcal{X} = \prod_{i=1}^n \mathcal{X}_i$  we denote the mixed strategy space for a game. For a mixed strategy profile  $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n) \in \prod_{i=1}^n \Delta(\mathcal{A}_i)$ , we denote by  $\mathbf{x}_{-i}$  the strategy profile of every player except player  $i$  and (by slight abuse of notation), we define the expected utility for player  $i$  as

$$u_i(\mathbf{x}_1, \dots, \mathbf{x}_n) := \sum_{(a_1, \dots, a_n) \in \mathcal{A}} u_i(a_1, \dots, a_n) \prod_{j=1}^n \mathbf{x}_j(a_j).$$

One of the most well-studied solution concepts in game theory is that of a (mixed) Nash Equilibrium, recalled below.

**Definition 2.1** (Nash equilibrium). A mixed strategy profile  $\mathbf{x}^* = (\mathbf{x}_1^*, \dots, \mathbf{x}_n^*) \in \mathcal{X}$  is called a (mixed) Nash equilibrium if it holds that  $u_i(\mathbf{x}_i^*, \mathbf{x}_{-i}^*) \geq u_i(\mathbf{x}_i, \mathbf{x}_{-i}^*)$  for all players  $i$  and all possible deviations  $\mathbf{x}_i \in \mathcal{X}_i$ . Equivalently, we can write  $u_i(\mathbf{x}_i^*, \mathbf{x}_{-i}^*) \geq u_i(a_i, \mathbf{x}_{-i}^*)$  for all players  $i$  and all possible pure strategy deviations  $a_i \in \mathcal{A}_i$ .



**Potential games** Our work focuses on refinements of the Nash equilibrium concept in the seminal class of *potential games*. Informally, a potential game is identified by a potential function which maps strategy profiles to real values and which increases by the same amount as the utility of any unilaterally deviating player. Identical interest games provide a canonical example of potential games, in which the common utility plays the role of the potential.

**Definition 2.2** (Potential game). A potential game is a game for which there exists a potential function,  $\Phi : \mathcal{A} \rightarrow \mathbb{R}$  such that for every unilateral deviation of any player  $i \in [n]$ , the change in the value of the potential function equals the change in utilities for that deviating player. Formally, for every player  $i \in [n]$ , any strategy  $(a_1, \dots, a_n) \in \mathcal{A}$ , and any unilateral deviation  $a'_i \in \mathcal{A}_i$ ,

$$\Phi(a'_i, a_{-i}) - \Phi(a_i, a_{-i}) = u_i(a'_i, a_{-i}) - u_i(a_i, a_{-i}).$$

A simple example of potential games is identical interest games. In any potential game, every local minimizer of the potential function corresponds to a Nash equilibrium of that game.

**Polymatrix games** Another class of games that we will refer to is *polymatrix games*.

**Definition 2.3** (Polymatrix game). We let  $G = (V, E)$  denote a simple, undirected graph where  $V = [n]$  is the set of vertices and  $E$  is the set of edges. Each player  $i \in [n]$  has a pure strategy set  $\mathcal{A}_i$  and for each edge  $\{i, j\} \in E$  defines a two-player normal form game with payoff matrices  $\mathbf{P}_{i,j} \in \mathbb{Q}^{|\mathcal{A}_i| \times |\mathcal{A}_j|}$  and  $\mathbf{P}_{j,i} \in \mathbb{Q}^{|\mathcal{A}_j| \times |\mathcal{A}_i|}$  for player  $i$  and  $j$  respectively. The polymatrix game  $\Gamma = (G, (\mathbf{P}_{i,j}, \mathbf{P}_{j,i})_{\{i,j\} \in E})$  is an  $n$ -player game with the following utility function,  $u_i : \mathcal{X} \rightarrow \mathbb{R}$ , for each player  $i \in V$

$$u_i(\mathbf{x}_1, \dots, \mathbf{x}_n) = \sum_{j \in N_i} \mathbf{x}_i^\top \mathbf{P}_{i,j} \mathbf{x}_j,$$

where  $\forall i \in V$   $\mathbf{x}_i \in \mathcal{X}_i$ , and  $N_i$  denotes the set of neighbors of  $i \in V$ .

**Congestion Games** A congestion game is a tuple  $\Gamma = (\mathcal{R}, (\mathcal{A}_i)_{i \in [n]}, (d_r)_{r \in \mathcal{R}})$ , where  $[n]$  is the set of players,  $\mathcal{R}$  is a set of resources, and the strategy space for a player  $i \in [n]$  is a collection of subsets of resources,  $\mathcal{A}_i \subseteq 2^{\mathcal{R}}$ . Each resource  $r \in \mathcal{R}$  has an associated delay function  $d_r : \mathbb{N} \rightarrow \mathbb{N}$ . If all players have the same strategy space, we call the congestion game symmetric.

We write a strategy profile as  $S = (S_1, \dots, S_n)$ , where each player  $i \in [n]$  is playing strategy  $S_i \in \mathcal{A}_i$ . Given such a strategy profile  $S$ , we define  $n_r(S) = |\{i \in [n] : r \in S_i\}|$  to be the number of players using resource  $r$  in  $S$ . We assume that each player  $i$  is rational and is trying to minimize her own cost, which for a strategy profile  $S$ , is described by the cost function  $c_i(S) = \sum_{r \in S_i} n_r(S)$ .

Given a strategy profile,  $S$ , a strategy  $S_i^* \in \mathcal{A}_i$  is a best-response of player  $i$  to  $S$  if  $c_i(S_i^*, S_{-i}) \leq c_i(S'_i, S_{-i})$  for all  $S'_i \in \mathcal{A}_i$ . A strategy profile,  $S^*$ , is a pure Nash equilibrium if no player can decrease her cost by changing her strategy. That is, for all players  $i \in [n]$  and all strategies  $S_i \in \mathcal{A}_i$ ,  $c_i(S_i, S_{-i}^*) \geq c_i(S^*)$ .

By Rosenthal [63], we know that congestion games are potential games with potential function

$$\Phi(S) = \sum_{r \in \mathcal{R}} \sum_{i=1}^{n_r(S)} d_r(i).$$

**Perfect equilibrium** A popular refinement of the Nash equilibrium concept is the (trembling-hand) perfect equilibrium [69], which accounts for small perturbations of players' strategies. This is achieved by defining a perturbed game in which every player has to play each action with some positive probability. We start with the definition of an  $\epsilon$ -perfect equilibrium, which requires that every player plays a fully mixed strategy. However, only pure strategies that are best responses get played with probability greater than  $\epsilon$ .

**Definition 2.4** ( $\epsilon$ -perfect equilibrium). For some  $\epsilon > 0$  a mixed strategy profile  $\mathbf{x}$  is an  $\epsilon$ -perfect equilibrium if it is fully mixed and

$$u_i(a_i, \mathbf{x}_{-i}) < u_i(a'_i, \mathbf{x}_{-i}) \implies \mathbf{x}_i(a_i) \leq \epsilon \text{ for all players } i \in [n] \text{ and actions } a_i, a'_i \in \mathcal{A}_i.$$

We will call a strategy  $\mathbf{x}_i \in \mathcal{X}_i^{(\epsilon)}$   $\epsilon$ -pure if it is as pure as possible given the perturbed game, that is, if  $\mathbf{x}_i(a_i) = \epsilon$  for all but one  $a_i \in \mathcal{A}_i$ . We will call an  $\epsilon$ -pure profile  $\mathbf{x}$  an  $\epsilon$ -pure equilibrium if it is  $\epsilon$ -perfect. Asserting that players must make mistakes with small probability gives rise to the following notion.

**Definition 2.5** (Perfect equilibrium). Consider a game  $\Gamma$ . A *trembling-hand perfect equilibrium*, or simply a *perfect equilibrium*, of  $\Gamma$  is a limit point of a sequence  $\{\mathbf{x}^{(\epsilon)} \in \mathcal{X}^{(\epsilon)}\}_{\epsilon \rightarrow 0+}$  where  $\mathbf{x}^{(\epsilon)}$  is an  $\epsilon$ -perfect equilibrium of  $\Gamma$ .

Due to Selten [69], we know that every normal form game has at least one perfect equilibrium and every perfect equilibrium must be a Nash equilibrium.

**Definition 2.6** ( $\epsilon$ -perturbed game for perfect equilibria). Consider a finite game  $\Gamma$ . For some  $\epsilon \in (0, 1)$  let

$$\mathcal{X}_i^{(\epsilon)} := \{\mathbf{x}_i \in \mathcal{X}_i \text{ such that } \mathbf{x}_i(a_i) \geq \epsilon \forall a_i \in \mathcal{A}_i\}$$

and  $\mathcal{X}^{(\epsilon)} = \prod_{i=1}^n \mathcal{X}_i^{(\epsilon)}$ . We define the perturbed game to be  $\Gamma^{(\epsilon)} = (\mathcal{X}_1^{(\epsilon)}, \dots, \mathcal{X}_n^{(\epsilon)}, u_1, \dots, u_n)$ .

From [Definitions 2.1](#) and [2.4](#) it follows that every Nash equilibrium of the perturbed game  $\Gamma^{(\epsilon)}$  is an  $\epsilon$ -perfect equilibrium of the original game  $\Gamma$ .

**Proper equilibrium** Proper equilibria [55] are refinements of perfect equilibria, in which the “perfection” property applies to *any* pair of actions where one action is better than the other. That is, if action  $a_i$  is worse than  $a'_i$ , then  $a_i$  must be played with probability at most  $\epsilon$  times that of  $a'_i$ , even if  $a'_i$  is not a best response. More formally, we have the following definition.

**Definition 2.7** ( $\epsilon$ -proper equilibrium). For some  $\epsilon > 0$  a mixed strategy profile  $\mathbf{x}$  is an  $\epsilon$ -proper equilibrium if it is fully mixed and

$$u_i(a_i, \mathbf{x}_{-i}) < u_i(a'_i, \mathbf{x}_{-i}) \implies \mathbf{x}_i(a_i) \leq \epsilon \mathbf{x}_i(a'_i) \text{ for all players } i \in [n] \text{ and actions } a_i, a'_i \in \mathcal{A}_i.$$

**Definition 2.8** (Proper equilibrium). A mixed strategy profile  $\mathbf{x}$  is a proper equilibrium if it is the limit point of a sequence  $\{\mathbf{x}^{(\epsilon)} \in \mathcal{X}^{(\epsilon)}\}_{\epsilon \rightarrow 0+}$  where  $\mathbf{x}^{(\epsilon)}$  is an  $\epsilon$ -proper equilibrium of  $\Gamma$ .

Due to Myerson [55], we know that in any normal-form game there exists at least one proper equilibrium. Furthermore, the proper equilibria form a subset of the perfect equilibria, which in turn are a subset of the Nash equilibria.

Kohlberg and Mertens [45] provide a constructive proof of the existence of a proper equilibrium in a normal-form game. This proof relies on computing a Nash equilibrium of a perturbed game,  $\Gamma^{(\epsilon)}$ , which in turn corresponds to an  $\epsilon$ -proper equilibrium of the original game,  $\Gamma$ .

For proper equilibria, we will use the following definition of a perturbed game due to Kohlberg and Mertens [45], which differs from that in [Definition 2.6](#).

**Definition 2.9** ( $\epsilon$ -perturbed game for proper equilibria). Consider a finite game  $\Gamma$ . For some  $\epsilon \in (0, 1)$  let the strategy space  $\mathcal{X}_i^{(\epsilon)}$  be

$$\mathcal{X}_i^{(\epsilon)} = \text{conv} \left( \left\{ \frac{1 - \epsilon}{1 - \epsilon^{m_i}} (\epsilon^{\pi(0)}, \epsilon^{\pi(1)}, \dots, \epsilon^{\pi(m_i-1)}) : \pi \in S_{m_i-1} \right\} \right),$$

where  $\text{conv}$  denotes the convex hull of a set of vectors,  $S_{m_i-1}$  is the set of all permutations of  $\{0, \dots, m_i - 1\}$  and  $\pi \in S_{m_i-1}$  is a permutation i.e., a bijection from  $\{0, \dots, m_i - 1\}$  to itself. We then define the perturbed game as  $\Gamma^{(\epsilon)} = (\mathcal{X}_1^{(\epsilon)}, \dots, \mathcal{X}_n^{(\epsilon)}, u_1, \dots, u_n)$ .

In the above definition, there is no need to redefine the utilities, as mixed strategies over mixed strategies of the original game are still just mixed strategies. We also remark that, with an overload in notation, we use  $\mathcal{X}_i^{(\epsilon)}$  to denote the perturbed strategy set for both perfect and proper equilibria, as the meaning will be clear from the context.

### 3 Search problems and modeling assumptions

Numbers in inputs will always be rational numbers represented as fractions in reduced form, given in binary. In general, we will use  $\text{size}(a)$  to denote the representation size of the object  $a$ , i.e., the number of bits needed to represent it as input. Our inputs, i.e., utility or potential functions, will be multilinear functions  $f : \mathcal{X} \rightarrow \mathbb{R}$ , usually provided as *arithmetic circuits*. To enforce multilinearity, we will only allow addition gates, multiplication gates, and rational constants; and we will insist that different inputs to any given multiplication gate must involve disjoint sets of players and rational constants (In particular, the sub-DAGs induced by the inputs from a multiplication gate must be disjoint). For the remainder of the paper, we will call circuits that satisfy the above assumptions *multilinear arithmetic circuits*.

**Lemma 3.1.** *Multilinear arithmetic circuits can be evaluated efficiently. That is, given a multilinear arithmetic circuit  $f : \mathbb{R}^M \rightarrow \mathbb{R}$  and input  $\mathbf{x} \in \mathbb{Q}^M$ , the output  $f(\mathbf{x}) \in \mathbb{Q}$  can be computed in time  $\text{poly}(\text{size}(f), \text{size}(\mathbf{x}))$ .*

*Proof.* We show the following claim inductively. For each gate  $G$ , let  $L_G$  be the description length of the sub-DAG induced by  $G$ , including any rational constants or inputs in that sub-DAG. Let  $P_G$  be the product of denominators of all rational constants and inputs in that sub-DAG. Clearly both  $L_G$  and  $\log_2 P_G$  are polynomial in the input length, and  $\log_2 P_G \leq L_G$ . We claim that the output of every gate  $G$  can be expressed as  $v_G = a_G / P_G$ , where  $a_G$  is an integer and  $|v_G| \leq 2^{L_G}$ . This would complete the proof immediately. We prove the claim inductively. The base case, for input or rational constant nodes, holds by definition. For the inductive case, let  $g$  and  $h$  be the two inputs of  $G$ .

If  $G$  is an addition gate, we have  $|v_G| \leq |v_g| + |v_h| \leq 2^{L_g} + 2^{L_h} \leq 2^{1+\max\{L_g, L_h\}} \leq 2^{L_G}$ , and the denominator of  $v_G$  is at most  $P_G$  because the inputs to  $G$  can be expressed with denominators  $P_g$  and  $P_h$  respectively, which by definition both divide  $P_G$ .

If  $G$  is a multiplication gate, then by definition we have  $P_G = P_g P_h$  and  $L_G > L_g + L_h$ . Therefore, we have  $|v_G| = |v_g| |v_h| \leq 2^{L_g} 2^{L_h} \leq 2^{L_G}$  and  $v_G = v_g v_h = a_g a_h / P_g P_h = a_g a_h / P_G$ , as desired.  $\square$

*Remark 3.2.* Multilinear arithmetic circuits immediately capture another natural representation:  $f$  is given as an explicit multivariate polynomial.

**Definition 3.3** (Concise game). A concise game  $\Gamma = (\mathcal{A}_1, \dots, \mathcal{A}_n, u_1, \dots, u_n)$  is a normal-form game in which each utility function,  $u_i : \mathcal{X} \rightarrow \mathbb{R}$ , is provided as a multilinear arithmetic circuit.

**Definition 3.4** (Concise potential game). A concise game  $\Gamma = (\mathcal{A}_1, \dots, \mathcal{A}_n, \Phi)$  is a normal-form potential game in which the potential function,  $\Phi : \mathcal{X} \rightarrow \mathbb{R}$ , is provided as a multilinear arithmetic circuit.

This assumption encompasses many classes of succinct, multi-player games [61], including poly-matrix games, bounded-degree graphical games, and anonymous games with a constant number of actions. We relax some of these assumptions, in Section 5.3, where we study general polytope games.

### 3.1 Search problems

Let  $\Sigma = \{0, 1\}$  and  $\Sigma^*$  denote the set of all finite binary strings. A *search problem* is defined by a relation  $R \subseteq \Sigma^* \times \Sigma^*$ . An algorithm *solves* the search problem  $R$  if, when given input  $x \in \Sigma^*$ , the algorithm outputs some  $y$  such that  $(x, y) \in R$ , or (correctly) asserts that no such  $y$  exists.

In the literature on complexity of search problems, it is standard to assume that the relation  $R$  is efficiently computable, that is, there is a polynomial-time algorithm that, on input  $(x, y)$ , checks whether  $(x, y) \in R$ . The set of search problems with this property is often called **FNP**. In this paper, however, we will commonly deal with search problems for which a polynomial-time algorithm exists, yet solutions cannot be efficiently verified. As a simple example of such a problem, consider

$$R_{\text{HALT}} = \{(x, y) : y \text{ is a Turing machine that halts on the empty tape}\}.$$

Then there is a constant-time algorithm for  $R_{\text{HALT}}$  (given any input  $x$ , output a trivial TM that halts immediately), but verifying a solution  $(x, y)$  is equivalent to the halting problem, which is undecidable. That is, under the usual definition of **FNP**,  $R_{\text{HALT}} \notin \text{FNP}$ . Therefore, in this paper we will use define the relevant complexity classes in a way that is slightly different from the usual definitions.<sup>3</sup>

**Definition 3.5** ([44]). A search problem  $R$  is in **FNP** if there exists a nondeterministic polynomial-time Turing machine  $M$  such that

1. the language recognized by  $M$  is  $D_R := \{x : \exists y \text{ s.t. } (x, y) \in R\}$ , and
2. on any accepting path of  $M$ , the TM outputs (to its tape) a  $y$  satisfying  $(x, y) \in R$ .

This definition does *not* imply the usual verifier definition of **FNP**: for example,  $R_{\text{HALT}} \in \text{FNP}$  by this definition, despite not being efficiently verifiable. However, we argue that it is in a sense more natural: is  $R_{\text{HALT}}$  really a hard problem (not even in **FNP** under the more common definition), when it has a trivial constant-time algorithm? In our paper, we will later see that many problems surrounding potential games are of this nature, for example, it is easy to find a perfect equilibrium of a potential game with a constant number of players (Proposition 5.1), but hard to verify such an equilibrium (Theorem 4.1).

Other classes of search problems can be defined by specifying complete problems for them. One way to define such problems is using Boolean circuits. We define a Boolean circuit  $C : \{0, 1\}^k \rightarrow \{0, 1\}^k$  as a function that can use the logic gates (AND), (OR), and (NOT), denoted as  $\wedge$ ,  $\vee$ , and  $\neg$  respectively.

---

<sup>3</sup>Our definition of **FNP** is also used by Johnson et al. [44], who call this class  $\text{NP}_S$ , where the  $S$  stands for *search*, but to our knowledge is rarely or never used by authors since then. Since reductions are defined the same way regardless, to our knowledge, all known results surrounding the complexity of search problems hold for these modified definitions as well.

Here we define the complexity class PLS [44] that contains *local search problems*. Informally, PLS contains all local search problems with neighborhoods that are searchable in polynomial time.

**Definition 3.6.** The complexity class PLS is the class of all problems that are reducible to **LOCALOPT**: given a finite set  $\mathcal{A} \subseteq [2^k]$  and two Boolean circuits  $S, V : \mathcal{A} \rightarrow [2^k]$ , find  $a \in \mathcal{A}$  such that  $V(S(a)) \leq V(a)$ .

The complexity class CLS [18, 29] contains all *continuous local search problems*. Informally, it contains all search problems seeking an approximate local optimum of a continuous function.

**Definition 3.7.** The class CLS is the class of all problems that are reducible to **CONTINUOUS-LOCALOPT**: given a precision parameter  $\epsilon > 0$ , well-behaved<sup>4</sup> circuits  $p : [0, 1]^k \rightarrow [0, 1]$  and  $g : [0, 1]^k \rightarrow [0, 1]$ , and a Lipschitz constant,  $L > 0$ , find  $\mathbf{x} \in [0, 1]^k$  such that  $p(g(\mathbf{x})) \geq p(\mathbf{x}) - \epsilon$ . Alternatively, find  $\mathbf{x}, \mathbf{y} \in [0, 1]^k$  such that  $|p(\mathbf{x}) - p(\mathbf{y})| \geq L\|\mathbf{x} - \mathbf{y}\|$  or  $\|g(\mathbf{x}) - g(\mathbf{y})\| \geq L\|\mathbf{x} - \mathbf{y}\|$ .

We now define the class PPAD [19, 60], which informally contains all problems reducible to the following: given a directed graph with  $[2^k]$  as the vertex set, and a source node, find another source or another sink. Each node has at most one incoming and at most one outgoing edge, which are identified by two Boolean circuits,  $S$  (successor) and  $P$  (predecessor).

**Definition 3.8.** The complexity class PPAD is the class of all problems that are reducible to **END-OF-LINE**: given a set  $\mathcal{A} \subseteq [2^k]$  and two Boolean circuits  $S, P : \mathcal{A} \rightarrow [2^k]$  with  $P(1) = 1 \neq S(1)$ , find  $v \in [2^k]$  such that  $P(S(v)) \neq v$  or  $S(P(v)) \neq v \neq 1$ .

The class FIXP [24] captures the complexity of fixed point problems with exact solutions. For many functions of interest, fixed points might be irrational, which motivates the definition of a discrete variant of FIXP, namely  $\text{FIXP}_a$  [24], which is most relevant to our work.

**Definition 3.9.** The complexity class  $\text{FIXP}_a$  is the class of all problems of the form: given a well-behaved arithmetic circuit  $f : D \rightarrow D$  defined on a convex and compact domain  $D$  and a precision parameter  $\epsilon > 0$  find  $\mathbf{x} \in D$  such that  $\|\mathbf{x} - \mathbf{x}^*\|_\infty \leq \epsilon$  for some  $\mathbf{x}^*$  where  $f(\mathbf{x}^*) = \mathbf{x}^*$  in time  $\text{poly}(\text{size}(C), \log(1/\epsilon))$ .

## 3.2 Evaluating circuits on polynomials

We will often want to evaluate circuits on inputs  $\mathbf{x}$  that are themselves functions of a single variable  $\epsilon$ . We first argue that such evaluations are well-behaved, in the sense that they also will not result in intermediate computations with high representation size.

**Lemma 3.10.** *Let  $f : \mathbb{R}^M \rightarrow \mathbb{R}$  be a polynomial of degree  $d_f$  represented by an arithmetic circuit, such that evaluating  $f(\mathbf{x})$  for a rational vector  $\mathbf{x} \in \mathbb{Q}^M$  takes time  $\text{poly}(\text{size}(f), \text{size}(\mathbf{x}))$ . Let  $\mathbf{x} \in \mathbb{Q}[\epsilon]^M$  be a vector of length  $M$  each of whose entries is a polynomial in  $\epsilon$  of degree at most  $d_x$ . Then the output  $f(\mathbf{x}) \in \mathbb{Q}[\epsilon]$  can also be computed in time  $\text{poly}(d_f, d_x, \text{size}(f), \text{size}(\mathbf{x}))$*

*Proof.* The composition  $g = f(\mathbf{x}) : \epsilon \mapsto f(\mathbf{x}(\epsilon))$  is a polynomial  $g : \mathbb{R} \rightarrow \mathbb{R}$ , of degree at most  $d_x \cdot d_f$ . It is thus uniquely determined by  $1 + d_x \cdot d_f$  values. Thus,  $g$  can be evaluated as follows: compute  $g(\epsilon) = f(\mathbf{x}(\epsilon))$  for  $\epsilon = 0, 1, 2, \dots, d_x \cdot d_f$ , and then recover  $g$  via polynomial interpolation, namely,

$$g(\epsilon) = \sum_{j=0}^{d_x \cdot d_f} g(j) \prod_{i \neq j} \frac{\epsilon - i}{j - i}. \quad \square$$

<sup>4</sup>Informally, a circuit is well-behaved if it does not allow repeated squaring. For more details, we refer the reader to Fearnley et al. [29] and <https://people.csail.mit.edu/costis/CLS-corrigendum.pdf>.

Let  $p, q$  be univariate polynomials. We use the notation  $p(\epsilon) <_{\epsilon \rightarrow 0^+} q(\epsilon)$  if there exists a sufficiently small  $\epsilon_0 > 0$  such that  $p(\epsilon) < q(\epsilon)$  for all  $\epsilon \in (0, \epsilon_0)$ .

**Lemma 3.11.** *For any  $L > 0$ , there is a function  $\psi : \mathbb{Q}[\epsilon] \rightarrow \mathbb{N}$  such that, for any two polynomials  $p, q \in \mathbb{Q}[\epsilon]$  of representation size at most  $L$ , (1)  $\psi(p), \psi(q)$  are computable in  $\text{poly}(L)$  time, and (2)  $p <_{\epsilon \rightarrow 0^+} q$  if and only if  $\psi(p) < \psi(q)$ .*

*Proof.* Comparing two polynomials  $p, q$  with  $<_{\epsilon \rightarrow 0^+}$  is equivalent to comparing their coefficients in lexicographic order. Since  $\text{size}(p) \leq L$ , the degree of  $p$  is also at most  $L$ . First, consider comparing rational numbers  $\alpha, \beta$  with size at most  $L$ . Such a rational number can have at most a  $L$ -bit denominator; therefore,  $\alpha - \beta$  has denominator at most  $4^L$ . Thus, if  $\alpha - \beta \neq 0$ , then  $\text{round}(4^L \alpha) - \text{round}(4^L \beta) \neq 0$ . Moreover, we have  $4^L |\alpha| < 4^L \cdot 2^L = 8^L$ , so  $0 \leq 4^L(\alpha + 2^L) < 2 \cdot 8^L \leq 16^L$ . Thus, it suffices to take the map  $\psi$  defined by

$$\psi\left(\sum_{i=0}^L \alpha_i \epsilon^i\right) = \sum_{i=0}^L \text{round}(4^L(\alpha_i + 2^L)) \cdot 16^{L(L-i)}. \quad \square$$

## 4 Lower bound on verification of perfect equilibrium

The first question is whether a purported perfect equilibrium in a potential game can be verified efficiently. Efficient verification is obvious for Nash equilibria, but, surprisingly, Hansen et al. [38] showed that even in three-player general-sum games the problem becomes NP-hard. In this section, we strengthen the result of Hansen et al. [38] by showing that even in a three-player *identical-interest* games NP-hardness persists.

**Theorem 4.1.** *It is NP-hard to check whether a given strategy profile is a perfect equilibrium, even when the game is an identical-interest game with three players and the given profile is pure.*

We dedicate the rest of this section to proving the theorem. Hansen et al. [38] showed that this problem is NP-hard for general-sum games; we adapt their proof. We reduce from the problem of computing the team minimax equilibrium in a three-player team game:

**Definition 4.2.** An *adversarial team game* is an  $n$ -player game in which  $n - 1$  players (the “team”) have the same utility function  $u$ , and the remaining player (the “adversary”) has utility  $-u$ . The *team minimax equilibrium (TME) value* is the smallest number  $r$  such that the team can force the adversary’s utility to be at most  $r$  by playing an uncorrelated mixed strategy profile.<sup>5</sup>

Computing even approximately the TME value of an adversarial team game is NP-hard.

**Theorem 4.3** ([12]). *It is NP-hard to approximate the TME value of a three-player adversarial team game with  $m$  actions per player to within an additive  $\Omega(1/m^2)$ . That is, given a game and a value  $r$ , it is NP-hard to distinguish whether the TME value is  $\geq r + \delta$  or  $\leq r - \delta$ , where  $\delta = \Omega(1/m^2)$ .*

We are now ready to prove Theorem 4.1. Given a three-player team game and a purported value  $r$  for the team, construct the following identical-interest game: add an action  $\perp$  for each player. Utilities are defined as follows.

1. If no one plays  $\perp$ , everyone gets the value that the *adversary* would have gotten in the team game.

---

<sup>5</sup>Following Hansen et al. [38], values will be from the perspective of the adversary.

2. Otherwise, if the adversary and *exactly* one team member plays  $\perp$ , then the utility is  $r - \delta/2$ .
3. Otherwise, the utility is  $r$ .

We now prove both directions of the reduction.

**Lemma 4.4.** *If the TME value is  $\leq r - \delta$ , then  $\mathbf{x}^\perp := (\perp, \perp, \perp)$  is a perfect equilibrium.*

*Proof.* Let  $\mathbf{x}^*$  be any strategy profile in which the team plays according to the TME. Consider the trembling sequence

$$\mathbf{x}^{(\epsilon)} := (1 - \epsilon - \epsilon^2) \cdot \mathbf{x}^\perp + \epsilon \cdot \mathbf{x}^* + \epsilon^2 \cdot \mathbf{f}$$

where  $\mathbf{f}$  is any fully mixed strategy profile. Clearly,  $\mathbf{x}^{(\epsilon)} \rightarrow \mathbf{x}^\perp$  as  $\epsilon \rightarrow 0$ . For team members,  $\perp$  is a strict best response to  $\mathbf{x}^\perp$  (since any other action has expected utility roughly  $r - \delta/2$ ), so team members are best responding in the limit  $\epsilon \rightarrow 0$ . For the adversary, the value of playing  $\perp$  is  $r \pm O(\epsilon^2)$ , and the value of playing any other action is upper-bounded by  $(1 - \epsilon)r + \epsilon(r - \delta) \pm O(\epsilon^2) = r - \epsilon\delta \pm O(\epsilon^2) < r \pm O(\epsilon^2)$  for  $\epsilon$  sufficiently small, so  $\perp$  is a best response. Thus,  $\mathbf{x}^\perp$  is a perfect equilibrium.  $\square$

**Lemma 4.5.** *If the TME value is  $\geq r + \delta$ , then  $\mathbf{x}^\perp$  is not a perfect equilibrium.*

*Proof.* Consider any fully mixed strategy profile  $\mathbf{x}$ . Let  $\mathbf{y}$  be the team's strategy in  $\mathbf{x}$ , conditioned on playing actions other than  $\perp$ . We claim that the adversary's best response to  $\mathbf{x}$  cannot be  $\perp$ , which would complete the proof since then, the adversary cannot play  $\perp$  with probability larger than  $\epsilon$  in any  $\epsilon$ -PE. Indeed, the expected value of action  $\perp$  for the adversary is at most  $r$ , but best-responding to  $\mathbf{y}$  gets expected value strictly larger than  $r$ , since it gets  $r$  when at least one team member plays  $\perp$ , and at least  $r + \delta$  otherwise.  $\square$

**Theorem 4.1** now follows immediately from the previous two lemmas.

## 5 Computing pure perfect equilibria

Notwithstanding **Theorem 4.1**, this section shows that computing pure perfect equilibria is PLS-complete even in general polytope games. We begin by examining the simple case of explicitly represented normal-form games (**Section 5.1**), where we think of the number of players as being a constant. In **Section 5.2** we then deal with concise potential games, and **Section 5.3** generalizes our approach to any polytope game.

### 5.1 Constant number of players

To begin with, we deal with perfect equilibria in potential games with a constant number of players. Perhaps the most natural approach is to identify any strategy corresponding to a maximum entry in the payoff tensor. However, this approach falls short: the example of **Figure 5** shows a game in which 3 different action profiles maximize the utility, but only one of them is a perfect equilibrium: if  $\mathbf{x}_1 \in \Delta(2) := \{\mathbf{x}_1 \in \mathbb{R}_{\geq 0}^2 : \mathbf{x}_1(\text{R1}) + \mathbf{x}_1(\text{R2}) = 1\}$  and  $\mathbf{x}_2 \in \Delta(2)$  are fully mixed, then  $u_1(\text{R1}, \mathbf{x}_2) > u_1(\text{R2}, \mathbf{x}_2)$  and  $u_2(\bar{\text{C1}}, \mathbf{x}_1) > u_2(\text{C2}, \mathbf{x}_2)$ , so  $\mathbf{x}_1(\text{R2}) \leq \epsilon$  and  $\mathbf{x}_2(\text{C2}) \leq \epsilon$  in the perturbed game. This means that  $(\text{R1}, \text{C1})$  is the unique perfect equilibrium.

Nevertheless, there is a simple algorithm that solves this problem, which is to compute the maximum entry of the potential function *in the perturbed game* in the limit where  $\epsilon$  is arbitrarily small. In the example of **Figure 5**, we have  $(\text{R1}, \text{C1}) \mapsto 1 - \epsilon^2$ ;  $(\text{R1}, \text{C2}) \mapsto 1 - \epsilon(1 - \epsilon)$ ;  $(\text{R2}, \text{C1}) \mapsto$



	C1	C2
R1	1, 1	1, 1
R2	1, 1	0, 0

Figure 5: A  $2 \times 2$  identical-interest game in normal form with a unique perfect equilibrium, namely (R1, C1).

$1 - \epsilon(1 - \epsilon)$ ;  $(R2, C2) \mapsto 1 - (1 - \epsilon)^2$ . As a result, (R1, C1) is to be returned. It is worth pointing out that this approach contains a simple but useful idea: one can transfer the perturbation from the strategy sets into the utilities. We will also make use of this transformation later on.

**Proposition 5.1.** *For any potential game with a constant number of players, there is a polynomial-time algorithm for computing a perfect equilibrium.*

For every joint action profile  $(a_1, \dots, a_n)$ , we compute the expected potential as a polynomial in  $\epsilon$  when each player  $i \in [n]$  plays  $a_i$  with probability  $1 - (m_i - 1)\epsilon$  and every other action with probability  $\epsilon$ . Each such polynomial can be written explicitly in polynomial time since the game is assumed to have a constant number of players. We then compute the largest such polynomial, where comparison of polynomials is done lexicographically. It is easy to see that the corresponding action profile is a pure perfect equilibrium of the game. (An analogous approach succeeds even under the stronger notion of proper equilibria, introduced in [Definition 2.8](#).)

Computing equilibrium refinements becomes more interesting in a concise potential games, which is the subject of the rest of this section.

## 5.2 Algorithmic approach and complexity implications

We consider best-response dynamics on the perturbed game,  $\Gamma^{(\epsilon)}$ , in which  $\epsilon$  is sufficiently small. At each timestep, we pick an arbitrary player,  $i \in [n]$  and let them best-respond to the current strategies of the other players. The player will change their strategy if there exists  $a_i \in \mathcal{A}_i$  such that

$$u_i(\mathbf{a}_i^{(\epsilon)}, \mathbf{x}_{-i}) > u_i(\mathbf{x}_i, \mathbf{x}_{-i}),$$

where with  $\mathbf{a}_i^{(\epsilon)}$  we denote the  $\epsilon$ -pure strategy that assigns maximum probability mass to action  $a_i$ .

In [Algorithm 1](#) we present the symbolic best-response dynamics and in [Theorem 5.2](#) we prove that they converge to a Nash equilibrium of the perturbed game.

---

### Algorithm 1 $\epsilon$ -symbolic best-response dynamics

---

- 1: Initialize for all  $i \in [n]$ ,  $\mathbf{x}_i^{(\epsilon)} \in \mathcal{X}_i^{(\epsilon)}$
  - 2: **while**  $\exists i \in [n], \tilde{\mathbf{x}}_i^{(\epsilon)} \in \mathcal{X}_i^{(\epsilon)}$  such that  $u_i(\mathbf{x}_i^{(\epsilon)}, \mathbf{x}_{-i}^{(\epsilon)}) <_{\epsilon \rightarrow 0^+} u_i(\tilde{\mathbf{x}}_i^{(\epsilon)}, \mathbf{x}_{-i}^{(\epsilon)})$  **do**
  - 3:     Set  $\mathbf{x}_i^{(\epsilon)} = \arg \max_{\mathbf{x} \in \mathcal{X}_i^{(\epsilon)}} u_i(\mathbf{x}, \mathbf{x}_{-i}^{(\epsilon)})$
  - 4: **return**  $\mathbf{x}^{(\epsilon)}$
- 

**Theorem 5.2.** *[Algorithm 1](#) returns a perfect equilibrium of  $\Gamma$  in finite time, and each iteration can be implemented in polynomial time in the size of the input.*

*Proof.* We first argue about the per-iteration complexity. [Algorithm 1](#) maintains the invariance that, for all  $i \in [n]$  and  $a_i \in \mathcal{A}_i$ ,  $\mathbf{x}_i(a_i) = \epsilon$  or  $\mathbf{x}_i(a_i) = 1 - (m_i - 1)\epsilon$ . Thus, for concise games, we

can determine in polynomial time the polynomials  $p(\epsilon) := u_i(\mathbf{x}_i, \mathbf{x}_{-i})$  and  $q(\epsilon) := u_i(\mathbf{a}_i^{(\epsilon)}, \mathbf{x}_{-i})$  in terms of  $\epsilon$  by [Lemma 3.10](#). Both of those polynomials have degree at most  $n$ . Let  $p(\epsilon) = \sum_{i=0}^n p_i \epsilon^i$  and  $q(\epsilon) = \sum_{i=0}^n q_i \epsilon^i$ . To ascertain whether  $p(\epsilon) <_{\epsilon \rightarrow 0+} q(\epsilon)$ , we determine whether  $p_i < q_i$  for some  $i \in [n] \cup \{0\}$  and  $p_{i'} = q_{i'}$  for all  $i' < i$ .

We now turn to the proof of convergence. For an  $\epsilon$ -pure strategy profile  $\mathbf{x}^{(\epsilon)} \in \mathcal{X}^{(\epsilon)}$ , the potential function value is a polynomial  $p(\epsilon)$ . This potential function increases (with respect to the ordering  $<_{\epsilon \rightarrow 0+}$ ) after every best-response step and it can only take on finitely many values, so the algorithm must eventually stop at some profile  $\mathbf{x}^{(\epsilon)}$ . At this point, no player has a symbolic best response, so  $\mathbf{x}^{(\epsilon)}$  is a Nash equilibrium of the perturbed game in the sense that  $u_i(\tilde{\mathbf{x}}_i^{(\epsilon)}, \mathbf{x}_{-i}^{(\epsilon)}) <_{\epsilon \rightarrow 0+} u_i(\mathbf{x}^{(\epsilon)})$  for all  $i \in [n]$  and  $\tilde{\mathbf{x}}_i^{(\epsilon)} \in \mathcal{X}_i^{(\epsilon)}$ . Therefore, there exists  $\epsilon_0 > 0$  small enough such that  $u_i(\tilde{\mathbf{x}}^{(\epsilon)}) < u_i(\mathbf{x}^{(\epsilon)})$  for all  $\epsilon \in (0, \epsilon_0)$ . Therefore,  $\{\mathbf{x}^{(\epsilon)}\}_{\epsilon \rightarrow 0+}$  defines a sequence of an  $\epsilon$ -perfect equilibria of  $\Gamma$ , the limit point of which is a perfect equilibrium.  $\square$

From a complexity standpoint, we prove PLS-membership for the problem of finding perfect equilibria in potential games.

**Theorem 5.3.** *Finding a perfect equilibrium of a concise potential game is in PLS.*

*Proof.* We construct the two required circuits. The feasible set  $\mathcal{A}$ , as suggested by the notation, is the set of pure strategy profiles. A pure strategy profile  $a \in \mathcal{A}$  is interpreted as corresponding to the  $\epsilon$ -pure strategy  $\mathbf{x}_a = (\mathbf{a}_1^{(\epsilon)}, \dots, \mathbf{a}_n^{(\epsilon)}) \in \mathcal{X}^{(\epsilon)}$ , where  $\epsilon$  is symbolic. Define the value function  $V : \mathcal{A} \rightarrow \mathbb{N}$  by  $V(a) = \psi(\Phi(\mathbf{x}_a))$ , where  $\psi$  is the function guaranteed by [Lemma 3.11](#). (By [Lemma 3.10](#),  $V$  is efficiently computable.) Define the step function  $S$  as a symbolic better response function: given  $a \in \mathcal{A}$ , find a player  $i$  and an action  $a'_i \in \mathcal{A}_i$  such that  $\Phi(\mathbf{x}_{(a'_i, a_{-i})}) >_{\epsilon \rightarrow 0+} \Phi(\mathbf{x}_a)$  and output  $(a'_i, a_{-i})$ ; if no such  $a'_i$  exists, output  $a$  itself. Since every local optimum of  $V$  is by construction also a local optimum of  $a \mapsto \Phi(\mathbf{x}_a)$  (with respect to  $<_{\epsilon \rightarrow 0+}$ ), it must also be a perfect equilibrium.  $\square$

### 5.3 Perfect equilibria of polytope games

Next, we generalize the results of the previous section beyond normal-form games, to *polytope games*. In this setting, the pure action sets  $\mathcal{A}_i$  are points in  $\mathbb{R}^d$ , and the polytope  $\mathcal{X}_i$  is the convex hull of  $\mathcal{A}_i$ . The utility functions  $u_i : \mathcal{X} \rightarrow \mathbb{R}$  are, as usual, assumed to be given as multilinear circuits. We will assume that  $\mathcal{X}_i$  is full-dimensional, and indeed we will assume that for each player  $i$  we are given a point  $\mathbf{o}_i$  that lies on the strict interior of  $\mathcal{X}_i$ .

Our assumptions on the set  $\mathcal{X}_i$  will be very general. Namely, we will only assume that we are given a best response (linear optimization) oracle for each player  $i$ , that is, an efficient Turing machine that, given a rational vector  $\mathbf{u} \in \mathbb{R}^d$ , outputs a rational point  $\mathbf{x}_i \in \mathcal{A}_i$  that maximizes the inner product  $\langle \mathbf{u}, \mathbf{x}_i \rangle$ . For polytopes, such an oracle can be implemented using typical oracles for convex sets such as a separation or membership oracle [\[35\]](#). We will also assume that the pure action sets  $\mathcal{A}_i$  are “nice”, in particular, that they have at most exponentially many vertices, each having polynomially bounded representation size.

Polytope games concisely describe many general and important classes of games, including Bayesian games (where  $\mathcal{X}_i$  is a product of simplices), network congestion games (where  $\mathcal{X}_i$  is the set of flows on a graph), and extensive-form games (where  $\mathcal{X}_i$  is the sequence-form polytope, which we will elaborate on in [Section A](#)).

We will be interested in computing perfect equilibria of polytope games. In this section, we will interpret the polytope game as merely a concise representation of an underlying normal-form game

in which the action sets are the  $\mathcal{A}_i$ s, so we will be interested in perfect equilibria of the underlying normal-form game. For extensive-form games in particular, there are several alternative ways of defining concepts related to perfect equilibria; we elaborate on those in [Section A](#) as well.

We now redo the definitions of [Section 5](#) in this more general setting.

**Definition 5.4** ( $\epsilon$ -perturbed game for polytope games). Let  $\epsilon > 0$ . The  $\epsilon$ -perturbed strategy set  $\mathcal{X}_i^{(\epsilon)}$  of player  $\mathcal{X}_i$  is defined as

$$\mathcal{X}_i^{(\epsilon)} := \{(1 - \epsilon)\mathbf{x}_i + \epsilon \mathbf{o}_i : \mathbf{x}_i \in \mathcal{X}_i\}.$$

As before, we define  $\mathcal{X}^{(\epsilon)} = \prod_{i=1}^n \mathcal{X}_i^{(\epsilon)}$  and the perturbed game  $\Gamma^{(\epsilon)} = (\mathcal{X}_1^{(\epsilon)}, \dots, \mathcal{X}_n^{(\epsilon)}, u_1, \dots, u_n)$ .

With these definitions, the remainder of the analysis from the normal-form section follows analogously, and we will not repeat it. We arrive at the following results, whose proofs follow analogously from the corresponding proofs from [Section 5](#):

**Theorem 5.5.** *[Algorithm 1](#) returns a perfect equilibrium in finite time, and each iteration can be implemented in polynomial time in the size of the input.*

**Theorem 5.6.** *Finding a perfect equilibrium of a concise polytope potential game is in PLS.*

Indeed, the same proof shows the following more general result, which we will reference in the remainder of the paper.

**Theorem 5.7.** *Let  $\{\mathcal{X}_i^{(\epsilon)}\}_{i \in [n]}$ , where  $\epsilon$  is symbolic, be a collection of strategy sets, one per player. Assume that for every player  $i$  there is an efficient algorithm for linear optimization over  $\mathcal{X}_i^{(\epsilon)}$ . Then the problem of computing a symbolic exact Nash equilibrium of a potential game in which each player  $i$  has strategy set  $\mathcal{X}_i^{(\epsilon)}$  is in PLS for potential games.*

## 6 Mixed perfect equilibria

Having fully characterized the complexity of pure perfect equilibria in potential games, we are concerned in this section with *mixed* perfect equilibria. [Section 6.1](#) deals with the special case of polymatrix games, while [Section 6.2](#) presents a lower bound in general potential games.

### 6.1 Almost implies near: a refinement in polymatrix games

We first restrict our attention to polymatrix games, a specific class of succinct games (introduced earlier in [Definition 2.3](#)). For such games, it follows from the result of Hansen and Lund [\[36\]](#) that computing a perfect equilibrium is in PPAD.<sup>6</sup> Combining with the PLS membership established in [Theorem 5.3](#), it follows that computing a perfect equilibrium in polymatrix potential games is in CLS since  $\text{CLS} = \text{PPAD} \cap \text{PLS}$  [\[29\]](#).

**Corollary 6.1.** *Computing a (mixed) perfect equilibrium in polymatrix potential games is in CLS.*

In terms of hardness results, it is generally believed that computing a Nash equilibrium is CLS-hard; for example, Hollender et al. [\[41\]](#) established CLS-hardness but in the presence of multiple independent adversaries; this CLS-hardness result was further strengthened by Anagnostides,

---

<sup>6</sup>More precisely, Hansen and Lund [\[36\]](#) showed that computing an  $\epsilon$ -symbolic proper equilibrium is in PPAD; this implies that computing a perfect equilibrium is also in PPAD ([Section 3.1](#)).

Panageas, Sandholm, and Yan [2]. In any event, in what follows, we will provide a certain equivalence between Nash and perfect equilibria in polymatrix potential games ([Proposition 6.8](#)).

The foregoing CLS membership in [Corollary 6.1](#) is established rather indirectly: having established PLS membership in [Theorem 5.3](#), we then make use of the existing PPAD membership together with the fact that  $\text{CLS} = \text{PPAD} \cap \text{PLS}$ . This begs the question of whether there is a more direct CLS membership through the use of gradient descent; this is the main subject of this subsection.

**Almost implies near in general games** The key step in the analysis is a refinement of the “almost implies near” paradigm of Anderson [3]. To begin with, we treat general games—without the polymatrix restriction. As observed by Etessami et al. [25], a direct application of the result of Anderson [3] implies that for any game  $\Gamma$  and  $\delta > 0$ , there is an  $\epsilon = \epsilon(\delta)$  such that any  $\epsilon$ -PE of  $\Gamma$  is within  $\ell_\infty$  distance  $\delta$  from an exact PE of  $\Gamma$ . Etessami et al. [25] provided a quantitative bound using techniques from real algebraic geometry [9], which is recalled below. (We use the shorthand notation  $M = \sum_{i=1}^n m_i$ .)

**Lemma 6.2** ([25]). *For any game  $\Gamma$ , a sufficiently small  $\delta \in \mathbb{Q}_{>0}$ , and  $\epsilon \leq \delta^{n^{O(M^3)}}$ , any  $\epsilon$ -PE of  $\Gamma$  is within  $\ell_\infty$  distance  $\delta$  from an exact PE of  $\Gamma$ .*

We first point out a refinement of the above lemma that accounts for  $\epsilon$ -perfect  $\epsilon'$ -almost equilibria of  $\Gamma$ , which means that  $u_i(a_i, \mathbf{x}_{-i}) < u_i(a'_i, \mathbf{x}_{-i}) - \epsilon' \implies \mathbf{x}_i(a_i) \leq \epsilon$  for all players  $i \in [n]$  and actions  $a_i, a'_i \in \mathcal{A}_i$ ; this is the  $\epsilon'$ -well-supported approximation of  $\epsilon$ -perfect equilibria, which is meaningful only when  $\epsilon' \ll \epsilon$ .<sup>7</sup> We will prove the following property.

**Lemma 6.3.** *For any game  $\Gamma$ ,  $\epsilon \in \mathbb{Q}_{>0}$ ,  $\delta \in \mathbb{Q}_{>0}$ , and  $\epsilon' \leq \min(\epsilon, \delta)^{n^{O(M^2)}}$ , any  $\epsilon$ -perfect  $\epsilon'$ -almost equilibrium of  $\Gamma$  is within  $\ell_\infty$  distance  $\delta$  from an  $\epsilon$ -PE of  $\Gamma$ .*

*Proof.* For a fixed  $\epsilon > 0$ , we define  $\text{ALMOSTPE}(\mathbf{x}, \epsilon, \epsilon')$  to be the quantifier-free first-order formula with free variables  $\mathbf{x} \in \mathbb{R}^M$  and  $\epsilon' > 0$  expressed as the conjunction of the following formulas:

$$\begin{aligned} & \mathbf{x}_i(a_i) > 0 \quad i \in [n], a_i \in \mathcal{A}_i, \\ & \sum_{a_i \in \mathcal{A}_i} \mathbf{x}_i(a_i) = 1 \quad i \in [n], \\ & (u_i(a_i, \mathbf{x}_{-i}) \geq u_i(a'_i, \mathbf{x}_{-i}) - \epsilon') \vee (\mathbf{x}_i(a_i) \leq \epsilon) \quad i \in [n], a_i, a'_i \in \mathcal{A}_i. \end{aligned} \quad (\text{ALMOSTPE}(\mathbf{x}, \epsilon, \epsilon'))$$

We also define  $\text{PE}(\mathbf{x}, \epsilon) \equiv \text{ALMOSTPE}(\mathbf{x}, \epsilon, 0)$  with free variable  $\mathbf{x} \in \mathbb{R}^M$ . What we want to prove is that

$$\forall \mathbf{x} \in \mathbb{R}^M \exists \mathbf{x}' \in \mathbb{R}^M : (\epsilon' > 0) \wedge (\neg \text{ALMOSTPE}(\mathbf{x}, \epsilon, \epsilon') \vee (\text{PE}(\mathbf{x}') \wedge \|\mathbf{x} - \mathbf{x}'\|_2^2 \leq \delta^2)).$$

( $\|\mathbf{x} - \mathbf{x}'\|_2 \leq \delta$  implies  $\|\mathbf{x} - \mathbf{x}'\|_\infty \leq \delta$ .) We denote by  $\text{PEBOUND}_{\epsilon, \delta}(\epsilon')$  the above first-order formula, with  $\epsilon'$  acting as a free variable. Suppose that each payoff can be represented with at most  $\tau$  bits,  $\epsilon = 2^{-r}$ , and  $\delta^2 = 2^{-k}$ . For  $\text{PEBOUND}_{\epsilon, \delta}(\epsilon')$  we have the following facts:

- The maximum degree of all involved polynomials in the formula is at most  $\max\{2, n - 1\}$ .
- Each coefficient requires at most  $\max(k, \tau, r)$  number of bits to be represented.

<sup>7</sup>Any Nash equilibrium can be trivially converted into an  $\epsilon$ -perfect  $O_\epsilon(\epsilon)$ -almost equilibrium by redistributing the probability mass of each strategy so that each action is allotted a probability of at least  $\epsilon$ .

- There is only a single free variable, namely  $\epsilon'$ .
- The formula is in prenex normal form, containing two blocks of quantifiers of size  $M$  and  $M$ , respectively.

Using quantifier elimination [9, Algorithm 14.5], we can convert  $\text{PEBOUND}_{\epsilon,\delta}(\epsilon')$  to an equivalent quantifier-free formula  $\text{PEBOUND}'_{\epsilon,\delta}(\epsilon')$ , again with a single free variable  $\epsilon$ . It follows, by Basu et al. [9, Theorem 14.16], that the degree of all involved polynomials (which are univariate polynomials in  $\epsilon$ ) is at most  $n^{O(M^2)}$ , and the bit complexity of each coefficient in the formula is at most  $\max(k, r, \tau)n^{O(M^2)}$ . By Anderson [3], we know that  $\text{PEBOUND}'_{\epsilon,\delta}(\epsilon')$  is satisfied for a sufficiently small  $\epsilon' > 0$ . It then follows that  $\text{PEBOUND}'_{\epsilon,\delta}(\epsilon')$  is satisfied for some  $\epsilon' = \epsilon^* \geq 2^{-\max(k,r,\tau)n^{O(M^2)}}$  [9], which in turn implies that  $\text{PEBOUND}'_{\epsilon,\delta}(\epsilon')$  is satisfied for any  $\epsilon' \leq \epsilon^*$  in view of the semantics of the formula.  $\square$

**Lemma 6.3** together with **Lemma 6.2** imply that, in principle, one can use gradient descent on the perturbed game  $\Gamma^{(\epsilon)}$ , for a concrete numerical value  $\epsilon$ , to find a strategy profile that is arbitrarily close to a perfect equilibrium in any potential game. The problem, of course, is that  $\epsilon$  needs to be doubly exponentially small, and the number of iterations of gradient descent will be doubly exponentially large since  $\epsilon' \leq \delta^{n^{O(M^3)}n^{O(M^2)}} = \delta^{n^{O(M^3)}}$  (**Lemma 6.3**).

More precisely, the previous claim follows from the usual analysis of gradient descent (**Proposition 6.4**), which we include below for completeness.

**Analysis of gradient descent** Under a constraint set  $\mathcal{X}$ , the update rule of (projected) gradient descent reads

$$\mathbf{x}(t+1) = \Pi_{\mathcal{X}}(\mathbf{x}(t) + \eta \nabla \Phi(\mathbf{x}(t))), \quad (\text{GD})$$

where  $\Pi_{\mathcal{X}}$  denotes the Euclidean projection. Now, if  $\Phi$  is  $L$ -smooth, meaning that  $\|\nabla \Phi(\mathbf{x}) - \nabla \Phi(\mathbf{x}')\|_2 \leq L\|\mathbf{x} - \mathbf{x}'\|_2$ , we have

$$\Phi(\mathbf{x}') \geq \Phi(\mathbf{x}) + \langle \nabla \Phi(\mathbf{x}), \mathbf{x}' - \mathbf{x} \rangle - \frac{L}{2}\|\mathbf{x} - \mathbf{x}'\|_2^2 \quad (1)$$

for any  $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ . By the nonexpansiveness of  $\Pi_{\mathcal{X}}$ , we have

$$\|\mathbf{x}(t+1) - \mathbf{x}(t)\|_2 = \|\Pi_{\mathcal{X}}(\mathbf{x}(t) + \eta \nabla \Phi(\mathbf{x}(t))) - \Pi_{\mathcal{X}}(\mathbf{x}(t))\|_2 \leq \eta \|\nabla \Phi(\mathbf{x}(t))\|_2 \leq \eta L, \quad (2)$$

where we assume that  $\|\nabla \Phi(\mathbf{x})\|_2 \leq L$  for all  $\mathbf{x} \in \mathcal{X}$ . Furthermore, by the first-order optimality condition of gradient descent, we have

$$\langle \eta \nabla \Phi(\mathbf{x}(t)) - (\mathbf{x}(t+1) - \mathbf{x}(t)), \mathbf{x}' - \mathbf{x}(t) \rangle \geq 0 \quad \forall \mathbf{x}' \in \mathcal{X},$$

which in turn implies

$$\langle \nabla \Phi(\mathbf{x}(t)), \mathbf{x}(t+1) - \mathbf{x}(t) \rangle \geq \frac{1}{\eta} \|\mathbf{x}(t+1) - \mathbf{x}(t)\|_2^2. \quad (3)$$

Combining (1), (2), and (3), we conclude that

$$\Phi(\mathbf{x}(t+1)) - \Phi(\mathbf{x}(t)) \geq \left( \frac{1}{\eta} - \frac{\eta^2 L^3}{2} \right) \|\mathbf{x}(t+1) - \mathbf{x}(t)\|_2^2 \geq \frac{1}{2\eta} \|\mathbf{x}(t+1) - \mathbf{x}(t)\|_2^2$$

when  $\eta \leq 1/L$ . The telescopic summation yields that after  $O_\epsilon(1/\epsilon^2)$  iterations there is a point  $\mathbf{x}(t)$  such that  $\|\mathbf{x}(t+1) - \mathbf{x}(t)\|_2 \leq \epsilon$ . By the first-order optimality condition, this implies

$$\langle \nabla \Phi(\mathbf{x}(t)), \mathbf{x}' - \mathbf{x}(t) \rangle \geq -\frac{D_{\mathcal{X}}}{\eta} \epsilon \quad \forall \mathbf{x}' \in \mathcal{X},$$

where  $D_{\mathcal{X}}$  denotes the  $\ell_2$  diameter of  $\mathcal{X}$ . We summarize this standard guarantee below.

**Proposition 6.4.** *Gradient descent on any  $L$ -smooth function on a bounded domain takes at most  $O_\epsilon(1/\epsilon^2)$  iterations to converge to a first-order stationary point  $\mathbf{x}(t)$ , that is,*

$$\langle \nabla \Phi(\mathbf{x}(t)), \mathbf{x}' - \mathbf{x}(t) \rangle \geq -\epsilon \quad \forall \mathbf{x}' \in \mathcal{X},$$

In our application, we set the constraint set to be the  $\epsilon$ -perturbed joint strategy set  $\mathcal{X}^{(\epsilon)}$ , and we execute gradient descent long enough so that there is a strategy  $\mathbf{x}(t)$  such that  $\langle \nabla \Phi(\mathbf{x}(t)), \mathbf{x}' - \mathbf{x}(t) \rangle \geq -\epsilon'$  for any  $\mathbf{x}' \in \mathcal{X}^{(\epsilon)}$ ; this can be converted into an  $\epsilon$ -perfect  $O_{\epsilon'}(\sqrt{\epsilon'})$ -almost equilibrium per the well-supported notion [16].

**Polymatrix games** We now observe that Lemma 6.2 can be significantly improved in polymatrix games. This refinement will allow us to obtain a more direct CLS membership for computing perfect equilibria in that class of games. We begin with the following simple observation relating perfect equilibria in polymatrix games and *linear complementarity problems (LCPs)*; we refer to Hansen and Lund [36] for an analogous connection pertaining to the more refined notion of proper equilibria.

**Lemma 6.5.** *Any perfect equilibrium of a polymatrix game can be obtained as a limit point, as  $\epsilon \rightarrow 0$ , of a solution to the perturbed standard-form LCP,  $\mathcal{P}^{(\epsilon)}$ , defined as*

$$\begin{aligned} & \text{find} && \mathbf{z}, \mathbf{w} \in \mathbb{R}^d \\ & \text{subject to} && \mathbf{z}^\top \mathbf{w} = 0, \\ & && \mathbf{w} = \mathbf{M}^{(\epsilon)} \mathbf{z} + \mathbf{b}^{(\epsilon)}, \\ & && \mathbf{z}, \mathbf{w} \geq 0. \end{aligned}$$

Furthermore, every entry of  $\mathbf{M}^{(\epsilon)}$  and  $\mathbf{b}^{(\epsilon)}$  has degree at most 1 as a polynomial in  $\epsilon$  and each coefficient can be represented with a polynomial number of bits.

We recall that a *basis*  $B$  for a standard-form LCP with constraints  $\mathbf{w} = \mathbf{M}\mathbf{z} + \mathbf{b}$  is a set of linearly independent columns of  $\mathbf{M}$  such that the associated (complementary) solution—called *basic solution*—is feasible. In this context, we borrow the following definition of Farina and Gatti [27].

**Definition 6.6** (Negligible positive perturbation; NPP). Let  $\mathcal{P}^{(\epsilon)}$  be an LCP parameterized on some perturbation parameter  $\epsilon$ . The value  $\epsilon^* > 0$  is a *negligible positive perturbation (NPP)* if any (complementary) basis  $B$  for  $\mathcal{P}^{(\epsilon^*)}$  remains a basis for  $\mathcal{P}^{(\epsilon)}$  for all  $0 \leq \epsilon \leq \epsilon^*$ .

Farina and Gatti [27] gave a general result regarding perturbed LCPs per Lemma 6.5.

**Lemma 6.7** ([27]). *Suppose that each entry of  $\mathbf{M}^{(\epsilon)}$  and  $\mathbf{b}^{(\epsilon)}$  has degree  $\text{poly}(d)$  and each coefficient can be represented with  $\text{poly}(d)$  bits. Then there exists some negligible positive perturbation  $\epsilon^* \in \mathbb{Q}_{>0}$  expressed in  $\text{poly}(d)$  bits for  $\mathcal{P}^{(\epsilon)}$ .*

We thus arrive at the following consequence.

**Proposition 6.8.** *For any polymatrix game  $\Gamma$ , there exists a polynomial and  $\epsilon \leq 2^{p(|\Gamma|)}$  such that any  $\epsilon$ -perfect equilibrium of  $\Gamma$  induces an exact perfect equilibrium of  $\Gamma$ .*

This makes the problem of computing an exact perfect equilibrium of  $\Gamma$  directly amenable to gradient descent, establishing an alternative proof of CLS membership. Indeed, one can run gradient descent (Proposition 6.4) on the polymatrix potential game  $\Gamma^{(\epsilon^*)}$  to identify an  $\epsilon^*$ -perfect equilibrium through a standard rounding procedure [30, Appendix A]; since  $\epsilon^* \leq 2^{p(\Gamma)}$ ,  $\Gamma^{(\epsilon^*)}$  can be equivalently expressed as a polymatrix game (without any perturbation in the strategy sets) where each entry in the payoff matrices has polynomially many bits. The same argument goes through for proper equilibria in polymatrix games.

**Symbolic gradient descent** While Proposition 6.8 shows that a numerical version of gradient descent, executed on the perturbed game for a sufficient number of iterations, will succeed, we now point out that *symbolic* gradient descent can fail. This can be seen even in the simple example of Figure 6, as we point out in Proposition 6.9.

We first clarify what we mean by “symbolic gradient descent.” Every player  $i \in [n]$  is assumed to initialize at some symbolic strategy  $\mathbf{x}_i^{(\epsilon)}(0) \in \mathcal{X}^{(\epsilon)}$ . Then  $\mathbf{x}^{(\epsilon)}(t+1)$  is obtained from  $\mathbf{x}^{(\epsilon)}(t)$  by applying symbolically a projected gradient descent step using  $\nabla \Phi(\mathbf{x}^{(\epsilon)}(t))$ . It is easy to see that a symbolic gradient descent step can be performed efficiently on the truncated probability simplex [77]. After executing  $T$  iterations, we return as output  $\mathbf{x}^{(0)}(T)$ , which is equal to  $\lim_{\epsilon \rightarrow 0^+} \mathbf{x}^{(\epsilon)}(T)$  by continuity.

**Proposition 6.9.** *For any  $T \in \mathbb{N}$ , there is a  $2 \times 2$  game in which symbolic gradient descent after  $T$  iterations converges to a strategy profile that is not a perfect equilibrium.*

*Proof.* We analyze the game given in Figure 6 when we initialize at  $\mathbf{x}_1^{(\epsilon)}(0) = (1 - \epsilon, \epsilon)$  and  $\mathbf{x}_2^{(\epsilon)}(0) = (1 - \epsilon, \epsilon)$ . We will prove the invariance  $\mathbf{x}_1^{(\epsilon)}(t) = (1 - \epsilon\kappa(t), \epsilon\kappa(t))$  and  $\mathbf{x}_2^{(\epsilon)}(t) = (1 - \epsilon\kappa(t), \epsilon\kappa(t))$  for some  $\kappa(t)$ . Inductively, if it holds for  $t$ , we have

$$\mathbf{x}_1^{(\epsilon)}(t+1) = \mathbf{x}_2^{(\epsilon)}(t+1) = \Pi_{\Delta(2)} \left( \begin{pmatrix} 1 - \epsilon\kappa(t) \\ \epsilon\kappa(t) \end{pmatrix} + \begin{pmatrix} 0 \\ \epsilon\kappa(t) \end{pmatrix} \right) = \begin{pmatrix} 1 - \frac{3}{2}\epsilon\kappa(t) \\ \frac{3}{2}\epsilon\kappa(t) \end{pmatrix}$$

for all  $\epsilon > 0$  small enough, establishing the induction for  $\kappa(t) = (3/2)^t$ . It thus follows that, no matter the choice of  $T$ ,  $\mathbf{x}_1^{(0)} = (1, 0) = \mathbf{x}_2^{(0)}$ , which is not a perfect equilibrium.  $\square$

	C1	C2
R1	0, 0	0, 0
R2	0, 0	1, 1

Figure 6: A  $2 \times 2$  identical-interest game in normal form where symbolic gradient descent can fail to converge to a perfect equilibrium.

## 6.2 Doubly exponentially small $\epsilon$ is necessary

We have now seen that, for polymatrix games, the almost-implies-near framework implies that computing perfect equilibria lies in the complexity class CLS. One may hope that this applies more generally to potential games, as potential games have more structure than generic normal-form games that may hypothetically allow a stronger almost-to-near result, even without the polymatrix assumption. Here, we show otherwise: we will explicitly exhibit normal-form games  $\Gamma$  in which  $\epsilon$  is required to be *doubly-exponentially small* (in the representation size of the game) before every (even exact) Nash equilibrium of  $\Gamma^{(\epsilon)}$  is close to a (even Nash) equilibrium of  $\Gamma$ .



**Theorem 6.10.** *For every positive integer  $n$  there exists a normal-form potential game  $\Gamma_n$  with  $O(n)$  players and two actions per player, or a polytope potential game with three players each with strategy set  $[0, 1]^{O(n)}$ , such that, for all  $\epsilon \in [1/2^{2^n}, 1/2]$ , the perturbed game  $\Gamma_n^{(\epsilon)}$  admits a Nash equilibrium that is distance  $1/2$  away in  $\ell_\infty$ -norm from any Nash equilibrium of  $\Gamma_n$ .*

*Proof.* We will construct a normal-form game  $\Gamma_n$  with  $4n + 1$  players, as follows. The first  $4n$  players are split into four groups of  $n$  players each, and the mixed strategies of the four groups will be denoted  $\mathbf{x}, \mathbf{x}', \mathbf{c}, \mathbf{d} \in [0, 1]^n$ , where for example  $c_i \in [0, 1]$  is the probability that the  $i$ th player of group  $\mathbf{c}$  plays its first action. The last player's strategy will be denoted  $t \in [0, 1]$ . The potential function is given by

$$\Phi(\mathbf{x}, \mathbf{x}', \mathbf{c}, \mathbf{d}, t) = \sum_{i=1}^n \left[ (t - c_i)(x_i - x_{i-1}x'_{i-1}) + \left(d_i - \frac{1}{2}\right)(x_i - x'_i) \right] - 2x_n - 2x'_n - 2n \cdot t \quad (4)$$

where for simplicity of notation we set  $x_0 = x'_0 := 1/2$ .

The construction of the three-player game is identical: the three players have strategies  $\mathbf{x}_i \in [0, 1]^n$ ,  $\mathbf{x}'_i \in [0, 1]^n$ , and  $(\mathbf{c}, \mathbf{d}, t) \in [0, 1]^{2n+1}$ , respectively.

For intuition,  $\Phi$  is constructed to achieve strategies at equilibrium that involve repeated squaring, as follows. If  $\mathbf{d} = \mathbf{1}/2$  and  $\mathbf{c} = t\mathbf{1}$ , then we must have  $x_i = x_{i-1}x'_{i-1}$  and  $x_i = x'_i$ , which implies  $x_i = 1/2^{2^i}$ . The sole purpose of the player  $t$  is to have a value that is guaranteed to be  $\epsilon$ , since by construction  $t$  always has negative gradient.

We claim first that, for  $\epsilon \in [1/2^{2^n}, 1/2]$ , the perturbed game  $\Gamma_n^{(\epsilon)}$  has the following equilibrium:

$$t = \epsilon, \quad \mathbf{d} = \frac{\mathbf{1}}{2}, \quad \mathbf{c} = \epsilon\mathbf{1}, \quad x_i = x'_i = \max \left\{ \epsilon, \frac{1}{2^{2^i}} \right\} = \max\{\epsilon, x_{i-1}^2\}.$$

We check this by checking the gradients for all players:

- The gradients for  $x_n$ ,  $x'_n$ , and  $t$  are always negative, and they are already set to the smallest possible value  $\epsilon$ .
- The gradient for  $c_i$  is nonpositive, because  $x_i \geq x_{i-1}^2$  holds for all  $i$ , and  $c_i$  is set to the smallest possible value,  $\epsilon$ .
- For  $i < n$ , the gradients for  $x_i$ ,  $x'_i$ , and  $d_i$  are all 0, because  $c_i = t$ ,  $d_i = 1/2$ , and  $x_i = x'_i$  for all  $i$ .

Thus, this is an equilibrium. It remains to show that this equilibrium is not close to any Nash equilibrium of  $\Gamma_n$ . Indeed, we will show that no equilibrium of  $\Gamma_n$  can satisfy  $0 < d_i < 1$  and  $c_i < 1$  for all  $i$ . Since  $0 < d_i < 1$ , we must have  $x_i = x'_i$  for all  $i$ . Since the gradients for  $x_n$ ,  $x'_n$ , and  $t$  are always negative, we must have  $x_n = x'_n = t = 0$ . But then there must be some  $i$  for which  $x_i < x_{i-1}^2$ . For that value of  $i$ , the gradient for  $c_i$  will be positive, so we must have  $c_i = 1$ . Thus, every Nash equilibrium of  $\Gamma_n$  must have  $c_i = 1$  or  $d_i = \{0, 1\}$  for some  $i$ , and therefore be at least  $\ell_\infty$  distance  $1/2$  from the previously established equilibrium of  $\Gamma_n^{(\epsilon)}$ .  $\square$

**Theorem 6.11.** *For every positive integer  $n$  there exists a normal-form potential game  $\Gamma_n$  with 3 players and  $O(n)$  actions per player, such that, for all  $\epsilon \in [1/2^{2^n}, 1/4n]$ , the perturbed game  $\Gamma_n^{(\epsilon)}$  admits a Nash equilibrium that is distance  $\Omega(1/n)$  away in  $\ell_\infty$ -norm from any Nash equilibrium of  $\Gamma_n$ .*

*Proof.* It suffices to only slightly modify the three-player counterexample in the previous proof; for the sake of completeness, we write out the full argument. We add one action for each of the three players, which will be denoted  $x_{n+1}, x'_{n+1}$ , and  $s$ , respectively. So, the strategy spaces of the three players are  $\mathbf{x} \in \Delta(n+1)$ ,  $\mathbf{x}' \in \Delta(n+1)$ , and  $(\mathbf{c}, \mathbf{d}, t, s) \in \Delta(2n+2)$ , respectively. The extra actions  $x_{n+1}, x'_{n+1}$ , and  $s$  serve only to ensure the existence of an extra degree of freedom in the strategy spaces, and will be essentially ignored for the remainder of the argument. The potential function is set to

$$\Phi(\mathbf{x}, \mathbf{x}', \mathbf{c}, \mathbf{d}, t, s) = \sum_{i=1}^n \left[ (t - c_i)(x_i - x_{i-1}x'_{i-1}) + \left( d_i - \frac{1}{4n} \right) (x_i - x'_i) \right] - 2x_n - 2x'_n - 2n \cdot t$$

The only change from (4) is that the  $d_i - 1/2$  has become a  $d_i - 1/2n$ , which is only necessary to ensure that P3's strategy is actually a valid strategy in the remainder of the proof. No attempt is made to optimize constant factors.

We claim first that, for  $\epsilon \in [1/2^{2^n}, 1/4n]$ , the perturbed game  $\Gamma_n^{(\epsilon)}$  has the following equilibrium:

$$t = \epsilon, \quad \mathbf{d} = \frac{\mathbf{1}}{4n}, \quad \mathbf{c} = \epsilon \mathbf{1}, \quad x_i = x'_i = \max \left\{ \epsilon, \frac{1}{2^{2^i}} \right\} = \max \{ \epsilon, x_{i-1}^2 \}$$

(To see that this is a valid strategy profile, notice that  $\sum_{i=1}^n \max \{ \epsilon, x_{i-1}^2 \} \leq \epsilon n + 1/2 \leq 1$ , so it suffices for  $\epsilon \leq 1/2n$ .) We check this by checking the gradients for all players:

- The gradients for  $x_n, x'_n$ , and  $t$  are always negative, and they are already set to the smallest possible value  $\epsilon$ .
- The gradient for  $c_i$  is nonpositive, because  $x_i \geq x_{i-1}^2$  holds for all  $i$ , and  $c_i$  is set to the smallest possible value,  $\epsilon$ .
- For  $i < n$ , the gradients for  $x_i, x'_i$ , and  $d_i$  are all 0, because  $c_i = t$ ,  $d_i = 1/2$ , and  $x_i = x'_i$  for all  $i$ .

Thus, this is an equilibrium. It remains to show that this equilibrium is not close to any Nash equilibrium of  $\Gamma_n$ . Indeed, we will show that no equilibrium of  $\Gamma_n$  can satisfy  $1/5n \leq d_i \leq 1/3n$  and  $c_i \leq 1/3n$  for all  $i$ . Suppose there were such an equilibrium. Since the gradients for  $x_n, x'_n$ , and  $t$  are always negative, we must have  $x_n = x'_n = t = 0$ . Thus, we have  $1/3 \leq s \leq 4/5$ , so the gradient for  $d_i$  must be zero and the gradient for  $c_i$  must be nonpositive for all  $i$  (else P3 profitably deviates by moving mass between  $s$  and the  $d_i$  or  $c_i$  whose gradient is nonzero.) But then we must have  $x_i = x'_i$  for all  $i$ . But then there must be some  $i$  for which  $x_i < x_{i-1}^2$ . For that value of  $i$ , the gradient for  $c_i$  will be positive which is a contradiction. Thus, every Nash equilibrium of  $\Gamma_n$  must have  $c_i = 1$  or  $d_i = \{0, 1\}$  for some  $i$ , and therefore be at least  $\ell_\infty$  distance  $1/2$  from the previously established equilibrium of  $\Gamma_n^{(\epsilon)}$ .  $\square$

## 7 Exponential path lengths for pure perfect equilibria

In this section, we investigate how many steps  $\epsilon$ -symbolic better-response dynamics might take in comparison to better-response dynamics. Of course, solution points to  $\epsilon$ -symbolic better-response dynamics are in particular solution points to better-response dynamics, that is, pure perfect equilibria are pure Nash equilibria. Since pure Nash equilibria are PLS-hard to compute in concise potential games (see, *e.g.*, Fabrikant et al. [26] for congestion games), pure perfect equilibria will inherit that complexity hardness. Therefore, one might intuitively hope to argue—and mistakenly

so, as we will see—that  $\epsilon$ -symbolic better-response dynamics should take at least as long to converge. We will show instead that *either* dynamics can take exponentially longer than the other. We remark that these results are *unconditional*, *i.e.*, true independent of whether  $\text{PLS} = \text{P}$  or not. We will first present the results of this section, and then discuss further background and the proofs.

For the specificity of our results, we recall our assumption that in both dynamics, players update their strategy in a round-robin fashion. On the positive side, we can then show that  $\epsilon$ -symbolic better-response dynamics can be exponentially faster than better-response dynamics in terms of improvement steps needed to reach an equilibrium.

**Theorem 7.1.** *There are families of concise potential games in which*

1.  *$\epsilon$ -symbolic better-response dynamics from any starting action profile takes at most  $n$  improvement steps to reach a pure perfect equilibrium, whereas*
2. *there exists starting action profiles from which better-response dynamics takes exponentially many improvement steps to reach a pure Nash equilibrium.*

*This already holds for games in which all players have exactly two actions.*

From the fact our proof constructs a family of 2-action games, we also obtain that (1) [Theorem 7.1](#) holds true independent of the pivoting rule a player might deploy for choosing a better response, and that (2) the results extend to proper equilibria as well (formally stated in [Corollary 7.5](#)).

We prove [Theorem 7.1](#) by reducing from the local search problem MAXCUT/FLIP. It is a PLS-complete problem that is known to possibly admit exponentially long local search paths. Interestingly enough, we will *also* use MAXCUT/FLIP—and a so-called *tight* PLS-reduction—to prove the contrary observation.

**Theorem 7.2.** *There is a tight PLS-reduction from MAXCUT/FLIP to finding a pure perfect equilibrium of concise potential games. This already holds for games in which all players have exactly two actions.*

**Corollary 7.3.** *There are families of concise potential games in which*

1. *better-response dynamics from any starting action profile takes at most  $n$  improvement steps to reach a pure Nash equilibrium, whereas*
2. *there exists starting action profiles from which  $\epsilon$ -symbolic better-response dynamics takes exponentially many improvement steps to reach a pure perfect equilibrium.*

**Corollary 7.4.** *It is PSPACE-complete to decide in concise potential games whether  $\epsilon$ -symbolic better-response dynamics reaches a pure perfect equilibrium from a given starting action profile within  $k \in \mathbb{N}$  number of improvement steps (where  $k$  is given in binary).*

Again, since our proof constructs a family of 2-action games, the above results hold true independent of the pivoting rule a player might deploy for choosing an  $\epsilon$ -symbolic better response. Also, when players only have two actions available, the notions of pure perfect equilibrium and pure proper equilibrium coincide. Therefore, we conclude with the same statements for the proper equilibrium refinement.

**Corollary 7.5.** *All results in this section also hold for pure proper equilibria instead of pure perfect equilibria. This includes [Theorem 7.1](#), [Theorem 7.2](#), [Corollary 7.3](#), and [Corollary 7.4](#).*

**Background on MAXCUT and PLS-tight reductions** To show these results, we use the local search problem MAXCUT/FLIP. It is a PLS-complete problem known to have exponentially long local search paths in the worst case. Let  $G = (V, E)$  be an undirected graph,  $w : E \rightarrow \mathbb{N}$  positive edge weights, and  $V = B \sqcup C$  a vertex partition into two sets. Then, the cut of  $B \sqcup C$  is defined as all the edges in between  $B$  and  $C$ :

$$E \cap (B, C) := \{\{u, v\} = e \in E : u \in B \wedge v \in C \text{ or } u \in C \wedge v \in B\}.$$

Its weight is defined as  $w(B, C) := \sum_{e \in E \cap (B, C)} w(e)$ . The FLIP neighbourhood of partition  $B \sqcup C$  is the set of partitions  $(B', C')$  that can be obtained from  $(B, C)$  by just moving one vertex from one part to the other:

$$\text{FLIP}(B, C) := \left\{ (B \cup \{c\}) \sqcup (C \setminus \{c\}) \right\}_{c \in C} \cup \left\{ (B \setminus \{b\}) \sqcup (C \cup \{b\}) \right\}_{b \in B}.$$

**Definition 7.6.** An instance of the search problem MAXCUT/FLIP consists of an undirected graph  $G = (V, E)$  with edge weights  $w : E \rightarrow \mathbb{N}$ . A (locally optimal) solution consists of a partition  $V = B \sqcup C$  that has maximal cut weight among its FLIP neighbourhood.

We are interested in its computational complexity in terms of  $|V|$ ,  $|E|$ , and a binary encoding of all weight values. The standard local search method with Flip iteratively checks for the current cut  $(B, C)$  whether there is a node to flip that would improve upon the current cut weight. A pivoting rule decides which node to flip for improvement if there are multiple ones available. The next lemma shows that local search fully captures the complexity of this problem.

**Lemma 7.7** (Schäffer and Yannakakis [67], Yannakakis [78]). *MAXCUT/FLIP is PLS-complete. Moreover, there are instances and initializations (starting partitions) for which—irrespective of the choice of pivoting rule—local search takes exponentially many iterations to reach a solution. Deciding whether one can reach a solution from a given initialization within  $k \in \mathbb{N}$  number of iterations ( $k$  given in binary) is PSPACE-complete.*

The latter two results in [Lemma 7.7](#) can be obtained with a so-called *tight* PLS-reduction. Since it forms a central pillar to our upcoming results, we shall quickly restate its definitions here (*cf.* [78](#), Definitions 2-4).

**Definition 7.8.** A PLS-reduction from a local search problem  $\Pi$  to a local search problem  $\Pi'$  consists of two polytime computable functions  $h$  and  $g$  such that

1.  $h$  maps instances  $x$  of  $\Pi$  to instances  $h(x)$  of  $\Pi'$ ,
2.  $g$  maps a pair consisting of an instance  $x$  of  $\Pi$  and a feasible point of  $h(x)$  to a feasible point of  $x$ , and
3. if  $s$  is a (locally optimal) solution to  $h(x)$ , then  $g(x, s)$  is a solution to  $x$ .

**Definition 7.9.** Let  $\Pi$  be a local search problem, and  $x$  be an instance of  $\Pi$ . Its transition graph  $\text{TG}_\Pi(x)$  is then defined as a directed graph with one node for each feasible point to  $x$ , and an arc  $s \rightarrow t$  whenever  $t$  is in the local neighborhood of  $s$  and the cost of  $t$  is strictly better than the cost of  $s$ .

For example, for an instance  $(G, w)$  of the local search problem MAXCUT/FLIP, the partitions are the feasible points, and there is an arc in the transition graph from a partition  $B \sqcup C$  to a partition  $B' \sqcup C'$ , if the latter has a strictly higher cut weight and was obtained from the former via a vertex flip.

**Definition 7.10.** Let  $(h, g)$  be a PLS-reduction from local search problem  $\Pi$  to local search problem  $\Pi'$ . We say it is *tight* if for any instance  $x$  of  $\Pi$  we can choose a subset  $R$  of feasible points of corresponding instance  $h(x)$  of  $\Pi'$  such that

1.  $R$  contains all (locally optimal) solutions to  $h(x)$ ,
2. for any feasible point  $p$  of  $x$  we can construct in polytime a preimage  $q$  of  $p$  under  $g(x, \cdot)$ , *i.e.*,  $g(x, q) = p$ , and
3. the following holds: If the transition graph  $\text{TG}_{\Pi'}(h(x))$  of  $h(x)$  contains a path from  $q \in R$  to  $q' \in R$  such that all of its internal path nodes are outside of the path, then either  $g(x, q) = g(x, q')$  already or  $\text{TG}_{\Pi}(x)$  contains an arc from  $g(x, q)$  to  $g(x, q')$ .

A tight PLS-reduction from  $\Pi$  to  $\Pi'$  implies that the transition graphs in the corresponding  $\Pi'$  instances are at least as long as the transition graphs in their respective  $\Pi$  instances. This enables lower bound proofs on the local search method as given in [Lemma 7.7](#) assuming they are known to hold for the problem  $\Pi$  one reduced from [\[78, Lemma 11\]](#). With that, we have the necessary tools to show the results of this section.

## 7.1 Proof of [Theorem 7.1](#)

This subsection is devoted to proving [Theorem 7.1](#). We first review a result known in the literature, as it forms the basis of our proof.

**Lemma 7.11.** *There is a tight PLS-reduction from MAXCUT/FLIP to finding a Nash equilibrium in identical-interest games. In particular, there exist identical-interest games and action profiles  $\mathbf{a}$  in these games from which all better-response paths are of exponential length.*

*Proof.* Let  $(V, E, w)$  be a MAXCUT/FLIP instance, and create an identical-interest game of it as follows: Introduce a player  $v \in V$  for each node. Each player has two actions  $\{b, c\}$ , which represent entering the subset  $B$  or  $C$  in the partition. Hence, a pure strategy profile  $\mathbf{a}$  corresponds to a partition  $(B_{\mathbf{a}}, C_{\mathbf{a}})$  of the vertex set. Define every players utility from that profile to be the weight of the partition's cut. Then, the Nash equilibria of that game corresponds to locally optimal partitions of the graph with respect to the FLIP neighborhood structure (“Can any vertex improve by switching to the other side of the partition?”). The tightness of this reduction follows from the fact that the transition graphs are exactly the same.  $\square$

To account for equilibrium refinements, we start from the identical-interest game construction above, and add two players  $w, w'$ , each with two actions  $d$  (default) and  $e$  (escape). If at least one of those players takes action  $d$ , we compute the payoff as we did before, based on the cut weight of the action profile of the other players. If  $w$  and  $w'$  play  $e$ , on the other hand, we give everyone a very high payoff  $M + \psi(\mathbf{a})$ —so high that they would not be able to achieve a comparable payoff through the weight of a cut. (For example, we can choose  $M = |V|^2 \cdot \max_e w(e) + 1$ .) We define the additional payoff  $\psi(\mathbf{a})$  to be  $|\{v \in V : \mathbf{a}_v = b\}|$ . Then, pure Nash equilibria of this game have one of two forms: either  $(\mathbf{b}, e, e)$ , where  $\mathbf{b}$  is the action profile where the original  $|V|$  players are playing  $b$ ; or  $(\mathbf{a}, d, d)$ , where  $\mathbf{a}$  represents a local max-cut of the graph. Only  $(\mathbf{b}, e, e)$  is also pure perfect equilibria, since for the latter type,  $w$  observes  $e$  to be better than  $d$  whenever  $w'$  plays  $e$  with some nonzero probability.

Let  $w$  and  $w'$  be first to update their strategy. Then all starting action profiles have an  $\epsilon$ -symbolic better-response paths of length at most  $|V|$  to a pure perfect equilibrium:  $w$  updates to  $e$  due to the first-order incentives from  $w'$  switching as well,  $w'$  follows suit due to zeroth order

incentives, and the remaining players update to  $b$ . Now consider better-response dynamics for the starting profile  $(\mathbf{a}, \mathbf{d}, \mathbf{d})$ . As long as both  $w$  and  $w'$  play  $\mathbf{d}$ , switching to action  $\mathbf{e}$  will never become a better response for either of those players. Hence, any better-response path starting from  $(\mathbf{a}, \mathbf{d}, \mathbf{d})$  will correspond to better-response paths in the original identical-interest game, appended by  $w$  and  $w'$  playing  $\mathbf{d}$ . All of these paths will be of exponential length if the identical-interest games and starting profile  $\mathbf{a}$  are chosen according to [Lemma 7.11](#).

## 7.2 Proof of [Theorem 7.2](#) and its subsequent corollaries

This subsection is mainly devoted to proving [Theorem 7.2](#). We reduce from MAXCUT/FLIP again, so let  $(V, E, w)$  be a MAXCUT/FLIP instance. We create a similar identical-interest game as in the proof of [Theorem 7.1](#), except with a different twist. For each node  $v \in V$ , introduce *three* players  $v^{(1)}, v^{(2)}$ , and  $v^{(3)}$ , each of them with the usual two actions  $\{b, c\}$ . Informally, we want the triplet to decide via majority vote whether  $v$  should enter the subset  $B$  or  $C$  in the partition, but if  $k$  triplets do not unanimously agree and  $k \geq 2$ , we will punish them slightly and proportionally to  $k$  (and everyone else equally so, since the game shall be of identical interest). For an action profile  $\mathbf{a} = (\mathbf{a}_v^{(1)}, \mathbf{a}_v^{(2)}, \mathbf{a}_v^{(3)})_{v \in V} \in A$  in the game, define its associated cut  $(B_{\mathbf{a}}, C_{\mathbf{a}})$  as follows: put vertex  $v \in B$  if and only if at least two players of the triplet  $\{v^{(1)}, v^{(2)}, v^{(3)}\}$  play  $b$  (note that otherwise, at least two must have played  $c$ ). Next, define  $\psi : A \rightarrow \mathbb{N}$  as the number non-unanimous triplets, that is,  $\psi(\mathbf{a}) = |\{v \in V : \exists i, j \in [3] \text{ with } \mathbf{a}_v^{(i)} \neq \mathbf{a}_v^{(j)}\}|$ . Lastly, let  $\lambda$  be a sufficiently large yet polynomially sized penalty multiplier; for example,  $\lambda = 6|V|$  suffices. We then define the utility function as

$$u(\mathbf{a}) = \begin{cases} w(B_{\mathbf{a}}, C_{\mathbf{a}}) & \text{if } \psi(\mathbf{a}) \leq 1, \\ w(B_{\mathbf{a}}, C_{\mathbf{a}}) - \psi(\mathbf{a})/\lambda & \text{otherwise.} \end{cases}$$

We denote this corresponding game instance with  $\Gamma$ . For purposes of analyzing its game dynamics, we assume that the player order has triplets appearing back to back. With this, we provide some intuition for the Nash equilibrium structure in the next lemma.

**Lemma 7.12.** *An action profile  $\mathbf{a}$  is a pure Nash equilibrium of this corresponding game  $\Gamma$  if and only if it satisfies  $\psi(\mathbf{a}) \leq 1$  and the following property: if there is a node  $v$  with a non-unanimous triplet, then its majority is playing the (weakly) better partition subset for  $v$  among  $B$  and  $C$ . Furthermore, better-response dynamics from any starting action profile in  $\Gamma$  takes at most as many improvement steps to reach a pure Nash equilibrium as there are players in  $\Gamma$ .*

*Proof.* Indeed, if the two properties are satisfied, then (a) players of unanimous triplets have no incentives to unilaterally deviate (according to better-response improvements) because we are using majority voting, and (b) the same is true for non-unanimous triplets if it already voted for the (weakly) better subset. For the other direction, let us discuss the situations not covered by the two situations. If there are at least two non-unanimous triplets, then the minority player of any non-unanimous triplet has incentives to join the other two player's decision in order to push down the punishment term. If there is only one non-unanimous triplet, and it is voting for a strictly worse subset, then any of the majority players have incentives to switch to the minority player's decision to turn the majority vote. This concludes the other direction of the Nash equilibrium characterization. Last but not least, the statement about better-response dynamics follows from the fact that non-unanimous triplets will come to unanimous agreement first until there is only one non-unanimous triplet left, and that one will turn its majority at most once.  $\square$

Next, we can show that the construction of corresponding  $\Gamma$  gives rise to a tight PLS-reduction from MAXCUT/FLIP to finding a pure perfect equilibrium of concise potential games, that is, [Theorem 7.2](#). First, we show it gives rise to a PLS-reduction in the first place. As mentioned earlier, we associate any action profile  $\mathbf{a}$  to the partition  $(B_{\mathbf{a}}, C_{\mathbf{a}})$  that one obtains by considering the majority vote for each triplet. Then we have the following.

**Lemma 7.13.** *If  $\mathbf{a}^*$  is a pure perfect equilibrium of  $\Gamma$  then  $\psi(\mathbf{a}^*) = 0$ .*

*Proof.* It cannot be  $\psi(\mathbf{a}^*) \geq 2$  because a minority player of any non-unanimous triplet would see zero-order incentives to deviate to reach unanimity. *Zero-order* here means a strict positive utility gain from a unilateral deviation. It cannot be  $\psi(\mathbf{a}^*) = 1$  either, because for the non-unanimous triplet  $\{v^{(1)}, v^{(2)}, v^{(3)}\}$  in that situation either (a) a majority player sees zero-order incentives to deviate because  $v$  is on currently on the worse side of the partition or, otherwise, (b) the minority player sees first-order incentives to reach unanimity. *First-order* here means zero utility gain from a unilateral deviation, and strictly positive utility gain from one owns deviation under the off-chance that a single other player deviated in line with perfect equilibrium reasoning. The reason the minority player sees first-order incentives here is because the minority player (weakly) wants to avoid a majority player deviating, resulting in a non-beneficial swap for  $v$  (since we are not in case (a)), and because the minority player strictly wants to avoid the penalty term from when a player of the other triplets deviated and created another non-unanimity.  $\square$

**Lemma 7.14.** *If  $\mathbf{a}^*$  is a pure perfect equilibrium of  $\Gamma$  then  $(B_{\mathbf{a}^*}, C_{\mathbf{a}^*})$  is a locally optimal solution to the original MAXCUT/FLIP instance  $(V, E, w)$ .*

*Proof.* By [Lemma 7.13](#),  $\psi(\mathbf{a}^*) = 0$ . To show that  $(B_{\mathbf{a}^*}, C_{\mathbf{a}^*})$  is locally optimal, we show that no vertex  $v$  can be flipped and yield a cut  $(B', C')$  with an improvement weight, since that would imply that  $\mathbf{a}^*$  could have been a pure perfect equilibrium in the first place. So let us study the deviation incentives of any player of the associated triplet, say, player  $v^{(1)}$ . That player does not see any zero-order deviation from deviating since we are deploying majority voting. On the first-order level, its incentives to deviate come either from a triplet partner possibly deviating or from player of another triplet deviating. If a player of another triplet deviates, the partition—and, therefore, the cut weight—would stay unchanged, but a penalty term of  $\lambda$  would incur on the players. If it is a triplet partner instead that deviates together with  $v^{(1)}$ , the majority for  $v$  turns, leading to a vertex flip in the partition but no penalty term. The first-order incentives summarize to

$$\epsilon \cdot \left( 2 \cdot (w(B', C') - w(B_{\mathbf{a}}, C_{\mathbf{a}})) + 3(|V| - 1) \cdot \left(-\frac{2}{\lambda}\right) \right). \quad (5)$$

Note that the second part is negative and, by our choice of  $\lambda$ , strictly smaller in magnitude than 1. Hence, the first-order incentives will be positive if and only if  $w(B', C') > w(B_{\mathbf{a}}, C_{\mathbf{a}})$ , i.e., if and only if the FLIP neighbor  $(B', C')$  is a local improvement over  $(B_{\mathbf{a}^*}, C_{\mathbf{a}^*})$  in the MaxCut instance. This concludes the proof that if  $\mathbf{a}^*$  is a pure perfect equilibrium, then  $(B_{\mathbf{a}^*}, C_{\mathbf{a}^*})$  must be locally optimal.  $\square$

Next, we can show that this PLS-reduction is tight. Define the subset  $R$  in [Definition 7.10](#) as all unanimous strategy profiles in  $\Gamma$ , that is,  $R := \{\mathbf{a} : \psi(\mathbf{a}) = 0\}$ . By [Lemma 7.13](#), this set contains all locally optimal solutions of  $\Gamma$ . Next, we show the third condition of a tight PLS-reduction.

**Lemma 7.15.** *If there is an  $\epsilon$ -symbolic better-response path  $P$  from a strategy profile  $\mathbf{a} \in R$  to another  $\mathbf{a}' \in R$  such that all profiles along that path are not in  $R$ , then there exists a vertex  $v$  such that one can obtain  $(B'_{\mathbf{a}}, C'_{\mathbf{a}})$  from  $(B_{\mathbf{a}}, C_{\mathbf{a}})$  by flipping  $v$ .*



*Proof.* Let  $v$  be the vertex of the player that deviated at the first edge of  $P$ . Refer to the player as  $v^{(i)}$ . By the proof of [Lemma 7.14](#), we obtain that the player deviation must have incurred because the cut from flipping  $v$  is a strict cut weight improvement over  $(B_{\mathbf{a}}, C_{\mathbf{a}})$ . So let us consider the strategy profile  $\mathbf{a}''$  in which  $v^{(i)}$  has changed their strategy. The associated partition remains unchanged, and we have  $\psi(\mathbf{a}'') = 1$  now. At this profile, the players of triplets not associated to  $v$  will observe negative zero-order deviation incentives because deviating yields no partition change and yields a penalty term by introducing a second non-unanimous triplet. The other players of the triplet associated to  $v$ , on the other hand, observe a positive zero-order deviation incentive to join  $v^{(i)}$  in order to turn the majority vote. Hence, the second edge of  $P$  must be one of these two players switching their strategy. At that new profile, still no player of another triplet wants to deviate by analogous reasoning. The other two players of the triplet that have switched already, do not want to switch back since that yields a worse cut weight. Only the third triplet player sees (first-order) deviation incentives to switch, for the reason to avoid cut weight losses or penalty terms in the off-chance where another player deviates. Hence, the third edge of  $P$  is the third player of the triplet joining the other two to obtain unanimity for  $v$  again, landing at a strategy profile in  $R$ . By assumption on the path  $P$ , we obtain that that strategy profile must be  $\mathbf{a}'$ . And indeed,  $(B'_{\mathbf{a}}, C'_{\mathbf{a}})$  is obtained from  $(B_{\mathbf{a}}, C_{\mathbf{a}})$  solely by flipping  $v$ .  $\square$

This concludes the proof on [Theorem 7.2](#). Informally, [Lemma 7.15](#) shows that  $\epsilon$ -symbolic better-response dynamics from an action profile  $\mathbf{a}$  with unanimous triplets corresponds bijectively to, and is three times as long as, local search in the MAXCUT/FLIP instance starting from the partition associated to  $\mathbf{a}$ . This implies that finding a pure perfect equilibrium inherits from MAXCUT/FLIP that improvement paths ( $\epsilon$ -symbolic better-response dynamics) can have exponential length and that it is PSPACE-complete to decide whether a solution can be reached within a fixed number of steps, *i.e.*, [Corollary 7.3](#) and [Corollary 7.4](#).

Finally, we highlight that we exclusively worked with two-action games in this reduction as well as the previous one for [Theorem 7.1](#), yielding [Corollary 7.5](#).

## 8 Positive results for structured games

In the previous section, we showed that computing pure perfect equilibria using better-response dynamics can take exponentially many steps. Therefore, one might naturally wonder whether there are structured classes of games, in which pure perfect equilibria can be computed in polynomial time. In this section, we show such positive result for two well-studied classes of games, namely symmetric matroid congestion games and symmetric network congestion games.

### 8.1 Symmetric matroid congestion games

Matroid congestion games are a broad class of congestion games, where the strategy space of each player consists of the bases of a matroid over the set of resources. Before we state the formal definition of such games, we recall some basic background about matroids.

**Definition 8.1** (Matroid). A tuple  $M = (\mathcal{R}, \mathcal{I})$  is a matroid, if  $\mathcal{R}$  is a finite set of resources and  $\mathcal{I}$  is a nonempty family of subsets of  $\mathcal{R}$  with the property that

1. if  $I \in \mathcal{I}$  and  $J \subseteq I$  then  $J \in \mathcal{I}$ , and
2. if  $I, J \in \mathcal{I}$  and  $|J| < |I|$  then there exists  $i \in I \setminus J$  with  $J \cup \{i\} \in \mathcal{I}$ .

Given a matroid  $M = (\mathcal{R}, \mathcal{I})$ , we say that a subset of resources  $I \subseteq \mathcal{R}$  is independent if  $I \in \mathcal{I}$ . Otherwise we call it dependent. A maximal independent subset of a matroid  $M$  is called a basis of  $M$  and its size is the rank of  $M$ , denoted as  $\text{rk}(M)$ .

**Definition 8.2** (Matroid congestion game). A matroid congestion game is a congestion game, where the strategy space of any player  $i$  is the set of bases  $B_i$  of a matroid  $M_i = (\mathcal{R}_i \subseteq \mathcal{R}, \mathcal{I}_i)$ .

For a common set of resources  $\mathcal{R}$ , a matroid  $M = (\mathcal{R}, \mathcal{I})$ , and a set of bases  $B$  of  $M$ , we denote a symmetric matroid congestion game as  $\Gamma = (\mathcal{R}, B, (d_r)_{r \in \mathcal{R}})$ , where the strategy space of each player is  $B$ .

Our approach to show fast convergence involves defining an  $\epsilon$ -perturbed game and showing that best-response dynamics will converge to a Nash equilibrium of that game in polynomial time. To show this, we will allow players to randomize over their strategy space. Specifically, we let  $\mathcal{X} = \Delta(B)$  denote the set of mixed strategies over the common strategy space,  $B$ . For a mixed strategy profile  $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathcal{X}^n$  (and by slight abuse of notation) we define the expected cost  $c_i$  of player  $i \in [n]$  as

$$\begin{aligned} c_i(\mathbf{x}_1, \dots, \mathbf{x}_n) &= \sum_{S_i \in B} \mathbf{x}_i(S_i) \cdot \mathbb{E}_{S_{-i}} \left[ \sum_{r \in S_i} d_r(n_r(S_i, S_{-i})) \right] \\ &= \sum_{S_i \in B} \mathbf{x}_i(S_i) \cdot \sum_{r \in S_i} \mathbb{E}_{S_{-i}} [d_r(1 + |\{j \neq i : r \in S_j\}|)] \\ &= \sum_{S_i \in B} \mathbf{x}_i(S_i) \cdot \sum_{r \in S_i} \sum_{k=0}^{n-1} d_r(1+k) \cdot \Pr_{S_{-i}} \left[ \sum_{j \neq i} \mathbb{1}\{r \in S_j\} = k \right]. \end{aligned} \quad (6)$$

More generally, we let  $P_r(k)$  denote the probability that  $\sum_{i \in [n]} \mathbb{1}\{i \text{ uses } r\} = k$ . It is easy to show that in general  $P_r(k)$  can be computed in polynomial time [59]. Specifically, given a mixed strategy profile,  $\mathbf{x}$ , we can first compute the probability,  $w_r(i)$ , that player  $i$  will use resource  $r$ . We have  $w_r(i) = \sum_{S_i \in B} \mathbf{x}_i(S_i) \mathbb{1}\{r \in S_i\}$ . Given this, we can compute  $P_r(k)$  via simple dynamic programming. We first define  $s_r(\ell, k)$  to be the probability that the total number of players, out of the first  $\ell$  players, using this resource is  $k$ . Knowing that  $s_r(0, 0) = 1$  and  $s_r(0, 1) = 0$ , we can define the recurrence relation as

$$s_r(\ell, k) = s_r(\ell - 1, k)(1 - w_r(\ell)) + s_r(\ell - 1, k - 1)w_r(\ell). \quad (7)$$

After polynomially many iterations we get the value of  $s_r(n, k)$ , which is equal to  $P_r(k)$ .

We are now ready to define our perturbed game. Given a symmetric, matroid congestion game,  $\Gamma = (\mathcal{R}, B, (d_r)_{r \in \mathcal{R}})$ , we define the  $\epsilon$ -perturbed game,  $\Gamma^{(\epsilon)} = (\mathcal{R}, \mathcal{X}^{(\epsilon)}, (D_r)_{r \in \mathcal{R}})$ , such that each player has strategy space  $\mathcal{X}^{(\epsilon)} = \{\mathbf{x} \in \mathcal{X} \text{ such that } (\mathbf{x}(S) = \epsilon \vee \mathbf{x}(S) = 1 - (|B| - 1)\epsilon) \ \forall S \in B\}$  i.e., each player is restricted to playing  $\epsilon$ -pure strategies. With  $D_r : (\mathcal{X}^{(\epsilon)})^n \rightarrow \mathbb{R}$ , we denote the expected delay of a resource  $r$ , defined as  $D_r(\mathbf{x}) = \mathbb{E}_S [d_r(|\{j \in [n] : r \in S_j\}|)] = \sum_{k=1}^n d_r(k)P_r(k)$ .

*Remark 8.3.* We note that it is *a priori* unclear that the perturbed game  $\Gamma^{(\epsilon)}$  is a matroid congestion game, which is a property that we will need to prove our main result. We circumvent this issue by defining  $\tilde{\Gamma}^{(\epsilon)}$ , a game that is equivalent to  $\Gamma^{(\epsilon)}$  (when players are only allowed to play  $\epsilon$ -pure strategies) and is also a matroid congestion game. Specifically, we let  $\tilde{\Gamma}^{(\epsilon)} = (\mathcal{R}, B, \tilde{d}_r^{(\epsilon)})$ , where  $\tilde{d}_r^{(\epsilon)} : \mathbb{N} \rightarrow \mathbb{N}[\epsilon]$ , defined as

$$\tilde{d}_r^{(\epsilon)}(k) = \sum_{i=1}^n \tilde{P}_{r,k}(i) d_r(i), \quad (8)$$

is the expected delay of resource  $r$ , when exactly  $k$  players are using  $r$  in their  $\epsilon$ -pure strategy.

To define  $\tilde{P}_{r,k}$ , we first introduce some additional notation. We let  $B_r = \sum_{S \in B} \mathbb{1}\{r \in S\}$  denote the number of occurrences of resource  $r$  in the set of bases  $B$ . If a player chooses to play a subset of resources that contains  $r$  with probability  $1 - |B|\epsilon$  then the probability with which this player is using resource  $r$  is

$$\begin{aligned} h_r^{(\epsilon)} &= 1 - (|B| - 1)\epsilon + (B_r - 1)\epsilon \\ &= 1 - (B_r - |B|)\epsilon. \end{aligned}$$

Otherwise, the player is playing resource  $r$  with probability  $f_r^{(\epsilon)} = B_r\epsilon$ . We define

$$\tilde{P}_{r,k}(i) = \sum_{j=\max\{0, i-(n-k)\}}^{\min\{i, k\}} \binom{k}{j} (h_r^{(\epsilon)})^j (1 - h_r^{(\epsilon)})^{k-j} \cdot \binom{n-k}{i-j} (f_r^{(\epsilon)})^{i-j} (1 - f_r^{(\epsilon)})^{n-k-(i-j)} \quad (9)$$

to be the probability that the sum of players playing resource  $r$  is  $i$ , when exactly  $k$  players are using  $r$  in their  $\epsilon$ -pure strategy. It is not hard to verify that  $\tilde{\Gamma}^{(\epsilon)}$  is indeed a symmetric matroid congestion game with potential  $\tilde{\Phi}^{(\epsilon)}(S) = \sum_{r \in \mathcal{R}} \sum_{k=1}^{n_r(S)} \tilde{d}_r^{(\epsilon)}(k)$ .

We are now ready to show that  $\epsilon$ -symbolic best-response dynamics in  $\tilde{\Gamma}^{(\epsilon)}$  converge to a Nash equilibrium in polynomial time.

**Theorem 8.4.** *Let  $\Gamma$  be a symmetric matroid congestion game. Then after at most  $n^2|\mathcal{R}|rk(M)$   $\epsilon$ -best response iterations in the perturbed game,  $\tilde{\Gamma}^{(\epsilon)}$ , a perfect equilibrium of  $\Gamma$  is reached. Moreover, each iteration can be implemented in time that is polynomial in the size of the input, assuming that we are given 1) a membership oracle (i.e., an oracle that given a subset  $I \in 2^{\mathcal{R}}$  decides whether  $I \in \mathcal{I}$ ) and 2) for each resource  $r \in \mathcal{R}$  the number  $B_r = \sum_{S \in B} \mathbb{1}\{r \in S\}$  of bases that contain  $r$ .*

*Proof.* We can assume without loss of generality that the best response of every player is a pure strategy in  $\tilde{\Gamma}^{(\epsilon)}$ , i.e., for a best-responding player  $i$  there exists an  $S_j \in B$  such that  $\mathbf{x}_j(S_j) = 1$  and  $\mathbf{x}_j(S_{j'}) = 0$  for all  $S_{j'} \in B \setminus S_j$ . By (9), we know that  $\tilde{P}_r(k)$  is a polynomial in  $\epsilon$  of degree at most  $n$ . Therefore, for concise games, given  $B_r$  for each  $r \in \mathcal{R}$ , the cost of each player for a given strategy profile can be determined in polynomial time. Moreover, it is known that computing best responses can be done efficiently using a greedy algorithm [22].

For the proof of convergence, we will first show that  $\tilde{\Phi}^{(\epsilon)}$  decreases (with respect to the ordering  $<_{\epsilon \rightarrow 0+}$ ) with every iteration. To see this, we can first rewrite the potential as  $\tilde{\Phi}^{(\epsilon)}(S) = \sum_{i=1}^n \sum_{r \in S_i} \tilde{d}_r^{(\epsilon)}(n_{r, \leq i}(S))$ , where  $n_{r, \leq i}(S)$  is the number of players  $j \leq i$  that use resource  $r$ .

Now for a given strategy profile  $S$  suppose that a player  $i \in [n]$  has a profitable deviation  $S'_i$ . Without loss of generality we can take this player to be player  $n$ . Therefore, we get that

$$\begin{aligned} \tilde{\Phi}^{(\epsilon)}(S') - \tilde{\Phi}^{(\epsilon)}(S) &= \sum_{r \in S'_n} \tilde{d}_r^{(\epsilon)}(n_{r, \leq i}(S')) - \sum_{r \in S_n} \tilde{d}_r^{(\epsilon)}(n_{r, \leq i}(S)) \\ &= \sum_{r \in S'_n} \tilde{d}_r^{(\epsilon)}(n_r(S')) - \sum_{r \in S_n} \tilde{d}_r^{(\epsilon)}(n_r(S)) \\ &= c_n(S') - c_n(S). \end{aligned}$$

Also, from Ackermann et al. [1], we know that in a matroid congestion game, the potential can only take  $n^2|\mathcal{R}|rk(M)$  many values, and therefore best-response dynamics in  $\tilde{\Gamma}^{(\epsilon)}$  converge to a Nash equilibrium,  $\mathbf{x}$ . Hence, the corresponding  $\epsilon$ -pure strategies,  $\mathbf{x}^{(\epsilon)}$  also form a Nash equilibrium in  $\Gamma^{(\epsilon)}$ . Lastly, we know that  $\{\mathbf{x}^{(\epsilon)}\}_{\epsilon \rightarrow 0+}$  defines a sequence of  $\epsilon$ -perfect equilibria of  $\Gamma$ , the limit point of which is a perfect equilibrium. This concludes our proof.  $\square$

## 8.2 Symmetric network congestion games

A network congestion game is a type of congestion game, defined on a graph, where the sets of resources available to each player correspond to paths in the graph. Concretely, we consider a graph  $G = (V, E)$ , two nodes  $a_i, b_i \in V$  for each player  $i \in [n]$ , and a delay function  $d_e : \mathbb{N} \rightarrow \mathbb{N}$  for each  $e \in E$ . A strategy,  $S_i$  of player  $i$  corresponds to a subset of edges that forms a valid path from  $a_i$  to  $b_i$ . We denote the set of such paths  $\mathcal{P}_{a_i, b_i}$ . As in general congestion games, the potential function associated with this class of games is  $\Phi(S) = \sum_{e \in E} \sum_{i=1}^{n_e(S)} d_e(i)$ .

In a symmetric network congestion game, every agent shares the same strategy space, which corresponds to all paths, between the same pair of vertices,  $a, b \in V$ .

As before, we will show the a perfect equilibrium can be found in polynomial time, by defining a perturbed game, and computing a Nash equilibrium of this game, this time by reducing the problem to a symbolic min-cost flow problem.

**Theorem 8.5.** *Let  $\Gamma$  be a symmetric network congestion game. Then there exists an algorithm that outputs a perfect equilibrium of  $\Gamma$  in time that is polynomial in the size of the input, assuming that we are given  $B_e = \sum_{S \in \mathcal{P}_{a,b}} \mathbb{1}\{e \in S\}$  for each edge  $e$ .*

*Proof.* First, given a symmetric network congestion game,  $\Gamma = (G = (V, E), \mathcal{P}_{a,b}, \{d_e\}_{e \in E})$ , where  $\mathcal{P}_{a,b}$  is the set of paths between vertices  $a$  and  $b$ , we define the  $\epsilon$ -perturbed game  $\Gamma^{(\epsilon)} = (G = (V, E), \mathcal{X}_{a,b}^{(\epsilon)}, \{d_e\}_{e \in E})$ , where  $\mathcal{X}_{a,b}^{(\epsilon)} = \{\mathbf{x} \in \mathcal{X} \text{ such that } \mathbf{x}(P) \geq \epsilon \ \forall P \in \mathcal{P}_{a,b}\}$ , i.e. each player is using each path with probability at least  $\epsilon$ . We can define an equivalent game to  $\Gamma^{(\epsilon)}$  by applying the perturbations directly to the delay functions. Specifically, we consider the game  $\tilde{\Gamma}^{(\epsilon)} = (G = (V, E), \mathcal{P}_{a,b}, \{\tilde{d}_e^{(\epsilon)}\}_{e \in E})$ , (where  $\tilde{d}_e^{(\epsilon)}$  are defined as in (8)), which is now a symmetric network congestion game.

The next step is to reduce the problem of finding an Nash equilibrium in  $\tilde{\Gamma}^{(\epsilon)}$  to a min-cost flow problem. The crux of this proof follows that of Fabrikant et al. [26], who show this reduction in the non-symbolic case. The idea is that given  $\tilde{\Gamma}^{(\epsilon)}$  with graph  $G = (V, E)$ , we construct a graph, where each edge  $e \in E$  is replaced with  $n$  parallel edges, each with capacity 1 and delays  $\tilde{d}_e^{(\epsilon)}(1), \dots, \tilde{d}_e^{(\epsilon)}(n)$ . We note that given  $B_e$  for each edge  $e$ , we can compute the total cost of each  $e$  in polynomial time. We can now write the symbolic min-cost flow problem as a symbolic linear program, as follows

$$\begin{aligned}
& \text{minimize} && \sum_{(u,v) \in E} f_{(u,v)} \cdot \tilde{d}_{(u,v)}^{(\epsilon)} && (10) \\
& \text{subject to} && f_{(u,v)} \leq 1 && \text{(capacity constraints)} \\
& && f_{(u,v)} = -f_{(v,u)} && \text{(skew symmetry)} \\
& && \sum_{v \in V} f_{(u,v)} = 0 \ \forall u \neq a, b && \text{(flow conservation)} \\
& && \sum_{v \in V} f_{(a,v)} = n \text{ and } \sum_{v \in V} f_{(v,b)} = n && \text{(required flow constraints)}
\end{aligned}$$

where with  $f_{(u,v)}$  we denote the flow of edge  $(u, v)$ . In terms of computational complexity, it is known that symbolic linear programs can be solved efficiently [28].

Since the solution to this linear program also minimizes the potential function  $\tilde{\Phi}^{(\epsilon)}$ , such a solution maps to a pure strategy profile,  $\mathbf{x}^{(\epsilon)}$ , in  $\tilde{\Gamma}^{(\epsilon)}$ , where no player can unilaterally change their path to decrease their cost. Hence,  $\{\mathbf{x}^{(\epsilon)}\}_{\epsilon \rightarrow 0+}$  defines a sequence of  $\epsilon$ -perfect equilibria of  $\Gamma$ , the limit point of which is a perfect equilibrium.  $\square$

### 8.3 Strongly polynomial algorithms and perturbed optimization

A recurring question within the equilibrium refinement literature is this: if an algorithm can be executed on numerical inputs, can it also be run  $\epsilon$ -symbolically for an entire family of inputs when  $\epsilon > 0$  is small enough? We have already encountered basic versions of this problem. The simplest example is implementing symbolic perfect best-response dynamics, which, on top of polynomial interpolation, rests on computing a maximum of a vector; this can be accomplished by a series of comparisons. A more interesting example pertains to symmetric network congestion games, which was covered a moment ago. There, the implementation rests on performing a series of additions and comparisons, which are again directly amenable to symbolic inputs—when  $\epsilon$  is small enough. This begs the question: is there a more general class of algorithms for which one can directly guarantee symbolic implementation? We provide one such answer here by making a connection with *strongly polynomial-time* algorithms.

A strongly polynomial-time algorithm is defined in the arithmetic model of computation. Here, the basic arithmetic operations—addition, subtraction, multiplication, division, and comparison—take one unit of time to execute, no matter the sizes of the operands. An algorithm is said to run in strongly polynomial time [35] if i) the number of operations in the arithmetic model of computation is bounded by a polynomial in the number of rationals given as part of the input, and ii) the space used by the algorithm is bounded by a polynomial in the size of the input.

We think of a strongly polynomial-time algorithm  $\mathcal{A}$  as a circuit with a polynomial number of gates  $|\mathcal{G}|$ . The degree of a rational function is the sum of the degrees of its numerator and denominator.

**Theorem 8.6.** *Let  $\mathbf{x} \in \mathbb{Q}[\epsilon]^M$  be a vector each of whose entries is a polynomial in  $\epsilon$  of degree at most  $d_x$ . Suppose further that the output of each gate of  $\mathcal{A}$  with input  $\mathbf{x}$  is a piecewise rational function such that each piece has degree at most  $d_A = \text{poly}(d_x, |\mathcal{G}|)$ . Then there is a polynomial-time algorithm that computes the output of the circuit symbolically for any sufficiently small  $\epsilon > 0$ .*

*Proof.* Suppose that the output of a gate is given by a rational function  $p(\epsilon)/q(\epsilon)$  when  $\epsilon > 0$  is small enough so that the outputs consistently lie within a single piece. We let  $p(\epsilon) = \alpha_0 + \alpha_1\epsilon + \alpha_2\epsilon^2 + \dots + \alpha_r\epsilon^r$  and  $q(\epsilon) = \beta_0 + \beta_1\epsilon + \beta_2\epsilon^2 + \dots + \beta_{d_A-r}\epsilon^{d_A-r}$  for some  $r \leq d_A$ ; we can always take  $\beta_j = 1$  for some  $j$  by scaling both the numerator and the denominator without affecting the underlying rational function. Consider now a series of distinct points  $\epsilon_1, \dots, \epsilon_{d_A+1}$ , each of which can be described with  $\text{poly}(d_A, |\mathcal{G}|)$ , such that  $q(\epsilon_j) \neq 0$  and  $y_j = p(\epsilon_j)/q(\epsilon_j)$ .<sup>8</sup> The coefficients of  $p(\epsilon)$  and  $q(\epsilon)$  can be determined through any solution to the linear system  $p(\epsilon_j) - q(\epsilon_j)y_j = 0$  for  $j = 1, \dots, d_A + 1$ . Any two rational functions  $p/q$  and  $p'/q'$  obtained through a solution to this linear system must be equal, excluding the roots of  $q$  and  $q'$ . Indeed, the degree- $d_A$  polynomial  $pq' - p'q$  must have  $d_A + 1$  roots, which can only happen if it is the zero polynomial. In particular, each coefficient can be expressed with  $\text{poly}(d_A, |\mathcal{G}|)$  bits since it is a solution to a linear system with  $\text{poly}(d_A, |\mathcal{G}|)$  bit complexity—each  $\epsilon_j$  and  $y_j$  can be represented with  $\text{poly}(d_A, |\mathcal{G}|)$  bits. This implies that each rational function  $p(\epsilon)/q(\epsilon)$  has coefficients whose bit complexity is  $\text{poly}(d_A, |\mathcal{G}|)$ . As a result, we conclude that, by taking  $\epsilon^* = 2^{p(d_A, |\mathcal{G}|)}$  sufficiently small, all outputs in the comparison gates will lie in the same piece for any  $\epsilon \leq \epsilon^*$ . The output of the circuit as a rational function can be thus determined by selecting a sequence of distinct points  $\epsilon_1, \dots, \epsilon_{d_A+1} \leq \epsilon^*$  and applying fractional interpolation based on the (numerical) output of the circuit in the corresponding inputs.  $\square$

<sup>8</sup>Whether  $\epsilon_j$  is small enough for the piece corresponding to  $p(\epsilon)/q(\epsilon)$  to be activated is moot for this argument.

## 9 Computing proper equilibria

We are concerned in this section with the complexity of computing proper equilibria in concise (normal-form) potential games (Section 9.1) and more generally extensive-form and polytope games (Sections 9.2 and 9.4).

### 9.1 Normal-form games

For ease of notation, in what follows, we will drop the normalizing factor  $\frac{1-\epsilon}{1-\epsilon^{m_i}}$  for each player  $i$ ; since normalizing amounts to a uniform scaling of the utilities, this will not affect our algorithm, which only requires a relative comparison of actions' payoffs.

**Theorem 9.1.** *Algorithm 1 returns a proper equilibrium of  $\Gamma$  in finite time. Further, for concise games, Algorithm 1 can be implemented in  $\text{poly}(n, m)$  time.*

*Proof.* We begin by analyzing the per-iteration time complexity. First, we will argue that for any current joint strategy  $\mathbf{x}^{(\epsilon)} \in \mathcal{X}^{(\epsilon)}$  of the perturbed game, an  $\epsilon$ -pure strategy best-response by any player, yields at least as much utility for that player as any mixed-strategy. To show this, for any given player  $i$  let  $\epsilon_i^{(\pi)} = (\epsilon^{\pi(0)}, \epsilon^{\pi(1)}, \dots, \epsilon^{\pi(m_i-1)})$  be player  $i$ 's  $\epsilon$ -pure strategy corresponding to permutation  $\pi$ . Since any  $\epsilon$ -perturbed strategy must be a convex combination of  $\{\epsilon_i^\pi : \pi \in S_{m_i-1}\}$ , it follows that the maximum utility that player  $i$  can guarantee when best responding to joint strategy  $\mathbf{x}^{(\epsilon)}$  is  $\max_{\pi \in S_{m_i-1}} u_i(\epsilon_i^{(\pi)}, \mathbf{x}_{-i}^{(\epsilon)})$ . Therefore, we know that throughout Algorithm 1, the probability mass that any player  $i \in [n]$  assigns to any given  $a_i \in \mathcal{A}_i$  is of the form  $\mathbf{x}_i^{(\epsilon)}(a_i) = \epsilon^k$  for some  $k \in \{0, \dots, m_i - 1\}$ . So for any such joint strategy,  $\mathbf{x}^{(\epsilon)} \in \mathcal{X}^{(\epsilon)}$ , and any two permutations,  $\pi, \pi' \in S_{m_i-1}$ ,  $p_i(\epsilon) := u_i(\epsilon_i^{(\pi)}, \mathbf{x}_{-i}^{(\epsilon)})$  and  $q_i(\epsilon) := u_i(\epsilon_i^{(\pi')}, \mathbf{x}_{-i}^{(\epsilon)})$  are polynomials of degree at most  $2m$ . For concise games  $p_i(\epsilon), q_i(\epsilon)$  can be evaluated in polynomial time using Lemma 3.10. To ascertain whether  $p(\epsilon) <_{\epsilon \rightarrow 0+} q(\epsilon)$ , we can compare their coefficients in lexicographic order. This concludes the analysis of Algorithm 1.

Now we show that Algorithm 1 can be implemented efficiently too. For player  $i$ , we consider the  $m_i$ -dimensional utility vector  $\mathbf{u}_i(\mathbf{x}_{-i}^{(\epsilon)}) = (u_i(a_i, \mathbf{x}_{-i}^{(\epsilon)}))_{a_i \in \mathcal{A}_i}$ . Also, we consider the permutation  $\pi_i \in S_{m_i-1}$  that corresponds to the sorted utility values of  $\mathbf{u}_i(\mathbf{x}_{-i}^{(\epsilon)})$ , that is,  $u_i(a_i, \mathbf{x}_{-i}^{(\epsilon)}) <_{\epsilon \rightarrow 0+} u_i(a'_i, \mathbf{x}_{-i}^{(\epsilon)}) \iff \pi_i(a_i) > \pi_i(a'_i)$ . Then the best-response of player  $i$  in the perturbed strategy space must be the one-hot vector that assigns with probability mass 1 to  $\epsilon^{(\pi)}$ . Indeed, if  $u_i(a_i, \mathbf{x}_{-i}^{(\epsilon)}) <_{\epsilon \rightarrow 0+} u_i(a'_i, \mathbf{x}_{-i}^{(\epsilon)})$ , then  $\pi(a_i) > \pi(a'_i)$  for any valid best-response  $\epsilon^{(\pi)}$ . So, Algorithm 1 in Algorithm 1 can be implemented in polynomial time by sorting the elements of  $\mathbf{u}_i(\mathbf{x}_{-i}^{(\epsilon)})$ .

We recall that from Kohlberg and Mertens [45] we know that a Nash equilibrium of an  $\epsilon$ -perturbed game  $\Gamma_\epsilon$  is an  $\epsilon$ -proper equilibrium of the original game  $\Gamma$ . Therefore, the proof of convergence follows a similar argument to the corresponding proof in Theorem 5.2.  $\square$

**Theorem 9.2.** *Finding a proper equilibrium of a concise potential game is in PLS.*

*Proof.* The proof follows similarly to Theorem 5.3.  $\square$

### 9.2 Hardness of proper equilibria in polytope games

We have seen that, for *perfect* equilibria, computing equilibria in polytope games has similar complexity to computing equilibria in normal-form games. One may hope that this continues



to hold for proper equilibria. Unfortunately, this is not the case—in fact, computing a proper equilibrium is a much harder problem than computing Nash or perfect equilibria.

For most of the remainder of this section, we will drop the subscript  $i$  indicating the player, since (following the framework given by [Theorem 5.7](#)) we will usually be interested in computing a single-player best response. Let  $\mathcal{X}$  be a polytope and  $\mathcal{V}$  be its set of vertices. Let  $\mathbf{u}$  be a utility vector. Our goal is to compute a *perturbed best response* to  $\mathbf{u}$  that is valid for normal-form proper equilibria ([Definition 2.8](#)). We will use  $\mathbf{x} \succeq \mathbf{x}'$  to mean  $\langle \mathbf{u}, \mathbf{x} \rangle \geq \langle \mathbf{u}, \mathbf{x}' \rangle$ . Formally, our goal is the following.

**Definition 9.3** ( $\epsilon$ -proper best response). Given polytope  $\mathcal{X} \subset \mathbb{R}^d$  and  $\mathbf{u} \in \mathbb{R}^d$ , an  $\epsilon$ -proper best response is a strategy  $\mathbf{x}^* \in \mathcal{X}$  such that  $\mathbf{x}^* = \mathbb{E}_{\mathbf{x} \sim \lambda} \mathbf{x}$  for *some* full-support distribution  $\lambda \in \Delta(\mathcal{V})$  such that  $\mathbf{x} \prec \mathbf{x}'$  implies  $\lambda_{\mathbf{x}} \leq \epsilon \cdot \lambda(\mathbf{x}')$ .

Our first result shows that the “naive”  $\epsilon$ -best response used for normal-form games, *i.e.*, the one implied by [Definition 2.9](#), cannot be computed in polytope games even when the polytope is the  $[0, 1]$ -hypercube.

**Theorem 9.4** (Hardness of computing the KM symbolic  $\epsilon$ -proper best response on the hypercube). *Given a vector  $\mathbf{u} \in \mathbb{R}^d$ , define a KM best response on the hypercube to be any vector<sup>9</sup>  $\mathbf{x}^{(\epsilon)} \in [0, 1]^d$  of the form*

$$\mathbf{x}^{(\epsilon)} = \sum_{i=0}^{2^d-1} \epsilon^i \mathbf{v}_i$$

*where  $\mathbf{v}_0 \succeq \cdots \succeq \mathbf{v}_{2^d-1}$  is the list of vertices of the hypercube  $\{0, 1\}^d$ , sorted in descending order of utility, and  $\epsilon$  is symbolic. The following promise problem is  $\#P$ -hard under Turing reductions: given a utility vector  $\mathbf{u}$ , compute the leading term in the polynomial  $\mathbf{x}^*[1]$ , given the promise that this leading term is the same for all KM best responses  $\mathbf{x}^*$ .*

*Proof.* We reduce from  $\#KNAPSACK$ . We are given a vector  $\mathbf{w} \in \mathbb{Z}^d$  and a total  $W \in \mathbb{Z}$ , and our goal is to count how many  $\mathbf{z} \in \{0, 1\}^d$  there are with  $\langle \mathbf{w}, \mathbf{z} \rangle \leq W$ . Consider the utility vector  $\mathbf{u} = (-(W + 1/2), -\mathbf{w}) \in \mathbb{R}^{d+1}$ . Then, by construction, we have that  $\mathbf{v}_1[1] = \mathbf{v}_2[1] = \cdots = \mathbf{v}_k[1] = 0 \neq \mathbf{v}_{k+1}[1] = 1$ , where  $k$  is the number of vectors  $\mathbf{z}$  for which  $\langle \mathbf{w}, \mathbf{z} \rangle \leq W$ . Thus, the leading term of  $\mathbf{x}^*[1]$  is exactly  $\epsilon^k$ , and computing  $\epsilon^k$  exactly solves the  $\#KNAPSACK$  instance.  $\square$

However, the KM best response is only one way to satisfy [Definition 9.3](#), and one might ask whether there are others. Here, we give strong evidence to the contrary, in the form of showing that computing a proper equilibrium of even a two-player polytope potential game is  $\text{NP-hard}$ . This creates a separation between proper and perfect equilibria in polytope games, since we have already seen that the latter problem lies in  $\text{CLS}$  for two-player potential games ([Corollary 6.1](#)).

**Theorem 9.5** (Hardness of computation). *The following promise problem is  $\text{NP-hard}$  (and  $\text{coNP-hard}$ , because it is trivially equivalent to its own negation): given a two-player identical-interest polytope game where  $P1$ ’s strategy set  $\mathcal{X}$  is given by explicit linear constraints and  $P2$ ’s pure strategy set is  $\{0, 1\}$ , decide whether*

- *$P2$  deterministically plays 0 in every normal-form proper equilibrium, or*
- *$P2$  deterministically plays 1 in every normal-form proper equilibrium*

---

<sup>9</sup>Note that there can be more than one such vector, due to tiebreaking.

given the promise that one of the two statements is true.

We dedicate the remainder of the subsection to the proof of the above theorem. We reduce from PARTITION. We are given a vector  $\mathbf{w} \in \mathbb{Z}^d$ , and our goal is to decide whether there is  $\mathbf{z} \in \{-1, 1\}^d$  with  $\langle \mathbf{w}, \mathbf{z} \rangle = 0$ . Define  $\mathcal{X}$  as

$$\begin{aligned} \mathcal{X} &= \left\{ (t, \mathbf{z}) \in [0, 1] \times \mathbb{R}^d : \|\mathbf{z}\|_\infty \leq 1 - t, \langle \mathbf{w}, \mathbf{z} \rangle \leq 1 - t \right\} \\ &= \text{conv}(\{(1, \mathbf{0})\} \cup (\{0\} \times \mathcal{Z})) \quad \text{where} \quad \mathcal{Z} = \{\mathbf{z} \in [-1, 1]^d : \langle \mathbf{w}, \mathbf{z} \rangle \leq 1\}. \end{aligned}$$

Let the utility function  $u : \mathcal{X} \times [0, 1] \rightarrow \mathbb{R}$  of both players be

$$u(\mathbf{x} = (t, \mathbf{z}), y) = \underbrace{-\frac{1}{2}t + \langle \mathbf{w}, \mathbf{z} \rangle}_{=: \tilde{u}(t, \mathbf{z})} + \underbrace{\frac{y}{3(1 + \|\mathbf{w}\|_1)}(1 - \langle \mathbf{w}, \mathbf{z} \rangle - 2t)}_{=: \delta(t, \mathbf{z}, y)}.$$

We will use the shorthand  $\mathbf{x} = (t, \mathbf{z})$  freely, and we will use  $\perp = (1, \mathbf{0}) \in \mathcal{X}$ . Intuitively, the rest of the proof proceeds as follows. The term  $\delta(\cdot)$  is small and can be ignored when thinking about P1's incentive. The ordering of P1's pure strategies in decreasing order of utility is as follows: first, all those pure strategies  $(0, \mathbf{z})$  where  $\langle \mathbf{w}, \mathbf{z} \rangle \geq 1$ ; second, all those pure strategies  $(0, \mathbf{z})$  where  $\langle \mathbf{w}, \mathbf{z} \rangle = 0$ ; third,  $\perp$ ; and finally, all other pure strategies, namely those of the form  $(0, \mathbf{z})$  with  $\langle \mathbf{w}, \mathbf{z} \rangle \leq -1$ . The second category are the solutions to the partition problem. Moreover,  $t$  is precisely the probability of playing  $\perp$ , and in any  $\epsilon$ -proper best response, for sufficiently small  $\epsilon$ , the probability of each category is much smaller than the probability of the previous category. Thus, by comparing  $1 - \langle \mathbf{w}, \mathbf{z} \rangle$  to  $t$ , we can understand whether the second category is empty. And P2's incentive is essentially to test which of these two quantities is bigger, so we can use P2's strategy in any proper equilibrium to determine whether a partition exists.

**Lemma 9.6.** *Let  $\mathbf{x}, \mathbf{x}'$  be two vertices of  $\mathcal{X}$ . If  $\tilde{u}(\mathbf{x}) > \tilde{u}(\mathbf{x}')$ , then  $u(\mathbf{x}, y) > u(\mathbf{x}', y)$  for all  $y$ .*

*Proof.*  $\tilde{u}$  can only take on half-integral values when  $t$  and  $\mathbf{z}$  are integral, and  $0 \leq \delta(t, \mathbf{z}, y) \leq 1/3$  for all  $(t, \mathbf{z}, y)$ , so adding  $\delta$  cannot affect the ordering of the  $\mathbf{x}$ s.  $\square$

Let  $(t^*, \mathbf{z}^*, y^*)$  be an  $\epsilon$ -proper equilibrium. Order the vertices of  $\mathcal{X}$  as  $\mathcal{V} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  where  $\mathbf{x}_i \succeq \mathbf{x}_{i+1}$  for all  $i$ , and  $\succeq$  is with respect to P1's utility  $u(\cdot, y^*)$ . Note that, by the previous lemma, this ordering also applies to  $\tilde{u}$ , that is, we have  $\tilde{u}(\mathbf{x}_i) \geq \tilde{u}(\mathbf{x}_{i+1})$  for all  $i$ . We will use this fact freely. By definition of  $\epsilon$ -proper best response, there is a full-support distribution  $\lambda \in \Delta(N)$  such that  $\mathbf{x}^* = \mathbb{E}_{i \sim \lambda} \mathbf{x}_i$  and  $\lambda_j \leq \epsilon \lambda_i$  whenever  $\mathbf{x}_j \prec \mathbf{x}_i$ . Let  $i^*$  be the smallest index for which  $\tilde{u}(\mathbf{x}_{i^*}) < 1$ . Notice that, for all  $i < i^*$ , we must have  $t_i = 0$  and  $\langle \mathbf{w}, \mathbf{z}_i \rangle = 1$ .

**Lemma 9.7.** *If the PARTITION instance has a solution, then  $y^* \geq 1 - \epsilon$ .*

*Proof.* In this case, we must have  $\tilde{u}(\mathbf{x}_{i^*}) = 0$ , and in fact  $\mathbf{z}_{i^*}$  must be a solution to the partition instance. Thus,  $1 - \langle \mathbf{w}, \mathbf{z}^* \rangle \geq \lambda_{i^*}$ . Moreover, since  $\tilde{u}(\perp) = -1/2 < 0$ , the weight  $\lambda$  places on  $\perp$  must be at most  $\epsilon \cdot \lambda_{i^*}$ , so  $t \leq \epsilon \cdot \lambda_{i^*}$ . Thus  $1 - \langle \mathbf{w}, \mathbf{z}^* \rangle - 2t^* > 0$ , and in order for  $y^*$  to be a best response, we must have  $y^* \geq 1 - \epsilon$ .  $\square$

**Lemma 9.8.** *If the PARTITION instance has no solution, then  $y^* \leq \epsilon$ .*

*Proof.* In this case, we have  $\mathbf{x}_{i^*} = \perp$ . Moreover, the very next strategy  $\mathbf{x}_{i^*+1}$  must be a vertex of  $\mathcal{Z}$  and thus must have  $\tilde{u}(\mathbf{x}_{i^*+1}) \leq -1$ , because  $\tilde{u}(\perp) = -1/2$  and  $\perp$  is the only strategy with a non-integral value of  $\tilde{u}$ . Thus,  $\lambda_j \leq \epsilon \lambda_{i^*}$  for every  $j > i^*$ , and therefore we have

$$1 - \langle \mathbf{w}, \mathbf{z}^* \rangle \leq \lambda_{i^*} + \sum_{j>i} \lambda_j \cdot \|\mathbf{w}\|_1 \leq \lambda_{i^*}(1 + \epsilon N \|\mathbf{w}\|_1) < 2\lambda_{i^*} = t$$

for sufficiently small  $\epsilon$ . Thus  $1 - \langle \mathbf{w}, \mathbf{z}^* \rangle - 2t^* < 0$ , and in order for  $y^*$  to be a best response, we must have  $y^* \leq \epsilon$ .  $\square$

Thus, passing to the limit, if the PARTITION instance has a solution, then every proper equilibrium has  $y = 1$ ; if the PARTITION instance has no solution, then every proper equilibrium has  $y = 0$ . This completes the proof.

*Remark 9.9.* The above result implies that, if a normal-form proper equilibrium can be computed in polynomial time, then  $P = NP$ . It does *not* imply that computing a normal-form proper equilibrium is FNP-hard, because the proof does not allow the extraction of the partition in the case where one exists, which would be required for FNP-hardness. We leave closing this gap to future research.

In the above proof, P1's order of pure strategies does not depend at all on P2's strategy. Thus, if  $\mathbf{x}^* = (t, \mathbf{z})$  is any optimizer of  $\langle \mathbf{w}, \mathbf{z} \rangle$ , then one of the two profiles  $(\mathbf{x}^*, 0)$  and  $(\mathbf{x}^*, 1)$  is a proper equilibrium. It thus follows that:

**Corollary 9.10.** *Checking whether a given profile is a proper equilibrium is NP-hard for polytope games.*

### 9.3 Optimal spanning sets

The previous section showed NP-hardness for computing normal-form proper equilibria for general polytope games, precluding any best-response-iteration-type algorithms in general. In this section, we will introduce a framework by which a certain subclass of polytope games *will* have efficiently-computable best responses. Notably, this subclass will include the extensive-form games.

Critically, computing an  $\epsilon$ -proper best response does *not* require the computation of the distribution  $\lambda$  itself, only the expectation of said distribution. This distinction will be crucial to efficient algorithms.

Before stating the formalism for this subsection, we give some intuition in the form of an informal example.

**Example 9.11.** Suppose  $\mathcal{V} = \{0, 1\}^d$  is a hypercube, let  $\mathbf{u} \in \mathbb{R}^d$  be a utility vector, and assume (WLOG) that  $\mathbf{x}^* = \mathbf{0}$  is a best response to  $\mathbf{u}$ . One can compute an  $\epsilon$ -proper best response by the following method: Let  $\mathbf{e}_1 \succeq \dots \succeq \mathbf{e}_d$  be the  $d$  one-hot vectors, ordered in descending order of utility. Then we claim that the vector  $\mathbf{x}^{(\hat{\epsilon})} := \sum_{i=1}^d \hat{\epsilon}^i \mathbf{e}_i$  is a valid  $\epsilon$ -proper best response, for sufficiently small  $\hat{\epsilon} \ll \epsilon$ . To see this, note that for any vector  $\mathbf{x} = \sum_{i \in S} \mathbf{e}_i \in \{0, 1\}^d$ ,  $\mathbf{x}$  lies in the span of  $\{\mathbf{e}_i\}_{i \in S}$ , and  $\mathbf{e}_i \succeq \mathbf{x}$  for all  $i \in S$ ; therefore, we can pretend there is a small amount of probability mass on  $\mathbf{x}$  by removing that amount of mass from each of the basis vectors  $\{\mathbf{e}_i\}_{i \in S}$ . This allows us to show that  $\mathbf{x}^{(\hat{\epsilon})}$  is an  $\epsilon$ -proper best response, without ever actually needing to know the order of all the vertices under  $\mathbf{u}$ .

This example shows the contrast between computing a KM best response and computing *any*  $\epsilon$ -proper best response: while the former problem was  $\#P$ -hard on the hypercube, the latter is easy.

We will now generalize and formalize this intuition. Let  $B \subseteq \mathcal{X}$ . For any  $\mathbf{x} \in \mathbb{R}^d$ , let  $B_{\succeq \mathbf{x}}$  be the subset of points in  $B$  that are at least as good as  $\mathbf{x}$ . That is,

$$B_{\succeq \mathbf{x}} = \{\mathbf{x}' \in B : \langle \mathbf{u}, \mathbf{x}' - \mathbf{x} \rangle \geq 0\}.$$

We call  $B$  an *optimal spanning set* of  $\mathbf{u}$  if every  $\mathbf{x} \in \mathcal{V}$  lies in the affine hull of  $B_{\succeq \mathbf{x}}$ . In particular,  $B$  must contain at least one optimizer of  $\langle \mathbf{u}, \mathbf{x} \rangle$ , and its affine hull must be exactly the affine hull of  $\mathcal{X}$ .

Very informally, an optimal spanning set should be sufficient information to construct a proper best response, because if every  $\mathbf{x} \in \mathcal{V}$  is in the affine hull of “better” vertices  $\mathbf{x}'$ , then the term  $\epsilon^k \mathbf{x}$ , where  $k$  is the power with which  $\mathbf{x}$  “should” appear, can be “absorbed into” the better  $\mathbf{x}'$ s.

We now formalize some basic properties of optimal spanning sets.

**Proposition 9.12.** *An optimal spanning set always exists.*

*Proof.*  $\mathcal{V}$  is always an optimal spanning set. □

**Proposition 9.13.** *Every inclusion-wise minimal optimal spanning set has size exactly  $d + 1$ , and is hence an affine basis of  $\mathcal{X}$ .*

*Proof.* (Lower bound) An optimal spanning set must satisfy  $\mathcal{X} \subset \text{aff } B$ , and this is only possible if  $|B| \geq d + 1$ .

(Upper bound) Let  $B = \{\mathbf{x}_0, \dots, \mathbf{x}_K\}$  where  $\mathbf{x}_0 \succeq \dots \succeq \mathbf{x}_K$ . If  $K = d$  we are done, so assume  $K > d$ . Let  $B^{(k)} = \{\mathbf{x}_0, \dots, \mathbf{x}_k\}$ . Let  $d_k$  be the dimension of  $\text{aff } B^{(k)}$ . Since  $0 \leq d_k \leq d$  for all  $k$  and  $K > d$ , there must be some  $k \in \{1, \dots, K\}$  for which  $d_k = d_{k-1}$ . Then, by definition, we have  $\mathbf{x}_k \in \text{aff } B^{(k-1)}$ . But then  $B \setminus \{\mathbf{x}_k\}$  is also optimal spanning set, so  $B$  was not inclusion-wise minimal. □

Let  $I(\mathcal{X})$  be the collection of affine bases of  $\mathcal{X}$ , that is, subsets  $B \subset \mathcal{X}$  of size exactly  $d + 1$  for which  $\mathcal{X} = \text{aff } B$ . Note that, by [Proposition 9.13](#), every inclusion-wise minimal optimal spanning set is in  $I(\mathcal{X})$ . Now fix some sufficiently small  $\epsilon > 0$ , and for any  $B = \{\mathbf{x}_0, \dots, \mathbf{x}_d\} \in I(\mathcal{X})$  with  $\mathbf{x}_0 \succeq \dots \succeq \mathbf{x}_d$ , let  $\mathbf{v}(B) = (\langle \mathbf{u}, \mathbf{x}_0 \rangle, \dots, \langle \mathbf{u}, \mathbf{x}_d \rangle) \in \mathbb{R}^{d+1}$ .

Now let  $B, B' \in I(\mathcal{V})$  where  $B$  is an optimal spanning set (but  $B'$  need not be).

**Lemma 9.14.**  $\mathbf{v}(B) \geq \mathbf{v}(B')$ , where the comparison is element-wise.

*Proof.* Let  $B$  be any affinely independent subset of  $\mathcal{X}$ . Then  $|B_{\succeq \mathbf{x}}| \leq 1 + \dim \text{aff } \mathcal{V}_{\succeq \mathbf{x}}$ , because otherwise  $B$  is not affinely independent on the subset  $\mathcal{V}_{\succeq \mathbf{x}}$ . We claim that if  $B$  is an optimal spanning set then in fact the inequality is an equality:  $|B_{\succeq \mathbf{x}}| = 1 + \dim \text{aff } \mathcal{V}_{\succeq \mathbf{x}}$  for all  $\mathbf{x}$ . This follows from the fact that, for any  $\mathbf{x}' \succeq \mathbf{x}$ , we have  $\mathbf{x}' \in \text{aff } B_{\succeq \mathbf{x}}$ , and therefore  $\text{aff } \mathcal{V}_{\succeq \mathbf{x}} = \text{aff } B_{\succeq \mathbf{x}}$ . But this is equivalent to the statement of the lemma. □

**Theorem 9.15.** *For every polytope  $\mathcal{X}$  and  $\epsilon > 0$ , there exists  $\epsilon^* > 0$  such that, for every  $\hat{\epsilon} \in (0, \epsilon^*]$  and every utility vector  $\mathbf{u}$ , the following holds: let  $B = \{\mathbf{x}_0, \dots, \mathbf{x}_K\}$  be any optimal spanning set of  $\mathbf{u}$ , where  $\mathbf{x}_0 \succeq \dots \succeq \mathbf{x}_K$ . Then the strategy*

$$\mathbf{x}^{(\hat{\epsilon})} := \mathbf{x}_0 + \sum_{i=1}^K \hat{\epsilon}^i (\mathbf{x}_i - \mathbf{x}_0) \in \mathcal{X}$$

*is an  $\epsilon$ -proper best response.*

That is, finding a proper best response for the set of vertices  $\mathcal{V}$  reduces to finding a proper best response over the (possibly much smaller) optimal spanning set  $B$ .

*Proof.* For any  $\mathbf{x} \in \mathcal{V}$ , by definition of optimal spanning set, we can write

$$\mathbf{x} = \sum_{\mathbf{x}' \in B_{\succeq \mathbf{x}}} c_{\mathbf{x}}(\mathbf{x}') \cdot \mathbf{x}'$$

where  $c_{\mathbf{x}}(\mathbf{x}') \in \mathbb{R}$  are coefficients (not necessarily nonnegative) satisfying  $\sum_{\mathbf{x}' \in B_{\succeq \mathbf{x}}} c_{\mathbf{x}}(\mathbf{x}') = 1$ . Then, for any  $\lambda \in \mathbb{R}[\epsilon]^{\mathcal{V}}$ , we have

$$\begin{aligned} \mathbf{x}^{(\epsilon)} &= \mathbf{x}_0 + \sum_{i=1}^K \hat{\epsilon}^i (\mathbf{x}_i - \mathbf{x}_0) - \sum_{\mathbf{x} \in \mathcal{V}} \lambda_{\mathbf{x}} \cdot \mathbf{x} + \sum_{\mathbf{x} \in \mathcal{V}} \lambda_{\mathbf{x}} \cdot \mathbf{x} \\ &= \left(1 - \sum_{i=1}^K \hat{\epsilon}^i\right) \mathbf{x}_0 + \sum_{i=1}^K \hat{\epsilon}^i \mathbf{x}_i - \sum_{\mathbf{x} \in \mathcal{V}} \lambda_{\mathbf{x}} \sum_{\mathbf{x}' \in B_{\succeq \mathbf{x}}} c_{\mathbf{x}}(\mathbf{x}') \cdot \mathbf{x}' + \sum_{\mathbf{x} \in \mathcal{V}} \lambda_{\mathbf{x}} \cdot \mathbf{x} \\ &= \left(1 - \sum_{i=1}^K \hat{\epsilon}^i\right) \mathbf{x}_0 + \sum_{i=1}^K \left( \hat{\epsilon}^i - \sum_{\mathbf{x} \in \mathcal{V} \setminus B: \mathbf{x}_i \succeq \mathbf{x}} \lambda_{\mathbf{x}} c_{\mathbf{x}}(\mathbf{x}_i) \right) \mathbf{x}_i + \sum_{\mathbf{x} \in \mathcal{V}} \lambda_{\mathbf{x}} \cdot \mathbf{x}. \end{aligned} \quad (11)$$

Notice that, for sufficiently small  $\epsilon$  and  $\lambda > \mathbf{0}$ , this is a fully-mixed probability distribution on  $\mathcal{V}$ : the coefficients sum to 1, and as long as  $\lambda_{\mathbf{x}} \ll \hat{\epsilon}^i$  whenever  $\mathbf{x}_i \succeq \mathbf{x}$ , they are all nonnegative. Thus, the following setting of  $\lambda$  completes the proof: number the pure strategies in  $\mathcal{V}$  as  $\mathbf{v}_1 \succeq \dots \succeq \mathbf{v}_{|\mathcal{V}|}$ , and define  $\lambda_{\mathbf{v}_j} = \hat{\epsilon}^{i(j)} \epsilon^{j+1} \geq \hat{\epsilon}^{i(j)+1} \epsilon^{-1}$ , where  $i(j)$  is the largest index for which  $\mathbf{x}_{i(j)} \succeq \mathbf{v}_j$ . The proof is now completed by taking any  $\hat{\epsilon} \leq \epsilon^* := \epsilon^{|\mathcal{V}|+3}$ .  $\square$

Since computing  $\mathbf{x}^{(\epsilon)}$  as in the above lemma is easy given  $B$ ,<sup>10</sup> the problem of computing a proper best response reduces to computing an optimal spanning set  $B$  given an objective  $\mathbf{u}$ .

We now show that there exists *some* perturbed polytope  $\mathcal{X}^{(\hat{\epsilon})}$  for which the proper best response in the previous lemma is a linear optimization oracle.

**Lemma 9.16.** *For any affinely independent set  $B \in I(\mathcal{V})$  and any  $\hat{\epsilon} < \epsilon^*$ , let  $\mathbf{x}^{(\hat{\epsilon})}(B)$  be the proper best response computed in [Theorem 9.15](#). Let*

$$\mathcal{X}^{(\hat{\epsilon})} := \text{conv}\{\mathbf{x}^{(\hat{\epsilon})}(B) : B \in I(\mathcal{V})\}. \quad (12)$$

*If  $B$  is an optimal spanning set, then  $\mathbf{x}^{(\hat{\epsilon})}(B)$  is a solution to  $\max_{\mathbf{x} \in \mathcal{X}^{(\hat{\epsilon})}} \langle \mathbf{u}, \mathbf{x} \rangle$ .*

*Proof.* We have  $\langle \mathbf{u}, \mathbf{x}^{(\hat{\epsilon})}(B) \rangle = \langle \mathbf{c}^{(\hat{\epsilon})}, \mathbf{v}(B) \rangle$ , where  $\mathbf{c} = (1 - \sum_{i=1}^d \hat{\epsilon}^i, \hat{\epsilon}, \hat{\epsilon}^2, \dots, \hat{\epsilon}^d) \in \mathbb{R}^{d+1}$ . Since  $\mathbf{c}$  has all nonnegative coefficients, it follows from [Lemma 9.14](#) that, over all subsets  $B \in I(\mathcal{V})$ , optimal spanning sets optimize the utility value.  $\square$

**Corollary 9.17.** *For any polytope game  $\Gamma$  and any  $\epsilon > 0$ , there is an  $\epsilon^* > 0$  such that, for all  $\hat{\epsilon} \in (0, \epsilon^*]$ , any Nash equilibrium of the game with perturbed strategy sets  $\mathcal{X}_i^{(\hat{\epsilon})}$  as defined in (12) is an  $\epsilon$ -proper equilibrium of  $\Gamma$ .*

We will refer to the above game as the  $\hat{\epsilon}$ -proper game  $\Gamma^{(\hat{\epsilon})}$ .

<sup>10</sup>We remark once again that the algorithm only needs to compute  $\mathbf{x}^{(\hat{\epsilon})}$ , not the distribution  $\lambda$ ; the latter problem is obviously intractable as it requires essentially sorting a list of size  $|\mathcal{V}|$ .

## 9.4 Efficient normal-form proper equilibria in extensive-form games

**Theorem 9.5** precludes the possibility of efficient computation of optimal spanning sets for all polytopes. However, we will now show that for *extensive-form games*, computing an optimal spanning set is easy. (For background on extensive-form games, see [Section A](#).)

**Theorem 9.18.** *For extensive-form games, there is an efficient algorithm for computing an optimal spanning set  $B$  given a utility vector  $\mathbf{u}$ .*

Before proving [Theorem 9.18](#), for intuition, we discuss two special cases.

- $\mathcal{X}$  is a simplex,  $\mathcal{X} = \Delta(d)$ . Then  $\mathcal{V}$  is itself small, so we can simply use  $B = \mathcal{V}$  as an optimal spanning set. (indeed, this holds for any set  $\mathcal{X}$  with a polynomial number of vertices.)
- $\mathcal{X}$  is a hypercube ([Example 9.11](#)),  $\mathcal{X} = [0, 1]^d$ . In this case, one can check that a valid choice of optimal spanning set consists of an optimal point  $\mathbf{x}^* = \text{sign}(\mathbf{u})$ , and all of the points  $\mathbf{x} \in \mathcal{V}$  that are a single bit-flip away from  $\mathbf{x}^*$ . This optimal spanning set has size  $K = d + 1$ , so it follows that a proper best response can be computed in polynomial time in  $d$ .

To our knowledge, [Theorem 9.18](#) is new even when  $\mathcal{X}$  is a hypercube.

*Proof.* For any strategy  $\mathbf{x}$  and decision point  $j$ , define  $\mathbf{x}_{\not\leq j}$  to be the sub-vector of  $\mathbf{x}$  indexed on all sequences that are not successors of  $j$ . We now construct  $B$  iteratively, as follows.  $B$  will consist of one strategy  $\mathbf{y}^\sigma \in \mathcal{X}$  for each sequence  $\sigma \in \Sigma$ . The strategy  $\mathbf{y}^\sigma$  is a best response. For each sequence  $\sigma = ja \neq \emptyset$ , define  $\mathbf{y}^\sigma$  to be a best response to  $\mathbf{u}$ , over all strategies that play to sequence  $\sigma$  with probability 1, under the constraint that  $\mathbf{y}^\sigma$  agrees with  $\mathbf{y}^{p_j}$  outside  $j$ . In symbols,  $\mathbf{y}^\sigma$  is an optimal solution to the linear program

$$\max \quad \langle \mathbf{y}^\sigma, \mathbf{u} \rangle \quad \text{s.t.} \quad \mathbf{y}^\sigma \in \mathcal{X}, \quad \mathbf{y}_{\not\leq j}^\sigma = \mathbf{y}_{\not\leq j}^{p_j}, \quad \mathbf{y}^\sigma(\sigma) = 1. \quad (13)$$

This program is always feasible, because  $\mathbf{y}^{p_j}(p_j) = 1$  inductively. Moreover, this program can be solved by a straightforward bottom-up recursive pass.

Notice that  $\mathbf{y}^\sigma$  is also a best response under the constraint that  $\mathbf{y}^\sigma(\sigma) = 1$ . That is, the value of the LP (13) would not change if the constraint  $\mathbf{y}_{\not\leq j}^\sigma = \mathbf{y}_{\not\leq j}^{p_j}$  were removed. Indeed, this follows by a straightforward top-down induction on decision points  $j$ , since  $\mathbf{y}^{p_j}$  is already best-responding at decision points  $j' \not\leq j$  by inductive hypothesis.

It therefore only remains to show that the set  $B := \{\mathbf{y}^\sigma : \sigma \in \Sigma\}$  is an optimal spanning set.

We need to show that every vertex  $\mathbf{x} \in \mathcal{V}$  lies in the affine hull of  $B_{\succeq \mathbf{x}}$ , so let  $\mathbf{x}$  be an arbitrary vertex. We will describe an algorithm that converts  $\mathbf{x}$  into  $\mathbf{x}^*$  by a series of iterative steps that only move in  $\text{aff } B_{\succeq \mathbf{x}}$ .

The algorithm is the following. For each decision point  $j$  in bottom-up (leaves-first) order, if  $\mathbf{x}(p_j) = 1$ , then perform the operation

$$\mathbf{x} \leftarrow \mathbf{x} + \mathbf{y}^{p_j} - \mathbf{y}^{ja} \quad (14)$$

where  $a$  is the action played by  $\mathbf{x}$  at  $j$ . We make the following inductive claims about this algorithm: when the operation (14) is executed at decision point  $j$ :

1. before the operation,  $\mathbf{x} \preceq \mathbf{y}^{ja} \preceq \mathbf{y}^{p_j}$ ; therefore,  $\langle \mathbf{x}, \mathbf{u} \rangle$  only (weakly) increases, and all changes to  $\mathbf{x}$  are moves within  $\text{aff } B_{\succeq \mathbf{x}}$ .

*Proof.* Both inequalities follow from the earlier observation that  $\mathbf{y}^\sigma$  is a best response under the constraint  $\mathbf{y}^\sigma(\sigma) = 1$ .



2. after the operation  $\mathbf{x}_{\succeq j} = \mathbf{y}_{\succeq j}^{p_j}$ .

*Proof.* When the operation is executed at  $j$ , we have  $\mathbf{x}_{\succeq j} = \mathbf{y}_{\succeq j}^{j^a}$ , where  $a$  is the action played by  $\mathbf{x}$ , by inductive hypothesis, and  $\mathbf{y}_{\succeq j}^{p_j}$  agrees with  $\mathbf{y}_{\succeq j}^{j^a}$  by construction of the  $\mathbf{y}^\sigma$ s.

But then, after the algorithm is finished, we have  $\mathbf{x} = \mathbf{y}^\varnothing = \mathbf{x}^*$ , so we are done.  $\square$

By [Theorem 5.7](#), we therefore have the following result.

**Theorem 9.19.** *In extensive-form potential games, the problem of computing an exact normal-form proper equilibrium is in PLS.*

We now use the same best-response oracle to show that finding an exact equilibrium of an extensive-form (not necessarily potential) game lies in the class FIXP. We start with a useful lemma, which can be viewed as a generalization of [Lemma 6.2](#) to arbitrary circuits.

**Lemma 9.20.** *Consider a bounded domain  $D \subset \mathbb{R}^d$  and a circuit  $D \ni \mathbf{x} \mapsto F_\epsilon(\mathbf{x})$  with a set of gates  $\mathcal{G}$ , each of which performs either addition, multiplication, maximum, or minimum. For any  $\delta > 0$  and sufficiently small  $\epsilon, \epsilon' \geq \delta^{2^{O(d^3|\mathcal{G}|)}}$ , any  $\epsilon'$ -almost fixed point of  $F_\epsilon(\mathbf{x})$  is  $\delta$ -close to a limit point of fixed points of  $F_{\epsilon''}(\cdot)$  when  $\epsilon'' \rightarrow 0^+$ .*

*Proof.* For a given  $\mathbf{x} \in \mathbb{R}^d$ ,  $\epsilon > 0$ , and  $\epsilon' > 0$ , we define the formula  $\text{ALMOSTFP}(\mathbf{x}, \epsilon, \epsilon')$  that represents whether  $\mathbf{x}$  is an  $\epsilon'$ -almost fixed point of  $F_\epsilon(\mathbf{x})$ . We consider a set of auxiliary variables  $\mathbf{y}$  with existential quantifiers. For notational convenience, we write  $\tilde{\mathbf{x}} = (\mathbf{x}, \epsilon)$ . We first add the clauses  $\mathbf{y}[i] = \tilde{\mathbf{x}}[i]$  for all  $i$  that appear as inputs in the first level of the circuit. We also add the following clauses for all gates  $G \in \mathcal{G}$ :

$$\begin{aligned} \mathbf{y}[o] &= \mathbf{y}[i_1] + \mathbf{y}[i_2] \text{ for } G = (+, i_1, i_2, o); \\ \mathbf{y}[o] &= \mathbf{y}[i_1] * \mathbf{y}[i_2] \text{ for } G = (*, i_1, i_2, o); \\ \mathbf{y}[o] &= \zeta * \mathbf{y}[i] \text{ for } G = (*, i, \zeta, o); \\ (\mathbf{y}[i_1] \geq \mathbf{y}[i_2] \implies \mathbf{y}[o] = \mathbf{y}[i_1]) \wedge (\mathbf{y}[i_2] \geq \mathbf{y}[i_1] \implies \mathbf{y}[o] = \mathbf{y}[i_2]) &\text{ for } G = (\max, i_1, i_2, o); \\ (\mathbf{y}[i_1] \geq \mathbf{y}[i_2] \implies \mathbf{y}[o] = \mathbf{y}[i_2]) \wedge (\mathbf{y}[i_2] \geq \mathbf{y}[i_1] \implies \mathbf{y}[o] = \mathbf{y}[i_1]) &\text{ for } G = (\min, i_1, i_2, o). \end{aligned}$$

We then add the clause  $\|\tilde{\mathbf{x}} - \mathbf{y}[\text{out}]\|_2^2 \leq (\epsilon')^2$ , where  $\mathbf{y}[\text{out}]$  collects all the output variables of  $\mathbf{y}$ . Finally, we add clauses expressing that  $\mathbf{x} \in D$ , assumed to be linear constraints. What we want to prove is that for a fixed  $\delta > 0$ , and free variables  $\epsilon, \epsilon'$ ,

$$\forall \mathbf{x} \in \mathbb{R}^d \exists \mathbf{x}' \in \mathbb{R}^d : (\epsilon, \epsilon' > 0) \wedge (\neg \text{ALMOSTFP}(\mathbf{x}, \epsilon, \epsilon') \vee (\text{REFINEDFP}(\mathbf{x}')) \wedge \|\mathbf{x} - \mathbf{x}'\|_2^2 \leq \delta^2),$$

where  $\text{REFINEDFP}(\mathbf{x}') := \forall \epsilon'' > 0 \exists \mathbf{x}'' \in \mathbb{R}^d : \text{ALMOSTFP}(\mathbf{x}'', \epsilon'', 0) \wedge (\|\mathbf{x}' - \mathbf{x}''\|_2^2 \leq (\epsilon'')^2)$ . We denote by  $\text{ALMOSTNEARBOUND}_\delta(\epsilon, \epsilon')$  the above formula. Following the proof of Etessami et al. [25, Lemma 4], it follows that  $\text{ALMOSTNEARBOUND}_\delta(\epsilon, \epsilon')$  is satisfied for some  $\epsilon, \epsilon' \geq \delta^{2^{O(d^3|\mathcal{G}|)}}$ .  $\square$

We will also need the following result, which is a special case of the framework of Filos-Ratsikas et al. [31, 32]:

**Definition 9.21** (Special case of [31]). A *linear-OPT gate* is parameterized by integers  $n, m > 0$ , a matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , and a radius  $R$ . It takes as input vectors  $\mathbf{b} \in \mathbb{R}^m$  and  $\mathbf{c} \in [-R, R]^n$ , and outputs a solution to the linear program<sup>11</sup>

$$\max \quad \mathbf{c}^\top \mathbf{x} \quad \text{s.t.} \quad \mathbf{A}\mathbf{x} \leq \mathbf{b}, \quad -R \leq \mathbf{x} \leq R. \quad (15)$$

<sup>11</sup>As discussed by Filos-Ratsikas et al. [31], if there are multiple optimal solutions, any solution is a valid output; and the matrix  $\mathbf{A}$  must not be input-dependent.

**Theorem 9.22** ([31]). *Linear-OPT gates can be used freely as part of the construction of an arithmetic circuit for FIXP proofs. More precisely, for any linear-OPT gate parameterized by  $\mathbf{A} \in \mathbb{R}^{m \times n}$  and  $R$ , we can construct in time  $\text{poly}(m, n, \text{size}(A), \text{size}(R))$  an arithmetic circuit with gates  $\{+, -, *c, \max, \min\}$  (where  $*c$  is multiplication by a rational constant)  $F : \mathbb{R}^m \times \mathbb{R}^n \times [0, 1]^t \rightarrow \mathbb{R}^n \times [0, 1]^t$ , such that the following holds: for all  $\mathbf{b} \in \mathbb{R}^m$  and  $\mathbf{c} \in \mathbb{R}^n$ , if*

1. *the LP (15) is feasible, and*
2.  *$\mathbf{c} \in [-R, R]^n$ ,*
3. *there is  $\boldsymbol{\alpha} \in [0, 1]^t$  such that  $F(\mathbf{b}, \mathbf{c}, \boldsymbol{\alpha}) = (\mathbf{x}, \boldsymbol{\alpha})$ ,*

*then  $\mathbf{x}$  is an optimal solution to the LP (15).*

**Theorem 9.23.** *In extensive-form potential games, the problem of computing a  $\delta$ -near normal-form proper equilibrium is in  $\text{FIXP}_a$ , that is, it can be expressed as the problem of finding a near-fixed point of an arithmetic circuit.*

*Proof.* Fix some  $\hat{\epsilon} > 0$ . We first consider the problem of creating a circuit for computing a perturbed best response. The computation of the spanning set (Theorem 9.18) consist of  $d := |\Sigma|$  linear optimizations over a symbolic utility  $\mathbf{u}$  whose degree is at most  $dn$ , one per sequence  $\sigma_i \in \Sigma_i$ , as per (13). Each optimization is over a polytope whose constraint matrix is fixed ahead of time, *i.e.*, only the right-hand side of the constraint  $\mathbf{y}_{\Sigma_j}^\sigma = \mathbf{y}_{\Sigma_j}^{p_j}$  is input-dependent, which is allowable in Theorem 9.22.

This creates an optimal spanning set  $\{\mathbf{y}^\sigma : \sigma \in \Sigma\}$ . Let  $\mathbf{Y} \in \mathbb{R}^{\Sigma \times \Sigma}$  be the matrix whose column is  $\mathbf{y}^\sigma$ . We now construct the perturbed best response via the formula in Theorem 9.15. We solve the linear program

$$\max_{\mathbf{Q} \in \mathbb{R}^{\Sigma \times \Sigma}} \mathbf{u}^\top \mathbf{V} \mathbf{Q} \hat{\epsilon} \quad \text{s.t.} \quad \mathbf{Q} \mathbf{1} = \mathbf{1}, \quad \mathbf{1}^\top \mathbf{Q} = \mathbf{1}, \quad \mathbf{Q} \geq \mathbf{0}$$

where the constraints on  $\mathbf{Q}$  enforce that  $\mathbf{Q}$  is a doubly stochastic matrix, *i.e.*, a convex combination of permutation matrices; and  $\hat{\epsilon} = (1 - \sum_{i=1}^{d-1} \hat{\epsilon}^i, \hat{\epsilon}^2, \dots, \hat{\epsilon}^d)$ . That is, an optimal solution to the above program permutes the columns of  $\mathbf{V}$  to optimize the perturbed best-response value. Then  $\mathbf{x} := \mathbf{V} \mathbf{Q} \hat{\epsilon}$  satisfies Theorem 9.15.<sup>12</sup> Thus, the problem of computing a best response—and hence also a Nash equilibrium—of the game  $\Gamma^{(\hat{\epsilon})}$  can be expressed as the fixed point of some arithmetic circuit. Finally, by Lemma 9.20, to compute a  $\delta$ -near equilibrium, it suffices to take  $\epsilon = \delta^{2^{\text{poly}(d, n)}}$ , which can be done using repeated squaring.  $\square$

## 10 Price of anarchy of perfect and proper equilibria

Perfect equilibria—and refinements thereof—constitute a subset of Nash equilibria; from a worst-case perspective, this means that they can only improve the solution quality. One can quantify this through the *price of anarchy* framework [46]. Namely, for a game  $\Gamma$  with nonnegative utilities and a solution concept  $\text{SolCon}$ , we define

$$\text{PoA}_{\text{SolCon}} := \frac{\max_{\mathbf{a} \in \mathcal{A}} \text{SW}(\mathbf{a})}{\inf_{\mathbf{x} \in \text{SolCon}(\Gamma)} \text{SW}(\mathbf{x})}. \quad (16)$$

Above,  $\text{SW}(\mathbf{a}) = \sum_{i=1}^n u_i(\mathbf{a})$  denotes the (utilitarian) social welfare and  $\text{SW}(\mathbf{x}) = \mathbb{E}_{\mathbf{a} \sim \mathbf{x}} \text{SW}(\mathbf{a})$ . It is further assumed that the set of outcomes of  $\Gamma$  per the solution concept  $\text{SolCon}$ , denoted by

<sup>12</sup>If  $\mathbf{Q}$  is not a permutation matrix but rather a doubly stochastic matrix with some entries in  $(0, 1)$ , this is still fine, because the proof of Theorem 9.15 does not break if probability is exchanged between spanning set elements  $\mathbf{x}_i, \mathbf{x}_j$  with the same utility.

$\text{SolCon}(\Gamma)$ , is such that  $\text{SolCon}(\Gamma) \neq \emptyset$ . We only treat the non-trivial case in which not all utilities are zero; as a result, if  $\inf_{\mathbf{x} \in \text{SolCon}(\Gamma)} \text{SW}(\mathbf{x}) = 0$ , then  $\text{PoA}_{\text{SolCon}} = \infty$ . In what follows,  $\text{NashEq}(\Gamma)$ ,  $\text{PerfEq}(\Gamma)$ , and  $\text{PropEq}(\Gamma)$  denote the set of Nash, perfect, and proper equilibria of  $\Gamma$ , respectively.

With this vocabulary at hand, we first point out that the simple example of [Figure 1](#) already establishes a stark separation between Nash and perfect equilibria:

**Proposition 10.1.** *There exists a  $2 \times 2$  identical-interest game such that  $\text{PoA}_{\text{NashEq}} = \infty$ , whereas  $\text{PoA}_{\text{PerfEq}} = 1$ .*

One criticism of the example of [Figure 1](#) is that the worst-case Nash equilibrium is brittle: gradient descent initialized at random would converge to the optimal equilibrium almost surely. But, as we pointed out in our introduction, an analogous dichotomy persists even if one initializes gradient descent at random ([Figure 4](#)), which can be formalized through the *average price of anarchy* notion introduced by Sakos et al. [[66](#)].

**Lower bounds for polynomial congestion games** While there has been a tremendous amount of interest in algorithmic game theory toward characterizing the price of anarchy with respect to Nash equilibria, less is known about equilibrium refinements from that perspective, with some notable exceptions highlighted in [Section 1.3](#).

Here, we make use of the elegant framework of Roughgarden [[64](#)] to establish lower bounds for both perfect and proper equilibria. As a proof of concept, we focus on congestion games, perhaps the most well-studied class of games from the perspective of price of anarchy; for consistency with prior work, in what follows, we switch to cost minimization as opposed to welfare maximization. The basic idea of Roughgarden [[64](#)] is that one can harness hardness results for approximating the optimal social cost to obtain lower bounds on the price of anarchy of Nash equilibria. For our purposes, we leverage our previous results on the complexity of perfect and proper equilibria to establish similar lower bounds. On the hardness of approximation front, we rely on the tight lower bounds of Paccagnan and Gairing [[58](#)], stated in [Theorem 10.2](#).

We first recall that in a congestion game we are given a finite set of resources  $\mathcal{R}$ . Players are to select a subset of resources that they intent to use. We denote by  $\mathcal{A}_i \subseteq 2^{\mathcal{R}}$  the action set of player  $i \in [n]$ . The cost of using a resource  $r \in \mathcal{R}$ ,  $\ell_r : \mathbb{N} \rightarrow \mathbb{R}_{>0}$ , depends solely on the number of other players simultaneously using that resource. We denote by  $n_r(S)$  the number of players using resource  $r$  under the joint action profile  $S \in \mathcal{A}_1 \times \dots \times \mathcal{A}_n$ . The social cost is defined as

$$\text{SC}(S) := \sum_{i=1}^n \sum_{r \in \mathcal{R}} \ell_r(n_r(S)).$$

A canonical example of a congestion game is a *network congestion game*, wherein resources correspond to edges of a graph and players are to select a path. We will restrict our attention to congestion games such that the size of each action set  $|\mathcal{A}_i|$  is polynomial in the representation of the congestion game, so that a best response can be directly computed in polynomial time for both perfect and proper equilibria.

**Theorem 10.2** ([[58](#)]). *Let  $\ell : \mathbb{N} \rightarrow \mathbb{R}_{>0}$  be a nondecreasing and semi-convex function, and*

$$\rho_\ell := \sup_{\nu \in \mathbb{N}} \frac{\mathbb{E}_{v \sim \text{Poi}(\nu)}[v\ell(v)]}{\nu\ell(\nu)}.$$

*For any  $\epsilon > 0$ , it is NP-hard to distinguish between  $\min_{\mathbf{a} \in \mathcal{A}} \text{SC}(\mathbf{a}) \leq p(|\mathcal{I}|)$  and  $\min_{\mathbf{a} \in \mathcal{A}} \text{SC}(\mathbf{a}) \geq (\rho_\ell - \epsilon)p(|\mathcal{I}|)$ , where  $p(|\mathcal{I}|)$  is some polynomial in the size of the input  $|\mathcal{I}|$ .*

Above, we remind that a function  $\ell : \mathbb{N} \rightarrow \mathbb{R}_{>0}$  is called semi-convex if  $\nu\ell(\nu)$  is convex; that is,  $(\nu+1)\ell(\nu+1) - \nu\ell(\nu) \geq \nu\ell(\nu) - (\nu-1)\ell(\nu-1)$  for all  $\nu \geq 2$ . Further,  $\text{Poi}(\nu)$  denotes the Poisson distribution with parameter  $\nu \in \mathbb{N}$ .

When  $\mathcal{L}_d$  contains all polynomials of degree  $d$  with nonnegative coefficients, it follows that  $\sup_{\ell \in \mathcal{L}} \rho_\ell$  is the  $(d+1)$ -Bell number, which we denote by  $\mathcal{B}(d+1)$ ; it follows from existing asymptotic bounds that  $\mathcal{B}(d) \gtrsim d^{d^{1-\epsilon}}$  for any  $\epsilon > 0$  [20]. By virtue of our PLS membership for both perfect (Theorem 5.6) and proper equilibria (Theorem 9.2), we can now show the following.

**Theorem 10.3.** *Consider the class of congestion games in which  $\ell_r \in \mathcal{L}_d$  for every  $r \in \mathcal{R}$ . If  $\text{NP} \neq \text{coNP}$ , we have  $\text{PoA}_{\text{PerfEq}} \geq \mathcal{B}(d+1) - \epsilon$  for any  $\epsilon > 0$ . Similarly, if  $\text{NP} \neq \text{coNP}$ , we have  $\text{PoA}_{\text{PropEq}} \geq \mathcal{B}(d+1) - \epsilon$  for any  $\epsilon > 0$ .*

In fact, those lower bounds hold even for *pure* perfect and *pure* proper equilibria. Theorem 10.3 follows because in the contrary case, a problem in PLS would also be NP-hard (by Theorem 10.2), which is precluded subject to  $\text{NP} \neq \text{coNP}$  [44]. It is likely that the lower bound in Theorem 10.3 can be pushed further through an explicit construction, as is the case for Nash equilibria; indeed, for Nash equilibria there is a certain gap between its price of anarchy and the best approximation ratio that can be achieved through a polynomial-time algorithm [58]. We leave this question for future research.

## 11 Conclusions and future research

Our main contribution in this paper was to characterize the complexity of equilibrium refinements in potential games. Our results span various game representations—ranging from normal-form games to general polytope games—and equilibrium refinements, primarily centered on (normal-form) perfect and (normal-form) proper equilibria. Taken together, our results paint a comprehensive picture of the complexity landscape.

Moreover, some of our technical results have important implications well beyond equilibrium refinements. Most notably, we showed that a (perturbed) proper best response can be computed in polynomial time in extensive-form games, resolving the complexity of normal-form proper equilibria. We also significantly strengthened and simplified a seminal result of Etessami and Yannakakis [24]: even in three-player potential games, we showed that a doubly exponentially small precision is necessary to be geometrically close to an equilibrium.

There are several interesting avenues for future research. First, the complexity of computing a strong approximation to a *mixed* Nash equilibrium in potential games remains an outstanding open problem.

**Question 11.1.** *Is computing an exact mixed Nash equilibrium in potential games  $\text{FIXP} \cap \text{PLS}$ -complete? Is the strong approximation thereof  $\text{FIXP}_a \cap \text{PLS}$ -complete.*

It is clearly in both  $\text{FIXP}$  and  $\text{PLS}$ ,<sup>13</sup> but nothing is known beyond that. In light of our results, a  $\text{FIXP} \cap \text{PLS}$ -hardness result would immediately resolve the complexity of both perfect and proper equilibria. To our knowledge, the relation between  $\text{FIXP}$  and  $\text{PLS}$  is entirely unexplored; what is the right analog of  $\text{FIXP}$  for problems in  $\text{CLS}$ ?

It would also be interesting to expand the positive results we obtained in Section 8. For example, what can be said about matroid congestion games without the symmetry assumption? And can those results for perfect equilibria be extended to proper equilibria?

<sup>13</sup>This is so when one adopts the definition of  $\text{PLS}$  we use in this paper, which does not rest on being able to verify any possible purported solution; otherwise, verifying whether a point is close to an exact mixed Nash equilibrium is a hard problem.

Finally, are there further implications of the connection we made between strongly polynomial-time algorithms and perturbed optimization ([Theorem 8.6](#))? We believe there is fertile ground in employing our techniques to other optimization problems that do not necessarily have a game-theoretic flavor. Our framework can be used to refine the solution quality in problems such as shortest paths and maximum flow by accounting for some uncertainty regarding the underlying costs. Naturally, there has been much work in that direction, but the viewpoint stemming from equilibrium refinements seems to be new.

## Acknowledgments

We are indebted to Ratip Emin Berker for numerous insightful discussions throughout this project. I.A. thanks Ioannis Panageas, Yingming Yan, and Alexandros Hollender for many discussions pertaining to [Question 11.1](#). K.F. thanks Vince Conitzer, whose course on “Foundations of Cooperative AI” inspired the initial ideas that led to this work. T.S. is supported by the Vanevar Bush Faculty Fellowship ONR N00014-23-1-2876, National Science Foundation grants RI-2312342 and RI-1901403, ARO award W911NF2210266, and NIH award A240108S001. Emanuel Tewolde thanks the Cooperative AI Foundation and PhD Fellowship, Macroscopic Ventures and Jaan Tallinn’s donor-advised fund at Founders Pledge for financial support.

## References

- [1] Heiner Ackermann, Heiko Röglin, and Berthold Vöcking. On the impact of combinatorial structure on congestion games. *Journal of the ACM*, 2008.
- [2] Ioannis Anagnostides, Ioannis Panageas, Tuomas Sandholm, and Jingming Yan. The complexity of symmetric equilibria in min-max optimization and team zero-sum games. In *Neural Information Processing Systems (NeurIPS)*, 2025.
- [3] Robert M. Anderson. “Almost” implies “near”. *Transactions of the American Mathematical Society*, 296(1):229–237, 1986.
- [4] Elliot Anshelevich, Anirban Dasgupta, Jon M. Kleinberg, Éva Tardos, Tom Wexler, and Tim Roughgarden. The price of stability for network design with fair cost allocation. *SIAM Journal on Computing*, 38(4):1602–1623, 2008.
- [5] Yakov Babichenko and Aviad Rubinstein. Settling the complexity of Nash equilibrium in congestion games. In *Symposium on Theory of Computing (STOC)*, 2021.
- [6] Maria-Florina Balcan, Avrim Blum, and Yishay Mansour. Circumventing the price of anarchy: Leading dynamics to good behavior. *SIAM Journal on Computing*, 42(1):230–264, 2013.
- [7] Maria-Florina Balcan, Avrim Blum, and Yishay Mansour. The price of uncertainty. *Transactions on Economics and Computation*, 1(3):15:1–15:29, 2013.
- [8] Maria-Florina Balcan, Avrim Blum, and Shang-Tse Chen. Diversified strategies for mitigating adversarial attacks in multiagent systems. In *Autonomous Agents and Multi-Agent Systems*, 2018.
- [9] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. *Algorithms in Real Algebraic Geometry*. Springer, 2008.

- [10] Martino Bernasconi, Alberto Marchesi, and Francesco Trovò. Learning extensive-form perfect equilibria in two-player zero-sum sequential games. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2024.
- [11] Lawrence Blume, Adam Brandenburger, and Eddie Dekel. Lexicographic probabilities and equilibrium refinements. *Econometrica: Journal of the Econometric Society*, pages 81–98, 1991.
- [12] Christian Borgs, Jennifer Chayes, Nicole Immorlica, Adam Tauman Kalai, Vahab Mirrokni, and Christos Papadimitriou. The myth of the folk theorem. In *Symposium on Theory of Computing (STOC)*, pages 365–372, 2008.
- [13] Ioannis Caragiannis, Angelo Fanelli, Nick Gravin, and Alexander Skopalik. Efficient computation of approximate pure Nash equilibria in congestion games. In *Symposium on Foundations of Computer Science (FOCS)*, 2011.
- [14] Ioannis Caragiannis, Angelo Fanelli, and Nick Gravin. Short sequences of improvement moves lead to approximate equilibria in constraint satisfaction games. *Algorithmica*, 77:1143–1158, 2017.
- [15] Oriol Carbonell-Nicolau and Richard P McLean. Refinements of Nash equilibrium in potential games. *Theoretical Economics*, 9(3):555–582, 2014.
- [16] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Settling the complexity of computing two-player Nash equilibria. *Journal of the ACM*, 2009.
- [17] Chuangyin Dang, Xiaoxuan Meng, and Dolf Talman. An interior-point path-following method for computing a perfect stationary point of a polynomial mapping on a polytope, 2015.
- [18] Constantinos Daskalakis and Christos Papadimitriou. Continuous local search. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2011.
- [19] Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. The complexity of computing a Nash equilibrium. *SIAM Journal on Computing*, 39(1), 2009.
- [20] Nicolaas Govert De Bruijn. *Asymptotic methods in analysis*, volume 4. Courier Corporation, 1981.
- [21] Jasper de Jong, Max Klimm, and Marc Uetz. Efficiency of equilibria in uniform matroid congestion games. In *International Symposium on Algorithmic Game Theory (SAGT)*, 2016.
- [22] Jack Edmonds. Matroids and the greedy algorithm. *Mathematical programming*, pages 127–136, 1971.
- [23] Kousha Etessami. The complexity of computing a (quasi-)perfect equilibrium for an  $n$ -player extensive form game. *Games and Economic Behavior*, 125:107–140, 2021.
- [24] Kousha Etessami and Mihalis Yannakakis. On the complexity of Nash equilibria and other fixed points (extended abstract). In *Symposium on Foundations of Computer Science (FOCS)*, 2007.
- [25] Kousha Etessami, Kristoffer Arnsfelt Hansen, Peter Bro Miltersen, and Troels Bjerre Sørensen. The complexity of approximating a trembling hand perfect equilibrium of a multi-player game in strategic form. In *International Symposium on Algorithmic Game Theory (SAGT)*, 2014.



- [26] Alex Fabrikant, Christos H. Papadimitriou, and Kunal Talwar. The complexity of pure Nash equilibria. In *Symposium on Theory of Computing (STOC)*, 2004.
- [27] Gabriele Farina and Nicola Gatti. Extensive-form perfect equilibrium computation in two-player games. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2017.
- [28] Gabriele Farina, Nicola Gatti, and Tuomas Sandholm. Practical exact algorithm for trembling-hand equilibrium refinements in games. In *Neural Information Processing Systems (NeurIPS)*, 2018.
- [29] John Fearnley, Paul Goldberg, Alexandros Hollender, and Rahul Savani. The complexity of gradient descent:  $\text{CLS} = \text{PPAD} \cap \text{PLS}$ . *Journal of the ACM*, 70(1):7:1–7:74, 2023.
- [30] John Fearnley, Paul W. Goldberg, Alexandros Hollender, and Rahul Savani. The complexity of computing KKT solutions of quadratic programs. In *Symposium on Theory of Computing (STOC)*, 2024.
- [31] Aris Filos-Ratsikas, Kristoffer Arnsfelt Hansen, Kasper Høgh, and Alexandros Hollender. Fixp-membership via convex optimization: Games, cakes, and markets. In *Symposium on Foundations of Computer Science (FOCS)*, 2021.
- [32] Aris Filos-Ratsikas, Kristoffer Arnsfelt Hansen, Kasper Høgh, and Alexandros Hollender. PPAD-membership for problems with exact rational solutions: A general approach via convex optimization. In *Symposium on Theory of Computing (STOC)*, 2024.
- [33] Nicola Gatti, Mario Gilli, and Alberto Marchesi. A characterization of quasi-perfect equilibria. *Games and Economic Behavior*, 2020.
- [34] Ian Gemp, Kevin R. McKee, Richard Everett, Edgar A. Duéñez-Guzmán, Yoram Bachrach, David Balduzzi, and Andrea Tacchetti. D3C: reducing the price of anarchy in multi-agent learning. In *Autonomous Agents and Multi-Agent Systems*, 2022.
- [35] M. Groetschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimizations*. Springer-Verlag, 1993.
- [36] Kristoffer Arnsfelt Hansen and Troels Bjerre Lund. Computational complexity of proper equilibrium. In *ACM Conference on Economics and Computation (EC)*, 2018.
- [37] Kristoffer Arnsfelt Hansen and Troels Bjerre Lund. Computational complexity of computing a quasi-proper equilibrium. In *Fundamentals of Computation Theory (FCT)*, 2021.
- [38] Kristoffer Arnsfelt Hansen, Peter Bro Miltersen, and Troels Bjerre Sørensen. The computational complexity of trembling hand perfection and other equilibrium refinements. In *International Symposium on Algorithmic Game Theory (SAGT)*, 2010.
- [39] Bainian Hao and Carla Michini. Price of anarchy in paving matroid congestion games. In *International Symposium on Algorithmic Game Theory (SAGT)*, 2024.
- [40] Tobias Harks, Tim Oosterwijk, and Tjark Vredeveld. A logarithmic approximation for polymatroid congestion games. *Operations Research Letters*, 44(6):712–717, 2016.
- [41] Alexandros Hollender, Gilbert Maystre, and Sai Ganesh Nagarajan. The complexity of two-team polymatrix games with independent adversaries. In *International Conference on Learning Representations (ICLR)*, 2025.

- [42] Samuel Ieong, Robert McGrew, Eugene Nudelman, Yoav Shoham, and Qixiang Sun. Fast and compact: A simple class of congestion games. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2005.
- [43] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, and Michael I. Jordan. How to escape saddle points efficiently. In *International Conference on Machine Learning (ICML)*, 2017.
- [44] David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37(1):79–100, 1988.
- [45] Elon Kohlberg and Jean-Francois Mertens. On the strategic stability of equilibria. *Econometrica*, 54:1003–1037, 1986.
- [46] Elias Koutsoupias and Christos Papadimitriou. Worst-case equilibria. In *Symposium on Theoretical Aspects in Computer Science*, 1999.
- [47] David M. Kreps and Robert Wilson. Sequential equilibria. *Econometrica*, 50(4):863–894, 1982.
- [48] Jason D. Lee, Ioannis Panageas, Georgios Piliouras, Max Simchowitz, Michael I. Jordan, and Benjamin Recht. First-order methods almost always avoid strict saddle points. *Mathematical Programming*, 176(1-2):311–337, 2019.
- [49] Renato Paes Leme, Vasilis Syrgkanis, and Éva Tardos. The curse of simultaneity. In Shafi Goldwasser, editor, *Innovations in Theoretical Computer Science*, 2012.
- [50] J.-F. Mertens. Two examples of strategic equilibrium. *Games and Economic Behavior*, 8(2):378–388, 1995.
- [51] Peter Bro Miltersen and Troels Bjerre Sørensen. Computing proper equilibria of zero-sum games. In *Computers and Games*, 2006.
- [52] Peter Bro Miltersen and Troels Bjerre Sørensen. Fast algorithms for finding proper strategies in game trees. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2008.
- [53] Peter Bro Miltersen and Troels Bjerre Sørensen. Computing a quasi-perfect equilibrium of a two-player game. *Economic Theory*, 42(1), 2010.
- [54] Dov Monderer and Lloyd S Shapley. Potential games. *Games and Economic Behavior*, 14(1):124–143, 1996.
- [55] Roger B. Myerson. Refinements of the Nash equilibrium concept. *International Journal of Game Theory*, 15:133–154, 1978.
- [56] John Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36:48–49, 1950.
- [57] Maher Nouiehed, Jason D Lee, and Meisam Razaviyayn. Convergence to second-order stationarity for constrained non-convex optimization. *arXiv:1810.02024*, 2018.
- [58] Dario Paccagnan and Martin Gairing. In congestion games, taxes achieve optimal approximation. *Operations Research*, 72(3):966–982, 2024.

- [59] Christos Papadimitriou and Tim Roughgarden. Computing equilibria in multi-player games. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 82–91, Vancouver, BC, Canada, 2005. SIAM.
- [60] Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and system Sciences*, 48(3):498–532, 1994.
- [61] Christos H. Papadimitriou and Tim Roughgarden. Computing correlated equilibria in multi-player games. *Journal of the ACM*, 55(3):14:1–14:29, 2008.
- [62] I. Romanovskii. Reduction of a game with complete memory to a matrix game. *Soviet Mathematics*, 3, 1962.
- [63] Robert W Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.
- [64] Tim Roughgarden. Barriers to near-optimal equilibria. In *Symposium on Foundations of Computer Science (FOCS)*, 2014.
- [65] Tim Roughgarden and Éva Tardos. How bad is selfish routing? In *Symposium on Foundations of Computer Science (FOCS)*, November 2000.
- [66] Iosif Sakos, Stefanos Leonardos, Stelios Andrew Stavroulakis, Will Overman, Ioannis Panageas, and Georgios Piliouras. Beating price of anarchy and gradient descent without regret in potential games. In *International Conference on Learning Representations (ICLR)*, 2024.
- [67] Alejandro A. Schäffer and Mihalis Yannakakis. Simple local search problems that are hard to solve. *SIAM Journal on Computing*, 20(1):56–87, 1991.
- [68] Thomas C. Schelling. *The Strategy of Conflict*. Havard University Press, 1960.
- [69] Reinhard Selten. Reexamination of the perfectness concept for equilibrium points in extensive games. *International journal of game theory*, 1975.
- [70] Yoav Shoham and Kevin Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.
- [71] Troels Bjerre Sørensen. Computing a proper equilibrium of a bimatrix game. In *ACM Conference on Economics and Computation (EC)*, 2012.
- [72] Eric van Damme. A relation between perfect equilibria in extensive form games and proper equilibria in normal form games. *International Journal of Game Theory*, 1984.
- [73] Eric van Damme. *Stability and perfection of Nash equilibria*, volume 339. Springer, 1991.
- [74] Gerard van der Laan, Dolf Talman, and Zaifu Yang. Existence and approximation of robust solutions of variational inequality problems over polytopes. *SIAM Journal on Control and Optimization*, 37(2):333–352, 1999.
- [75] Bernhard von Stengel. Efficient computation of behavior strategies. *Games and Economic Behavior*, 14(2):220–246, 1996.
- [76] Bernhard Von Stengel, Antoon Van Den Elzen, and Dolf Talman. Computing normal form perfect equilibria for extensive two-person games. *Econometrica*, 70(2):693–715, 2002.

- [77] Weiran Wang and Miguel Á. Carreira-Perpiñán. Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application. *CoRR*, arXiv:1309.1541, 2013.
- [78] Mihalis Yannakakis. *Computational Complexity*, pages 19–56. Princeton University Press, 2003.

## A Equilibrium refinements for games in extensive form

### A.1 Notation and background

Deciding which strategy each player should play in an extensive-form game of perfect recall can be encoded as a *tree-form decision problem*. A tree-form decision problem for any player  $i \in [n]$  is represented as a rooted tree, where each root-to-leaf path alternates between two types of nodes: *decision points*, denoted as  $j \in \mathcal{J}_i$  (which correspond to the infosets of the game), and *observation points* (or *sequences*), denoted as  $\sigma \in \Sigma_i$ . The root node of the tree, denoted as  $\emptyset \in \Sigma_i$ , is always an observation point. The edges of decision points are called *actions*, and the player must choose one of them before continuing. We denote the set of actions at decision point  $j$  as  $\mathcal{A}_j$ . At observation points, the labels are called *observations*, one of which the player observes. We let  $d_i = |\Sigma_i|$  denote the number of sequences in the tree. The set of decision points that follow an observation point  $\sigma \in \Sigma_i$ , is denoted as  $C_\sigma$ . An observation point is denoted as  $ja$ , where  $j$  is the parent decision point and  $a$  is the action that connects the two points. For any node  $s$ , we denote its parent node as  $p_s$ . Finally, we will use  $\succeq$  to denote the partial order induced by the tree, *e.g.*,  $\emptyset \prec j$  for every decision point  $j$ .

A *behavioral strategy*  $\beta_i$  for player  $i$  is a collection of distributions one for each decision point. That is,  $\beta_i(\cdot|j) \in \Delta(\mathcal{A}_j)$  denotes the probability distribution from which player  $i$  samples their strategy. The *sequence form* of player  $i \in [n]$  is a vector  $\mathbf{x}_i \in \mathbb{R}^{d_i}$ , where  $\mathbf{x}_i(\sigma)$  denotes the product of probabilities on the path to sequence  $\sigma \in \Sigma_i$ . We write  $\mathcal{X}_i$  to denote the set of all possible sequence form strategies for player  $i$ . Romanovskii [62] and von Stengel [75] showed that  $\mathcal{X}_i$  for player  $i$  can be described by a system of linear constraints

$$\mathbf{x}_i[\emptyset] = 1, \quad \mathbf{x}_i(p_j) = \sum_{a \in \mathcal{A}_j} \mathbf{x}_i[ja] \quad \forall j \in \mathcal{J}_i, \quad \mathbf{x}_i[\sigma] \geq 0 \quad \forall \sigma \in \Sigma_i.$$

For our purposes, an *extensive-form game* is a concise polytope game where each player’s strategy set  $\mathcal{X}_i$  is a sequence-form strategy set.<sup>14</sup>

### A.2 Computing extensive-form perfect equilibria

**Definition A.1.** ( $\epsilon$ -perturbed EFG for extensive-form perfect equilibria) Let  $\Gamma$  be an  $n$ -player extensive-form game and  $\mathcal{X}_i$  be the strategy set of player  $i$ . For some  $\epsilon > 0$ , we let  $\Gamma^{(\epsilon)}$  denote the perturbed sequence form game with the same structure as  $\Gamma$  with the additional following restricted set of players’ strategies. For player  $i \in [n]$  we the perturbed sequence-form mixed strategies as

$$\mathcal{X}_i^{(\epsilon)} = \{\mathbf{x}_i \in \mathcal{X}_i : \mathbf{x}_i[ja] \geq \epsilon \cdot \mathbf{x}_i[p_j] \quad \forall \sigma \in \Sigma_i\}.$$

---

<sup>14</sup>Technically, this class of games is actually slightly more expressive than extensive-form games: extensive-form games would be equivalent to the class of games in which, additionally, utility functions are given explicitly as a (weighted) sum of monomials.

That is, in a perturbation for extensive-form perfect equilibria, every action  $a$  is played at every decision point  $j$  with probability at least  $\epsilon$ .

An extensive-form perfect equilibrium is now, as usual, defined as the limit of Nash equilibria of the above kind of  $\epsilon$ -perturbed games. The following can be easily shown by performing a bottom-up pass through the sequence-form tree:

**Theorem A.2.** *Best responses in  $\mathcal{X}_i^{(\epsilon)}$  can be computed in time polynomial in  $d$  and the representation size of the utility vector. As a consequence, computing an EFPE of a potential game is in PLS.*

In extensive-form games, one can also define notions of equilibrium refinements by viewing the game as a polytope game with strategy set  $\mathcal{X}_i$  for each player, and then applying the definitions in the main body for perfect and proper equilibria in polytope games. These give the notions, respectively, of normal-form perfect and normal-form proper equilibria. In extensive-form games, it turns out that extensive-form perfect equilibria are *incomparable* to both of these notions [50]; therefore, Theorem A.2 is not implied directly by any of the other results in this paper.

Similarly, a symbolic best-response for the perturbation that arises under QPEs can also be computed efficiently, implying the following result.<sup>15</sup>

**Theorem A.3.** *Computing a QPE of a potential game is in PLS.*

## B Quasi-perfect versus normal-form proper equilibria

Here, we elaborate on the difference between QPEs and normal-form proper equilibria. As stated above, QPEs are a superset of normal-form proper equilibria. To see that the inclusion can be strict, it suffices to consider normal-form games, where QPEs are equivalent to normal-form perfect equilibria. Indeed, this implies that Figure 2 is already an example of a potential game in which QPEs and proper equilibria do not coincide.

For completeness, we describe an extensive-form game, due to Miltersen and Sørensen [52], in which there is a non-sensible quasi-perfect equilibrium, whereas the sensible equilibrium is the unique normal-form proper equilibrium of the game. The game, referred to as “matching pennies on Christmas day” by Miltersen and Sørensen [52], is given in Figure 7. It is a variation of the usual matching pennies games. Both Player 1 and Player 2 select either Heads or Tails. Player 2 wins \$1 if it correctly guesses the action of Player 1. It is a zero-sum game, so Player 2 has diametrically opposing utilities. On top of that, Player 1 has the option of offering a gift of \$1 dollar to Player 2 (Gift in Figure 7). Modulo symmetries, there are 4 possible outcomes:

- Player 2 guesses correctly and Player 1 offers the gift—Player 2 receives \$2;
- Player 2 guesses correctly and Player 1 does not offer the gift—Player 2 receives \$1;
- Player 2 guesses incorrectly and Player 1 offers the gift—Player 2 receives \$1; and
- Player 2 guesses incorrectly and Player 1 does not offer the gift—Player 2 receives \$0.

<sup>15</sup>By virtue of Theorem 9.19 and the fact that the set of normal-form proper equilibria is a subset of QPEs, PLS membership for computing QPEs in potential extensive-form games follows. What we want to point out here is that one does not have to go through the complicated machinery we developed to obtain Theorem 9.19 to argue about QPEs.

The only sensible equilibrium in this game is for Player 1 to refrain from offering the gift and each player to play **Heads** or **Tails** with equal probability. This is indeed the unique normal-form proper equilibrium of the game [52]. On the other hand, the following pair of strategies can be shown to be a quasi-perfect equilibrium:

- Player 1 plays **Heads** with probability  $\frac{1}{2}$  and **Tails** with probability  $\frac{1}{2}$ , and always plays **NoGift**.
- Player 2 plays **Heads** with probability  $\frac{1}{2}$  and **Tails** with probability  $\frac{1}{2}$  at the information set upon observing **NoGift**, and plays **Heads** with probability 1 at the other information set.

At the same, we remind that a normal-form proper equilibrium is a refinement of a quasi-proper equilibrium [72]; that is, from a worst-case perspective, one should always strive for computing a normal-form proper equilibrium. In light of our results, computing a normal-form proper equilibrium is polynomial-time equivalent to computing a quasi-proper equilibrium.

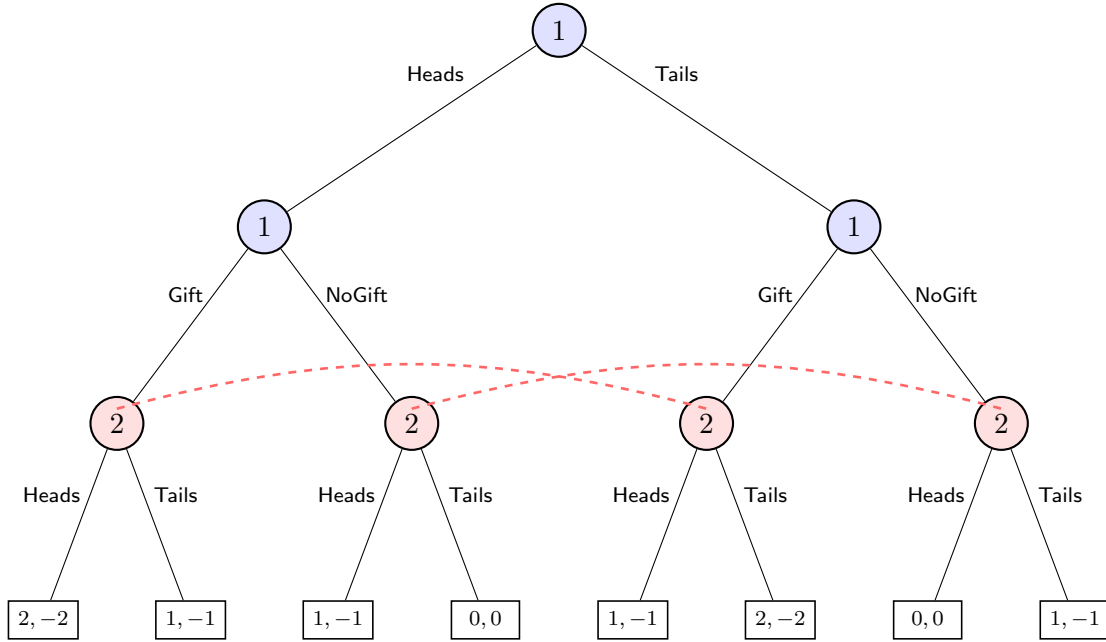


Figure 7: The “matching pennies on Christmas day” game of Miltersen and Sørensen [52].

## C Relations between different equilibrium refinements

For completeness, we provide a basic diagram (Figure 8) illustrating the relations between some basic equilibrium refinements in extensive-form games. All notions appearing in Figure 8 are guaranteed to exist; we do not cover equilibrium concepts that may or may not exist, such as *stable equilibria* [73].



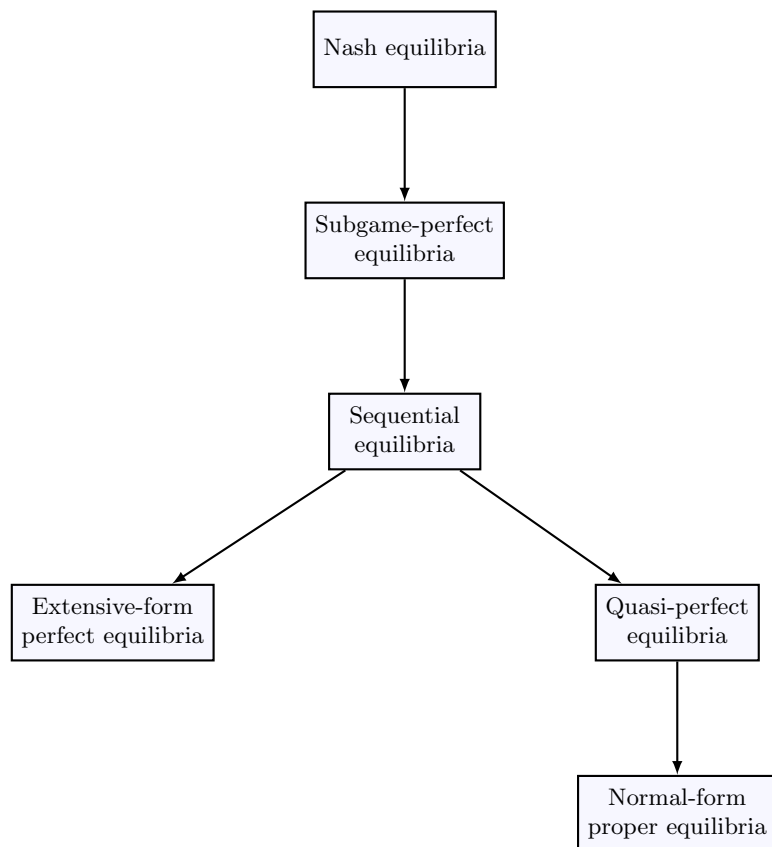


Figure 8: Relation between different equilibrium refinements in extensive-form games. An arrow  $A \rightarrow B$  between two equilibrium concepts A and B means that B refines A; that is, every element in B is also in A.