GAMES and Economic Behavior

New complexity results about Nash equilibria ☆

Vincent Conitzer [a,*], Tuomas Sandholm [b]

[a] *Department of Computer Science and Department of Economics, Duke University, Durham, NC 27708, USA*
[b] *Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA*

Abstract

We provide a single reduction that demonstrates that in normal-form games: (1) it is $\mathcal{NP}$-complete to determine whether Nash equilibria with certain natural properties exist (these results are similar to those obtained by Gilboa and Zemel [Gilboa, I., Zemel, E., 1989. Nash and correlated equilibria: Some complexity considerations. Games Econ. Behav. 1, 80–93]), (2) more significantly, the problems of maximizing certain properties of a Nash equilibrium are inapproximable (unless $\mathcal{P} = \mathcal{NP}$), and (3) it is #$\mathcal{P}$-hard to count the Nash equilibria. We also show that determining whether a pure-strategy Bayes–Nash equilibrium exists in a Bayesian game is $\mathcal{NP}$-complete, and that determining whether a pure-strategy Nash equilibrium exists in a Markov (stochastic) game is $\mathcal{PSPACE}$-hard even if the game is unobserved (and that this remains $\mathcal{NP}$-hard if the game has finite length). All of our hardness results hold even if there are only two players and the game is symmetric.
© 2008 Elsevier Inc. All rights reserved.

* Corresponding author at: Duke University, Department of Computer Science and Department of Economics, Box 90129, LRSC, Durham, NC 27708, USA.

E-mail addresses: conitzer@cs.duke.edu (V. Conitzer), sandholm@cs.cmu.edu (T. Sandholm).

1. Introduction

Game theory provides a normative framework for analyzing strategic interactions. However, in order for anyone to play according to the solutions that it prescribes, these solutions must be *computed*. There are many different ways in which this can happen: a player can consciously solve the game (possibly with the help of a computer[1]); some players can perhaps eyeball the game and find the solution by intuition, even without being aware of the general solution concept; and in some cases, the players can converge to the solution by following simple learning rules. In each case, some computational machinery (respectively, one player's conscious brain, a computer, one player's subconscious brain, or the system consisting of all players together) arrives at the solution using some procedure, or *algorithm*.

Some of the most basic computational problems in game theory concern the computation of Nash equilibria of a finite normal-form game. An example problem is to compute *one* Nash equilibrium—any equilibrium will do. What are good algorithms for solving such a problem? Certainly, we want the algorithm to always return a correct solution. Moreover, we are interested in how *fast* the algorithm returns a solution. Generally, as the size of the game (more generally, the *problem instance*) increases, so does the running time of the algorithm. Whether the algorithm is practical for solving larger instances depends on how rapidly its running time increases. An algorithm is generally considered *efficient* if its running time is at most a polynomial function of the size of the instance (game). There are certainly other properties that one may want the algorithm to have—for example, one may be interested in learning algorithms that are simple enough for people to use—but the algorithm should *at least* be correct and computationally efficient.

The same computational problem may admit both efficient and inefficient algorithms. The theory of *computational complexity* aims to analyze the inherent complexity of the problem itself: how fast is the *fastest* (correct) algorithm for a given problem? $\mathcal{P}$ is the class of problems that admit at least one efficient (polynomial-time) algorithm.[2] While many problems have been proved to be in $\mathcal{P}$ (generally by explicitly giving an algorithm and proving a bound on its running time), it is extremely rare that someone proves that a problem is *not* in $\mathcal{P}$. Instead, to show that a problem is hard, computer scientists generally prove results of the form: "If this problem can be solved efficiently, then so can every member of the class $\mathcal{X}$ of problems." This is usually shown using a *reduction* from one problem to another (we will give more detail on reductions in Section 2). If this has been proven, the problem is said to be $\mathcal{X}$-*hard* (and $\mathcal{X}$-*complete* if, additionally, the problem has also been shown to lie in $\mathcal{X}$). The strength of such a *hardness result* depends on the class $\mathcal{X}$ used. Usually, the class $\mathcal{NP}$ is used (we will describe $\mathcal{NP}$ in more detail in Section 2), and most problems of interest turn out to be either in $\mathcal{P}$ or $\mathcal{NP}$-hard. $\mathcal{NP}$ contains $\mathcal{P}$, and it is generally considered unlikely that $\mathcal{P} = \mathcal{NP}$. Exhibiting a polynomial-time algorithm for an $\mathcal{NP}$-hard problem (thereby showing $\mathcal{P} = \mathcal{NP}$) would constitute a truly major upset: among other things, it would (at least in a theoretical sense, and possibly in a practical

---

[1] The player might also *be* a computer, for example, a poker-playing computer program. Indeed, at least for some variants of poker, the top computer programs are based around computing a game-theoretic solution (usually, a minimax strategy).

[2] To define $\mathcal{P}$ formally (which we will not do here), one must also formally define a model of computation. Fortunately, the class of polynomial-time solvable problems is quite robust to changes in the model of computation. Nevertheless, it is in principle possible that humans have a more powerful computational architecture, and hence that they can solve problems outside $\mathcal{P}$ efficiently.

sense) break current approaches to cryptography, and it would allow a computer to find a proof of any theorem that has a proof of reasonable length.

The problem of finding just one Nash equilibrium of a finite normal-form game is one of the rare interesting problems that have neither been shown to be in $\mathcal{P}$, nor shown to be $\mathcal{NP}$-hard. Not too long ago, it was dubbed "a most fundamental computational problem whose complexity is wide open" and "together with factoring, [...] the most important concrete open question on the boundary of $\mathcal{P}$ today" (Papadimitriou, 2001). A recent sequence of break-through papers (Chen and Deng, 2005, 2006; Daskalakis et al., 2006; Daskalakis and Papadimitriou, 2005) shows that the problem is $\mathcal{PPAD}$-complete, even in the two-player case. [An earlier result shows that the problem is no easier if all utilities are required to be in $\{0, 1\}$ (Abbott et al., 2005).] This gives some evidence that the problem is indeed hard, although not nearly as much is known about the class $\mathcal{PPAD}$ as about $\mathcal{NP}$. The best-known algorithm for finding a Nash equilibrium, the *Lemke–Howson* algorithm (Lemke and Howson, 1964), has been shown to indeed have exponential running time on some instances (and is therefore not a polynomial-time algorithm) (Savani and von Stengel, 2006). More recent algorithms for computing Nash equilibria have focused on guessing which of the players' pure strategies receive positive probability in the equilibrium: after this guess, only a simple linear feasibility problem needs to be solved (Dickhaut and Kaplan, 1991; Porter et al., 2008; Sandholm et al., 2005). These algorithms clearly require exponentially many guesses, and hence exponential time, on some instances, although they are often quite fast in practice.

The interest in the problem of computing a single Nash equilibrium has in large part been driven by the fact that it posed a challenge to complexity theorists. However, from the perspective of a game theorist, this is not always the relevant computational problem. One may, for example, be more interested in what the *best* equilibrium of the game is (for some definition of "best"), or whether a given pure strategy is played in any equilibrium, etc. Gilboa and Zemel (1989) have demonstrated that many of these problems are in fact $\mathcal{NP}$-hard. In Section 3, we continue this line of research by providing a single reduction that proves many results of this type. One important improvement over Gilboa and Zemel's results is that our reduction also shows *inapproximability* results: for example, not even an equilibrium that is *approximately* optimal can be found in polynomial time, unless $\mathcal{P} = \mathcal{NP}$.[3] We also use the reduction to show that counting the number of Nash equilibria (or connected sets of Nash equilibria) is #$\mathcal{P}$-hard.

We proceed to prove some additional results (not based on the main reduction). In Section 4, we consider Bayesian games and show that determining whether a pure-strategy Bayes–Nash equilibrium exists is $\mathcal{NP}$-complete. Finally, in Section 5 we show that determining whether a pure-strategy Nash equilibrium exists in a Markov game is $\mathcal{PSPACE}$-hard even if the game is unobserved, and that this remains $\mathcal{NP}$-hard if the game has finite length. ("Unobserved" means that the players never receive any information about what happened earlier in the game.) All of the hardness results in this paper hold even if there are only two players and the game is symmetric. These results suggest that for sufficiently large games, we cannot expect the players to always play according to these solution concepts, whether they are naïve learning players or sophisticated game theorists armed with state-of-the-art computing equipment.

---

[3] It should be noted that this is different from the problem of computing an *approximate equilibrium* (Daskalakis et al., 2007; Lipton et al., 2003), that is, a strategy profile from which individual players have only a small incentive to deviate. The problems that we consider require an *exact* equilibrium that approximately optimizes some objective.

2. Brief review of reductions and complexity

A key concept in computational complexity theory is that of a *reduction* from one problem A to another problem B. Informally, a reduction maps every instance of computational problem A to a corresponding instance of computational problem B, in such a way that the answer to the former instance can be easily inferred from the answer to the latter instance. Moreover, we require that this mapping is itself easy to compute. If such a reduction exists, then we know that, in a sense, problem A is computationally *at most* as hard to solve as problem B: if we had an efficient algorithm for problem B, then we could use the reduction together with this algorithm to solve problem A.

The most directly useful reductions are those that reduce a problem of interest to a problem for which we already have an efficient algorithm. However, another (backward) use of reductions is to reduce a problem that is known or conjectured to be hard *to* the problem of interest. Such a reduction tells us that we cannot hope to find an efficient algorithm for the problem of interest without (implicitly) also finding such an algorithm for the hard problem.

Certain problems have been shown to be *hard for* a large class of problems (such as $\mathcal{NP}$). Problem A is hard for class $\mathcal{X}$ if any problem in $\mathcal{X}$ can be reduced to problem A. Thus, exhibiting an efficient algorithm for the hard problem entails exhibiting an efficient algorithm for *every* problem in the class. Once one problem A has been shown hard for a class, the task of proving that another problem B is hard for the same class generally becomes much easier: we can do so by reducing A to B. A problem is *complete for* a class if (1) it is hard for the class and (2) the problem is itself in the class.

The class for which problems are most often shown to be hard (or complete) is $\mathcal{NP}$. $\mathcal{NP}$ is the class of all *decision* problems (problems that require a "yes" or "no" answer) such that if the answer to a problem instance is "yes", then there exists a polynomial-sized *certificate* for that instance that proves that the answer is "yes." More precisely, such a certificate can be used to check that the answer is "yes" in polynomial time. The most famous complete problem for $\mathcal{NP}$ is *satisfiability (SAT)*. An instance of satisfiability is given by a Boolean formula in *conjunctive normal form (CNF)*—that is, an "AND" of "ORs" of ground *literals* (Boolean variables and their negations). We are asked whether there exists some assignment of truth values to the variables such that the formula evaluates to *true*. For example, the formula $(x_1 \vee x_2) \wedge (-x_1) \wedge (x_1 \vee -x_2 \vee -x_3)$ is satisfiable by setting x_1 to *false*, x_2 to *true*, and x_3 to *false*. (This assignment is also a certificate for the instance, since it is easy to check that it makes the formula evaluate to *true*.) However, if we add a fourth clause $(x_1 \vee -x_2 \vee x_3)$, then the formula is no longer satisfiable. Satisfiability was the first problem shown to be $\mathcal{NP}$-complete (Cook, 1971), but many other problems have been shown $\mathcal{NP}$-complete since then (often by reducing satisfiability to them).

There are other classes of problems that are even larger[4] than $\mathcal{NP}$, and for which natural problems are sometimes shown to be hard, constituting even stronger evidence that there is no efficient algorithm for the problem. One of these classes is #$\mathcal{P}$, the class of problems counting *how many* solutions a particular instance has. (It is required that solutions can be verified efficiently.) An example problem in #$\mathcal{P}$ is counting how many satisfying assignments a CNF formula has. (This problem is in fact #$\mathcal{P}$-complete; see Valiant, 1979.) Another class is $\mathcal{PSPACE}$, the class of problems that can be solved using only polynomial *space*.

---

[4] Technically, for the classes we mention here, all we know is that they are no smaller than $\mathcal{NP}$—they may in fact coincide with $\mathcal{NP}$. However, exhibiting such a coincidence would again constitute a major upset.

3. The main reduction and its implications

In this section, we give our main reduction, which maps every instance of satisfiability (given by a formula in conjunctive normal form) to a finite symmetric two-player normal-form game. This reduction has no direct complexity implications for the problem of finding one (any) Nash equilibrium. However, it has significant implications for many related problems. Most significantly, it shows that, for many properties, deciding whether an equilibrium *with that property* exists is $\mathcal{NP}$-hard. For example, it shows that deciding whether an equilibrium with social welfare at least k is $\mathcal{NP}$-hard (hence it is also hard to find the social-welfare maximizing equilibrium, arguably a key problem in equilibrium selection). As another example, it shows that deciding whether a certain pure strategy occurs in the support of at least one Nash equilibrium is $\mathcal{NP}$-hard. This has indirect implications for the problem of finding one Nash equilibrium: several recent algorithms for that problem operate by guessing the equilibrium supports and subsequently checking whether the guess is correct (Dickhaut and Kaplan, 1991; Porter et al., 2008; Sandholm et al., 2005). The result above implies that it is $\mathcal{NP}$-hard to determine whether such an algorithm can safely restrict attention to guesses in which a particular pure strategy is included in the support.

These are not the first results of this nature; Gilboa and Zemel provide a number of $\mathcal{NP}$-hardness results in the same spirit (Gilboa and Zemel, 1989). Our reduction demonstrates (sometimes stronger versions of) most of their hardness results, as well as some new ones. Significantly, for the problems that concern an optimization (e.g., maximizing social welfare), we show not only $\mathcal{NP}$-hardness but also *inapproximability*: unless $\mathcal{P} = \mathcal{NP}$, there is no polynomial-time algorithm that always returns a Nash equilibrium that is *close* to obtaining the optimal value. We also use the reduction to show that counting the number of equilibria of a game is #$\mathcal{P}$-hard. (One may argue that it is impossible to have a good overview of all the Nash equilibria of a game if one cannot even count them.)

For completeness, we review the following basic definitions.

Definition 1. In a *normal-form game*, we are given a set of *players A*, and for each player $i \in A$, a *(pure) strategy* set Σ_i and a *utility function* $u_i : \Sigma_1 \times \Sigma_2 \times \ldots \times \Sigma_{|A|} \to \mathbb{R}$.

We will assume throughout that games have finite size.

Definition 2. A *mixed strategy* σ_i for player i is a probability distribution over Σ_i. A special case of a mixed strategy is a *pure strategy*, where all of the probability mass is on one element of Σ_i.

Definition 3. (See Nash, 1950.) Given a normal-form game, a *Nash equilibrium (NE)* is vector of mixed strategies, one for each player i, such that no player has an incentive to deviate from her mixed strategy given that the others do not deviate. That is, for any i and any alternative mixed strategy σ_i', we have $E[u_i(s_1, s_2, \ldots, s_i, \ldots, s_{|A|})] \geqslant E[u_i(s_1, s_2, \ldots, s_i', \ldots, s_{|A|})]$, where each s_j is drawn from σ_j, and s_i' from σ_i'.

It is well known that every finite game has at least one Nash equilibrium (Nash, 1950). We are now ready to present our reduction.[5]

---

[5] The reduction presented here is somewhat different from the reduction given in the earlier (IJCAI-03) version of this work. The reason is that the new reduction presented here implies inapproximability results that the original reduction does not.

Definition 4. Let ϕ be a Boolean formula in conjunctive normal form (representing a SAT instance). Let V be its set of variables (with $|V| = n$), L the set of corresponding literals (a positive and a negative one for each variable[6]), and C its set of clauses. The function $v : L \to V$ gives the variable corresponding to a literal, *e.g.*, $v(x_1) = v(-x_1) = x_1$. We define $G_\epsilon(\phi)$ to be the following finite symmetric 2-player game in normal form. Let $\Sigma = \Sigma_1 = \Sigma_2 = L \cup V \cup C \cup \{f\}$. Let the utility functions be

- $u_1(l^1, l^2) = u_2(l^2, l^1) = n - 1$ for all $l^1, l^2 \in L$ with $l^1 \neq -l^2$;
- $u_1(l, -l) = u_2(-l, l) = n - 4$ for all $l \in L$;
- $u_1(l, x) = u_2(x, l) = n - 4$ for all $l \in L$, $x \in \Sigma - L - \{f\}$;
- $u_1(v, l) = u_2(l, v) = n$ for all $v \in V$, $l \in L$ with $v(l) \neq v$;
- $u_1(v, l) = u_2(l, v) = 0$ for all $v \in V$, $l \in L$ with $v(l) = v$;
- $u_1(v, x) = u_2(x, v) = n - 4$ for all $v \in V$, $x \in \Sigma - L - \{f\}$;
- $u_1(c, l) = u_2(l, c) = n$ for all $c \in C$, $l \in L$ with $l \notin c$;
- $u_1(c, l) = u_2(l, c) = 0$ for all $c \in C$, $l \in L$ with $l \in c$;
- $u_1(c, x) = u_2(x, c) = n - 4$ for all $c \in C$, $x \in \Sigma - L - \{f\}$;
- $u_1(x, f) = u_2(f, x) = 0$ for all $x \in \Sigma - \{f\}$;
- $u_1(f, f) = u_2(f, f) = \epsilon$;
- $u_1(f, x) = u_2(x, f) = n - 1$ for all $x \in \Sigma - \{f\}$.

We will show in Theorem 1 that each satisfying assignment of ϕ corresponds to a Nash equilibrium of $G_\epsilon(\phi)$, and that there is one additional equilibrium. The following example illustrates this.

Example 1. The following table shows the game $G_\epsilon(\phi)$ where $\phi = (x_1 \vee -x_2) \wedge (-x_1 \vee x_2)$.

	x_1	x_2	$+x_1$	$-x_1$	$+x_2$	$-x_2$	$(x_1 \vee -x_2)$	$(-x_1 \vee x_2)$	f
x_1	$-2, -2$	$-2, -2$	$0, -2$	$0, -2$	$2, -2$	$2, -2$	$-2, -2$	$-2, -2$	$0, 1$
x_2	$-2, -2$	$-2, -2$	$2, -2$	$2, -2$	$0, -2$	$0, -2$	$-2, -2$	$-2, -2$	$0, 1$
$+x_1$	$-2, 0$	$-2, 2$	$1, 1$	$-2, -2$	$1, 1$	$1, 1$	$-2, 0$	$-2, 2$	$0, 1$
$-x_1$	$-2, 0$	$-2, 2$	$-2, -2$	$1, 1$	$1, 1$	$1, 1$	$-2, 2$	$-2, 0$	$0, 1$
$+x_2$	$-2, 2$	$-2, 0$	$1, 1$	$1, 1$	$1, 1$	$-2, -2$	$-2, 2$	$-2, 0$	$0, 1$
$-x_2$	$-2, 2$	$-2, 0$	$1, 1$	$1, 1$	$-2, -2$	$1, 1$	$-2, 0$	$-2, 2$	$0, 1$
$(x_1 \vee -x_2)$	$-2, -2$	$-2, -2$	$0, -2$	$2, -2$	$2, -2$	$0, -2$	$-2, -2$	$-2, -2$	$0, 1$
$(-x_1 \vee x_2)$	$-2, -2$	$-2, -2$	$2, -2$	$0, -2$	$0, -2$	$2, -2$	$-2, -2$	$-2, -2$	$0, 1$
f	$1, 0$	$1, 0$	$1, 0$	$1, 0$	$1, 0$	$1, 0$	$1, 0$	$1, 0$	ϵ, ϵ

The only two solutions to the SAT instance defined by ϕ is to either set both variables to *true*, or both to *false*. The only equilibria of the game $G_\epsilon(\phi)$ are those where: (1) both players randomize uniformly over $\{+x_1, +x_2\}$; (2) both players randomize uniformly over $\{-x_1, -x_2\}$; (3) both players play f.

We are now ready to prove the result in general.

---

[6] Thus, if x_i is a variable, $+x_i$ and $-x_i$ are literals. Often, the $+$ is dropped from the positive literal (especially when writing CNF formulas), but it is helpful for distinguishing positive literals from variables.

Theorem 1. *If $(l_1, l_2, \ldots, l_n)$ (where $v(l_i) = x_i$) satisfies ϕ, then there is a Nash equilibrium of $G_\epsilon(\phi)$ where both players play l_i with probability $\frac{1}{n}$, with expected utility $n - 1$ for each player. The only other Nash equilibrium is the one where both players play f, and receive expected utility ϵ each.*

Proof. We first demonstrate that these combinations of mixed strategies indeed do constitute Nash equilibria. If $(l_1, l_2, \ldots, l_n)$ (where $v(l_i) = x_i$) satisfies ϕ and the other player plays l_i with probability $\frac{1}{n}$, playing one of these l_i as well gives utility $n - 1$. On the other hand, playing the negation of one of these l_i gives utility $\frac{1}{n}(n - 4) + \frac{n-1}{n}(n - 1) < n - 1$. Playing some variable v gives utility $\frac{1}{n}(0) + \frac{n-1}{n}(n) = n - 1$ (since one of the l_i that the other player sometimes plays has $v(l_i) = v$). Playing some clause c gives utility at most $\frac{1}{n}(0) + \frac{n-1}{n}(n) = n - 1$ (since at least one of the l_i that the other player sometimes plays occurs in clause c, since the l_i satisfy ϕ). Finally, playing f gives utility $n - 1$. It follows that playing any one of the l_i that the other player sometimes plays is an optimal response, and hence that both players playing each of these l_i with probability $\frac{1}{n}$ is a Nash equilibrium. Clearly, both players playing f is also a Nash equilibrium since playing anything else when the other plays f gives utility 0.

Now we demonstrate that there are no other Nash equilibria. If the other player always plays f, the unique best response is to also play f since playing anything else will give utility 0. Otherwise, given a mixed strategy for the other player, consider a player's expected utility given that the other player does not play f. (That is, the probability distribution over the other player's strategies is proportional to the probability distribution constituted by that player's mixed strategy, except f occurs with probability 0.) If this expected utility is less than $n - 1$, the player is strictly better off playing f (which gives utility $n - 1$ when the other player does not play f, and also performs better than the original strategy when the other player does play f). So this cannot occur in equilibrium.

As we pointed out, here are no Nash equilibria where one player always plays f but the other does not, so suppose both players play f with probability less than one. Consider the expected social welfare ($E[u_1 + u_2]$), given that neither player plays f. It is easily verified that there is no outcome with social welfare greater than $2n - 2$. Also, any outcome in which one player plays an element of V or C has social welfare at most $n - 4 + n < 2n - 2$. It follows that if either player ever plays an element of V or C, the expected social welfare given that neither player plays f is strictly below $2n - 2$. By linearity of expectation it follows that the expected utility of at least one player is strictly below $n - 1$ given that neither player plays f, and by the above reasoning, this player would be strictly better off playing f instead of her randomization over strategies other than f. It follows that no element of V or C is ever played in a Nash equilibrium.

So, we can assume both players only put positive probability on strategies in $L \cup \{f\}$. Then, if the other player puts positive probability on f, playing f is a strictly better response than any element of L (since f does as at least as well against any strategy in L, and strictly better against f). It follows that the only equilibrium where f is ever played is the one where both players always play f.

Now we can assume that both players only put positive probability on elements of L. Suppose that for some $l \in L$, the probability that a given player plays either l or $-l$ is less than $\frac{1}{n}$. Then the expected utility for the other player of playing $v(l)$ is strictly greater than $\frac{1}{n}(0) + \frac{n-1}{n}(n) = n - 1$, and hence this cannot be a Nash equilibrium. So we can assume that for any $l \in L$, the probability that a given player plays either l or $-l$ is precisely $\frac{1}{n}$.

If there is an element of L such that player 1 puts positive probability on it and player 2 on its negation, both players have expected utility less than $n - 1$ and would be better off switching to f. So, in a Nash equilibrium, if player 1 plays l with some probability, player 2 must play l with probability $\frac{1}{n}$, and thus player 1 must play l with probability $\frac{1}{n}$. Thus we can assume that for each variable, exactly one of its corresponding literals is played with probability $\frac{1}{n}$ by both players. It follows that in any Nash equilibrium (besides the one where both players play f), literals that are sometimes played indeed correspond to an assignment to the variables.

All that is left to show is that if this assignment does not satisfy ϕ, it does not correspond to a Nash equilibrium. Let $c \in C$ be a clause that is not satisfied by the assignment, that is, none of its literals are ever played. Then playing c would give utility n, and both players would be better off playing this. $\quad\square$

From Theorem 1, it follows that there exists a Nash equilibrium in $G_\epsilon(\phi)$ where each player gets utility $n - 1$ if and only if ϕ is satisfiable; otherwise, the only equilibrium is the one where both players play f and each of them gets ϵ. Suppose $n - 1 > \epsilon$. Then, any sensible definition of welfare optimization would prefer the first kind of equilibrium. Because determining whether ϕ is satisfiable is $\mathcal{NP}$-hard, it follows that determining whether a "good" equilibrium exists is $\mathcal{NP}$-hard for any such definition. Additionally, the first kind of equilibrium is, in various senses, an optimal outcome for the game, even if the players were to cooperate; hence, finding out whether such an optimal equilibrium exists is $\mathcal{NP}$-hard. More significantly, given that $n - 1$ is significantly larger than ϵ, there is no efficient algorithm that always returns an equilibrium that is "close" to optimal (assuming $\mathcal{P} \neq \mathcal{NP}$): either an optimal equilibrium is found, or we have to settle for the equilibrium that gives each player ϵ.

In the remainder of this section, we prove a variety of corollaries of Theorem 1 that illustrate these and other points. We start with corollaries that do not involve an optimization problem. All of these corollaries show $\mathcal{NP}$-completeness of a problem, meaning that the problem is both $\mathcal{NP}$-hard and in $\mathcal{NP}$. Technically, only the $\mathcal{NP}$-hardness part is a corollary of Theorem 1 in each case. Membership in $\mathcal{NP}$ follows because, for the case of two players, if an equilibrium with the desired property exists, then the supports in this equilibrium constitute a polynomial-length certificate. This is because given the supports, the remainder of the problem can be solved using linear programming (and linear programs can be solved in polynomial time; see Khachiyan, 1979).

Corollary 1. *Even in symmetric 2-player games, it is $\mathcal{NP}$-complete to determine whether there exists a* Pareto-optimal *Nash equilibrium.* (*A distribution over outcomes is Pareto-optimal if there is no other distribution over outcomes such that every player has at least the same expected utility, and at least one player has strictly greater expected utility.*)

Proof. For $\epsilon < 1$ and $n \geqslant 2$, any Nash equilibrium in $G_\epsilon(\phi)$ corresponding to a satisfying assignment is Pareto-optimal, whereas the Nash equilibrium that always exists is not Pareto-optimal. Thus, a Pareto optimal Nash equilibrium exists if and only if ϕ is satisfiable. $\quad\square$

Corollary 2. *(Gilboa and Zemel, 1989) Even in symmetric 2-player games, it is $\mathcal{NP}$-complete to determine whether there is more than one Nash equilibrium.*

Proof. For any ϕ, $G_\epsilon(\phi)$ has additional Nash equilibria (besides the one that always exists) if and only if ϕ is satisfiable. $\quad\square$

Corollary 3. *(Gilboa and Zemel, 1989)[7] Even in symmetric 2-player games, it is $\mathcal{NP}$-complete to determine whether there is a Nash equilibrium where player 1 sometimes plays a given $x \in \Sigma_1$.*

Proof. For any ϕ, in $G_\epsilon(\phi)$, there is a Nash equilibrium where player 1 sometimes plays $+x_1$ if and only if there is a satisfying assignment to ϕ with x_1 set to *true*. But determining whether this is the case is $\mathcal{NP}$-complete. $\square$

Corollary 4. *(Gilboa and Zemel, 1989)[8] Even in symmetric 2-player games, it is $\mathcal{NP}$-complete to determine whether there is a Nash equilibrium where player 1 never plays a given $x \in \Sigma_1$.*

Proof. For any ϕ, in $G_\epsilon(\phi)$, there is a Nash equilibrium where player 1 never plays f if and only if ϕ is satisfiable. $\square$

Definition 5. A *strong Nash equilibrium* (Aumann, 1959) is a vector of mixed strategies for the players so that no nonempty subset of the players can change their strategies to make all players in the subset better off.

Corollary 5. *Even in symmetric 2-player games, it is $\mathcal{NP}$-complete to determine whether a strong Nash equilibrium exists.*

Proof. For $\epsilon < 1$ and $n \geqslant 2$, any Nash equilibrium in $G_\epsilon(\phi)$ corresponding to a satisfying assignment is a strong Nash equilibrium, whereas the Nash equilibrium that always exists is not strong. Thus, a strong Nash equilibrium exists if and only if ϕ is satisfiable. $\square$

The next few corollaries concern optimization problems, such as maximizing social welfare, or maximizing the number of pure strategies in the supports of the equilibrium. For such problems, an important question is whether they can be approximately solved. For example, is it possible to find, in polynomial time, a Nash equilibrium that has at least half as great a social welfare as the social-welfare maximizing Nash equilibrium? Or—a nonconstructive version of the same problem—can we, in polynomial time, find a number k such that there exists a Nash equilibrium with social welfare at least k, and there is no Nash equilibrium with social welfare greater than $2k$? (The latter problem does not require constructing a Nash equilibrium, so it is conceivable that there is a polynomial-time algorithm for this problem even if it is hard to construct any Nash equilibrium.) We will not give approximation algorithms in this subsection; rather, we will derive certain inapproximability results from Theorem 1. In each case, we will show that even the nonconstructive problem is hard (and therefore the constructive problem is hard as well).

Before presenting our results, we first make one subtle technical point, namely that it is unreasonable to expect an approximation algorithm to work even when the game has some negative utilities in it. For suppose we had an algorithm that approximated (say) social welfare to some positive ratio, even when there are some negative utilities in the game. Then we can "boost" its

---

[7] Gilboa and Zemel (1989) only stated weaker versions of Corollaries 3 and 4, but their proof technique can in fact be used to prove the results in their full strength.

[8] See previous footnote.

results, as follows. Suppose the algorithm returns a social welfare of $2r$ on a game, and suppose this is less than the social welfare of the best Nash equilibrium. If we subtract r from all utilities in the game, the game remains the same for all strategic purposes (it has the same set of Nash equilibria). But now the result returned by the approximation algorithm on the original game corresponds to a social welfare of 0, which does not satisfy the approximation ratio. It follows that running the approximation algorithm on the transformed game must give a better result (which we can easily transform back to the original game).

For this reason, we require our hardness results to only use reductions to games where 0 is the lowest possible utility in the game. Strictly speaking, our main reduction does not have this property, as can be seen from Example 1. Nevertheless, $G_\epsilon(\phi)$ does have this property whenever $n \geqslant 4$. (We recall that n is the number of variables in ϕ.) Hence, our reduction does in fact suffice, because satisfiability remains an $\mathcal{NP}$-hard problem even under the restriction $n \geqslant 4$.[9]

We are now ready to present the remaining corollaries.

Corollary 6. *Unless $\mathcal{P} = \mathcal{NP}$, there does not exist a polynomial-time algorithm that approximates (to any positive ratio) the maximum social welfare obtained in a Nash equilibrium, even in symmetric 2-player games. (This holds even if the ratio is allowed to be a function of the size of the game.)*

Proof. Suppose such an algorithm did exist. For any formula ϕ (with number of variables $n \geqslant 4$), consider the game $G_\epsilon(\phi)$ where ϵ is set so that $2\epsilon < r(2n - 2)$ (here, r is the approximation ratio that the algorithm guarantees for games of the size of $G_\epsilon(\phi)$). If ϕ is satisfiable, then by Theorem 1, there exists an equilibrium with social welfare $2n - 2$, and thus the approximation algorithm should return a social welfare of at least $r(2n - 2) > 2\epsilon$. Otherwise, by Theorem 1, the only equilibrium has social welfare 2ϵ, and thus the approximation algorithm should return a social welfare of at most 2ϵ. Thus we can use the algorithm to solve arbitrary SAT instances. □

Corollary 7. *Unless $\mathcal{P} = \mathcal{NP}$, there does not exist a polynomial-time algorithm that approximates (to any positive ratio) the maximum* egalitarian *social welfare obtained in a Nash equilibrium, even in symmetric 2-player games. (This holds even if the ratio is allowed to be a function of the size of the game. The egalitarian social welfare is the expected utility of the worse-off player.)*

Proof. The proof is similar to that of Corollary 6. □

Corollary 8. *Unless $\mathcal{P} = \mathcal{NP}$, there does not exist a polynomial-time algorithm that approximates (to any positive ratio) the maximum utility for player 1 obtained in a Nash equilibrium,*

---

[9] Incidentally, the Gilboa and Zemel (1989) reduction uses negative utilities, and, unlike in the reduction in this paper, those utilities become *more* negative as the size of the instance increases. Specifically, their game contains utilities of $-nk^2$ (their reduction is from CLIQUE, where an instance consists of a graph with n vertices and a target clique size of k). Of course, we can add nk^2 to every utility in their game so that all utilities become nonnegative, and doing this will not change the game strategically. If we do this, then, in the resulting game, there exists a Nash equilibrium with utility $nk^2 + 1 + 1/(nk^2)$ for each player if there is a clique of size k, but in any case there exists a Nash equilibrium with utility nk^2 for each player. Hence, the reduction by Gilboa and Zemel does not imply any (significant) inapproximability. Similarly, our earlier (IJCAI-03) reduction contained utilities of $2 - n$, and could therefore not be used to obtain any (significant) inapproximability result.

even in symmetric 2-player games. (This holds even if the ratio is allowed to be a function of the size of the game.)

Proof. The proof is similar to that of Corollary 6. □

The next few corollaries use the notation $o(x)$, which refers to functions that grow slower than linearly in x, and $\Omega(x)$, which refers to functions that grow at least as fast as linearly in x. The corollaries state that it is hard to maximize (even approximately) the number of pure strategies played with positive probability (respectively, for both players together, for the player with the smaller support, and for one player only) in a Nash equilibrium.

Corollary 9. *Unless $\mathcal{P} = \mathcal{NP}$, there does not exist a polynomial-time algorithm that approximates (to any ratio $1/o(|\Sigma|)$) the maximum number, in a Nash equilibrium, of pure strategies in the players' strategies' supports, even in symmetric 2-player games.*

Proof. Suppose such an algorithm did exist. For any formula ϕ, consider the game $G_\epsilon(\phi)$ where ϵ is set arbitrarily. If ϕ is not satisfiable, then by Theorem 1, the only equilibrium has only one pure strategy in each player's support, and thus the algorithm can return a number of strategies of at most 2. On the other hand, if ϕ is satisfiable, then by Theorem 1, there is an equilibrium where each player's support has size $\Omega(|\Sigma|)$. (This is assuming that n, the number of variables in ϕ, is $\Omega(|\Sigma|)$. This is only true if the number of clauses in ϕ is at most linear in the number of variables, but it is known that SAT remains $\mathcal{NP}$-hard under this restriction—for example, SAT is known to remain $\mathcal{NP}$-hard even if each variable occurs in at most 3 clauses.) Because by assumption our algorithm has an approximation ratio of $1/o(|\Sigma|)$, this means that for large enough $|\Sigma|$, the algorithm must return a support size strictly greater than 2. Thus we can use the algorithm to solve arbitrary SAT instances (given that the instances are large enough to produce large enough $|\Sigma|$). □

Corollary 10. *Unless $\mathcal{P} = \mathcal{NP}$, there does not exist a polynomial-time algorithm that approximates (to any ratio $1/o(|\Sigma|)$) the maximum number, in a Nash equilibrium, of pure strategies in the support of the player that uses fewer pure strategies than the other, even in symmetric 2-player games.*

Proof. The proof is similar to that of Corollary 9. □

Corollary 11. *Unless $\mathcal{P} = \mathcal{NP}$, there does not exist a polynomial-time algorithm that approximates (to any ratio $1/o(|\Sigma|)$) the maximum number, in a Nash equilibrium, of pure strategies in player 1's support, even in symmetric 2-player games.*

Proof. The proof is similar to that of Corollary 9. □

Versions of Corollaries 7 and 10 that do not mention inapproximability were proven by Gilboa and Zemel (1989).

The final corollary goes beyond $\mathcal{NP}$-hardness, to #$\mathcal{P}$-hardness. Determining whether equilibria with certain properties exist is not always sufficient: sometimes, we are interested in

characterizing *all* the equilibria of a game. One rather weak such characterization is the number of equilibria.[10] We can use Theorem 1 to show that determining this number is #$\mathcal{P}$-hard.

Corollary 12. *Even in symmetric 2-player games, counting the number of Nash equilibria is #$\mathcal{P}$-hard.*

Proof. The number of Nash equilibria in our game $G_\epsilon(\phi)$ is the number of satisfying assignments of ϕ, plus one. Counting the number of satisfying assignments to a CNF formula is #$\mathcal{P}$-hard (Valiant, 1979).   □

In a sense, the most interesting #$\mathcal{P}$-hardness results are the ones where the corresponding existence problem (does there exist at least one solution?) and search problem (construct one solution, if one exists) are easy. This is the case, for example, for the problem of counting the perfect matchings in a bipartite graph (Valiant, 1979). For the problem of counting the Nash equilibria in a finite normal-form game, the corresponding existence problem is trivial (at least one Nash equilibrium always exists, so the answer is always "yes"), but the search problem is $\mathcal{PPAD}$-complete.

4. Pure-strategy Nash equilibria in Bayesian games

Equilibria in pure strategies are particularly desirable because they avoid the uncomfortable requirement that players randomize over strategies among which they are indifferent. In normal-form games, it is easy to determine the existence of pure-strategy equilibria: one can simply check, for each combination of pure strategies, whether it constitutes a Nash equilibrium. This trivial algorithm runs in time that is polynomial in the size of the normal form. However, this approach is not computationally efficient in *Bayesian* games where the players have private information about their own preferences (this private information is known as the player's *type*). In such games, players can condition their actions on their types, resulting in a strategy space that is exponential in the number of types (whereas the natural representation of the Bayesian game is not exponential in the number of types).

In this section, we show that determining whether a pure-strategy Bayes–Nash equilibrium exists is in fact $\mathcal{NP}$-complete even in symmetric two-player Bayesian games. (A mixed-strategy equilibrium always exists, although constructing one is $\mathcal{PPAD}$-hard because normal-form games are a special case of Bayesian games.) First, we review the standard definitions of Bayesian games and Bayes–Nash equilibrium.

Definition 6. In a *Bayesian game*, we are given a set of players A; for each player i, a set of types Θ_i; a commonly known prior distribution ϕ over $\Theta_1 \times \Theta_2 \times \ldots \times \Theta_{|A|}$; for each player i, a set of actions Σ_i; and for each player i, a utility function $u_i : \Theta_i \times \Sigma_1 \times \Sigma_2 \times \ldots \times \Sigma_{|A|} \to \mathbb{R}$.

We emphasize again that we only consider finite games; in particular, we only consider finite type spaces.

---

[10] The number of equilibria in normal-form games has been studied both in the worst case (McLennan and Park, 1999) and in the average case (McLennan, 2005).

Definition 7. (See Harsanyi, 1967–1968.) Given a Bayesian game, a *Bayes–Nash equilibrium (BNE)* is a vector of probability distributions over actions, one distribution (over Σ_i) for each pair $i, \theta_i \in \Theta_i$, such that no player has an incentive to deviate, for any of her types, given that the others do not deviate. That is, for any $i, \theta_i \in \Theta_i$, and any alternative probability distribution σ'_{i,θ_i} over Σ_i, we have

$$E_{\theta_{-i}|\theta_i}\big[E\big[u_i(\theta_i, s_{1,\theta_1}, s_{2,\theta_2}, \ldots, s_{i,\theta_i}, \ldots, s_{|A|,\theta_{|A|}})\big]\big]$$
$$\geq E_{\theta_{-i}|\theta_i}\big[E\big[u_i(\theta_i, s_{1,\theta_1}, s_{2,\theta_2}, \ldots, s'_{i,\theta_i}, \ldots, s_{|A|,\theta_{|A|}})\big]\big]$$

where each s_{i,θ_i} is drawn from σ_{i,θ_i}, and s'_{i,θ_i} from σ'_{i,θ_i}.

A Bayesian game can be converted to a normal-form game as follows. For every player i, let *every* mapping $s'_i : \Theta_i \to \Sigma_i$ be a pure strategy in the new normal-form game. Then, the utility function for the normal-form game is given by $u'_i(s'_1, \ldots, s'_{|A|}) = E_{\theta_1, \ldots, \theta_{|A|}}[u_i(\theta_i, s'_1(\theta_1), \ldots, s'_{|A|}(\theta_{|A|}))]$. Assuming that no type receives 0 probability under the prior, the Nash equilibria of this normal-form game correspond exactly to the Bayes–Nash equilibria of the original game. However, the normal-form game is exponentially larger (player i has $|\Sigma_i|^{|\Theta_i|}$ pure strategies in it), so this conversion is of little use for solving computational problems efficiently.

We can now define the computational problem.

Definition 8 *(PURE-STRATEGY-BNE)*. We are given a Bayesian game. We are asked whether there exists a BNE where every distribution σ_{i,θ_i} places all its mass on a single action.

To show our $\mathcal{NP}$-hardness result, we will reduce from the $\mathcal{NP}$-complete SET-COVER problem.

Definition 9 *(SET-COVER)*. We are given a set $S = \{s_1, \ldots, s_n\}$, subsets $S_1, S_2, \ldots, S_m$ of S with $\bigcup_{1 \leqslant i \leqslant m} S_i = S$, and an integer k. We are asked whether there exist $c_1, c_2, \ldots, c_k \in \{1, \ldots, m\}$ such that $\bigcup_{1 \leqslant i \leqslant k} S_{c_i} = S$.

Theorem 2. *PURE-STRATEGY-BNE is $\mathcal{NP}$-complete, even in symmetric 2-player games where the priors over types are uniform.*

Proof. To show membership in $\mathcal{NP}$, we observe that, given an action for each type for each player, it is easy to verify whether these constitute a BNE: we merely need to check that for each player i, for each type θ_i, the corresponding action maximizes i's expected utility (with respect to θ_i, given the (conditional) distribution over $-i$'s types and given $-i$'s strategy). This is done by computing the expected utility for θ_i for each possible action for i. (As an aside, we cannot simply examine every (pure) *strategy* for each player, since there are exponentially many pure strategies. Effectively, the above only examines the strategies that deviate for only a single type, and this is sufficient.)

To show $\mathcal{NP}$-hardness, we reduce an arbitrary SET-COVER instance to the following PURE-STRATEGY-BNE instance. Let there be two players, with $\Theta = \Theta_1 = \Theta_2 = \{\theta^1, \ldots, \theta^k\}$. The priors over types are uniform. Furthermore, $\Sigma = \Sigma_1 = \Sigma_2 = \{S_1, S_2, \ldots, S_m, s_1, s_2, \ldots, s_n\}$. The utility functions we choose actually do not depend on the types, so we omit the type argument in their definitions. They are as follows:

- $u_1(S_i, S_j) = u_2(S_j, S_i) = 1$ for all S_i and S_j;

- $u_1(S_i, s_j) = u_2(s_j, S_i) = 1$ for all S_i and $s_j \notin S_i$;
- $u_1(S_i, s_j) = u_2(s_j, S_i) = 2$ for all S_i and $s_j \in S_i$;
- $u_1(s_i, s_j) = u_2(s_j, s_i) = -3k$ for all s_i and s_j;
- $u_1(s_j, S_i) = u_2(S_i, s_j) = 3$ for all S_i and $s_j \notin S_i$;
- $u_1(s_j, S_i) = u_2(S_i, s_j) = -3k$ for all S_i and $s_j \in S_i$.

We now show the two instances are equivalent. First suppose there exist $c_1, c_2, \ldots, c_k \in \{1, \ldots, m\}$ such that $\bigcup_{1 \leqslant i \leqslant k} S_{c_i} = S$. Suppose both players play as follows: when their type is θ_i, they play S_{c_i}. We claim that this is a BNE. For suppose the other player employs this strategy. Then, because for any s_j, there is at least one S_{c_i} such that $s_j \in S_{c_i}$, we have that the expected utility of playing s_j is at most $\frac{1}{k}(-3k) + \frac{k-1}{k}3 < 0$. It follows that playing any of the S_j (which gives utility 1) is optimal. So there is a pure-strategy BNE.

On the other hand, suppose that there is a pure-strategy BNE. We first observe that in no pure-strategy BNE, both players play some element of S for some type: for if the other player sometimes plays some s_j, the utility of playing some s_i is at most $\frac{1}{k}(-3k) + \frac{k-1}{k}3 < 0$, whereas playing some S_i instead guarantees a utility of at least 1. So there is at least one player who never plays any element of S. Now suppose the other player sometimes plays some s_j. We know there is some S_i such that $s_j \in S_i$. If the former player plays this S_i, this will give her a utility of at least $\frac{1}{k}2 + \frac{k-1}{k}1 = 1 + \frac{1}{k}$. Since she must do at least this well in the equilibrium, and she never plays elements of S, she must sometimes receive utility 2. It follows that there exist S_a and $s_b \in S_a$ such that the former player sometimes plays S_a and the latter sometimes plays s_b. But then, playing s_b gives the latter player a utility of at most $\frac{1}{k}(-3k) + \frac{k-1}{k}3 < 0$, and she would be better off playing some S_i instead. This contradiction implies that no element of S is ever played in any pure-strategy BNE.

Now, in our given pure-strategy equilibrium, consider the set of all the S_i that are played by player 1 for some type. Clearly there can be at most k such sets. We claim they cover S. For if they do not cover some element s_j, the expected utility of playing s_j for player 2 is 3 (because player 1 never plays any element of S). But this means that player 2 (who never plays any element of S either) is not playing optimally. This contradiction implies that there exists a set cover. $\quad\square$

5. Pure-strategy Nash equilibria in stochastic (Markov) games

We now shift our attention from one-shot games to games with multiple stages. There has already been some research into the complexity of playing repeated and sequential games. For example, determining whether a particular automaton is a best response is $\mathcal{NP}$-complete (Ben-Porath, 1990); it is $\mathcal{NP}$-complete to compute a best-response automaton when the automata under consideration are bounded (Papadimitriou, 1992); the problem of whether a given player with imperfect recall can guarantee herself a given payoff using pure strategies is $\mathcal{NP}$-complete (Koller and Megiddo, 1992); and in general, best-responding to an arbitrary strategy can even be noncomputable (Knoblauch, 1994; Nachbar and Zame, 1996). In this section, we present a $\mathcal{PSPACE}$-hardness result on the existence of a pure-strategy equilibrium.

Markov (or *stochastic*) games constitute an important type of multi-stage games. In such games, there is an underlying set of states, and the game shifts between these states from stage to stage (Fudenberg and Tirole, 1991; Shapley, 1953; Sobel, 1971). At every stage, each player's payoff depends not only on the players' actions, but also on the state. Furthermore, the probability of *transitioning* to a given state is determined by the current state and the players' current actions. It should be noted that $\mathcal{PSPACE}$-hardness results are known for alternating-move games such as

generalized Go (Lichtenstein and Sipser, 1980) or QSAT (Stockmeyer and Meyer, 1973); however, if we were to formulate such a game as a Markov game, we would require an exponential number of states, so these results do not imply $\mathcal{PSPACE}$-hardness for (straightforwardly represented) Markov games. Still, one might suspect that Markov games are hard to solve because the strategy spaces are extremely rich. However, in this section we show $\mathcal{PSPACE}$-hardness for a variant where the strategy spaces are quite simple: in this variant, the players cannot condition their actions on events in the game.

Definition 10. A *Markov game* consists of

- A set of players A;
- A set of *states* S, among which the game transits, one of which is the *starting state*;
- For each player i, a set of actions Σ_i that can be played in any state;
- A *transition probability function* $p : S \times \Sigma_1 \times \ldots \times \Sigma_{|A|} \times S \to [0, 1]$, where $p(s_1, a_1, \ldots, a_{|A|}, s_2)$ gives the probability of the game being in state s_2 in the next stage, given that the current state of the game is s_1 and the players play actions $a_1, \ldots, a_{|A|}$;
- For each player i, a payoff function $u_i : S \times \Sigma_1 \times \ldots \Sigma_{|A|} \to \mathbb{R}$, where $u_i(s, a_1, \ldots, a_{|A|})$ gives the payoff to player i when the players play actions $a_1, \ldots, a_{|A|}$ in state s;
- A *discount factor* δ such that the total utility of player i is $\sum_{k=0}^{\infty} \delta^k u_i(s^k, a_1^k, \ldots, a_{|A|}^k)$, where s^k is the state of the game at stage k and the players play actions $a_1^k, \ldots, a_{|A|}^k$ in stage k.

In general, a player is not always aware of the current state of the game, the actions the others played in previous stages, or even the payoffs that the player has accumulated. In the extreme case, players never receive any information about any of these. We call such a Markov game *unobserved*. It is relatively easy to specify a pure strategy in an unobserved Markov game, because there is nothing on which the player can condition her actions. Hence, a strategy for player i is "simply" an infinite sequence of actions $\{a_i^k\}$. In spite of this apparent simplicity of the game, we show that determining whether pure-strategy equilibria exist is extremely hard. We do not need to worry about issues of credible threats and subgame perfection in this setting, so we can simply use Nash equilibrium as our solution concept.

Definition 11 *(PURE-STRATEGY-UNOBSERVED-MARKOV-NE).* We are given an unobserved Markov game. We are asked whether there exists a Nash equilibrium where all the strategies are pure.

We show that this computational problem is $\mathcal{PSPACE}$-hard, by reducing from PERIODIC-SAT, which is $\mathcal{PSPACE}$-complete (Orlin, 1981).

Definition 12 *(PERIODIC-SAT).* We are given a CNF formula $\phi(0)$ over the variables $\{x_1^0, \ldots, x_n^0\} \cup \{x_1^1, \ldots, x_n^1\}$. For any $k \in \mathbb{N}$, let $\phi(k)$ be the same formula, except that all the superscripts are incremented by k (so that each $\phi(k)$ is implicitly defined by $\phi(0)$). We are asked whether there exists an assignment of truth values to the variables $\bigcup_{k \in \mathbb{N}} \{x_1^k, \ldots, x_n^k\}$ such that $\phi(k)$ is satisfied for every $k \in \mathbb{N}$.

Theorem 3. *PURE-STRATEGY-UNOBSERVED-MARKOV-NE is $\mathcal{PSPACE}$-hard, even when the game is symmetric and 2-player, and the transition process is deterministic.*

Proof. We reduce an arbitrary PERIODIC-SAT instance to the following symmetric 2-player PURE-STRATEGY-UNOBSERVED-MARKOV-NE instance. The state space is $S = \{s_i\}_{1 \leqslant i \leqslant n} \cup \{t^1_{i,c}\}_{1 < i \leqslant 2n; c \in C} \cup \{t^2_{i,c}\}_{1 < i \leqslant 2n; c \in C} \cup \{r\}$, where C is the set of clauses in $\phi(0)$. s_1 is the starting state. Furthermore, $\Sigma = \Sigma_1 = \Sigma_2 = \{t, f\} \cup C$. The transition probabilities are given by

- $p(s_i, x^1, x^2, s_{i+1 (\text{mod } n)}) = 1$ for $1 < i \leqslant n$ and all $x^1, x^2 \in \Sigma$;
- $p(s_1, b^1, b^2, s_2) = 1$ for all $b^1, b^2 \in \{t, f\}$;
- $p(s_1, c, b, t^1_{2,c}) = 1$ for all $b \in \{t, f\}$ and $c \in C$;
- $p(s_1, b, c, t^2_{2,c}) = 1$ for all $b \in \{t, f\}$ and $c \in C$;
- $p(s_1, c^1, c^2, r) = 1$ for all $c^1, c^2 \in C$;
- $p(t^j_{i,c}, x^1, x^2, t^j_{i+1,c}) = 1$ for all $1 < i < 2n$, $j \in \{1, 2\}$, $c \in C$, and $x^1, x^2 \in \Sigma$;
- $p(t^j_{2n,c}, x^1, x^2, r) = 1$ for all $j \in \{1, 2\}$, $c \in C$, and $x^1, x^2 \in \Sigma$;
- $p(r, x^1, x^2, r) = 1$ for all $x^1, x^2 \in \Sigma$.

Some of the utilities obtained in a given stage are as follows (we do not specify utilities irrelevant to our analysis):

- $u_1(s_i, x^1, x^2) = u_2(s_i, x^2, x^1) = 0$ for $1 < i \leqslant n$ and all $x^1, x^2 \in \Sigma$;
- $u_1(s_1, b^1, b^2) = u_2(s_1, b^2, b^1) = 0$ for all $b^1, b^2 \in \{t, f\}$;
- $u_1(s_1, c, b) = u_2(s_1, b, c) = 1$ for all $b \in \{t, f\}$ and $c \in C$, when setting variable x^0_1 to b does not satisfy c;
- $u_1(s_1, c, b) = u_2(s_1, b, c) = -1$ for all $b \in \{t, f\}$ and $c \in C$, when setting variable x^0_1 to b does satisfy c;
- $u_1(s_1, c^1, c^2) = u_2(s_1, c^2, c^1) = -1$ for all $c^1, c^2 \in C$;
- $u_1(t^1_{kn+i,c}, x, b) = u_2(t^2_{kn+i,c}, b, x) = 0$ for $k \in \{0, 1\}$, $1 \leqslant i \leqslant n$, all $c \in C$ and $b \in \{t, f\}$ such that setting variable x^k_i to b does not satisfy c, and all $x \in \Sigma$;
- $u_1(t^1_{kn+i,c}, x, b) = u_2(t^2_{kn+i,c}, b, x) = -4$ for $k \in \{0, 1\}$, $1 \leqslant i \leqslant n$, all $c \in C$ and $b \in \{t, f\}$ such that setting variable x^k_i to b does satisfy c, and all $x \in \Sigma$;
- $u_1(t^1_{kn+i,c}, x, c') = u_2(t^2_{kn+i,c}, c', x) = 0$ for $k \in \{0, 1\}$, $1 \leqslant i \leqslant n$, all $c, c' \in C$, and all $x \in \Sigma$.

Additionally, the game played in state r is some symmetric zero-sum game without a pure-strategy equilibrium (for example, a generalization of rock-paper-scissors) with very small pay-offs. Finally, the discount factor is $\delta = (\frac{1}{2})^{\frac{1}{2n+1}}$ (so that $\delta^{2n} > \frac{1}{2}$).

We start our analysis with a few observations. First, there can be no pure-strategy equilibrium in which state r is reached at some point, because (since r is an absorbing state) this would require that some pure-strategy equilibrium of the game in state r were played whenever state r occurred. (Otherwise, a player who is not best-responding in one of these stages could simply switch to a best response in this stage, and because the game is unobserved, the rest of the game would remain unaffected, so this would give higher utility. This is using the fact that in a pure-strategy equilibrium, on the path of play, every player always knows the current state, because the transition process is deterministic.) But such an equilibrium does not exist. Second, if we ever reach one of the $t^j_{i,c}$ states, we will inevitably reach state r at some point after this. It follows that all pure-strategy Nash equilibria never leave the s_i states.

Now suppose an assignment satisfying the periodic SAT formula exists. Let both players play as follows: in stage $kn + i$ (with $1 \leqslant i \leqslant n$), $b \in \{t, f\}$ is played, where b is the value that the variable x_i^k is set to. Clearly, both players receive utility 0 with these strategies. Does either player have an incentive to deviate? The only deviation of any significance is to play some $c \in C$ when the current state is s_1. So, without loss of generality (because of the symmetry of the game), say player 2 deviates to playing $c \in C$ in stage $kn + 1$ (when the state is s_1). We know that in the satisfying assignment, some variable x_i^l among $x_1^k, \ldots, x_n^k, x_1^{k+1}, \ldots, x_n^{k+1}$ is set to some b such that setting x_i^{l-k} to b satisfies c. If it is x_1^k, which is set to b, then in stage $kn + 1$ player 1 plays b, and player 2 gets payoff -1 in this stage since we are in state s_1 and setting x_1^0 to b satisfies c. Otherwise, if it is x_i^l with $l = k + 1$ or $i \neq 1$, which is set to b, then player 2 will get payoff 1 in stage $kn + 1$, but in stage $ln + i$ player 1 plays b, and player 2 gets payoff -4 in this stage since we are in state $t_{(l-k)n+i,c}^2$ and setting x_i^{l-k} to b satisfies c. The discounting is insignificant enough that this more than cancels out the 1 earned in stage $kn + 1$. Player 2 will get (at most) 0 in the other stages up to the first stage in state r, and given that we made the payoffs in the game in state r sufficiently small relative to δ, player 2 will not earn enough in the remaining stages to cancel out her losses so far. So there is no incentive to deviate. Thus, a pure-strategy NE exists.

On the other hand, suppose that no assignment satisfying the periodic SAT formula exists. Let us investigate whether a Nash equilibrium could exist. We know that in such a Nash equilibrium we never leave the s_i, so both players receive utility 0, and no c is ever played in a stage with state s_1. Since playing a c in one of the other stages can have no deterrent value, we may suppose that only elements of $\{t, f\}$ are played. Now consider the following assignment to the x_i^k: if player 1 plays b in stage $kn + i$, x_i^k is set to b. Since no assignment satisfying the periodic SAT formula exists, we know there is some clause c and some k such that no variable x_i^l among $x_1^k, \ldots, x_n^k, x_1^{k+1}, \ldots, x_n^{k+1}$ is set to some b such that setting x_i^{l-k} to b satisfies c. But then, if player 2 deviates to play this c in stage $kn + 1$, she will receive payoff 1 in this stage, and payoff 0 in all the remaining stages up to the first stage in state r. Furthermore, player 2 can guarantee herself at least payoff 0 in each stage in state r, as this state corresponds to a zero-sum symmetric game. It follows that this deviation gives player 2 positive utility and is hence beneficial. Thus, no pure-strategy NE exists. $\quad\square$

A slightly simpler argument of a similar nature shows a weaker form of hardness for the case where the game is restricted to have only finitely (linearly) many stages:

Theorem 4. *PURE-STRATEGY-UNOBSERVED-MARKOV-NE is $\mathcal{NP}$-hard, even when the game is symmetric, 2-player, the transition process is deterministic, the number of stages in the game is finite* (*in fact, linear in the number of states*), *and there is no discounting* ($\delta = 1$).

Proof. We reduce an arbitrary SAT instance to the following PURE-STRATEGY-UNOBSERVED-MARKOV-NE instance. The state space is $S = \{s_1\} \cup \{s_{i,c}^1\}_{1 < i \leqslant n; c \in C} \cup \{s_{i,c}^2\}_{1 < i \leqslant n; c \in C} \cup \{r\}$ (where n is the number of variables). s_1 is the starting state. Furthermore, $\Sigma = \Sigma_1 = \Sigma_2 = \{t, f\} \cup C$. The game always ends after n stages. (If desired, the information of how many stages we have had can be captured in the state without incurring an exponential blowup.) The transition probabilities are:

- $p(s_1, b^1, b^2, r) = 1$ for all $b^1, b^2 \in \{t, f\}$;
- $p(s_1, c, b, s_{2,c}^1) = 1$ for all $b \in \{t, f\}$ and $c \in C$;

- $p(s_1, b, c, s^2_{2,c}) = 1$ for all $b \in \{t, f\}$ and $c \in C$;
- $p(s_1, c^1, c^2, r) = 1$ for all $c^1, c^2 \in C$;
- $p(s^j_{i,c}, x^1, x^2, s^j_{i+1,c}) = 1$ for all $1 < i < n$, $j \in \{1, 2\}$, $c \in C$, and $x^1, x^2 \in \Sigma$;
- $p(r, x^1, x^2, r) = 1$ for all $x^1, x^2 \in \Sigma$.

The utility functions are as follows:

- $u_1(s_1, c, b) = u_2(s_1, b, c) = 1$ for all $b \in \{t, f\}$ and $c \in C$, when setting variable x_1 to b does not satisfy c;
- $u_1(s_1, c, b) = u_2(s_1, b, c) = -1$ for all $b \in \{t, f\}$ and $c \in C$, when setting variable x_1 to b does satisfy c;
- $u_1(s_1, b, c) = u_2(s_1, c, b) = 1$ for all $b \in \{t, f\}$ and $c \in C$, when setting variable x_1 to b does satisfy c;
- $u_1(s_1, c^1, c^2) = u_2(s_1, c^2, c^1) = -1$ for all $c^1, c^2 \in C$;
- $u_1(s^1_{i,c}, x, b) = u_2(s^1_{i,c}, b, x) = -2$ for all $1 < i \leqslant n$, all $c \in C$ and $b \in \{t, f\}$ such that setting variable x_i to b does satisfy c, and all $x \in \Sigma$;
- $u_1(s^2_{i,c}, b, x) = u_2(s^1_{i,c}, x, b) = 1$ for all $1 < i \leqslant n$, all $c \in C$ and $b \in \{t, f\}$ such that setting variable x_i to b does satisfy c, and all $x \in \Sigma$;
- All other utilities are 0.

We now proceed to show that the instances are equivalent. First suppose there exists an assignment of truth values to the variables such that every clause is satisfied. Then, if variable x_i is set to $b_i \in \{t, f\}$ in this assignment, let each player play b_i in the ith stage. This will give both players a total utility of 0. The only deviation for a player that may change this is to play some clause c in the first stage. However, some variable x_i occurring in that clause must be set to a value that satisfies c. If it is x_1, the deviating player will receive utility -1 in the first stage, and no positive utilities after that. Otherwise, it is some x_i with $i > 1$, and the deviating player will receive 1 in the first stage, but -2 in the ith stage, and no positive utilities anywhere else. It follows that there is no incentive to deviate, and this is a pure-strategy Nash equilibrium.

Now suppose there exists a pure strategy Nash equilibrium. If both players play a clause in the first stage, then both players would receive a utility of -1, and either player would be better off playing some $b \in \{t, f\}$ in the first stage, to get a total utility of at least 0. So this cannot be the case in a pure-strategy equilibrium. If only one player plays a clause c in the first stage, any best response for the other player plays a truth value in the first stage, and plays b_i in stage i whenever setting x_i to b_i satisfies c. But then, the clause-playing player receives negative utility overall, and is better off playing some $b \in \{t, f\}$ in the first stage, to get a total utility of at least 0. So this also cannot be the case in a pure-strategy equilibrium. It follows that in any pure-strategy equilibrium, both players play a truth value in the first stage, and thus both players receive a total utility of 0. However, if there were no satisfying solution to the SAT instance, then there must be some clause c such that whenever setting x_i to b_i satisfies c, player 2 does not play b_i in stage i. But then, player 1 is better off playing c in the first stage, to get a total utility of 1, contradicting the fact that we have a pure-strategy Nash equilibrium. It follows that there exists a satisfying solution to the SAT instance. □

It is instructive to compare these two hardness results to known hardness results for partially observable Markov decision processes (POMDPs). A *Markov decision process* is a Markov

game with a single player. A *partially observable* Markov decision process is a Markov decision process in which the current state is not directly observable, but a player may observe noisy signals about the state. Papadimitriou and Tsitsiklis (1987) show that computing the optimal policy (strategy) for a POMDP is $\mathcal{PSPACE}$-hard even with a finite horizon. (In fact, they show this for a special kind of POMDP in which the states are partitioned, and the player always observes the element of the partition to which the current state belongs.) Unlike Theorem 3, their reduction makes use of both probabilistic transitions and nontrivial observations about the current state. Papadimitriou and Tsitsiklis also mention that their reduction can be modified to show $\mathcal{NP}$-hardness for the unobserved case, leading to a result that is more similar to our Theorem 4 (though neither result directly implies the other).

6. Conclusions and future research

We provided a single reduction that demonstrates that in normal-form games:

(1) it is $\mathcal{NP}$-complete to determine whether Nash equilibria with certain natural properties exist (these results are similar to those obtained by Gilboa and Zemel, 1989),
(2) more significantly, the problems of maximizing certain properties of a Nash equilibrium are inapproximable (unless $\mathcal{P} = \mathcal{NP}$), and
(3) it is #$\mathcal{P}$-hard to count the Nash equilibria.

We also showed that determining whether a pure-strategy Bayes–Nash equilibrium exists in a Bayesian game is $\mathcal{NP}$-complete, and that determining whether a pure-strategy Nash equilibrium exists in a Markov (stochastic) game is $\mathcal{PSPACE}$-hard even if the game is unobserved (and that this remains $\mathcal{NP}$-hard if the game has finite length). All of our hardness results hold even if there are only two players and the game is symmetric.

This paper is undoubtedly not the last word on computing game-theoretic solutions. Solution concepts other than Nash equilibrium are also receiving attention: examples include CURB sets (Benisch et al., 2006), (iterated) dominance (Conitzer and Sandholm, 2005a; Gilboa et al., 1993), other elimination criteria (Conitzer and Sandholm, 2005b), and correlated equilibria (Gilboa and Zemel, 1989; Papadimitriou, 2005). There is also a significant body of research on solving extensive-form games (Gilpin et al., 2007; Gilpin and Sandholm, 2007; Koller and Megiddo, 1992; Koller et al., 1996; Miltersen and Sørensen, 2006; Romanovskii, 1962; von Stengel, 1996; von Stengel et al., 2002).

Another topic of interest is how the game is *represented*, that is, in what form the game is presented to the solver. A polynomial-time algorithm for normal-form games is of little use if the normal form is too large for the computer to store. In this case, the computer needs to operate directly on a more concise representation of the game. Examples of such representations (other than the extensive form) include graphical games (Kearns et al., 2001), action-graph games (Bhat and Leyton-Brown, 2004; Leyton-Brown and Tennenholtz, 2003), and multiagent influence diagrams (Koller and Milch, 2001). While changing the way the game is represented does not change it strategically,[11] it does affect the computational complexity of solving the game (Gottlob et al., 2003; Schoenebeck and Vadhan, 2006). However, as long as the representation can capture

---

[11] This is assuming that no strategic sinformation is added or lost in the conversion. For example, converting an extensive-form game to a normal-form game does change the game strategically, but this is because information is lost in the conversion.

any game, the computational problem cannot become any easier than under the straightforward representation. Therefore, our hardness results apply to such other representations as well.

Finally, we should consider the implications of complexity results in game theory for the modeling of human behavior. It seems unreasonable to expect humans to play according to solutions that are too hard for computers to find, so perhaps we should consider new solution concepts. On the other hand, as rationality and computational resources increase, it seems that the standard concepts should result in the limit.

Acknowledgments

References

Abbott, T., Kane, D., Valiant, P., 2005. On the complexity of two-player win-lose games. In: Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS), pp. 113–122.

Aumann, R.J., 1959. Acceptable points in general cooperative *n*-person games. In: Contributions to the Theory of Games. In: Ann. of Math. Study, 40, vol. IV. Princeton Univ. Press, pp. 287–324.

Ben-Porath, E., 1990. The complexity of computing a best response automaton in repeated games with mixed strategies. Games Econ. Behav. 2 (1), 1–12.

Benisch, M., Davis, G., Sandholm, T., 2006. Algorithms for rationalizability and CURB sets. In: Proceedings of the National Conference on Artificial Intelligence (AAAI).

Bhat, N.A.R., Leyton-Brown, K., 2004. Computing Nash equilibria of action-graph games. In: Proceedings of the 20th Annual Conference on Uncertainty in Artificial Intelligence (UAI), pp. 35–42.

Chen, X., Deng, X., 2005. 3-Nash is PPAD-complete. In: Electronic Colloquium on Computational Complexity, Report No. 134.

Chen, X., Deng, X., 2006. Settling the complexity of two-player Nash equilibrium. In: Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS), pp. 261–272.

Conitzer, V., Sandholm, T., 2005a. Complexity of (iterated) dominance. In: Proceedings of the ACM Conference on Electronic Commerce (EC), pp. 88–97.

Conitzer, V., Sandholm, T., 2005b. A generalized strategy eliminability criterion and computational methods for applying it. In: Proceedings of the National Conference on Artificial Intelligence (AAAI), pp. 483–488.

Cook, S., 1971. The complexity of theorem proving procedures. In: Proceedings of the Third Annual ACM Symposium on Theory of Computing, pp. 151–158.

Daskalakis, C., Papadimitriou C.H., 2005. Three-player games are hard. In: Electronic Colloquium on Computational Complexity, Report 139.

Daskalakis, C., Goldberg, P., Papadimitriou, C.H., 2006. The complexity of computing a Nash equilibrium. In: Proceedings of the Annual Symposium on Theory of Computing (STOC), pp. 71–78.

Daskalakis, C., Mehta, A., Papadimitriou, C.H., 2007. Progress in approximate Nash equilibria. In: Proceedings of the ACM Conference on Electronic Commerce (EC), pp. 355–358.

Dickhaut, J., Kaplan, T., 1991. A program for finding Nash equilibria. Mathematica Journal, 87–93.

Fudenberg, D., Tirole, J., 1991. Game Theory. MIT Press.

Gilboa, I., Zemel, E., 1989. Nash and correlated equilibria: Some complexity considerations. Games Econ. Behav. 1, 80–93.

Gilboa, I., Kalai, E., Zemel, E., 1993. The complexity of eliminating dominated strategies. Math. Operation Res. 18, 553–565.

Gilpin, A., Sandholm, T., 2007. Lossless abstraction of imperfect information games. J. ACM 54 (5).

Gilpin, A., Hoda, S., Peña, J., Sandholm, T., 2007. Gradient-based algorithms for finding Nash equilibria in extensive form games. In: Workshop on Internet and Network Economics (WINE). San Diego, CA, USA.

Gottlob, G., Greco, G., Scarcello, F., 2003. Pure Nash equilibria: hard and easy games. In: Theoretical Aspects of Rationality and Knowledge (TARK), pp. 215–230.

Harsanyi, J., 1967–1968. Game with incomplete information played by Bayesian players. Manage. Sci. 14, 159–182. 320–334; 486–502.

Kearns, M., Littman, M., Singh, S., 2001. Graphical models for game theory. In: Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI).

Khachiyan, L., 1979. A polynomial algorithm in linear programming. Soviet Math. Doklady 20, 191–194.

Knoblauch, V., 1994. Computable strategies for repeated prisoner's dilemma. Games Econ. Behav. 7, 381–389.

Koller, D., Megiddo, N., 1992. The complexity of two-person zero-sum games in extensive form. Games Econ. Behav. 4 (4), 528–552.

Koller, D., Milch, B., 2001. Multi-agent influence diagrams for representing and solving games. In: Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI), pp. 1027–1034.

Koller, D., Megiddo, N., von Stengel, B., 1996. Efficient computation of equilibria for extensive two-person games. Games Econ. Behav. 14 (2), 247–259.

Lemke, C., Howson, J., 1964. Equilibrium points of bimatrix games. J. Soc. Ind. Appl. Math. 12, 413–423.

Leyton-Brown, K., Tennenholtz, M., 2003. Local-effect games. In: Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI), pp. 772–780.

Lichtenstein, D., Sipser, M., 1980. GO is polynomial-space hard. J. ACM 27, 393–401.

Lipton, R., Markakis, E., Mehta, A., 2003. Playing large games using simple strategies. In: Proceedings of the ACM Conference on Electronic Commerce (EC), pp. 36–41.

McLennan, A., 2005. The expected number of Nash equilibria of a normal form game. Econometrica 73 (1), 141–174.

McLennan, A., Park, I.-U., 1999. Generic 4×4 two person games have at most 15 Nash equilibria. Games Econ. Behav. 26 (1), 111–130.

Miltersen, P.B., Sørensen, T.B., 2006. Computing sequential equilibria for two-player games. In: Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 107–116.

Nachbar, J.H., Zame, W.R., 1996. Non-computable strategies and discounted repeated games. Econ. Theory 8 (1), 103–122. June.

Nash, J., 1950. Equilibrium points in n-person games. Proc. Natl. Acad. Sci. 36, 48–49.

Orlin, J., 1981. The complexity of dynamic languages and dynamic optimization problems. In: Proceedings of the Annual Symposium on Theory of Computing (STOC), pp. 218–227.

Papadimitriou, C.H., 1992. On players with a bounded number of states. Games Econ. Behav. 4, 122–131.

Papadimitriou, C.H., 2001. Algorithms, games and the Internet. In: Proceedings of the Annual Symposium on Theory of Computing (STOC), pp. 749–753.

Papadimitriou, C.H., 2005. Computing correlated equilibria in multi-player games. In: Proceedings of the Annual Symposium on Theory of Computing (STOC), pp. 49–56.

Papadimitriou, C.H., Tsitsiklis, J.N., 1987. The complexity of Markov decision processes. Math. Operations Res. 12, 441–450.

Porter, R., Nudelman, E., Shoham, Y., 2008. Simple search methods for finding a Nash equilibrium. Games Econ. Behav. 63 (2), 642–662 (in this issue).

Romanovskii, I., 1962. Reduction of a game with complete memory to a matrix game. Soviet Math. 3, 678–681.

Sandholm, T., Gilpin, A., Conitzer, V., 2005. Mixed-integer programming methods for finding Nash equilibria. In: Proceedings of the National Conference on Artificial Intelligence (AAAI), pp. 495–501.

Savani, R., von Stengel, B., 2006. Hard-to-solve bimatrix games. Econometrica 74, 397–429. Early version in FOCS-04.

Schoenebeck, G., Vadhan, S., 2006. The computational complexity of Nash equilibria in concisely represented games. In: Proceedings of the ACM Conference on Electronic Commerce (EC), pp. 270–279.

Shapley, L.S., 1953. Stochastic games. Proc. Natl. Acad. Sci. 39, 1095–1100.

Sobel, M., 1971. Noncooperative stochastic games. Ann. Math. Statist. 42, 1930–1935.

Stockmeyer, L., Meyer, A., 1973. Word problems requiring exponential time. In: Proceedings of the Annual Symposium on Theory of Computing (STOC), pp. 1–9.

Valiant, L., 1979. The complexity of computing the permanent. Theoretical Comput. Sci. 8, 189–201.

von Stengel, B., 1996. Efficient computation of behavior strategies. Games Econ. Behav. 14 (2), 220–246.

von Stengel, B., van den Elzen, A., Talman, D., 2002. Computing normal form perfect equilibria for extensive two-person games. Econometrica 70 (2), 693–715. March.