

Complexity of (Iterated) Dominance*

Vincent Conitzer
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213, USA
conitzer@cs.cmu.edu

Tuomas Sandholm
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213, USA
sandholm@cs.cmu.edu

ABSTRACT

We study various computational aspects of solving games using dominance and iterated dominance. We first study both strict and weak dominance (not iterated), and show that checking whether a given strategy is dominated by some mixed strategy can be done in polynomial time using a single linear program solve. We then move on to iterated dominance. We show that determining whether there is some path that eliminates a given strategy is NP-complete with iterated weak dominance. This allows us to also show that determining whether there is a path that leads to a unique solution is NP-complete. Both of these results hold both with and without dominance by mixed strategies. (A weaker version of the second result (only without dominance by mixed strategies) was already known [7].) Iterated strict dominance, on the other hand, is path-independent (both with and without dominance by mixed strategies) and can therefore be done in polynomial time.

We then study what happens when the dominating strategy is allowed to place positive probability on only a few pure strategies. First, we show that finding the dominating strategy with minimum support size is NP-complete (both for strict and weak dominance). Then, we show that iterated strict dominance becomes path-dependent when there is a limit on the support size of the dominating strategies, and that deciding whether a given strategy can be eliminated by iterated strict dominance under this restriction is NP-complete (even when the limit on the support size is 3).

Finally, we study Bayesian games. We show that, unlike in normal form games, deciding whether a given pure strategy is dominated by another pure strategy in a Bayesian game is NP-complete (both with strict and weak dominance); however, deciding whether a strategy is dominated by some mixed strategy can still be done in polynomial time with a single linear program solve (both with strict and weak

dominance). Finally, we show that iterated dominance using pure strategies can require an exponential number of iterations in a Bayesian game (both with strict and weak dominance).

Categories and Subject Descriptors

F.2 [Theory of Computation]: Analysis of Algorithms and Problem Complexity; J.4 [Computer Applications]: Social and Behavioral Sciences—*Economics*

General Terms

Algorithms, Economics, Theory

Keywords

Game Theory, (Iterated) Dominance

1. INTRODUCTION

In multiagent systems with self-interested agents, the optimal action for one agent may depend on the actions taken by other agents. In such settings, the agents require tools from game theory to rationally decide on an action. Game theory offers various formal models of strategic settings—the best-known of which is a game in *normal* (or *matrix*) *form*, specifying a utility (payoff) for each agent for each combination of strategies that the agents choose—as well as *solution concepts*, which, given a game, specify which outcomes are reasonable (under various assumptions of rationality and common knowledge).

Probably the best-known (and certainly the most-studied) solution concept is that of *Nash equilibrium*. A Nash equilibrium specifies a strategy for each player, in such a way that no player has an incentive to (unilaterally) deviate from the prescribed strategy. Recently, numerous papers have studied computing Nash equilibria in various settings [9, 4, 12, 3, 13, 14], and the complexity of constructing a Nash equilibrium in normal form games has been labeled one of the two most important open problems on the boundary of P today [20].

The problem of computing solutions according to the perhaps more elementary solution concepts of *dominance* and *iterated dominance* has received much less attention. (After an early short paper on an easy case [11], the main computational study of these concepts has actually taken place in a paper in the game theory community [7].¹) A strategy *strictly dominates* another strategy if it performs strictly

*This material is based upon work supported by the National Science Foundation under ITR grants IIS-0121678 and IIS-0427858, and a Sloan Fellowship.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

EC'05, June 5–8, 2005, Vancouver, British Columbia, Canada.
Copyright 2005 ACM 1-59593-049-3/05/0006 ...\$5.00.

¹This is not to say that computer scientists have ignored

better against all vectors of opponent strategies, and *weakly dominates* it if it performs at least as well against all vectors of opponent strategies, and strictly better against at least one. The idea is that dominated strategies can be eliminated from consideration. In *iterated dominance*, the elimination proceeds in rounds, and becomes easier as more strategies are eliminated: in any given round, the dominating strategy no longer needs to perform better than or as well as the dominated strategy against opponent strategies that were eliminated in earlier rounds. Computing solutions according to (iterated) dominance is important for at least the following reasons: 1) it can be computationally easier than computing (for instance) a Nash equilibrium (and therefore it can be useful as a preprocessing step in computing a Nash equilibrium), and 2) (iterated) dominance requires a weaker rationality assumption on the players than (for instance) Nash equilibrium, and therefore solutions derived according to it are more likely to occur.

In this paper, we study some fundamental computational questions concerning dominance and iterated dominance, including how hard it is to check whether a given strategy can be eliminated by each of the variants of these notions. The rest of the paper is organized as follows. In Section 2, we briefly review definitions and basic properties of normal form games, strict and weak dominance, and iterated strict and weak dominance. In the remaining sections, we study computational aspects of dominance and iterated dominance. In Section 3, we study one-shot (not iterated) dominance. In Section 4, we study iterated dominance. In Section 5, we study dominance and iterated dominance when the dominating strategy can only place probability on a few pure strategies. Finally, in Section 6, we study dominance and iterated dominance in Bayesian games.

2. DEFINITIONS AND BASIC PROPERTIES

In this section, we briefly review normal form games, as well as dominance and iterated dominance (both strict and weak). An n -player normal form game is defined as follows.

DEFINITION 1. A normal form game is given by a set of players $\{1, 2, \dots, n\}$; and, for each player i , a (finite) set of pure strategies Σ_i and a utility function $u_i : \Sigma_1 \times \Sigma_2 \times \dots \times \Sigma_n \rightarrow \mathbb{R}$ (where $u_i(\sigma_1, \sigma_2, \dots, \sigma_n)$ denotes player i 's utility when each player j plays action σ_j).

The two main notions of dominance are defined as follows.

DEFINITION 2. Player i 's strategy σ'_i is said to be strictly dominated by player i 's strategy σ_i if for any vector of strategies σ_{-i} for the other players, $u_i(\sigma_i, \sigma_{-i}) > u_i(\sigma'_i, \sigma_{-i})$. Player i 's strategy σ'_i is said to be weakly dominated by player i 's strategy σ_i if for any vector of strategies σ_{-i} for the other players, $u_i(\sigma_i, \sigma_{-i}) \geq u_i(\sigma'_i, \sigma_{-i})$, and for at least one vector of strategies σ_{-i} for the other players, $u_i(\sigma_i, \sigma_{-i}) > u_i(\sigma'_i, \sigma_{-i})$.

In this definition, it is sometimes allowed for the dominating strategy σ_i to be a *mixed strategy*, that is, a probability distribution over pure strategies. In this case, the utilities in

dominance altogether. For example, simple dominance checks are sometimes used as a subroutine in searching for Nash equilibria [21].

the definition are the expected utilities.² There are other notions of dominance, such as *very weak* dominance (in which no strict inequality is required, so two strategies can dominate each other), but we will not study them here. When we are looking at the dominance relations for player i , the other players ($-i$) can be thought of as a single player.³ Therefore, in the rest of the paper, when we study one-shot (not iterated) dominance, we will focus without loss of generality on two-player games.⁴ In two-player games, we will generally refer to the players as r (row) and c (column) rather than 1 and 2.

In *iterated dominance*, dominated strategies are removed from the game, and no longer have any effect on future dominance relations. Iterated dominance can eliminate more strategies than dominance, as follows. σ_r may originally not dominate σ'_r because the latter performs better against σ'_c ; but then, once σ'_c is removed because it is dominated by σ_c , σ_r dominates σ'_r , and the latter can be removed. For example, in the following game, R can be removed first, after which D is dominated.

	L	R
U	1, 1	0, 0
D	0, 1	1, 0

Either strict or weak dominance can be used in the definition of iterated dominance. We note that the process of iterated dominance is never helped by removing a dominated mixed strategy, for the following reason. If σ'_i gives player i a higher utility than σ_i against mixed strategy σ'_j for player $j \neq i$ (and strategies $\sigma_{-\{i,j\}}$ for the other players), then for at least one pure strategy σ_j that σ'_j places positive probability on, σ'_i must perform better than σ_i against σ_j (and strategies $\sigma_{-\{i,j\}}$ for the other players). Thus, removing the mixed strategy σ'_j does not introduce any new dominances.

More detailed discussions and examples can be found in standard texts on microeconomics or game theory [17, 5]. We are now ready to move on to the core of this paper.

3. DOMINANCE (NOT ITERATED)

In this section, we study the notion of one-shot (not iterated) dominance. As a first observation, checking whether a given strategy is strictly (weakly) dominated by some *pure* strategy is straightforward, by checking, for every pure strategy for that player, whether the latter strategy performs strictly better against all the opponent's strategies (at least as well against all the opponent's strategies, and strictly

²The dominated strategy σ'_i is, of course, also allowed to be mixed, but this has no technical implications for the paper: when we study one-shot dominance, we ask whether a given strategy is dominated, and it does not matter whether the given strategy is pure or mixed; when we study iterated dominance, there is no use in eliminating mixed strategies, as we will see shortly.

³This player may have a very large strategy space (one pure strategy for every vector of pure strategies for the players that are being replaced). Nevertheless, this will not result in an increase in our representation size, because the original representation already had to specify utilities for each of these vectors.

⁴We note that a restriction to two-player games would not be without loss of generality for *iterated* dominance. This is because for iterated dominance, we need to look at the dominated strategies of each individual player, so we cannot merge any players.

better against at least one).⁵ Next, we show that checking whether a given strategy is dominated by some *mixed* strategy can be done in polynomial time by solving a single linear program. (Similar linear programs have been given before [18]; we present the result here for completeness, and because we will build on the linear programs given below in Theorem 6.)

PROPOSITION 1. *Given the row player’s utilities, a subset D_r of the row player’s pure strategies Σ_r , and a distinguished strategy σ_r^* for the row player, we can check in time polynomial in the size of the game (by solving a single linear program of polynomial size) whether there exists some mixed strategy σ_r , that places positive probability only on strategies in D_r and dominates σ_r^* , both for strict and for weak dominance.*

PROOF. Let p_{d_r} be the probability that σ_r places on $d_r \in D_r$. We will solve a single linear program in each of our algorithms; linear programs can be solved in polynomial time [10]. For strict dominance, the question is whether the p_{d_r} can be set so that for every pure strategy for the column player $\sigma_c \in \Sigma_c$, $\sum_{d_r \in D_r} p_{d_r} u_r(d_r, \sigma_c) > u_r(\sigma_r^*, \sigma_c)$. Because the inequality must be strict, we cannot solve this directly by linear programming. We proceed as follows. Because the game is finite, we may assume without loss of generality that all utilities are positive (if not, simply add a constant to all utilities.) Solve the following linear program:

$$\begin{aligned} & \text{minimize} && \sum_{d_r \in D_r} p_{d_r} \\ & \text{such that} && \\ & \text{for any } \sigma_c \in \Sigma_c, && \sum_{d_r \in D_r} p_{d_r} u_r(d_r, \sigma_c) \geq u_r(\sigma_r^*, \sigma_c). \end{aligned}$$

If σ_r^* is strictly dominated by some mixed strategy, this linear program has a solution with objective value < 1 . (The dominating strategy is a feasible solution with objective value exactly 1. Because no constraint is binding for this solution, we can reduce one of the probabilities slightly without affecting feasibility, thereby obtaining a solution with objective value < 1 .) Moreover, if this linear program has a solution with objective value < 1 , there is a mixed strategy strictly dominating σ_r^* , which can be obtained by taking the LP solution and adding the remaining probability to any strategy (because all the utilities are positive, this will add to the left side of any inequality, so all inequalities will become strict).

For weak dominance, we can solve the following linear program:

$$\begin{aligned} & \text{maximize} && \sum_{\sigma_c \in \Sigma_c} \left(\sum_{d_r \in D_r} p_{d_r} u_r(d_r, \sigma_c) \right) - u_r(\sigma_r^*, \sigma_c) \\ & \text{such that} && \\ & \text{for any } \sigma_c \in \Sigma_c, && \sum_{d_r \in D_r} p_{d_r} u_r(d_r, \sigma_c) \geq u_r(\sigma_r^*, \sigma_c); \\ & && \sum_{d_r \in D_r} p_{d_r} = 1. \end{aligned}$$

If σ_r^* is weakly dominated by some mixed strategy, then that mixed strategy is a feasible solution to this program with objective value > 0 , because for at least one strategy $\sigma_c \in \Sigma_c$ we have $(\sum_{d_r \in D_r} p_{d_r} u_r(d_r, \sigma_c)) - u_r(\sigma_r^*, \sigma_c) > 0$. On the other hand, if this program has a solution with objective value > 0 , then for at least one strategy $\sigma_c \in \Sigma_c$ we

⁵Recall that the assumption of a single opponent (that is, the assumption of two players) is without loss of generality for one-shot dominance.

must have $(\sum_{d_r \in D_r} p_{d_r} u_r(d_r, \sigma_c)) - u_r(\sigma_r^*, \sigma_c) > 0$, and thus the linear program’s solution is a weakly dominating mixed strategy. \square

4. ITERATED DOMINANCE

We now move on to iterated dominance. It is well-known that iterated strict dominance is path-independent [6, 19]—that is, if we remove dominated strategies until no more dominated strategies remain, in the end the remaining strategies for each player will be the same, regardless of the order in which strategies are removed. Because of this, to see whether a given strategy can be eliminated by iterated strict dominance, all that needs to be done is to repeatedly remove strategies that are strictly dominated, until no more dominated strategies remain. Because we can check in polynomial time whether any given strategy is dominated (whether or not dominance by mixed strategies is allowed, as described in Section 3), this whole procedure takes only polynomial time. In the case of iterated dominance by pure strategies with two players, Knuth *et al.* [11] slightly improve on (speed up) the straightforward implementation of this procedure by keeping track of, for each ordered pair of strategies for a player, the number of opponent strategies that prevent the first strategy from dominating the second. Hereby the runtime for an $m \times n$ game is reduced from $O((m+n)^4)$ to $O((m+n)^3)$. (Actually, they only study very weak dominance (for which no strict inequalities are required), but the approach is easily extended.)

In contrast, iterated weak dominance is known to be path-dependent.⁶ For example, in the following game, using iterated weak dominance we can eliminate M first, and then D , or R first, and then U .

	L	M	R
U	1, 1	0, 0	1, 0
D	1, 1	1, 0	0, 0

Therefore, while the procedure of removing weakly dominated strategies until no more weakly dominated strategies remain can certainly be executed in polynomial time, which strategies survive in the end depends on the order in which we remove the dominated strategies. We will investigate two questions for iterated weak dominance: whether a given strategy is eliminated in some path, and whether there is a path to a unique solution (one pure strategy left per player). We will show that both of these problems are computationally hard.

DEFINITION 3. *Given a game in normal form and a distinguished strategy σ^* , IWD-STRATEGY-ELIMINATION asks whether there is some path of iterated weak dominance that eliminates σ^* . Given a game in normal form, IWD-UNIQUE-SOLUTION asks whether there is some path of iterated weak dominance that leads to a unique solution (one strategy left per player).*

The following lemma shows a special case of normal form games in which allowing for weak dominance by mixed strategies (in addition to weak dominance by pure strategies) does

⁶There is, however, a restriction of weak dominance called *nice weak dominance* which is path-independent [15, 16]. For an overview of path-independence results, see Apt [1].

not help. We will prove the hardness results in this setting, so that they will hold whether or not dominance by mixed strategies is allowed.

LEMMA 1. *Suppose that all the utilities in a game are in $\{0, 1\}$. Then every pure strategy that is weakly dominated by a mixed strategy is also weakly dominated by a pure strategy.*

PROOF. Suppose pure strategy σ is weakly dominated by mixed strategy σ^* . If σ gets a utility of 1 against some opponent strategy (or vector of opponent strategies if there are more than 2 players), then all the pure strategies that σ^* places positive probability on must also get a utility of 1 against that opponent strategy (or else the expected utility would be smaller than 1). Moreover, at least one of the pure strategies that σ^* places positive probability on must get a utility of 1 against an opponent strategy that σ gets 0 against (or else the inequality would never be strict). It follows that this pure strategy weakly dominates σ . \square

We are now ready to prove the main results of this section.

THEOREM 1. *IWD-STRATEGY-ELIMINATION is NP-complete, even with 2 players, and with 0 and 1 being the only utilities occurring in the matrix—whether or not dominance by mixed strategies is allowed.*

PROOF. The problem is in NP because given a sequence of strategies to be eliminated, we can easily check whether this is a valid sequence of eliminations (even when dominance by mixed strategies is allowed, using Proposition 1). To show that the problem is NP-hard, we reduce an arbitrary satisfiability instance (given by a nonempty set of clauses C over a nonempty set of variables V , with corresponding literals $L = \{+v : v \in V\} \cup \{-v : v \in V\}$) to the following IWD-STRATEGY-ELIMINATION instance. (In this instance, we will specify that certain strategies are *uneliminable*. A strategy σ_r can be made uneliminable, even when 0 and 1 are the only allowed utilities, by adding another strategy σ'_r and another opponent strategy σ_c , so that: 1. σ_r and σ'_r are the only strategies that give the row player a utility of 1 against σ_c . 2. σ_r and σ'_r always give the row player the same utility. 3. σ_c is the only strategy that gives the column player a utility of 1 against σ'_r , but otherwise σ_c always gives the column player utility 0. This makes it impossible to eliminate any of these three strategies. We will not explicitly specify the additional strategies to make the proof more legible.)

In this proof, we will denote row player strategies by s , and column player strategies by t , to improve legibility. Let the row player's pure strategy set be given as follows. For every variable $v \in V$, the row player has corresponding strategies $s_{+v}^1, s_{+v}^2, s_{-v}^1, s_{-v}^2$. Additionally, the row player has the following 2 strategies: s_0^1 and s_0^2 , where $s_0^2 = \sigma_r^*$ (that is, it is the strategy we seek to eliminate). Finally, for every clause $c \in C$, the row player has corresponding strategies s_c^1 (uneliminable) and s_c^2 . Let the column player's pure strategy set be given as follows. For every variable $v \in V$, the column player has a corresponding strategy t_v . For every clause $c \in C$, the column player has a corresponding strategy t_c , and additionally, for every literal $l \in L$ that occurs in c , a strategy $t_{c,l}$. For every variable $v \in V$, the column player has corresponding strategies t_{+v}, t_{-v} (both uneliminable). Finally, the column player has three additional strategies: t_0^1 (uneliminable), t_0^2 , and t_1 .

The utility function for the row player is given as follows:

- $u_r(s_{+v}^1, t_v) = 0$ for all $v \in V$;
- $u_r(s_{+v}^2, t_v) = 1$ for all $v \in V$;
- $u_r(s_{-v}^1, t_v) = 1$ for all $v \in V$;
- $u_r(s_{-v}^2, t_v) = 0$ for all $v \in V$;
- $u_r(s_{+v}^1, t_1) = 1$ for all $v \in V$;
- $u_r(s_{+v}^2, t_1) = 0$ for all $v \in V$;
- $u_r(s_{-v}^1, t_1) = 0$ for all $v \in V$;
- $u_r(s_{-v}^2, t_1) = 1$ for all $v \in V$;
- $u_r(s_{+v}^b, t_{+v}) = 1$ for all $v \in V$ and $b \in \{1, 2\}$;
- $u_r(s_{-v}^b, t_{-v}) = 1$ for all $v \in V$ and $b \in \{1, 2\}$;
- $u_r(s_l, t) = 0$ otherwise for all $l \in L$ and $t \in S_2$;
- $u_r(s_0^1, t_c) = 0$ for all $c \in C$;
- $u_r(s_0^2, t_c) = 1$ for all $c \in C$;
- $u_r(s_0^b, t_0^1) = 1$ for all $b \in \{1, 2\}$;
- $u_r(s_0^1, t_0^2) = 1$;
- $u_r(s_0^2, t_0^2) = 0$;
- $u_r(s_0^b, t) = 0$ otherwise for all $b \in \{1, 2\}$ and $t \in S_2$;
- $u_r(s_c^b, t) = 0$ otherwise for all $c \in C$ and $b \in \{1, 2\}$;

and the row player's utility is 0 in every other case. The utility function for the column player is given as follows:

- $u_c(s, t_v) = 0$ for all $v \in V$ and $s \in S_1$;
- $u_c(s, t_1) = 0$ for all $s \in S_1$;
- $u_c(s_l^i, t_c) = 1$ for all $c \in C$ and $l \in L$ where $l \in c$ (literal l occurs in clause c);
- $u_c(s_{l_2}^2, t_{c,l_1}) = 1$ for all $c \in C$ and $l_1, l_2 \in L, l_1 \neq l_2$ where $l_2 \in c$;
- $u_c(s_c^1, t_c) = 1$ for all $c \in C$;
- $u_c(s_c^2, t_c) = 0$ for all $c \in C$;
- $u_c(s_c^b, t_{c,l}) = 1$ for all $c \in C, l \in L$, and $b \in \{1, 2\}$;
- $u_c(s_2, t_c) = u_c(s_2, t_{c,l}) = 0$ otherwise for all $c \in C$ and $l \in L$;

and the column player's utility is 0 in every other case. We now show that the two instances are equivalent.

First, suppose there is a solution to the satisfiability instance: that is, a truth-value assignment to the variables in V such that all clauses are satisfied. Then, consider the following sequence of eliminations in our game: 1. For every variable v that is set to *true* in the assignment, eliminate t_v (which gives the column player utility 0 everywhere). 2. Then, for every variable v that is set to *true* in the assignment, eliminate s_{+v}^2 using s_{+v}^1 (which is possible because t_v has been eliminated, and because t_1 has not been eliminated (yet)). 3. Now eliminate t_1 (which gives the column player utility 0 everywhere). 4. Next, for every variable v that is set to *false* in the assignment, eliminate s_{-v}^2 using s_{-v}^1 (which is possible because t_1 has been eliminated, and because t_v has not been eliminated (yet)). 5. For every clause c which has the variable corresponding to one of its positive literals $l = +v$ set to *true* in the assignment, eliminate t_c using $t_{c,l}$ (which is possible because s_l^2 has been eliminated, and s_c^2 has not been eliminated (yet)). 6. For every clause c which has the variable corresponding to one of its negative literals $l = -v$ set to *false* in the assignment, eliminate t_c using $t_{c,l}$

(which is possible because s_l^2 has been eliminated, and s_c^2 has not been eliminated (yet)). 7. Because the assignment satisfied the formula, all the t_c have now been eliminated. Thus, we can eliminate $s_0^2 = \sigma_r^*$ using s_0^1 . It follows that there is a solution to the IWD-STRATEGY-ELIMINATION instance.

Now suppose there is a solution to the IWD-STRATEGY-ELIMINATION instance. By Lemma 1, we can assume that all the dominances are by pure strategies. We first observe that only s_0^1 can eliminate $s_0^2 = \sigma_r^*$, because it is the only other strategy that gets the row player a utility of 1 against t_0^1 , and t_0^1 is uneliminable. However, because s_0^2 performs better than s_0^1 against the t_c strategies, it follows that all of the t_c strategies must be eliminated. For each $c \in C$, the strategy t_c can only be eliminated by one of the strategies $t_{c,l}$ (with the same c), because these are the only other strategies that get the column player a utility of 1 against s_c^1 , and s_c^1 is uneliminable. But, in order for some $t_{c,l}$ to eliminate t_c , s_l^2 must be eliminated first. Only s_l^1 can eliminate s_l^2 , because it is the only other strategy that gets the row player a utility of 1 against t_l , and t_l is uneliminable. We next show that for every $v \in V$ only one of s_{+v}^2, s_{-v}^2 can be eliminated. This is because in order for s_{+v}^1 to eliminate s_{+v}^2 , t_v needs to have been eliminated and t_1 , not (so t_v must be eliminated before t_1); but in order for s_{-v}^1 to eliminate s_{-v}^2 , t_1 needs to have been eliminated and t_v , not (so t_1 must be eliminated before t_v). So, set v to *true* if s_{+v}^2 is eliminated, and to *false* otherwise. Because by the above, for every clause c , one of the s_l^2 with $l \in c$ must be eliminated, it follows that this is a satisfying assignment to the satisfiability instance. \square

Using Theorem 1, it is now (relatively) easy to show that IWD-UNIQUE-SOLUTION is also NP-complete under the same restrictions.

THEOREM 2. *IWD-UNIQUE-SOLUTION is NP-complete, even with 2 players, and with 0 and 1 being the only utilities occurring in the matrix—whether or not dominance by mixed strategies is allowed.*

PROOF. Again, the problem is in NP because we can non-deterministically choose the sequence of eliminations and verify whether it is correct. To show NP-hardness, we reduce an arbitrary IWD-STRATEGY-ELIMINATION instance to the following IWD-UNIQUE-SOLUTION instance. Let all the strategies for each player from the original instance remain part of the new instance, and let the utilities resulting from the players playing a pair of these strategies be the same. We add three additional strategies $\sigma_r^1, \sigma_r^2, \sigma_r^3$ for the row player, and three additional strategies $\sigma_c^1, \sigma_c^2, \sigma_c^3$ for the column player. Let the additional utilities be as follows:

- $u_r(\sigma_r, \sigma_c^j) = 1$ for all $\sigma_r \notin \{\sigma_r^1, \sigma_r^2, \sigma_r^3\}$ and $j \in \{2, 3\}$;
- $u_r(\sigma_r^i, \sigma_c) = 1$ for all $i \in \{1, 2, 3\}$ and $\sigma_c \notin \{\sigma_c^2, \sigma_c^3\}$;
- $u_r(\sigma_r^i, \sigma_c^2) = 1$ for all $i \in \{2, 3\}$;
- $u_r(\sigma_r^1, \sigma_c^3) = 1$;
- and the row player's utility is 0 in all other cases involving a new strategy.
- $u_c(\sigma_r^3, \sigma_c) = 1$ for all $\sigma_c \notin \{\sigma_c^1, \sigma_c^2, \sigma_c^3\}$;
- $u_c(\sigma_r^*, \sigma_c^j) = 1$ for all $j \in \{2, 3\}$ (σ_r^* is the strategy to be eliminated in the original instance);
- $u_c(\sigma_r^i, \sigma_c^1) = 1$ for all $i \in \{1, 2\}$;
- $u_r(\sigma_r^1, \sigma_c^2) = 1$;

- $u_r(\sigma_r^2, \sigma_c^3) = 1$;
- and the column player's utility is 0 in all other cases involving a new strategy.

We proceed to show that the two instances are equivalent.

First suppose there exists a solution to the original IWD-STRATEGY-ELIMINATION instance. Then, perform the same sequence of eliminations to eliminate σ_r^* in the new IWD-UNIQUE-SOLUTION instance. (This is possible because at any stage, any weak dominance for the row player in the original instance is still a weak dominance in the new instance, because the two strategies' utilities for the row player are the same when the column player plays one of the new strategies; and the same is true for the column player.) Once σ_r^* is eliminated, let σ_c^1 eliminate σ_c^2 . (It performs better against σ_r^2 .) Then, let σ_r^1 eliminate all the other remaining strategies for the row player. (It always performs better against either σ_c^1 or σ_c^3 .) Finally, σ_c^1 is the unique best response against σ_r^1 among the column player's remaining strategies, so let it eliminate all the other remaining strategies for the column player. Thus, there exists a solution to the IWD-UNIQUE-SOLUTION instance.

Now suppose there exists a solution to the IWD-UNIQUE-SOLUTION instance. By Lemma 1, we can assume that all the dominances are by pure strategies. We will show that none of the new strategies ($\sigma_r^1, \sigma_r^2, \sigma_r^3, \sigma_c^1, \sigma_c^2, \sigma_c^3$) can either eliminate another strategy, or be eliminated before σ_r^* is eliminated. Thus, there must be a sequence of eliminations ending in the elimination of σ_r^* , which does not involve any of the new strategies, and is therefore a valid sequence of eliminations in the original game (because all original strategies perform the same against each new strategy). We now show that this is true by exhausting all possibilities for the *first* elimination before σ_r^* is eliminated that involves a new strategy. None of the σ_r^i can be eliminated by a $\sigma_r \notin \{\sigma_r^1, \sigma_r^2, \sigma_r^3\}$, because the σ_r^i perform better against σ_c^1 . σ_r^1 cannot eliminate any other strategy, because it always performs poorer against σ_c^2 . σ_r^2 and σ_r^3 are equivalent from the row player's perspective (and thus cannot eliminate each other), and cannot eliminate any other strategy because they always perform poorer against σ_c^3 . None of the σ_c^j can be eliminated by a $\sigma_c \notin \{\sigma_c^1, \sigma_c^2, \sigma_c^3\}$, because the σ_c^j always perform better against either σ_r^1 or σ_r^2 . σ_c^1 cannot eliminate any other strategy, because it always performs poorer against either σ_r^* or σ_r^3 . σ_c^2 cannot eliminate any other strategy, because it always performs poorer against σ_r^2 or σ_r^3 . σ_c^3 cannot eliminate any other strategy, because it always performs poorer against σ_r^1 or σ_r^3 . Thus, there exists a solution to the IWD-STRATEGY-ELIMINATION instance. \square

A slightly weaker version of the part of Theorem 2 concerning dominance by pure strategies only is the main result of Gilboa *et al.* [7]. (Besides not proving the result for dominance by mixed strategies, the original result was weaker because it required utilities $\{0, 1, 2, 3, 4, 5, 6, 7, 8\}$ rather than just $\{0, 1\}$ (and because of this, our Lemma 1 cannot be applied to it to get the result for mixed strategies).)

5. (ITERATED) DOMINANCE USING MIXED STRATEGIES WITH SMALL SUPPORTS

When showing that a strategy is dominated by a mixed strategy, there are several reasons to prefer exhibiting a

dominating strategy that places positive probability on as few pure strategies as possible. First, this will reduce the number of bits required to specify the dominating strategy (and thus the proof of dominance can be communicated quicker): if the dominating mixed strategy places positive probability on only k strategies, then it can be specified using k real numbers for the probabilities, plus $k \log m$ (where m is the number of strategies for the player under consideration) bits to indicate which strategies are used. Second, the proof of dominance will be “cleaner”: for a dominating mixed strategy, it is typically (always in the case of strict dominance) possible to spread some of the probability onto any unused pure strategy and still have a dominating strategy, but this obscures which pure strategies are the ones that are key in making the mixed strategy dominating. Third, because (by the previous) the argument for eliminating the dominated strategy is simpler and easier to understand, it is more likely to be accepted. Fourth, the level of risk neutrality required for the argument to work is reduced, at least in the extreme case where dominance by a single pure strategy can be exhibited (no risk neutrality is required here).

This motivates the following problem.

DEFINITION 4 (MINIMUM-DOMINATING-SET). *We are given the row player’s utilities of a game in normal form, a distinguished strategy σ^* for the row player, a specification of whether the dominance should be strict or weak, and a number k . We are asked whether there exists a mixed strategy σ for the row player that places positive probability on at most k pure strategies, and dominates σ^* in the required sense.*

Unfortunately, this problem is NP-complete.

THEOREM 3. *MINIMUM-DOMINATING-SET is NP-complete, both for strict and for weak dominance.*

PROOF. The problem is in NP because we can nondeterministically choose a set of at most k strategies to give positive probability, and decide whether we can dominate σ^* with these k strategies as described in Proposition 1. To show NP-hardness, we reduce an arbitrary SET-COVER instance (given a set S , subsets S_1, S_2, \dots, S_r , and a number t , can all of S be covered by at most t of the subsets?) to the following MINIMUM-DOMINATING-SET instance. For every element $s \in S$, there is a pure strategy σ_s for the column player. For every subset S_i , there is a pure strategy σ_{S_i} for the row player. Finally, there is the distinguished pure strategy σ^* for the row player. The row player’s utilities are as follows: $u_r(\sigma_{S_i}, \sigma_s) = t + 1$ if $s \in S_i$; $u_r(\sigma_{S_i}, \sigma_s) = 0$ if $s \notin S_i$; $u_r(\sigma^*, \sigma_s) = 1$ for all $s \in S$. Finally, we let $k = t$. We now proceed to show that the two instances are equivalent.

First suppose there exists a solution to the SET-COVER instance. Without loss of generality, we can assume that there are exactly k subsets in the cover. Then, for every S_i that is in the cover, let the dominating strategy σ place exactly $\frac{1}{k}$ probability on the corresponding pure strategy σ_{S_i} . Now, if we let $n(s)$ be the number of subsets in the cover containing s (we observe that that $n(s) \geq 1$), then for every strategy σ_s for the column player, the row player’s expected utility for playing σ when the column player is playing σ_s is $u(\sigma, \sigma_s) = \frac{n(s)}{k}(k + 1) \geq \frac{k+1}{k} > 1 = u(\sigma^*, \sigma_s)$. So σ strictly (and thus also weakly) dominates σ^* , and there exists a solution to the MINIMUM-DOMINATING-SET instance.

Now suppose there exists a solution to the MINIMUM-DOMINATING-SET instance. Consider the (at most k) pure strategies of the form σ_{S_i} on which the dominating mixed strategy σ places positive probability, and let \mathcal{T} be the collection of the corresponding subsets S_i . We claim that \mathcal{T} is a cover. For suppose there is some $s \in S$ that is not in any of the subsets in \mathcal{T} . Then, if the column player plays σ_s , the row player (when playing σ) will always receive utility 0—as opposed to the utility of 1 the row player would receive for playing σ^* , contradicting the fact that σ dominates σ^* (whether this dominance is weak or strict). It follows that there exists a solution to the SET-COVER instance. \square

On the other hand, if we require that the dominating strategy only places positive probability on a *very* small number of pure strategies, then it once again becomes easy to check whether a strategy is dominated. Specifically, to find out whether player i ’s strategy σ^* is dominated by a strategy that places positive probability on only k pure strategies, we can simply check, for every subset of k of player i ’s pure strategies, whether there is a strategy that places positive probability only on these k strategies and dominates σ^* , using Proposition 1. This requires only $O(|\Sigma_i|^k)$ such checks. Thus, if k is a constant, this constitutes a polynomial-time algorithm.

A natural question to ask next is whether *iterated* strict dominance remains computationally easy when dominating strategies are required to place positive probability on at most k pure strategies, where k is a small constant. (We have already shown in Section 4 that iterated weak dominance is hard even when $k = 1$, that is, only dominance by pure strategies is allowed.) Of course, if iterated strict dominance were path-independent under this restriction, computational easiness would follow as it did in Section 4. However, it turns out that this is not the case.

OBSERVATION 1. *If we restrict the dominating strategies to place positive probability on at most two pure strategies, iterated strict dominance becomes path-dependent.*

PROOF. Consider the following game:

7, 1	0, 0	0, 0
0, 0	7, 1	0, 0
3, 0	3, 0	0, 0
0, 0	0, 0	3, 1
1, 0	1, 0	1, 0

Let (i, j) denote the outcome in which the row player plays the i th row and the column player plays the j th column. Because $(1, 1)$, $(2, 2)$, and $(4, 3)$ are all Nash equilibria, none of the column player’s pure strategies will ever be eliminated, and neither will rows 1, 2, and 4. We now observe that randomizing uniformly over rows 1 and 2 dominates row 3, and randomizing uniformly over rows 3 and 4 dominates row 5. However, if we eliminate row 3 first, it becomes impossible to dominate row 5 without randomizing over at least 3 pure strategies. \square

Indeed, iterated strict dominance turns out to be hard even when $k = 3$.

THEOREM 4. *If we restrict the dominating strategies to place positive probability on at most three pure strategies, it becomes NP-complete to decide whether a given strategy can be eliminated using iterated strict dominance.*

PROOF. The problem is in NP because given a sequence of strategies to be eliminated, we can check in polynomial time whether this is a valid sequence of eliminations (for any strategy to be eliminated, we can check, for every subset of three other strategies, whether there is a strategy placing positive probability on only these three strategies that dominates the strategy to be eliminated, using Proposition 1). To show that the problem is NP-hard, we reduce an arbitrary satisfiability instance (given by a nonempty set of clauses C over a nonempty set of variables V , with corresponding literals $L = \{+v : v \in V\} \cup \{-v : v \in V\}$) to the following two-player game.

For every variable $v \in V$, the row player has strategies $s_{+v}, s_{-v}, s_v^1, s_v^2, s_v^3, s_v^4$, and the column player has strategies $t_v^1, t_v^2, t_v^3, t_v^4$. For every clause $c \in C$, the row player has a strategy s_c , and the column player has a strategy t_c , as well as, for every literal l occurring in c , an additional strategy t_c^l . The row player has two additional strategies s_1 and s_2 . (s_2 is the strategy that we are seeking to eliminate.) Finally, the column player has one additional strategy t_1 .

The utility function for the row player is given as follows (where ϵ is some sufficiently small number):

- $u_r(s_{+v}, t_v^j) = 4$ if $j \in \{1, 2\}$, for all $v \in V$;
- $u_r(s_{+v}, t_v^j) = 1$ if $j \in \{3, 4\}$, for all $v \in V$;
- $u_r(s_{-v}, t_v^j) = 1$ if $j \in \{1, 2\}$, for all $v \in V$;
- $u_r(s_{-v}, t_v^j) = 4$ if $j \in \{3, 4\}$, for all $v \in V$;
- $u_r(s_{+v}, t) = u_r(s_{-v}, t) = 0$ for all $v \in V$ and $t \notin \{t_v^1, t_v^2, t_v^3, t_v^4\}$;
- $u_r(s_v^i, t_v^i) = 13$ for all $v \in V$ and $i \in \{1, 2, 3, 4\}$;
- $u_r(s_v^i, t) = \epsilon$ for all $v \in V$, $i \in \{1, 2, 3, 4\}$, and $t \neq t_v^i$;
- $u_r(s_c, t_c) = 2$ for all $c \in C$;
- $u_r(s_c, t) = 0$ for all $c \in C$ and $t \neq t_c$;
- $u_r(s_1, t_1) = 1 + \epsilon$;
- $u_r(s_1, t) = \epsilon$ for all $t \neq t_1$;
- $u_r(s_2, t_1) = 1$;
- $u_r(s_2, t_c) = 1$ for all $c \in C$;
- $u_r(s_2, t) = 0$ for all $t \notin \{t_1\} \cup \{t_c : c \in C\}$.

The utility function for the column player is given as follows:

- $u_c(s_v^i, t_v^i) = 1$ for all $v \in V$ and $i \in \{1, 2, 3, 4\}$;
- $u_c(s, t_v^i) = 0$ for all $v \in V$, $i \in \{1, 2, 3, 4\}$, and $s \neq s_v^i$;
- $u_c(s_c, t_c) = 1$ for all $c \in C$;
- $u_c(s_l, t_c) = 1$ for all $c \in C$ and $l \in L$ occurring in c ;
- $u_c(s, t_c) = 0$ for all $c \in C$ and $s \notin \{s_c\} \cup \{s_l : l \in c\}$;
- $u_c(s_c, t_c^l) = 1 + \epsilon$ for all $c \in C$;
- $u_c(s_{l'}, t_c^l) = 1 + \epsilon$ for all $c \in C$ and $l' \neq l$ occurring in c ;
- $u_c(s, t_c^l) = \epsilon$ for all $c \in C$ and $s \notin \{s_c\} \cup \{s_{l'} : l' \in c, l' \neq l\}$;
- $u_c(s_2, t_1) = 1$;
- $u_c(s, t_1) = 0$ for all $s \neq s_2$.

We now show that the two instances are equivalent. First, suppose that there is a solution to the satisfiability instance. Then, consider the following sequence of eliminations in our game: 1. For every variable v that is set to *true* in the satisfying assignment, eliminate s_{+v} with the mixed strategy σ_r that places probability $1/3$ on s_{-v} , probability $1/3$ on s_v^1 , and probability $1/3$ on s_v^2 . (The expected utility of playing σ_r against t_v^1 or t_v^2 is $14/3 > 4$; against t_v^3 or t_v^4 , it is $4/3 > 1$; and against anything else it is $2\epsilon/3 > 0$. Hence the dominance is valid.) 2. Similarly, for every variable v that is set to *false* in the satisfying assignment, eliminate s_{-v} with the mixed strategy σ_r that places probability $1/3$ on s_{+v} , probability $1/3$ on s_v^3 , and probability $1/3$ on s_v^4 . (The expected utility of playing σ_r against t_v^1 or t_v^2 is $4/3 > 1$; against t_v^3 or t_v^4 , it is $14/3 > 4$; and against anything else it is $2\epsilon/3 > 0$. Hence the dominance is valid.) 3. For every $c \in C$, eliminate t_c with any t_c^l for which l was set to *true* in the satisfying assignment. (This is a valid dominance because t_c^l performs better than t_c against any strategy other than s_l , and we eliminated s_l in step 1 or in step 2.) 4. Finally, eliminate s_2 with s_1 . (This is a valid dominance because s_1 performs better than s_2 against any strategy other than those in $\{t_c : c \in C\}$, which we eliminated in step 3.) Hence, there is an elimination path that eliminates s_2 .

Now, suppose that there is an elimination path that eliminates s_2 . The strategy that eventually dominates s_2 must place most of its probability on s_1 , because s_1 is the only other strategy that performs well against t_1 , which cannot be eliminated before s_2 . But, s_1 performs significantly worse than s_2 against any strategy t_c with $c \in C$, so it follows that all these strategies must be eliminated first. Each strategy t_c can only be eliminated by a strategy that places most of its weight on the corresponding strategies t_c^l with $l \in c$, because they are the only other strategies that perform well against s_c , which cannot be eliminated before t_c . But, each strategy t_c^l performs significantly worse than t_c against s_l , so it follows that for every clause c , for one of the literals l occurring in it, s_l must be eliminated first. Now, strategies of the form t_v^j will never be eliminated because they are the unique best responses to the corresponding strategies s_v^j (which are, in turn, the best responses to the corresponding t_v^j). As a result, if strategy s_{+v} (respectively, s_{-v}) is eliminated, then its opposite strategy s_{-v} (respectively, s_{+v}) can no longer be eliminated, for the following reason. There is no other pure strategy remaining that gets a significant utility against more than one of the strategies $t_v^1, t_v^2, t_v^3, t_v^4$, but s_{-v} (respectively, s_{+v}) gets significant utility against all 4, and therefore cannot be dominated by a mixed strategy placing positive probability on at most 3 strategies. It follows that for each $v \in V$, at most one of the strategies s_{+v}, s_{-v} is eliminated, in such a way that for every clause c , for one of the literals l occurring in it, s_l must be eliminated. But then setting all the literals l such that s_l is eliminated to *true* constitutes a solution to the satisfiability instance. \square

In the next section, we return to the setting where there is no restriction on the number of pure strategies on which a dominating mixed strategy can place positive probability.

6. (ITERATED) DOMINANCE IN BAYESIAN GAMES

So far, we have focused on normal form games that are flatly represented (that is, every matrix entry is given ex-

plicitly). However, for many games, the flat representation is too large to write down explicitly, and instead, some representation that exploits the structure of the game needs to be used. *Bayesian games*, besides being of interest in their own right, can be thought of as a useful structured representation of normal form games, and we will study them in this section.

In a Bayesian game, each player first receives privately held preference information (the player's *type*) from a distribution, which determines the utility that that player receives for every outcome of (that is, vector of actions played in) the game. After receiving this type, the player plays an action based on it.⁷

DEFINITION 5. *A Bayesian game is given by a set of players $\{1, 2, \dots, n\}$; and, for each player i , a (finite) set of actions A_i , a (finite) type space Θ_i with a probability distribution π_i over it, and a utility function $u_i : \Theta_i \times A_1 \times A_2 \times \dots \times A_n \rightarrow \mathbb{R}$ (where $u_i(\theta_i, a_1, a_2, \dots, a_n)$ denotes player i 's utility when i 's type is θ_i and each player j plays action a_j). A pure strategy in a Bayesian game is a mapping from types to actions, $\sigma_i : \Theta_i \rightarrow A_i$, where $\sigma_i(\theta_i)$ denotes the action that player i plays for type θ_i .*

Any vector of pure strategies in a Bayesian game defines an (expected) utility for each player, and therefore we can translate a Bayesian game into a normal form game. In this normal form game, the notions of dominance and iterated dominance are defined as before. However, the normal form representation of the game is exponentially larger than the Bayesian representation, because each player i has $|A_i|^{|\Theta_i|}$ distinct pure strategies. Thus, any algorithm for Bayesian games that relies on expanding the game to its normal form will require exponential time. Specifically, our easiness results for normal form games do not directly transfer to this setting. In fact, it turns out that checking whether a strategy is dominated by a pure strategy is hard in Bayesian games.

THEOREM 5. *In a Bayesian game, it is NP-complete to decide whether a given pure strategy $\sigma_r : \Theta_r \rightarrow A_r$ is dominated by some other pure strategy (both for strict and weak dominance), even when the row player's distribution over types is uniform.*

PROOF. The problem is in NP because it is easy to verify whether a candidate dominating strategy is indeed a dominating strategy. To show that the problem is NP-hard, we reduce an arbitrary satisfiability instance (given by a set of clauses C using variables from V) to the following Bayesian game. Let the row player's action set be $A_r = \{t, f, 0\}$ and let the column player's action set be $A_c = \{a_c : c \in C\}$. Let the row player's type set be $\Theta_r = \{\theta_v : v \in V\}$, with a distribution π_r that is uniform. Let the row player's utility function be as follows:

- $u_r(\theta_v, 0, a_c) = 0$ for all $v \in V$ and $c \in C$;
- $u_r(\theta_v, b, a_c) = |V|$ for all $v \in V$, $c \in C$, and $b \in \{t, f\}$ such that setting v to b satisfies c ;
- $u_r(\theta_v, b, a_c) = -1$ for all $v \in V$, $c \in C$, and $b \in \{t, f\}$ such that setting v to b does not satisfy c .

⁷In general, a player can also receive a signal about the other players' preferences, but we will not concern ourselves with that here.

Let the pure strategy to be dominated be the one that plays 0 for every type. We show that the strategy is dominated by a pure strategy if and only if there is a solution to the satisfiability instance.

First, suppose there is a solution to the satisfiability instance. Then, let σ_r^d be given by: $\sigma_r^d(\theta_v) = t$ if v is set to *true* in the solution to the satisfiability instance, and $\sigma_r^d(\theta_v) = f$ otherwise. Then, against any action a_c by the column player, there is at least one type θ_v such that either $+v \in c$ and $\sigma_r^d(\theta_v) = t$, or $-v \in c$ and $\sigma_r^d(\theta_v) = f$. Thus, the row player's expected utility against action a_c is at least $\frac{|V|}{|V|} - \frac{|V|-1}{|V|} = \frac{1}{|V|} > 0$. So, σ_r^d is a dominating strategy.

Now, suppose there is a dominating pure strategy σ_r^d . This dominating strategy must play t or f for at least one type. Thus, against any a_c by the column player, there must at least be some type θ_v for which $u_r(\theta_v, \sigma_r^d(\theta_v), a_c) > 0$. That is, there must be at least one variable v such that setting v to $\sigma_r^d(\theta_v)$ satisfies c . But then, setting each v to $\sigma_r^d(\theta_v)$ must satisfy all the clauses. So a satisfying assignment exists. \square

However, it turns out that we can modify the linear programs from Proposition 1 to obtain a polynomial time algorithm for checking whether a strategy is dominated by a *mixed* strategy in Bayesian games.

THEOREM 6. *In a Bayesian game, it can be decided in polynomial time whether a given (possibly mixed) strategy σ_r is dominated by some other mixed strategy, using linear programming (both for strict and weak dominance).*

PROOF. We can modify the linear programs presented in Proposition 1 as follows. For strict dominance, again assuming without loss of generality that all the utilities in the game are positive, use the following linear program (in which $p_r^{\sigma_r}(\theta_r, a_r)$ is the probability that σ_r , the strategy to be dominated, places on a_r for type θ_r):

$$\begin{aligned} & \text{minimize } \sum_{\theta_r \in \Theta_r} \sum_{a_r \in A_r} p_r(a_r) \\ & \text{such that} \\ & \text{for any } a_c \in A_c, \sum_{\theta_r \in \Theta_r} \sum_{a_r \in A_r} \pi(\theta_r) u_r(\theta_r, a_r, a_c) p_r(\theta_r, a_r) \geq \\ & \sum_{\theta_r \in \Theta_r} \sum_{a_r \in A_r} \pi(\theta_r) u_r(\theta_r, a_r, a_c) p_r^{\sigma_r}(\theta_r, a_r); \\ & \text{for any } \theta_r \in \Theta_r, \sum_{a_r \in A_r} p_r(\theta_r, a_r) \leq 1. \end{aligned}$$

Assuming that $\pi(\theta_r) > 0$ for all $\theta_r \in \Theta_r$, this program will return an objective value smaller than $|\Theta_r|$ if and only if σ_r is strictly dominated, by reasoning similar to that done in Proposition 1.

For weak dominance, use the following linear program:

$$\begin{aligned} & \text{maximize } \sum_{a_c \in A_c} \left(\sum_{\theta_r \in \Theta_r} \sum_{a_r \in A_r} \pi(\theta_r) u_r(\theta_r, a_r, a_c) p_r(\theta_r, a_r) - \right. \\ & \left. \sum_{\theta_r \in \Theta_r} \sum_{a_r \in A_r} \pi(\theta_r) u_r(\theta_r, a_r, a_c) p_r^{\sigma_r}(\theta_r, a_r) \right) \\ & \text{such that} \\ & \text{for any } a_c \in A_c, \sum_{\theta_r \in \Theta_r} \sum_{a_r \in A_r} \pi(\theta_r) u_r(\theta_r, a_r, a_c) p_r(\theta_r, a_r) \geq \\ & \sum_{\theta_r \in \Theta_r} \sum_{a_r \in A_r} \pi(\theta_r) u_r(\theta_r, a_r, a_c) p_r^{\sigma_r}(\theta_r, a_r); \\ & \text{for any } \theta_r \in \Theta_r, \sum_{a_r \in A_r} p_r(\theta_r, a_r) = 1. \end{aligned}$$

This program will return an objective value greater than 0 if and only if σ_r is weakly dominated, by reasoning similar to that done in Proposition 1. \square

We now turn to iterated dominance in Bayesian games. Naïvely, one might argue that iterated dominance in Bayesian

games *always* requires an exponential number of steps when a significant fraction of the game's pure strategies can be eliminated, because there are exponentially many pure strategies. However, this is not a very strong argument because oftentimes we can eliminate exponentially many pure strategies *in one step*. For example, if for some type $\theta_r \in \Theta_r$ we have, for all $a_c \in A_c$, that $u(\theta_r, a_r^1, a_c) > u(\theta_r, a_r^2, a_c)$, then any pure strategy for the row player which plays action a_r^2 for type θ_r is dominated (by the strategy that plays action a_r^1 for type θ_r instead)—and there are exponentially many ($|A_r|^{|\Theta_r|-1}$) such strategies. It is therefore conceivable that we need only polynomially many eliminations of *collections* of a player's strategies. However, the following theorem shows that this is not the case, by giving an example where an exponential number of *iterations* (that is, alternations between the players in eliminating strategies) is required. (We emphasize that this is not a result about computational complexity.)

THEOREM 7. *Even in symmetric 3-player Bayesian games, iterated dominance by pure strategies can require an exponential number of iterations (both for strict and weak dominance), even with only three actions per player.*

PROOF. Let each player $i \in \{1, 2, 3\}$ have $n + 1$ types $\theta_i^1, \theta_i^2, \dots, \theta_i^{n+1}$. Let each player i have 3 actions a_i, b_i, c_i , and let the utility function of each player be defined as follows. (In the below, $i + 1$ and $i + 2$ are shorthand for $i + 1 \pmod{3}$ and $i + 2 \pmod{3}$ when used as player indices. Also, $-\infty$ can be replaced by a sufficiently negative number. Finally, δ and ϵ should be chosen to be very small (even compared to $2^{-(n+1)}$), and ϵ should be more than twice as large as δ .)

- $u_i(\theta_i^1; a_i, c_{i+1}, c_{i+2}) = -1$;
- $u_i(\theta_i^1; a_i, s_{i+1}, s_{i+2}) = 0$ for $s_{i+1} \neq c_{i+1}$ or $s_{i+2} \neq c_{i+2}$;
- $u_i(\theta_i^1; b_i, s_{i+1}, s_{i+2}) = -\epsilon$ for $s_{i+1} \neq a_{i+1}$ and $s_{i+2} \neq a_{i+2}$;
- $u_i(\theta_i^1; b_i, s_{i+1}, s_{i+2}) = -\infty$ for $s_{i+1} = a_{i+1}$ or $s_{i+2} = a_{i+2}$;
- $u_i(\theta_i^1; c_i, s_{i+1}, s_{i+2}) = -\infty$ for all s_{i+1}, s_{i+2} ;
- $u_i(\theta_i^j; a_i, s_{i+1}, s_{i+2}) = -\infty$ for all s_{i+1}, s_{i+2} when $j > 1$;
- $u_i(\theta_i^j; b_i, s_{i+1}, s_{i+2}) = -\epsilon$ for all s_{i+1}, s_{i+2} when $j > 1$;
- $u_i(\theta_i^j; c_i, s_{i+1}, c_{i+2}) = \delta - \epsilon - 1/2$ for all s_{i+1} when $j > 1$;
- $u_i(\theta_i^j; c_i, s_{i+1}, s_{i+2}) = \delta - \epsilon$ for all s_{i+1} and $s_{i+2} \neq c_{i+2}$ when $j > 1$.

Let the distribution over each player's types be given by $p(\theta_i^j) = 2^{-j}$ (with the exception that $p(\theta_i^2) = 2^{-2} + 2^{-(n+1)}$). We will be interested in eliminating strategies of the following form: play b_i for type θ_i^1 , and play one of b_i or c_i otherwise. Because the utility function is the same for any type θ_i^j with $j > 1$, these strategies are effectively defined by the total probability that they place on c_i ,⁸ which is $t_i^2(2^{-2} + 2^{-(n+1)}) + \sum_{j=3}^{n+1} t_i^j 2^{-j}$ where $t_i^j = 1$ if player i

⁸Note that the strategies are still pure strategies; the probability placed on an action by a strategy here is simply the sum of the probabilities of the types for which the strategy chooses that action.

plays c_i for type θ_i^j , and 0 otherwise. This probability is different for any two different strategies of the given form, and we have exponentially many different strategies of the given form. For any probability q which can be expressed as $t_2(2^{-2} + 2^{-(n+1)}) + \sum_{j=3}^{n+1} t_j 2^{-j}$ (with all $t_j \in \{0, 1\}$), let $\sigma_i(q)$ denote the (unique) strategy of the given form for player i which places a total probability of q on c_i . Any strategy that plays c_i for type θ_i^1 or a_i for some type θ_i^j with $j > 1$ can immediately be eliminated. We will show that, after that, we must eliminate the strategies $\sigma_i(q)$ with high q first, slowly working down to those with lower q .

Claim 1: If $\sigma_{i+1}(q')$ and $\sigma_{i+2}(q')$ have not yet been eliminated, and $q < q'$, then $\sigma_i(q)$ cannot yet be eliminated. *Proof:* First, we show that no strategy $\sigma_i(q'')$ can eliminate $\sigma_i(q)$. Against $\sigma_{i+1}(q'''), \sigma_{i+2}(q''')$, the utility of playing $\sigma_i(p)$ is $-\epsilon + p \cdot \delta - p \cdot q'''/2$. Thus, when $q''' = 0$, it is best to set p as high as possible (and we note that $\sigma_{i+1}(0)$ and $\sigma_{i+2}(0)$ have not been eliminated), but when $q''' > 0$, it is best to set p as low as possible because $\delta < q'''/2$. Thus, whether $q > q''$ or $q < q''$, $\sigma_i(q)$ will always do strictly better than $\sigma_i(q'')$ against some remaining opponent strategies. Hence, no strategy $\sigma_i(q'')$ can eliminate $\sigma_i(q)$. The only other pure strategies that could dominate $\sigma_i(q)$ are strategies that play a_i for type θ_i^1 , and b_i or c_i for all other types. Let us take such a strategy and suppose that it plays c with probability p . Against $\sigma_{i+1}(q'), \sigma_{i+2}(q')$ (which have not yet been eliminated), the utility of playing this strategy is $-(q')^2/2 - \epsilon/2 + p \cdot \delta - p \cdot q'/2$. On the other hand, playing $\sigma_i(q)$ gives $-\epsilon + q \cdot \delta - q \cdot q'/2$. Because $q' > q$, we have $-(q')^2/2 < -q \cdot q'/2$, and because δ and ϵ are small, it follows that $\sigma_i(q)$ receives a higher utility. Therefore, no strategy dominates $\sigma_i(q)$, proving the claim.

Claim 2: If for all $q' > q$, $\sigma_{i+1}(q')$ and $\sigma_{i+2}(q')$ have been eliminated, then $\sigma_i(q)$ can be eliminated. *Proof:* Consider the strategy for player i that plays a_i for type θ_i^1 , and b_i for all other types (call this strategy σ'_i); we claim σ'_i dominates $\sigma_i(q)$. First, if either of the other players k plays a_k for θ_k^1 , then σ'_i performs better than $\sigma_i(q)$ (which receives $-\infty$ in some cases). Because the strategies for player k that play c_k for type θ_k^1 , or a_k for some type θ_k^j with $j > 1$, have already been eliminated, all that remains to check is that σ'_i performs better than $\sigma_i(q)$ whenever both of the other two players play strategies of the following form: play b_k for type θ_k^1 , and play one of b_k or c_k otherwise. We note that among these strategies, there are none left that place probability greater than q on c_k . Letting q_k denote the probability with which player k plays c_k , the expected utility of playing σ'_i is $-q_{i+1} \cdot q_{i+2}/2 - \epsilon/2$. On the other hand, the utility of playing $\sigma_i(q)$ is $-\epsilon + q \cdot \delta - q \cdot q_{i+2}/2$. Because $q_{i+1} \leq q$, the difference between these two expressions is at least $\epsilon/2 - \delta$, which is positive. It follows that σ'_i dominates $\sigma_i(q)$.

From Claim 2, it follows that all strategies of the form $\sigma_i(q)$ will eventually be eliminated. However, Claim 1 shows that we cannot go ahead and eliminate multiple such strategies for one player, unless at least one other player simultaneously “keeps up” in the eliminated strategies: every time a $\sigma_i(q)$ is eliminated such that $\sigma_{i+1}(q)$ and $\sigma_{i+2}(q)$ have not yet been eliminated, we need to eliminate one of the latter two strategies before any $\sigma_i(q')$ with $q' > q$ can be eliminated—that is, we need to alternate between players. Because there are exponentially many strategies of the form $\sigma_i(q)$, it follows that iterated elimination will require exponentially many iterations to complete. \square

It follows that an efficient algorithm for iterated dominance (strict or weak) by pure strategies in Bayesian games, if it exists, must somehow be able to perform (at least part of) many iterations in a single step of the algorithm (because if each step only performed a single iteration, we would need exponentially many steps). Interestingly, Knuth *et al.* [11] argue that iterated dominance appears to be an inherently sequential problem (in light of their result that iterated very weak dominance is P-complete, that is, apparently not efficiently parallelizable), suggesting that aggregating many iterations may be difficult.

7. CONCLUSIONS

While the Nash equilibrium solution concept is studied more and more intensely in our community, the perhaps more elementary concept of (iterated) dominance has received much less attention. In this paper we studied various computational aspects of this concept.

We first studied both strict and weak dominance (not iterated), and showed that checking whether a given strategy is dominated by some mixed strategy can be done in polynomial time using a single linear program solve. We then moved on to iterated dominance. We showed that determining whether there is some path that eliminates a given strategy is NP-complete with iterated weak dominance. This allowed us to also show that determining whether there is a path that leads to a unique solution is NP-complete. Both of these results hold both with and without dominance by mixed strategies. (A weaker version of the second result (only without dominance by mixed strategies) was already known [7].) Iterated strict dominance, on the other hand, is path-independent (both with and without dominance by mixed strategies) and can therefore be done in polynomial time.

We then studied what happens when the dominating strategy is allowed to place positive probability on only a few pure strategies. First, we showed that finding the dominating strategy with minimum support size is NP-complete (both for strict and weak dominance). Then, we showed that iterated strict dominance becomes path-dependent when there is a limit on the support size of the dominating strategies, and that deciding whether a given strategy can be eliminated by iterated strict dominance under this restriction is NP-complete (even when the limit on the support size is 3).

Finally, we studied dominance and iterated dominance in Bayesian games, as an example of a concise representation language for normal form games that is interesting in its own right. We showed that, unlike in normal form games, deciding whether a given pure strategy is dominated by another pure strategy in a Bayesian game is NP-complete (both with strict and weak dominance); however, deciding whether a strategy is dominated by some mixed strategy can still be done in polynomial time with a single linear program solve (both with strict and weak dominance). Finally, we showed that iterated dominance using pure strategies can require an exponential number of iterations in a Bayesian game (both with strict and weak dominance).

There are various avenues for future research. First, there is the open question of whether it is possible to complete iterated dominance in Bayesian games in polynomial time (even though we showed that an exponential number of alternations between the players in eliminating strategies is sometimes required). Second, we can study computational

aspects of (iterated) dominance in concise representations of normal form games other than Bayesian games—for example, in graphical games [9] or local-effect/action graph games [12, 2]. (How to efficiently perform iterated very weak dominance has already been studied for partially observable stochastic games [8].) Finally, we can ask whether some of the algorithms we described (such as the one for iterated strict dominance with mixed strategies) can be made faster.

8. REFERENCES

- [1] Krzysztof R. Apt. Uniform proofs of order independence for various strategy elimination procedures. *Contributions to Theoretical Economics*, 4(1), 2004.
- [2] Nivan A. R. Bhat and Kevin Leyton-Brown. Computing Nash equilibria of action-graph games. In *UAI*, 2004.
- [3] Ben Blum, Christian R. Shelton, and Daphne Koller. A continuation method for Nash equilibria in structured games. In *IJCAI*, 2003.
- [4] Vincent Conitzer and Tuomas Sandholm. Complexity results about Nash equilibria. In *IJCAI*, pages 765–771, 2003.
- [5] Drew Fudenberg and Jean Tirole. *Game Theory*. MIT Press, 1991.
- [6] Itzhak Gilboa, Ehud Kalai, and Eitan Zemel. On the order of eliminating dominated strategies. *Operations Research Letters*, 9:85–89, 1990.
- [7] Itzhak Gilboa, Ehud Kalai, and Eitan Zemel. The complexity of eliminating dominated strategies. *Mathematics of Operation Research*, 18:553–565, 1993.
- [8] Eric A. Hansen, Daniel S. Bernstein, and Shlomo Zilberstein. Dynamic programming for partially observable stochastic games. In *AAAI*, pages 709–715, 2004.
- [9] Michael Kearns, Michael Littman, and Satinder Singh. Graphical models for game theory. In *UAI*, 2001.
- [10] Leonid Khachiyan. A polynomial algorithm in linear programming. *Soviet Math. Doklady*, 20:191–194, 1979.
- [11] Donald E. Knuth, Christos H. Papadimitriou, and John N Tsitsiklis. A note on strategy elimination in bimatrix games. *Operations Research Letters*, 7(3):103–107, 1988.
- [12] Kevin Leyton-Brown and Moshe Tennenholtz. Local-effect games. In *IJCAI*, 2003.
- [13] Richard Lipton, Evangelos Markakis, and Aranyak Mehta. Playing large games using simple strategies. In *ACM-EC*, pages 36–41, 2003.
- [14] Michael Littman and Peter Stone. A polynomial-time Nash equilibrium algorithm for repeated games. In *ACM-EC*, pages 48–54, 2003.
- [15] Leslie M. Marx and Jeroen M. Swinkels. Order independence for iterated weak dominance. *Games and Economic Behavior*, 18:219–245, 1997.
- [16] Leslie M. Marx and Jeroen M. Swinkels. Corrigendum, order independence for iterated weak dominance. *Games and Economic Behavior*, 31:324–329, 2000.
- [17] Andreu Mas-Colell, Michael Whinston, and Jerry R. Green. *Microeconomic Theory*. Oxford University Press, 1995.
- [18] Roger Myerson. *Game Theory: Analysis of Conflict*. Harvard University Press, Cambridge, 1991.
- [19] Martin J Osborne and Ariel Rubinstein. *A Course in Game Theory*. MIT Press, 1994.
- [20] Christos Papadimitriou. Algorithms, games and the Internet. In *STOC*, pages 749–753, 2001.
- [21] Ryan Porter, Eugene Nudelman, and Yoav Shoham. Simple search methods for finding a Nash equilibrium. In *AAAI*, pages 664–669, 2004.