Chapter 9

Computing Game-Theoretic Solutions

In the previous chapter, we saw that in certain settings, it is computationally hard for an agent to act in a strategically optimal way even when the agent already knows the actions of all the other agents—that is, even the *best-response* problem is hard. There are also many settings in which this is not the case, *i.e.* in which it is computationally easy to find a best response to specific actions of the other agents. However, there is much more to optimal strategic behavior than merely best-responding to given actions of the other agents. In most settings, the agents (1) do not know each other's types, but rather only a distribution over them; and (2) must somehow deduce the other agents' strategies, which map types to actions, using game-theoretic analysis (such as equilibrium reasoning). To some extent, we already discussed (1) in the previous chapter: we saw that uncertainty about other agents' votes makes the manipulation problem more difficult. But those results assumed that the manipulator(s) somehow knew a probability distribution over the other agents' votes. This is sweeping (2) under the rug, because to obtain such a distribution, some strategic assessment needs to be made about the strategies that the other agents are likely to use. The traditional assumption in mechanism design has been that strategic agents will play according to some solution concept (such as Bayes-Nash equilibrium), and if this assumption is accurate, then by the revelation principle, we can restrict our attention to truthful mechanisms. But what if such solutions are too hard for the agents to compute? If that is the case, then the agents cannot play according to these solutions, and the revelation principle loses its relevance.[1] Thus, the complexity of computing solutions according to these concepts becomes a key issue when considering how to design mechanisms for bounded agents. This chapter investigates that issue. (An even more difficult question is how a mechanism designer should proceed when solutions do turn out to be hard to compute, but that

[1] It should be emphasized here that it only loses its relevance in the sense that we may be able to achieve better results with non-truthful mechanisms (due to the agents' computational boundedness). However, the revelation principle still holds in the sense that using truthful mechanisms, we can achieve any result that we would have achieved under a non-truthful mechanism if agents *had* acted according to game-theoretic solution concepts (even if this would have been computationally infeasible for them). Thus, if we want to take the perspective that we want to help the agents act strategically optimally, and that we do not want them to feel any regret about having failed to misreport their preferences in the optimal way, then the revelation principle still applies and we may as well restrict our attention to truthful mechanisms—thereby relieving the agents of the burden of acting strategically. The view taken in this dissertation, however, is that there is nothing bad about an agent failing to manipulate the mechanism if the overall outcome is better as a result. This view is what motivates the interest in non-truthful mechanisms in this chapter.

remains outside the scope of this chapter.)

Certainly, intuitively, it seems that computing an equilibrium is typically much harder than computing a best response, and we have already seen in the previous chapter that the latter is hard in some complex settings. This chapter will therefore focus on more basic settings: in fact, it will focus only on normal-form games and Bayesian games that are *flatly represented* (all types and actions are listed explicitly). We note that even in (say) a straightforward voting setting, the type and action spaces are exponential in size, so that flat representation is not reasonable. (In fact, this is why finding a best response can be computationally hard in those settings. By contrast, computing best responses in flatly represented games is easy.) If computing game-theoretic solutions is hard even under flat representation, this makes it seem even more unlikely that agents will be able to compute such solutions in the richer settings that we are interested in.

If we are in fact able to compute certain game-theoretic solutions, that is of interest for other reasons as well. It can be helpful in predicting the outcomes of non-truthful mechanisms. It also allows us to build computer players for game-theoretically nontrivial games such as poker [Koller and Pfeffer, 1997; Shi and Littman, 2001; Billings *et al.*, 2003; Gilpin and Sandholm, 2006b,a] or potentially even RoboSoccer. Finally, it can also potentially be helpful in other settings where computer systems are interacting with other agents (human or computer) whose interests are not aligned with the computer system, such as surveillance and fraud detection.

The rest of this chapter is layed out as follows. In Section 9.1, we characterize the complexity of some basic computational questions about dominance and iterated dominance in both normal-form and Bayesian games [Conitzer and Sandholm, 2005c], and in Section 9.2 we do the same for Nash equilibrium [Conitzer and Sandholm, 2003e]. In Section 9.3 we provide a parameterized definition of strategy eliminability that is more general than dominance, and give an algorithm for computing whether a strategy is eliminable whose running time is exponential in only one parameter of the definition [Conitzer and Sandholm, 2005e].

9.1 Dominance and iterated dominance

While an ever-increasing amount of research focuses on computing Nash equilibria, the arguably simpler concept of (iterated) dominance has received much less attention. After an early short paper on a special case [Knuth *et al.*, 1988], the main computational study of these concepts has taken place in a paper in the game theory community [Gilboa *et al.*, 1993].[2] 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 section, 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. We study both strict and weak dominance, by both pure and mixed strategies, in both normal-form and Bayesian games.

[2] This is not to say that computer scientists have ignored dominance altogether. For example, simple dominance checks are sometimes used as a subroutine in searching for Nash equilibria [Porter *et al.*, 2004].

9.1.1 Dominance (not iterated)

In this subsection, we study the notion of one-shot (not iterated) dominance. 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 this section, 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.

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 pure strategies (at least as well against all the opponent's pure strategies, and strictly 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 [Myerson, 1991]; we present the result here for completeness, and because we will build on the linear programs given below in Theorem 98.)

Proposition 11 *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 [Khachiyan, 1979]. 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:

minimize $\sum_{d_r \in D_r} p_{d_r}$
such that
for all $\sigma_c \in \Sigma_c$, $\sum_{d_r \in D_r} p_{d_r} u_r(d_r, \sigma_c) \geq u_r(\sigma_r^*, \sigma_c)$.

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

[3] This 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.

[4] We 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.

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

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). Thus, we have strict dominance if and only if the linear program has a solution with objective value < 1.

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

maximize $\sum_{\sigma_c \in \Sigma_c} ((\sum_{d_r \in D_r} p_{d_r} u_r(d_r, \sigma_c)) - u_r(\sigma_r^*, \sigma_c))$

such that

for all $\sigma_c \in \Sigma_c$, $\sum_{d_r \in D_r} p_{d_r} u_r(d_r, \sigma_c) \geq u_r(\sigma_r^*, \sigma_c)$;

$\sum_{d_r \in D_r} p_{d_r} = 1$.

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 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. ■

9.1.2 Iterated dominance

We now move on to iterated dominance. It is well-known that iterated strict dominance is path-independent [Gilboa *et al.*, 1990; Osborne and Rubinstein, 1994]—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 Subsection 9.1.1), this whole procedure takes only polynomial time. In the case of iterated dominance by pure strategies with two players, Knuth *et al.* [1988] 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.

[6]There is, however, a restriction of weak dominance called *nice weak dominance* which is path-independent [Marx and Swinkels, 1997, 2000]. For an overview of path-independence results, see Apt [2004].

	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 52 *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 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 21 *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. ■

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

Theorem 93 *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 11). 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 $s^1_{+v}, s^2_{+v}, s^1_{-v}, s^2_{-v}$. Additionally, the row player has the following 2 strategies: s^1_0 and s^2_0, where $s^2_0 = \sigma^*_r$ (that is, it is the strategy we seek to eliminate). Finally, for every clause $c \in C$, the row player has corresponding strategies s^1_c (uneliminable) and s^2_c. 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^1_0 (uneliminable), t^2_0, and t_1.

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

- $u_r(s^1_{+v}, t_v) = 0$ for all $v \in V$;
- $u_r(s^2_{+v}, t_v) = 1$ for all $v \in V$;
- $u_r(s^1_{-v}, t_v) = 1$ for all $v \in V$;
- $u_r(s^2_{-v}, t_v) = 0$ for all $v \in V$;
- $u_r(s^1_{+v}, t_1) = 1$ for all $v \in V$;
- $u_r(s^2_{+v}, t_1) = 0$ for all $v \in V$;
- $u_r(s^1_{-v}, t_1) = 0$ for all $v \in V$;
- $u_r(s^2_{-v}, t_1) = 1$ for all $v \in V$;
- $u_r(s^b_{+v}, t_{+v}) = 1$ for all $v \in V$ and $b \in \{1, 2\}$;
- $u_r(s^b_{-v}, t_{-v}) = 1$ for all $v \in V$ and $b \in \{1, 2\}$;
- $u_r(s_l, t) = 0$ otherwise for all $l \in L$ and $t \in S_2$;
- $u_r(s^1_0, t_c) = 0$ for all $c \in C$;
- $u_r(s^2_0, t_c) = 1$ for all $c \in C$;
- $u_r(s^b_0, t^1_0) = 1$ for all $b \in \{1, 2\}$;
- $u_r(s^1_0, t^2_0) = 1$;
- $u_r(s^2_0, t^2_0) = 0$;
- $u_r(s^b_0, t) = 0$ otherwise for all $b \in \{1, 2\}$ and $t \in S_2$;
- $u_r(s^b_c, t) = 0$ 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 $s \in S_1$;
- $u_c(s_l^2, t_c) = 1$ 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 21, 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. ■

Using Theorem 93, it is now (relatively) easy to show that IWD-UNIQUE-SOLUTION is also NP-complete under the same restrictions.

Theorem 94 *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 nondeterministically 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 21, 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. From this, it follows that there exists a solution to the IWD-STRATEGY-ELIMINATION instance. ■

A slightly weaker version of the part of Theorem 94 concerning dominance by pure strategies only is the main result of Gilboa *et al.* [1993]. (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 21 cannot be applied to the original result to get the result for mixed strategies, giving us an additional motivation to prove the result for the case where utilities are in $\{0, 1\}$).)

9.1.3 (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 53 (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 (not necessarily a constant). 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 95 *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 11. 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. ∎

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 11. 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 Subsection 9.1.2 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 Subsection 9.1.2. 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 ith row and the column player plays the jth 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. ■

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

Theorem 96 *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 11). 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. ■

In the next subsection, 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.

9.1.4 (Iterated) dominance in Bayesian games

In this subsection, we study Bayesian games. Because Bayesian games have a representation that is exponentially more concise than their normal-form representation, questions that are easy for normal-form games can be hard for Bayesian games. In fact, it turns out that checking whether a strategy is dominated by a pure strategy is hard in Bayesian games.

Theorem 97 *In a Bayesian game, it is NP-complete to decide whether a given pure strategy $\sigma_r : \Theta_r \to A_r$ 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.

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. ■

However, it turns out that we can modify the linear programs from Proposition 11 to obtain a polynomial time algorithm for checking whether a strategy is dominated by a *mixed* strategy in Bayesian games.

Theorem 98 *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 11 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):

minimize $\sum_{\theta_r \in \Theta_r} \sum_{a_r \in A_r} p_r(a_r)$

such that

for all $a_c \in A_c$, $\sum_{\theta_r \in \Theta_r} \sum_{a_r \in A_r} \pi(\theta_r) u_r(\theta_r, a_r, a_c) p_r(\theta_r, a_r) \geq \sum_{\theta_r \in \Theta_r} \sum_{a_r \in A_r} \pi(\theta_r) u_r(\theta_r, a_r, a_c) p_r^{\sigma_r}(\theta_r, a_r)$;

for all $\theta_r \in \Theta_r$, $\sum_{a_r \in A_r} p_r(\theta_r, a_r) \leq 1$.

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 11.

For weak dominance, use the following linear program:

maximize $\sum_{a_c \in A_c} (\sum_{\theta_r \in \Theta_r} \sum_{a_r \in A_r} \pi(\theta_r) u_r(\theta_r, a_r, a_c) p_r(\theta_r, a_r) - \sum_{\theta_r \in \Theta_r} \sum_{a_r \in A_r} \pi(\theta_r) u_r(\theta_r, a_r, a_c) p_r^{\sigma_r}(\theta_r, a_r))$

such that

for all $a_c \in A_c$, $\sum_{\theta_r \in \Theta_r} \sum_{a_r \in A_r} \pi(\theta_r) u_r(\theta_r, a_r, a_c) p_r(\theta_r, a_r) \geq \sum_{\theta_r \in \Theta_r} \sum_{a_r \in A_r} \pi(\theta_r) u_r(\theta_r, a_r, a_c) p_r^{\sigma_r}(\theta_r, a_r)$;

for all $\theta_r \in \Theta_r$, $\sum_{a_r \in A_r} p_r(\theta_r, a_r) = 1$.

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 11. ■

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 99 *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 $\theta_i^1, \theta_i^2, \ldots, \theta_i^{n+1}$. 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,[7] 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 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 $\sigma_i(q)$, iterated elimination will require exponentially many iterations to complete. ∎

[7] Note 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.

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.* [1988] 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.

This concludes the part of this dissertation studying the complexity of dominance and iterated dominance. In the next section, we study the complexity of computing Nash equilibria.

9.2 Nash equilibrium

In recent years, there has been a large amount of research on computing Nash equilibria. The question of how hard it is to compute just a single Nash equilibrium especially drew attention, and 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 P today" [Papadimitriou, 2001]. A recent breakthrough series of papers [Daskalakis *et al.*, 2005; Chen and Deng, 2005a; Daskalakis and Papadimitriou, 2005; Chen and Deng, 2005b] shows that the problem is 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 suggests that the problem is indeed hard, although not as much is known about the class PPAD as about (say) NP. The best-known algorithm for finding a Nash equilibrium, the *Lemke-Howson* algorithm [Lemke and Howson, 1964], has recently been shown to have a worst-case exponential running time [Savani and von Stengel, 2004]. 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.*, 2004; Sandholm *et al.*, 2005b]. (These algorithms clearly require exponential time in the worst case, but are often quite fast in practice.) Also, there has been growing interest in computing equilibria of games with special structure that allows them to be represented concisely [Kearns *et al.*, 2001; Leyton-Brown and Tennenholtz, 2003; Blum *et al.*, 2003; Gottlob *et al.*, 2003; Bhat and Leyton-Brown, 2004; Schoenebeck and Vadhan, 2006].

In this section, we focus mostly on computing equilibria *with certain properties*: for example, computing an equilibrium with maximal social welfare, or one that places probability on a given pure strategy. We also consider the complexity of counting the number of equilibria and computing a pure-strategy Bayes-Nash equilibirium of a Bayesian game.

9.2.1 Equilibria with certain properties in normal-form games

When one analyzes the strategic structure of a game, especially from the viewpoint of a mechanism designer who tries to construct good rules for a game, finding a single equilibrium is far from satisfactory. More desirable equilibria may exist: in this case the game becomes more attractive, especially if one can coax the players into playing a desirable equilibrium. Also, less desirable equilibria may exist: in this case the game becomes less attractive (if there is some chance that these equilibria will end up being played). Before we can make a definite judgment about the quality of the game, we would like to know the answers to questions such as: What is the game's most desirable equilibrium? Is there a unique equilibrium? If not, how many equilibria are there? Algorithms that tackle these questions would be useful both to players and to the mechanism designer.

Furthermore, algorithms that answer certain existence questions may pave the way to designing algorithms that construct a Nash equilibrium. For example, if we had an algorithm that told us whether there exists any equilibrium where a certain player plays a certain strategy, this could be useful in eliminating possibilities in the search for a Nash equilibrium.

However, all the existence questions that we have investigated turn out to be NP-hard. These are not the first results of this nature; most notably, Gilboa and Zemel [1989] provide some NP-hardness results in the same spirit. We provide a single reduction which in demonstrates (sometimes stronger versions of) most of their hardness results, and interesting new results. More significantly, as we show in Subsection 9.2.2, our reduction shows that the problems of maximizing certain properties of Nash equilibria are *inapproximable* (unless P=NP). Additionally, as we show in Subsection 9.2.3, the reduction shows #P-hardness of counting the number of equilibria.

We now present our reduction.[8]

Definition 54 *Let ϕ be a Boolean formula in conjunctive normal form. 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)*[9]*, 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 symmetric 2-player game in normal form. Let $\Sigma \equiv \Sigma_1 = \Sigma_2 = L \cup V \cup C \cup \{f\}$. 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* $x \in \Sigma - \{f\}$*;*
- $u_1(f, f) = u_2(f, f) = \epsilon$*;*

[8]The reduction presented here is somewhat different from the reduction given in the IJCAI version of this work. The reason is that the new reduction presented here implies inapproximability results that the original reduction did not.

[9]Thus, if x_1 is a variable, x_1 and $-x_1$ are literals. We make a distinction between the variable x_1 and the literal x_1.

- $u_1(f, x) = u_2(x, f) = n - 1$ *for all* $x \in \Sigma - \{f\}$.

Theorem 100 *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 its 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. ■

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. *Indeed, 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. So the example is consistent with Theorem 100.*

Thus, in general, 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. So, it follows that determining whether a "good" equilibrium exists is 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, so even finding out whether such an optimal equilibrium exists is hard. The corollaries below illustrate these points.

All the corollaries show NP-completeness of a problem, meaning that the problem is both NP-hard and in NP. Technically, only the NP-hardness part is a corollary of Theorem 100 in each case. Membership in NP follows in each case because we can nondeterministically generate strategies for the players, and verify whether these constitute a Nash equilibrium with the desired property.

Alternatively, for the case of two players, we can nondeterministically generate only the supports of the players' strategies. At this point, determining whether a Nash equilibrium with the given supports exists is a simple linear feasibility program (see, for example, Dickhaut and Kaplan [1991]; Porter *et al.* [2004]), to which we can add an objective to maximize (such as, for example, social welfare). The resulting linear program can be solved in polynomial time [Khachiyan, 1979].

Corollary 9 *Even in symmetric 2-player games, it is NP-complete to determine whether there exists a NE with expected (standard) social welfare ($E[\sum_{1 \leq i \leq |A|} u_i]$) at least k, even when k is the maximum social welfare that could be obtained in the game.*

Proof: For any ϕ, in $G_\epsilon(\phi)$, the social welfare of a Nash equilibrium corresponding to any satisfying assignment is $2(n-1)$. On the other hand, the social welfare of the Nash equilibrium that always exists is only 2ϵ. Thus, for $\epsilon < 1$ and $n \geq 2$, $G_\epsilon(\phi)$ has a Nash equilibrium with a social welfare of at least $2(n-1)$ if and only if ϕ is satisfiable. ■

Corollary 10 *Even in symmetric 2-player games, it is NP-complete to determine whether there exists a NE where all players have expected utility at least k (that is, the* egalitarian social welfare *is at least k), even when k is the largest number such that there exists a distribution over outcomes of the game such that all players have expected utility at least k.*

Proof: For any ϕ, in $G_\epsilon(\phi)$, the egalitarian social welfare of a Nash equilibrium corresponding to any satisfying assignment is $n-1$. On the other hand, the egalitarian social welfare of the Nash equilibrium that always exists is only ϵ. Thus, for $\epsilon < 1$ and $n \geq 2$, $G_\epsilon(\phi)$ has a Nash equilibrium with an egalitarian social welfare of at least $n-1$ if and only if ϕ is satisfiable. ■

Corollary 11 *Even in symmetric 2-player games, it is NP-complete to determine whether there exists a* Pareto-optimal *NE. (A distribution over outcomes is Pareto-optimal if there is no other distribution over outcomes such that every player has at least equal expected utility, and at least one player has strictly greater expected utility.)*

Proof: For $\epsilon < 1$ and $n \geq 2$, 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. ■

Corollary 12 *Even in symmetric 2-player games, it is NP-complete to determine whether there exists a NE where player 1 has expected utility at least k.*

Proof: For any ϕ, in $G_\epsilon(\phi)$, player 1's utility in a Nash equilibrium corresponding to any satisfying assignment is $(n-1)$. On the other hand, player 1's utility in the Nash equilibrium that always exists is only ϵ. Thus, for $\epsilon < 1$ and $n \geq 2$, $G_\epsilon(\phi)$ has a Nash equilibrium with a utility for player 1 of at least $n-1$ if and only if ϕ is satisfiable. ■

Some additional corollaries are:

Corollary 13 *Even in symmetric 2-player games, it is 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. ■

Corollary 14 *Even in symmetric 2-player games, it is NP-complete to determine whether there is an 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 NP-complete. ■

Corollary 15 *Even in symmetric 2-player games, it is NP-complete to determine whether there is an 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. ■

Corollary 16 *Even in symmetric 2-player games, it is NP-complete to determine whether there is an equilibrium where player 1's strategy has at least k pure strategies in its support (even when $k = 2$).*

Proof: For any ϕ, in $G_\epsilon(\phi)$, any Nash equilibrium corresponding to a satisfying assignment uses a support of n strategies for player 1. On the other hand, the Nash equilibrium that always exists uses a support of only 1 strategy for player 1. Thus, for $n \geq 2$, $G_\epsilon(\phi)$ has a Nash equilibrium using a support of at least 2 strategies for player 1 if and only if ϕ is satisfiable. ■

Corollary 17 *Even in symmetric 2-player games, it is NP-complete to determine whether there is an equilibrium where the players' strategies together have at least k pure strategies in their supports (even when $k = 3$).*

Proof: For any ϕ, in $G_\epsilon(\phi)$, any Nash equilibrium corresponding to a satisfying assignment uses a support of n strategies for each player, for a total of $2n$ strategies. On the other hand, the Nash equilibrium that always exists uses a support of only 1 strategy for each player, for a total of only 2 strategies. Thus, for $n \geq 2$, $G_\epsilon(\phi)$ has a Nash equilibrium using at least 3 strategies in the supports of the players if and only if ϕ is satisfiable. ■

Corollary 18 *Even in symmetric 2-player games, it is NP-complete to determine whether there is an equilibrium where each player's strategy has at least k pure strategies in its support (even when $k = 2$).*

Proof: For any ϕ, in $G_\epsilon(\phi)$, any Nash equilibrium corresponding to a satisfying assignment uses a support of n strategies for each player. On the other hand, the Nash equilibrium that always exists uses a support of only 1 strategy for each player. Thus, for $n \geq 2$, $G_\epsilon(\phi)$ has a Nash equilibrium using at least 2 strategies in the supports of each player if and only if ϕ is satisfiable. ■

Definition 55 *A* strong Nash equilibrium *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 19 *Even in symmetric 2-player games, it is NP-complete to determine whether a strong Nash equilibrium exists.*

Proof: For $\epsilon < 1$ and $n \geq 2$, 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. ■

All of these results indicate that it is hard to obtain summary information about a game's Nash equilibria. (Corollaries 13, 18, and weaker[10] versions of Corollaries 10, 14, and 15 were first proven by Gilboa and Zemel [1989].)

9.2.2 Inapproximability results

Some of the corollaries of the previous subsection state that it is NP-complete to find the Nash equilibrium that maximizes a certain property (such as social welfare). For such properties, an important question is to ask whether they can be approximated. For instance, 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—the same question, asked nonconstructively—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 nonconstructive question does not require constructing a Nash equilibrium, so it is perhaps possible that there is a polynomial-time algorithm for this question even if it is hard to construct any Nash equilibrium.) We will not give approximation algorithms in this subsection, but we will derive certain inapproximability results from Theorem 100. In each case, we will show that even the nonconstructive question is hard (and therefore also the constructive question).

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 approximation 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 results, as follows. Suppose it returned a social welfare of $2r$ on a game, and suppose it were 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 provided by the approximation algorithm on the original game corresponds

[10] Our results prove hardness in a slightly more restricted setting. Corollaries 14 and 15 in their full strength can in fact also be obtained using Gilboa and Zemel's proof technique, even though they stated the result in a weaker form.

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. To do so, we will simply use the fact that $G_\epsilon(\phi)$ satisfies this property whenever $n \geq 4$. (We recall that n is the number of variables in ϕ.)

We are now ready to present our results. The first one is a stronger version of Corollary 9.

Corollary 20 *Unless P = NP, there does not exist a polynomial-time algorithm that approximates (to any positive ratio) the maximum social welfare obtained by a Nash equilibrium, even in symmetric 2-player games. (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 $n \geq 4$), 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, by Theorem 100, 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 100, 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. ■

The next result is a stronger version of Corollary 10.

Corollary 21 *Unless P = NP, there does not exist a polynomial-time algorithm that approximates (to any positive ratio) the maximum egalitarian social welfare (minimum utility) obtained by a Nash equilibrium, even in symmetric 2-player games. (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 $n \geq 4$), consider the game $G_\epsilon(\phi)$ where ϵ is set so that $\epsilon < r(n-1)$ (here, r is the approximation ratio that the algorithm guarantees for games of the size of $G_\epsilon(\phi)$). If ϕ is satisfiable, by Theorem 100, there exists an equilibrium with egalitarian social welfare $n-1$, and thus the approximation algorithm should return an egalitarian social welfare of at least $r(n-1) > \epsilon$. Otherwise, by Theorem 100, the only equilibrium has egalitarian social welfare ϵ, and thus the approximation algorithm should return an egalitarian social welfare of at most ϵ. Thus we can use the algorithm to solve arbitrary SAT instances. ■

The next result is a stronger version of Corollary 12.

Corollary 22 *Unless P = NP, there does not exist a polynomial-time algorithm that approximates (to any positive ratio) the maximum utility for player 1 obtained by a Nash equilibrium, even in symmetric 2-player games. (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 $n \geq 4$), consider the game $G_\epsilon(\phi)$ where ϵ is set so that $\epsilon < r(n-1)$ (here, r is the approximation ratio that the algorithm guarantees for games of the size of $G_\epsilon(\phi)$). If ϕ is satisfiable, by Theorem 100, there exists an equilibrium with a utility of $n-1$ for player 1, and thus the approximation algorithm should return a utility of at least $r(n-1) > \epsilon$. Otherwise, by Theorem 100, the only equilibrium has a utility of ϵ for player 1, and thus the approximation algorithm should return a utility of at most ϵ. Thus we can use the algorithm to solve arbitrary SAT instances. ■

The next result is a stronger version of Corollary 16.

Corollary 23 *Unless P = NP, there does not exist a polynomial-time algorithm that approximates (to any ratio $o(|\Sigma|)$) the maximum number of pure strategies in player 1's support in a Nash equilibrium, 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, by Theorem 100, the only equilibrium has only one pure strategy in player 1's support, and thus the algorithm can return a maximum support size of at most 1. On the other hand, if ϕ is satisfiable, by Theorem 100, there is an equilibrium where player 1's support has size $\Omega(|\Sigma|)$. Because by assumption our approximation algorithm has an approximation ratio of $o(|\Sigma|)$, this means that for large enough $|\Sigma|$, the approximation ratio must return a support size strictly greater than 1. Thus we can use the algorithm to solve arbitrary SAT instances (given that the instances are large enough to produce large enough $|\Sigma|$). ■

The next result is a stronger version of Corollary 17.

Corollary 24 *Unless P = NP, there does not exist a polynomial-time algorithm that approximates (to any ratio $o(|\Sigma|)$) the maximum number of pure strategies in the players' strategies' supports in a Nash equilibrium, 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, by Theorem 100, 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, by Theorem 100, there is an equilibrium where each player's support has size $\Omega(|\Sigma|)$. Because by assumption our approximation algorithm has an approximation ratio of $o(|\Sigma|)$, this means that for large enough $|\Sigma|$, the approximation ratio 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 $|\Sigma|$). ■

The next result is a stronger version of Corollary 18.

Corollary 25 *Unless P = NP, there does not exist a polynomial-time algorithm that approximates (to any ratio $o(|\Sigma|)$) 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: 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, by Theorem 100, 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

1. On the other hand, if ϕ is satisfiable, by Theorem 100, there is an equilibrium where each player's support has size $\Omega(|\Sigma|)$. Because by assumption our approximation algorithm has an approximation ratio of $o(|\Sigma|)$, this means that for large enough $|\Sigma|$, the approximation ratio must return a support size strictly greater than 1. Thus we can use the algorithm to solve arbitrary SAT instances(given that the instances are large enough to produce large enough $|\Sigma|$). ∎

9.2.3 Counting the number of equilibria in normal-form games

Existence questions do not tell the whole story. In general, we are interested in characterizing all the equilibria of a game. One rather weak such characterization is the number of equilibria[11]. We can use Theorem 100 to show that even determining this number in a given normal-form game is hard.

Corollary 26 *Even in symmetric 2-player games, counting the number of Nash equilibria is #P-hard.*

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

It is easy to construct games where there is a continuum of Nash equilibria. In such games, it is more meaningful to ask how many distinct continuums of equilibria there are. More formally, one can ask how many maximal connected sets of equilibria a game has (a maximal connected set is a connected set which is not a proper subset of a connected set).

Corollary 27 *Even in symmetric 2-player games, counting the number of maximal connected sets of Nash equilibria is #P-hard.*

Proof: Every Nash equilibrium in $G_\epsilon(\phi)$ constitutes a maximal connected set by itself, so the number of maximal connected sets is the number of satisfying assignments to the variables of ϕ, plus one. ∎

The most interesting #P-hardness results are the ones where the corresponding existence and search questions are easy, such as counting the number of perfect bipartite matchings. In the case of Nash equilibria, the existence question is trivial: it has been analytically shown (by Kakutani's fixed point theorem) that a Nash equilibrium always exists [Nash, 1950]. The complexity of the search question remains open.

9.2.4 Pure-strategy Bayes-Nash equilibria

Equilibria in pure strategies are particularly desirable because they avoid the uncomfortable requirement that players randomize over strategies among which they are indifferent [Fudenberg and

[11] 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, 1999].

Tirole, 1991]. In normal-form games with small numbers of players, 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.[12] However, this is not feasible in Bayesian games, where the players have private information about their own preferences (represented by types). Here, players may condition their actions on their types, so the strategy space of each player is exponential in the number of types.

In this subsection, we show that the question of whether a pure-strategy Bayes-Nash equilibrium exists is in fact NP-hard even in symmetric two-player games.

We study the following computational problem.

Definition 56 (PURE-STRATEGY-BNE) *We are given a Bayesian game. We are asked whether there exists a Bayes-Nash equilibrium (BNE) where all the strategies σ_{i,θ_i} are pure.*

To show our NP-hardness result, we will reduce from the SET-COVER problem.

Definition 57 (SET-COVER) *We are given a set $S = \{s_1, \ldots, s_n\}$, subsets $S_1, S_2, \ldots, S_m$ of S with $\bigcup_{1 \leq i \leq m} S_i = S$, and an integer k. We are asked whether there exist $S_{c_1}, S_{c_2}, \ldots, S_{c_k}$ such that $\bigcup_{1 \leq i \leq k} S_{c_i} = S$.*

Theorem 101 *PURE-STRATEGY-BNE is NP-complete, even in symmetric 2-player games where the priors over types are uniform.*

Proof: To show membership in NP, we observe that we can nondeterministically choose a pure strategy for each type for each player, and verify whether these constitute a BNE.

To show NP-hardness, we reduce an arbitrary SET-COVER instance to the following PURE-STRATEGY-BNE instance. Let there be two players, with $\Theta \equiv \Theta_1 = \Theta_2 = \{\theta^1, \ldots, \theta^k\}$. The priors over types are uniform. Furthermore, $\Sigma \equiv \Sigma_1 = \Sigma_2 = \{S_1, S_2, \ldots, S_m, s_1, s_2, \ldots, s_n\}$. The utility functions we choose in fact 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 $s_j \in S_i$.

[12] Computing pure-strategy Nash equilibria for more concise representations of normal-form games has been systematically studied [Gottlob *et al.*, 2003; Schoenebeck and Vadhan, 2006].

We now show the two instances are equivalent. First suppose there exist $S_{c_1}, S_{c_2}, \ldots, S_{c_k}$ such that $\bigcup_{1 \leq i \leq k} S_{c_i} = S$. 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 it a utility of at least $\frac{1}{k}2 + \frac{k-1}{k}1 = 1 + \frac{1}{k}$. Since it must do at least this well in the equilibrium, and it never plays elements of S, it 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 it would be better off playing some S_i instead. (Contradiction.) It follows that in no pure-strategy BNE, any element of S is ever played.

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. (Contradiction.) Hence, there exists a set cover. ■

If one allows for general mixed strategies, a Bayes-Nash equilibrium always exists [Fudenberg and Tirole, 1991]. Computing a single mixed-strategy Bayes-Nash equilibrium is of course at least as hard as computing a single mixed-strategy Nash equilibrium in a normal-form game (since that is the special case where each agent has a single type).

This concludes the part of this dissertation studying the complexity of computing Nash equilibria. The next section introduces a parameterized strategy eliminability criterion that generalizes both dominance and Nash equilibrium, and studies how hard it is to apply computationally.

9.3 A generalized eliminability criterion

The concept of (iterated) dominance is often too strong for the purpose of solving games: it cannot eliminate enough strategies. But, if possible, we would like a stronger argument for eliminating a strategy than (mixed-strategy) Nash equilibrium. Similarly, in mechanism design (where one gets to create the game), implementation in dominant strategies is often excessively restrictive, but implementation in (Bayes-)Nash equilibrium may not be sufficiently strong for the designer's purposes. Hence, it is desirable to have eliminability criteria that are *between* these concepts in strength. In this section, we will introduce such a criterion. This criterion considers whether a given strategy is eliminable relative to given dominator & eliminee subsets of the players' strategies. The criterion spans an entire *spectrum* of strength between Nash equilibrium and strict dominance (in terms of which strategies it can eliminate), and in the extremes can be made to coincide with either of these two concepts, depending on how the dominator & eliminee sets are set. It can also be used for iterated elimination of strategies. We will also study the computational complexity of applying the new eliminability criterion, and provide a mixed integer programming approach for it.

One of the benefits of the new criterion is that when a strategy cannot be eliminated by dominance (but it can be eliminated by the Nash equilibrium concept), the new criterion may provide a stronger argument than Nash equilibrium for eliminating the strategy, by using dominator & eliminee sets smaller than the entire strategy set. To get the strongest possible argument for eliminating a strategy, the dominator & eliminee sets should be chosen to be as small as possible while still having the strategy be eliminable relative to these sets.[13] Iterated elimination of strategies using the new criterion is also possible, and again, to get the strongest possible argument for eliminating a strategy, the sequence of eliminations leading up to it should use dominator & eliminee sets that are as small as possible.[14]

As another benefit, the algorithm that we provide for checking whether a strategy is eliminable according to the new criterion can also be used as a subroutine in the computation of Nash equilibria. Specifically, any strategy that is eliminable (even using iterated elimination) according to the criterion is guaranteed not to occur in any Nash equilibrium. Current state-of-the-art algorithms for computing Nash equilibria already use a subroutine that eliminates (conditionally) dominated strategies [Porter *et al.*, 2004]. Because the new criterion can eliminate more strategies than dominance, the algorithm we provide may speed up the computation of Nash equilibria. (For purposes of speed, it is probably desirable to only apply special cases of the criterion that can be computed fast—in particular, as we will show, eliminability according to the criterion can be computed fast when the eliminee sets are small. Even these special cases are more powerful than dominance.)

Throughout, we focus on two-player games only. The eliminability criterion itself can be generalized to more players, but the computational tools we introduce do not straightforwardly generalize to more players. Moreover, we restrict attenton to normal-form games only.

9.3.1 A motivating example

Because the definition of the new eliminability criterion is complex, we will first illustrate it with an example. Consider the following (partially specified) game.

	σ_c^1	σ_c^2	σ_c^3	σ_c^4
σ_r^1	?, ?	?, 2	?, 0	?, 0
σ_r^2	2, ?	2, 2	2, 0	2, 0
σ_r^3	0, ?	0, 2	3, 0	0, 3
σ_r^4	0, ?	0, 2	0, 3	3, 0

[13] There may be multiple minimal vectors of dominator & eliminee sets relative to which the strategy is eliminable; in this dissertation, we will not attempt to settle which of these minimal vectors, if any, constitutes the most powerful argument for eliminating the strategy.

[14] Here, there may also be a tradeoff with the length of the elimination path. For example, there may be a path of several eliminations using dominator & eliminee sets that are small, as well as a single elimination using dominator & eliminee sets that are large, both of which eliminate a given strategy. (In fact, we will *always* be confronted with this situation, as Corollary 30 will show.) Again, in this dissertation, we will not attempt to settle which argument for eliminating the strategy is stronger.

A quick look at this game reveals that strategies σ_r^3 and σ_r^4 are both *almost* dominated by σ_r^2—but they perform better than σ_r^2 against σ_c^3 and σ_c^4, respectively. Similarly, strategies σ_c^3 and σ_c^4 are both almost dominated by σ_c^2—but they perform better than σ_c^2 against σ_r^4 and σ_c^3, respectively. So we are unable to eliminate any strategies using (even weak) dominance.

Now consider the following reasoning. In order for it to be worthwhile for the row player to ever play σ_r^3 rather than σ_r^2, the column player should play σ_c^3 at least $\frac{2}{3}$ of the time. (If it is exactly $\frac{2}{3}$, then switching from σ_r^2 to σ_r^3 will cost the row player 2 exactly $\frac{1}{3}$ of the time, but the row player will gain 1 exactly $\frac{2}{3}$ of the time, so the expected benefit is 0.) But, similarly, in order for it to be worthwhile for the column player to ever play σ_c^3, the row player should play σ_r^4 at least $\frac{2}{3}$ of the time. But again, in order for it to be worthwhile for the row player to ever play σ_r^4, the column player should play σ_c^4 at least $\frac{2}{3}$ of the time. Thus, if both the row and the column player accurately assess the probabilities that the other places on these strategies, and their strategies are rational with respect to these assessments (as would be the case in a Nash equilibrium), then, if the row player puts positive probability on σ_r^3, by the previous reasoning, the column player should be playing σ_c^3 at least $\frac{2}{3}$ of the time, and σ_c^4 at least $\frac{2}{3}$ of the time. Of course, this is impossible; so, in a sense, the row player should not play σ_r^3.

It may appear that all we have shown is that σ_r^3 is not played in any Nash equilibrium. But, to some extent, our argument for not playing σ_r^3 did not make use of the full elimination power of the Nash equilibrium concept. Most notably, we only reasoned about a small part of the game: we never mentioned strategies σ_r^1 and σ_c^1, and we did not even specify most of the utilities for these strategies. (It is easy to extend this example so that the argument only uses an arbitrarily small fraction of the strategies and of the utilities in the matrix, for instance by adding many copies of σ_r^1 and σ_c^1.) The locality of the reasoning that we did is more akin to the notion of dominance, which is perhaps the extreme case of local reasoning about eliminability—only two strategies are mentioned in it. So, in this sense, the argument for eliminating σ_r^3 is somewhere between dominance and Nash equilibrium in strength.

9.3.2 Definition of the eliminability criterion

We are now ready to give the formal definition of the generalized eliminability criterion. To make the definition a bit simpler, we define its negation—when a strategy is *not* eliminable relative to certain sets of strategies. Also, we only define when one of the *row player's* strategies is eliminable, but of course the definition is analogous for the column player.

The definition, which considers when a strategy e_r^* is eliminable relative to subsets D_r, E_r of the row player's pure strategies (with $e_r^* \in E_r$) and subsets D_c, E_c of the column player's pure strategies, can be stated informally as follows. To protect e_r^* from elimination, we should be able to specify the probabilities that the players' mixed strategies place on the E_i sets in such a way that 1) e_r^* receives nonzero probability, and 2) for every pure strategy e_i that receives nonzero probability, for every mixed strategy d_i using only strategies in D_i, it is conceivable that player $-i$'s mixed strategy[15] is completed so that e_i is no worse than d_i.[16] The formal definition follows.

[15] As is common in the game theory literature, $-i$ denotes "the player other than i."

[16] This description may sound similar to the concept of *rationalizability*. However, in two-player games (the subject of this section), rationalizability is known to coincide with iterated strict dominance [Pearce, 1984].

Definition 58 *Given a two-player game in normal form, subsets D_r, E_r of the row player's pure strategies Σ_r, subsets D_c, E_c of the column player's pure strategies Σ_c, and a distinguished strategy $e_r^* \in E_r$, we say that e_r^* is* not eliminable relative to *D_r, E_r, D_c, E_c, if there exist functions (partial mixed strategies) $p_r : E_r \rightarrow [0,1]$ and $p_c : E_c \rightarrow [0,1]$ with $p_r(e_r^*) > 0$, $\sum_{e_r \in E_r} p_r(e_r) \leq 1$, and $\sum_{e_c \in E_c} p_c(e_c) \leq 1$, such that the following holds. For both $i \in \{r, c\}$, for any $e_i \in E_i$ with $p_i(e_i) > 0$, for any mixed strategy d_i placing positive probability only on strategies in D_i, there is some pure strategy $\sigma_{-i} \in \Sigma_{-i} - E_{-i}$ such that (letting $p_{-i} \diamond \sigma_{-i}$ denote the mixed strategy that results from placing the remaining probability $1 - \sum_{e_{-i} \in E_{-i}} p_{-i}(e_{-i})$ that is not used by the partial mixed strategy p_{-i} on σ_{-i}), we have: $u_i(e_i, p_{-i} \diamond \sigma_{-i}) \geq u_i(d_i, p_{-i} \diamond \sigma_{-i})$. (If p_{-i} already uses up all the probability, we simply have $u_i(e_i, p_{-i}) \geq u_i(d_i, p_{-i})$—no σ_{-i} needs to be chosen.)*[17]

In the example from the previous subsubsection, we can set $D_r = \{\sigma_r^2\}$, $D_c = \{\sigma_c^2\}$, $E_r = \{\sigma_r^3, \sigma_r^4\}$, $E_c = \{\sigma_c^3, \sigma_c^4\}$, and $e_r^* = \sigma_r^3$. Then, by the reasoning that we did, it is impossible to set p_r and p_c so that the conditions are satisfied, and hence σ_r^3 is eliminable relative to these sets.

9.3.3 The spectrum of strength

In this subsection we show that the generalized eliminability criterion we defined in in the previous subsection spans a spectrum of strength all the way from Nash equilibrium (when the sets D_r, E_r, D_c, E_c are chosen as large as possible), to strict dominance (when the sets are chosen as small as possible). First, we show that the criterion is monotonically increasing, in the sense that the larger we make the sets D_r, E_r, D_c, E_c, the more strategies are eliminable.

Proposition 12 *If e_r^* is eliminable relative to $D_r^1, E_r^1, D_c^1, E_c^1$, and $D_r^1 \subseteq D_r^2, E_r^1 \subseteq E_r^2, D_c^1 \subseteq D_c^2, E_c^1 \subseteq E_c^2$, then e_r^* is eliminable relative to $D_r^2, E_r^2, D_c^2, E_c^2$.*

Proof: We will prove this by showing that if e_r^* is not eliminable relative to $D_r^2, E_r^2, D_c^2, E_c^2$, then e_r^* is not eliminable relative to $D_r^1, E_r^1, D_c^1, E_c^1$. It is straightforward that making the D_i sets smaller only weakens the condition on strategies e_i with $p_i(e_i) > 0$ in Definition 58. Hence, if e_r^* is not eliminable relative to $D_r^2, E_r^2, D_c^2, E_c^2$, then e_r^* is not eliminable relative to $D_r^1, E_r^2, D_c^1, E_c^2$. All that remains to show is that making the E_i sets smaller will not make e_r^* eliminable. To show this, we first observe that, if in its last step Definition 58 allowed for distributing the remaining probability arbitrarily over the strategies in $\Sigma_{-i} - E_{-i}$ (rather than requiring a single one of these strategies to receive all the remaining probability), this would not change the definition, because we might as well place all the remaining probability on the strategy $\sigma_{-i} \in \Sigma_{-i} - E_{-i}$ that maximizes $u_i(e_i, \sigma_{-i}) - u_i(d_i, \sigma_{-i})$. Now, let p_r and p_c be partial mixed strategies over E_r^2 and E_c^2 that prove that e_r^* is not eliminable relative to $D_r^1, E_r^2, D_c^1, E_c^2$. Then, to show that e_r^* is not eliminable relative to $D_r^1, E_r^1, D_c^1, E_c^1$, use the partial mixed strategies p_r' and p_c', which are simply the restrictions of p_r and p_c to E_r^1 and E_c^1, respectively. For any $e_i \in E_i^1$ with $p_i'(e_i) > 0$ and for any mixed strategy d_i over D_i^1, we know that there exists some $\sigma_{-i} \in \Sigma_{-i} - E_{-i}^2$ such that $u_i(e_i, p_{-i} \diamond \sigma_{-i}) \geq u_i(d_i, p_{-i} \diamond \sigma_{-i})$ (because the p_i prove that e_r^* is not eliminable relative to $D_r^1, E_r^2, D_c^1, E_c^2$). But,

[17] We need to make this case explicit for the case $E_{-i} = \Sigma_{-i}$.

the distribution $p_{-i} \diamond \sigma_{-i}$ is a legitimate completion of the partial mixed strategy p'_{-i} as well (albeit one that distributes the remaining probability over multiple strategies), and hence the p'_i prove that e^*_r is not eliminable relative to $D^1_r, E^1_r, D^1_c, E^1_c$. ■

Next, we show that the Nash equilibrium concept is weaker[18] than our generalized eliminability criterion—in the sense that the generalized criterion can never eliminate a strategy that is in some Nash equilibrium. So, if a strategy can be eliminated by the generalized criterion, it can be eliminated by the Nash equilibrium concept.

Proposition 13 *If there is some Nash equilibrium that places positive probability on pure strategy σ^*_r, then σ^*_r is not eliminable relative to any D_r, E_r, D_c, E_c.*

Proof: Let σ'_r be the row player's (mixed) strategy in the Nash equilibrium (which places positive probability on σ^*_r), and let σ'_c be the column player's (mixed) strategy in the Nash equilibrium. For any D_r, E_r, D_c, E_c with $\sigma^*_r \in E_r$, to prove that σ^*_r is not eliminable relative to these sets, simply let p_r coincide with σ'_r on E_r—that is, let p_r be the probabilities that the row player places on the strategies in E_r in the equilibrium. (Thus, $p_r(\sigma^*_r) > 0$). Similarly, let p_c coincide with σ'_c on E_c. We will prove that the condition on strategies with positive probability is satisfied for the row player; the case of the column player follows by symmetry. For any $e_r \in E_r$ with $p_r(e_r) > 0$, for any mixed strategy d_r, we have $u_r(e_r, \sigma'_c) - u_r(d_r, \sigma'_c) \geq 0$, by the Nash equilibrium condition. Now, let pure strategy $\sigma_c \in \arg\max_{\sigma \in \Sigma_c - E_c}(u_r(e_r, p_c \diamond \sigma) - u_r(d_r, p_c \diamond \sigma))$. Then we must have $u_r(e_r, p_c \diamond \sigma_c) - u_r(d_r, p_c \diamond \sigma_c) \geq u_r(e_r, \sigma'_c) - u_r(d_r, \sigma'_c) \geq 0$ (because $p_c \diamond \sigma_c$ and σ'_c coincide on E_c, and for the former, the remainder of the distribution is chosen to maximize this expression). It follows that σ^*_r is not eliminable relative to any D_r, E_r, D_c, E_c. ■

We next show that by choosing the sets D_r, E_r, D_c, E_c as large as possible, we can make the generalized eliminability criterion coincide with the Nash equilibrium concept.[19]

Proposition 14 *Let $D_r = E_r = \Sigma_r$ and $D_c = E_c = \Sigma_c$. Then e^*_r is eliminable relative to these sets if and only if there is no Nash equilibrium that places positive probability on e^*_r.*

Proof: The "only if" direction follows from Proposition 13. For the "if" direction, suppose e^*_r is not eliminable relative to $D_r = E_r = \Sigma_r$ and $D_c = E_c = \Sigma_c$. The partial distributions p_r and p_c with $p_r(e^*_r) > 0$ that show that e^*_r is not eliminable must use up all the probability (the probabilities must sum to one), because there are no strategies outside $E_c = \Sigma_c$ and $E_r = \Sigma_r$ to place any remaining probability on. Hence, we must have, for any strategy $e_r \in E_r = \Sigma_r$ with $p_r(e_r) > 0$, that for any mixed strategy d_r, $u_r(e_r, p_c) \geq u_r(d_r, p_c)$ (and the same for the column player). But these are precisely the conditions for p_r and p_c to constitute a Nash equilibrium. It follows that there is a

[18]When discussing elimination of strategies, it is tempting to say that the stronger criterion is the one that can eliminate more strategies. However, when discussing solution concepts, the convention is that the stronger concept is the one that implies the other. Therefore, the criterion that can eliminate fewer strategies is actually the stronger one. For example, strict dominance is stronger than weak dominance, even though weak dominance can eliminate more strategies.

[19]Unlike Nash equilibrium, the generalized eliminability criterion does not discuss what probabilities should be placed on strategies that are not eliminated, so it only "coincides" with Nash equilibrium in terms of what it can eliminate.

Nash equilibrium with positive probability on e_r^*. ■

Moving to the other side of the spectrum, we now show that the concept of strict dominance is stronger than the generalized eliminability criterion—in the sense that the generalized eliminability criterion can always eliminate a strictly dominated strategy (as long as the dominating strategy is in D_r).

Proposition 15 *If pure strategy σ_r^* is strictly dominated by some mixed strategy d_r, then σ_r^* is eliminable relative to any D_r, E_r, D_c, E_c such that 1) $\sigma_r^* \in E_r$, and 2) all the pure strategies on which d_r places positive probability are in D_r.*

Proof: To show that σ_r^* is not eliminable relative to these sets, we must set $p_r(\sigma_r^*) > 0$, and thus we must demonstrate that for some pure strategy $\sigma_c \in \Sigma_c - E_c$, $u_r(\sigma_r^*, p_c \diamond \sigma_c) \geq u_r(d_r, p_c \diamond \sigma_c)$ (or, if all the probability is used up, $u_r(\sigma_r^*, p_c) \geq u_r(d_r, p_c)$), because d_r only places positive probability on strategies in D_r. But this is impossible, because by strict dominance, $u_r(\sigma_r^*, \sigma_c) < u_r(d_r, \sigma_c)$ for any mixed strategy σ_c. ■

Finally, we show that by choosing the sets E_r, E_c as small as possible, we can make the generalized eliminability criterion coincide with the strict dominance concept.

Proposition 16 *Let $E_c = \{\}$ and $E_r = \{e_r\}$. Then e_r is eliminable relative to D_r, E_r, D_c, E_c if and only if it is strictly dominated by some mixed strategy that places positive probability only on elements of D_r.*

Proof: The "if" direction follows from Proposition 15. For the "only if" direction, suppose that e_r is eliminable relative to these sets. That means that there exists a mixed strategy d_r that places positive probability only on strategies in D_r such that for any pure strategy $\sigma_c \in \Sigma_c - E_c = \Sigma_c$, $u(e_r, \sigma_c) < u(d_r, \sigma_c)$ (because $E_c = \{\}$ and $E_r = \{e_r\}$, this is the only way in which an attempt to prove that e_r is not eliminable could fail). But this is precisely the condition for d_r to strictly dominate e_r. ■

We are now ready to turn to computational aspects of the new eliminability criterion.

9.3.4 Applying the new eliminability criterion can be computationally hard

In this subsection, we demonstrate that applying the eliminability criterion can be computationally hard, in the sense of worst-case complexity.[20] We show that applying the eliminability criterion is coNP-complete in two key special cases (subclasses of the problem). The first case is the one in which the D_r, E_r, D_c, E_c sets are set to be as large as possible. Here, the hardness follows directly from Proposition 14 and a result from Section 9.2.

Theorem 102 *Deciding whether a given strategy is eliminable relative to $D_r = E_r = \Sigma_r$ and $D_c = E_c = \Sigma_c$ is coNP-complete, even when the game is symmetric.*

[20] Because we only show hardness in the worst case, it is possible that many (or even most) instances are in fact easy to solve.

Proof: By Proposition 14, this is the converse of asking whether there exists a Nash equilibrium with positive probability on the given strategy. As we saw in Section 9.2, this is NP-complete. ∎

While this shows that the eliminability criterion is, in general, computationally hard to apply, we may wonder if there are special cases in which it is computationally easy to apply. Natural special cases to look at include those in which some of the sets D_r, E_r, D_c, E_c are small. The next theorem shows that applying the eliminability criterion remains coNP-complete even when $|D_r| = |D_c| = 1$.

Theorem 103 *Deciding whether a given strategy is eliminable relative to given D_r, E_r, D_c, E_c is coNP-complete, even when $|D_r| = |D_c| = 1$.*

Proof: We will show later (Corollary 28) that the problem is in coNP. To show that the problem is coNP-hard, we reduce an arbitrary KNAPSACK instance (given by m cost-value pairs (c_i, v_i), a cost constraint C and a value target V; we assume without loss of generality that $C = 1 - \epsilon$, for some ϵ small enough that it is impossible for a subset of the c_i to sum to a value strictly between C and 1,[21] that $c_i > 0$ for all i, and that $\sum_{i=1}^{m} v_i \leq 1$) to the following eliminability question. Let the game be as follows. The row player has $m+2$ distinct pure strategies: $e_r^1, e_r^2, \ldots, e_r^m, e_r^*, d_r$ (where $E_r = \{e_r^1, e_r^2, \ldots, e_r^m, e_r^*\}$ and $D_r = \{d_r\}$). The column player has $m+1$ distinct pure strategies: $e_c^1, e_c^2, \ldots, e_c^m, d_c$ (where $E_c = \{e_c^1, e_c^2, \ldots, e_c^m\}$ and $D_c = \{d_c\}$). Let the utilities be as follows:

- $u_r(e_r^i, e_c^j) = 1$ for all $i \neq j$;
- $u_r(e_r^i, e_c^i) = 1 - \frac{1}{v_i}$ for all i;
- $u_r(e_r^i, d_c) = 1$ for all i;
- $u_r(e_r^*, e_c^i) = \frac{1}{V} - 1$ for all i;
- $u_r(e_r^*, d_c) = -1$;
- $u_r(d_r, e_c^i) = 0$ for all i;
- $u_r(d_r, d_c) = 0$;
- $u_c(e_r^i, e_c^j) = 0$ for all $i \neq j$;
- $u_c(e_r^i, e_c^i) = \frac{1}{c_i}$ for all i;
- $u_c(e_r^i, d_c) = 1$ for all i;
- $u_c(e_r^*, e_c^i) = 0$ for all i;
- $u_c(e_r^*, d_c) = 1$;
- $u_c(d_r, e_c^i) = 0$ for all i;

[21] Because we may assume that the c_i and C are all integers divided by some number K, it is sufficient if $\epsilon < \frac{1}{K}$.

- $u_c(d_r, d_c) = 1$.

Thus, the matrix is as follows:

	e_c^1	e_c^2	$\cdots$	e_c^m	d_c
e_r^1	$1 - \frac{1}{v_1}, \frac{1}{c_1}$	$1, 0$	$\cdots$	$1, 0$	$1, 1$
e_r^2	$1, 0$	$1 - \frac{1}{v_2}, \frac{1}{c_2}$	$\cdots$	$1, 0$	$1, 1$
$\vdots$					
e_r^m	$1, 0$	$1, 0$	$\cdots$	$1 - \frac{1}{v_m}, \frac{1}{c_m}$	$1, 1$
e_r^*	$\frac{1}{V} - 1, 0$	$\frac{1}{V} - 1, 0$	$\cdots$	$\frac{1}{V} - 1, 0$	$-1, 1$
d_r	$0, 0$	$0, 0$	$\cdots$	$0, 0$	$0, 1$

We now show that e_r^* is eliminable relative to D_r, E_r, D_c, E_c if and only if there is no solution to the KNAPSACK instance.

First suppose there is a solution to the KNAPSACK instance. Then, for every i such that (c_i, v_i) is included in the KNAPSACK solution, let $p_r(e_r^i) = c_i$; for every i such that (c_i, v_i) is not included in the KNAPSACK solution, let $p_r(e_r^i) = 0$. Also, let $p_r(e_r^*) = 1 - \sum_{i=1}^{m} p_r(e_r^i)$. (We note that $\sum_{i=1}^{m} p_r(e_r^i) \leq C = 1 - \epsilon$, so that $p_r(e_r^*) \geq \epsilon > 0$.) Also, for every i such that (c_i, v_i) is included in the KNAPSACK solution, let $p_c(e_c^i) = v_i$. We now show that p_r and p_c satisfy the conditions of Definition 58. If the column player places the remaining probability on d_c, then the utility for the row player of playing any e_r^i with $p_r(e_r^i) > 0$ is $1 - \frac{v_i}{v_i} = 0$; the utility of playing e_r^* is $-1 + \frac{1}{V} \sum_{i=1}^{m} p_c(e_c^i) \geq -1 + \frac{V}{V} = 0$; and the utility of playing d_r is also 0. Thus, the condition is satisfied for all elements of E_r that have positive probability. As for E_c, we note that all of the row player's probability has already been used up. The utility of playing any e_c^i with $p_c(e_c^i) > 0$ is $\frac{c_i}{c_i} = 1$, whereas the utility for playing d_c is also 1. Thus, the condition is satisfied for all elements of E_c that have positive probability. It follows that p_r and p_c satisfy the conditions of Definition 58 and e_r^* is not eliminable relative to D_r, E_r, D_c, E_c.

Now suppose that e_r^* is not eliminable relative to D_r, E_r, D_c, E_c. Let p_r and p_c be partial mixed strategies on E_r and E_c satisfying the conditions of Definition 58. We must have that $p_r(e_r^*) > 0$. The utility for the row player of playing e_r^* is $-1 + \frac{1}{V} \sum_{i=1}^{m} p_c(e_c^i)$, which must be at least 0 (the utility of playing d_r); hence $\sum_{i=1}^{m} p_c(e_c^i) \geq V$. The utility for the column player of playing e_c^i is $\frac{p_r(e_r^i)}{c_i}$, which must be at least 1 (the utility of playing d_c) if $p_c(e_c^i) > 0$; hence $p_r(e_r^i) \geq c_i$ if $p_c(e_c^i) > 0$. Finally, the utility for the row player of playing e_r^i is $1 - \frac{p_c(e_c^i)}{v_i}$, which must be at least 0 (the utility of playing d_r) if $p_r(e_r^i) > 0$; hence $p_c(e_c^i) \leq v_i$ if $p_r(e_r^i) > 0$. Because we must have $p_r(e_r^i) \geq c_i > 0$ if $p_c(e_c^i) > 0$, it follows that we must always have $p_c(e_c^i) \leq v_i$. Let $S = \{i : p_c(e_c^i) > 0\}$. We must have $\sum_{i \in S} v_i \geq \sum_{i \in S} p_c(e_c^i) \geq V$. Also, we must have $\sum_{i \in S} c_i \leq \sum_{i \in S} p_r(e_r^i) < 1$ (because we must have $p_r(e_r^*) > 0$). Because it is impossible that $C < \sum_{i \in S} c_i < 1$, it follows that $\sum_{i \in S} c_i \leq C$. But then, S is a solution to the KNAPSACK instance. ■

However, we will show later that the eliminability criterion can be applied in polynomial time if the E_i sets are small (regardless of the size of the D_i sets). To do so, we first need to introduce an alternative version of the definition.

9.3.5 An alternative, equivalent definition of the eliminability criterion

In this subsection, we will give an alternative definition of eliminability, and we will show it is equivalent to the one presented in Definition 58. While the alternative definition is slightly less intuitve than the original one, it is easier to work with computationally, as we will show in the next subsection. Informally, the alternative definition differs from the original one as follows: in the alternative definition, the completion of player $-i$'s mixed strategy has to be chosen *before* player i's strategy d_i is chosen (but after player i's strategy e_i with $p_i(e_i) > 0$ is chosen). The formal definition follows.

Definition 59 *Given a two-player game in normal form, subsets D_r, E_r of the row player's pure strategies Σ_r, subsets D_c, E_c of the column player's pure strategies Σ_c, and a distinguished strategy $e_r^* \in E_r$, we say that e_r^* is* not eliminable relative to *D_r, E_r, D_c, E_c, if there exist functions (partial mixed strategies) $p_r : E_r \to [0,1]$ and $p_c : E_c \to [0,1]$ with $p_r(e_r^*) > 0$, $\sum_{e_r \in E_r} p_r(e_r) \leq 1$, and $\sum_{e_c \in E_c} p_c(e_c) \leq 1$, such that the following holds. For both $i \in \{r, c\}$, for any $e_i \in E_i$ with $p_i(e_i) > 0$, there exists some completion of the probability distribution over $-i$'s strategies, given by $p_{-i}^{e_i} : \Sigma_{-i} \to [0,1]$ (with $p_{-i}^{e_i}(e_{-i}) = p_{-i}(e_{-i})$ for all $e_{-i} \in E_{-i}$, and $\sum_{\sigma_{-i} \in \Sigma_{-i}} p_{-i}^{e_i}(\sigma_{-i}) = 1$), such that for any pure strategy $d_i \in D_i$, we have $u_i(e_i, p_{-i}^{e_i}) \geq u_i(d_i, p_{-i}^{e_i})$.*

We now show that the two definitions are equivalent.

Theorem 104 *The notions of eliminability put forward in Definitions 58 and 59 are equivalent. That is, e_r^* is eliminable relative to D_r, E_r, D_c, E_c according to Definition 58 if and only if e_r^* is eliminable relative to (the same) D_r, E_r, D_c, E_c according to Definition 59.*

Proof: The definitions are identical up to the condition that each strategy with positive probability (each $e_r \in E_r$ with $p_r(e_r) > 0$ and each $e_c \in E_c$ with $p_c(e_c) > 0$) must satisfy. We will show that these conditions are equivalent across the two definitions, thereby showing that the definitions are equivalent.

To show that the conditions are equivalent, we introduce another, zero-sum game that is a function of the original game, the sets D_r, E_r, D_c, E_c, the chosen partial probability distributions p_r and p_c, and the strategy e_i for which we are checking whether the conditions are satisfied. (Without loss of generality, assume that we are checking it for some strategy $e_r \in E_r$ with $p_r(e_r) > 0$.)

The zero-sum game has two players, 1 and 2 (not to be confused with the row and column players of the original game). Player 1 chooses some $d_r \in D_r$, and player 2 chooses some $\sigma_c \in \Sigma_c - E_c$. The utility to player 1 is $u_r(d_r, p_c \diamond \sigma_c) - u_r(e_r, p_c \diamond \sigma_c)$ (and the utility to player 2 is the negative of this). (We assume without loss of generality that p_c does not already use up all the probability, because in this case the conditions are trivially equivalent across the two definitions.)

First, suppose that player 1 must declare her probability distribution (mixed strategy) over D_r first, after which player 2 best-responds. Then, letting $\Delta(X)$ denote the set of probability distributions over set X, player 1 will receive $\max_{\delta_r \in \Delta(D_r)} \min_{\sigma_c \in \Sigma_c - E_c} \sum_{d_r \in D_r} \delta_r(d_r)(u_r(d_r, p_c \diamond \sigma_c) - u_r(e_r, p_c \diamond \sigma_c)) = \max_{\delta_r \in \Delta(D_r)} \min_{\sigma_c \in \Sigma_c - E_c} u_r(\delta_r, p_c \diamond \sigma_c) - u_r(e_r, p_c \diamond \sigma_c)$. This expression is at most 0 if and only if the condition in Definition 58 is satisfied.

Second, suppose that player 2 must declare his probability distribution (mixed strategy) over $\Sigma_c - E_c$ first, after which player 1 best-responds. Then, player 1 will receive $\min_{\delta_c \in \Delta(\Sigma_c - E_c)} \max_{d_r \in D_r} \sum_{\sigma_c \in \Sigma_c - E_c} \delta_c(\sigma_c)(u_r(d_r, p_c \diamond \sigma_c) - u_r(e_r, p_c \diamond \sigma_c)) = \min_{\delta_c \in \Delta(\Sigma_c - E_c)} \max_{d_r \in D_r} \sum_{e_c \in E_c} p_c(e_c)(u_r(d_r, e_c) - u_r(e_r, e_c)) + \sum_{\sigma_c \in \Sigma_c - E_c} (1 - \sum_{e_c \in E_c} p_c(e_c))\delta_c(\sigma_c)(u_r(d_r, \sigma_c) - u_r(e_r, \sigma_c)) = \min_{\delta_c \in \Delta(\Sigma_c - E_c)} \max_{d_r \in D_r} u_r(d_r, p_c \diamond \delta_c) - u_r(e_r, p_c \diamond \delta_c)$. This expression is at most 0 if and only if the condition in Definition 59 is satisfied.

However, by the Minimax Theorem [von Neumann, 1927], the two expressions must have the same value, and hence the two conditions are equivalent. ■

Informally, the reason that Definition 59 is easier to work with computationally is that all of the continuous variables (the values of the functions $p_r, p_c, p_c^{e_r}, p_r^{e_c}$) are set by the party that is trying to prove that the strategy is not eliminable; whereas in Definition 58, some of the continuous variables (the probabilities defining the mixed strategies d_r, d_c) are set by the party trying to refute the proof that the strategy is not eliminable. This will become more precise in the next subsection.

9.3.6 A mixed integer programming approach

In this subsection, we show how to translate Definition 59 into a mixed integer program that determines whether a given strategy e_r^* is eliminable relative to given sets D_r, E_r, D_c, E_c. The variables in the program, which are all restricted to be nonnegative, are the $p_i(e_i)$ for all $e_i \in E_i$; the $p_i^{e_{-i}}(\sigma_i)$ for all $e_{-i} \in E_{-i}$ and all $\sigma_i \in \Sigma_i - E_i$; and *binary* indicator variables $b_i(e_i)$ for all $e_i \in E_i$ which can be set to zero if and only if $p_i(e_i) = 0$. The program is the following:

maximize $p_r(e_r^*)$ **subject to**

(probability constraints): for both $i \in \{r, c\}$, for all $e_i \in E_i$, $\sum_{e_{-i} \in E_{-i}} p_{-i}(e_{-i}) + \sum_{\sigma_{-i} \in \Sigma_{-i} - E_{-i}} p_{-i}^{e_i}(\sigma_{-i}) = 1$

(binary constraints): for both $i \in \{r, c\}$, for all $e_i \in E_i$, $p_i(e_i) \leq b_i(e_i)$

(main constraints): for both $i \in \{r, c\}$, for all $e_i \in E_i$ and all $d_i \in D_i$, $\sum_{e_{-i} \in E_{-i}} p_{-i}(e_{-i})(u_i(e_i, e_{-i}) - u_i(d_i, e_{-i})) + \sum_{\sigma_{-i} \in \Sigma_{-i} - E_{-i}} p_{-i}^{e_i}(\sigma_{-i})(u_i(e_i, \sigma_{-i}) - u_i(d_i, \sigma_{-i})) \geq (b_i(e_i) - 1)U_i$

In this program, the constant U_i is the maximum difference between two different utilities that player i may receive in the game, that is, $U_i = \max_{\sigma_r, \sigma_r' \in \Sigma_r, \sigma_c, \sigma_c' \in \Sigma_c} u_i(\sigma_r, \sigma_c) - u_i(\sigma_r', \sigma_c')$.

Theorem 105 *The mixed integer program has a solution with objective value greater than zero if and only if e_r^* is not eliminable relative to D_r, E_r, D_c, E_c.*

Proof: For any $e_i \in E_i$ with $p_i(e_i) > 0$, $b_i(e_i)$ must be 1, and thus the corresponding main constraints become: for any $d_i \in D_i$, $\sum_{e_{-i} \in E_{-i}} p_{-i}(e_{-i})(u_i(e_i, e_{-i}) - u_i(d_i, e_{-i})) +$ $\sum_{\sigma_{-i} \in \Sigma_{-i} - E_{-i}} p_{-i}^{e_i}(\sigma_{-i})(u_i(e_i, \sigma_{-i}) - u_i(d_i, \sigma_{-i})) \geq 0$. These are equivalent to the constraints given on strategies $e_i \in E_i$ with $p_i(e_i) > 0$ in Definition 59. On the other hand, for any $e_i \in E_i$ with $p_i(e_i) = 0$, $b_i(e_i)$ can be set to 0, in which case the constraints become: for any $d_i \in D_i$, $\sum_{e_{-i} \in E_{-i}} p_{-i}(e_{-i})(u_i(e_i, e_{-i}) - u_i(d_i, e_{-i})) + \sum_{\sigma_{-i} \in \Sigma_{-i} - E_{-i}} p_{-i}^{e_i}(\sigma_{-i})(u_i(e_i, \sigma_{-i}) - u_i(d_i, \sigma_{-i})) \geq -U_i$. Because the probabilities in each of these constraints must sum to one by the probability constraints, and U_i is the maximum difference between two different utilities that player i may receive in the game, these constraints are vacuous. Therefore the main constraints correspond exactly to those in Definition 59. ■

We obtain the following corollaries:

Corollary 28 *Checking whether a given strategy can be eliminated relative to given D_r, E_r, D_c, E_c is in coNP.*

Proof: To see whether the strategy can be protected from elimination, we can nondeterministically choose the values for the binary variables $b_r(e_r)$ and $b_c(e_c)$. After this, only a linear program remains to be solved, which can be done in polynomial time [Khachiyan, 1979]. ■

Corollary 29 *Using the mixed integer program above, the time required to check whether a given strategy can be eliminated relative to given D_r, E_r, D_c, E_c is exponential only in $|E_r| + |E_c|$ (and not in $|D_r|, |D_c|, |\Sigma_r|$, or $|\Sigma_c|$).*

Proof: Any mixed integer program whose only integer variables are binary variables can be solved in time exponential only in its number of binary variables (for example, by searching over all settings of its binary variables and solving the remaining linear program in each case). The number of binary variables in this program is $|E_r| + |E_c|$. ■

9.3.7 Iterated elimination

In this subsection, we study what happens when we eliminate strategies *iteratively* using the new criterion. The criterion can be iteratively applied by removing an eliminated strategy from the game, and subsequently checking for new eliminabilities in the game with the strategy removed, *etc.* (as in the more elementary, conventional notion of iterated dominance). First, we show that this procedure is, in a sense, sound.

Theorem 106 *Iterated elimination according to the generalized criterion will never remove a strategy that is played with positive probability in some Nash equilibrium of the original game.*

Proof: We will prove this by induction on the elimination round (that is, the number of strategies eliminated so far). The claim is true for the first round by Proposition 13. Now suppose it is true up to and including round k; we must show it is true for round $k+1$. Suppose that the claim is false for round $k+1$, that is, there exists some game G and some pure strategy σ such that 1) σ is played with positive probability in some Nash equilibrium of G, and 2) using k elimination rounds, G can be reduced to G^{k+1}, in which σ is eliminable. Now consider the game G^k which preceded G^{k+1} in the elimination sequence, that is, the game obtained by undoing the last elimination before G^{k+1}. Also, let σ' be the strategy removed from G^k to obtain G^{k+1}. Now, in G^k, σ cannot be eliminated by the induction assumption. However, by Proposition 14, any strategy that is not played with positive probability in any Nash equilibrium can be eliminated, so it follows that there is some Nash equilibrium of G^k in which σ is played with positive probability. Moreover, this Nash equilibrium cannot place positive probability on σ' (because otherwise, by Proposition 13, we would not be able to eliminate it). But then, this Nash equilibrium must also be a Nash equilibrium of G^{k+1}: it does not place any probability on strategies that are not in G^{k+1}, and the set of strategies that the players can switch to in G^{k+1} is a subset of those in G^k. Hence, by Proposition 13, we cannot eliminate σ from G^{k+1}, and we have achieved the desired contradiction. ■

Because (the single-round version of) the eliminability criterion extends all the way to Nash equilibrium by Proposition 14, we get the following corollary.

Corollary 30 *Any strategy that can be eliminated using iterated elimination can also be eliminated in a single round (that is, without iterated application of the criterion).*

Proof: By Proposition 14, all strategies that are not played with positive probability in any Nash equilibrium can be eliminated in a single round; but by Theorem 106, this is the only type of strategy that iterated elimination can eliminate. ■

Interestingly, iterated elimination is in a sense incomplete:

Proposition 17 *Removing an eliminated strategy from a game sometimes decreases the set of strategies that can be eliminated.*

Proof: Consider the following game:

	L	M	R
U	2, 2	0, 1	0, 5
D	1, 0	1, 1	1, 0

The unique Nash equilibrium of this game is (D, M), for the following reasons. In order for it to be worthwhile for the row player to play U with positive probability, the column player should play L with probability at least $1/2$. But, in order for it to be worthwhile for the column player to play L with positive probability (rather than M), the row player should play U with probability at least $1/2$. However, if the row player plays U with probability at least $1/2$, then the column player's unique best response is to play R. Hence, the row player must play D in any Nash equilibrium, and the unique best response to D is M.

Thus, by Proposition 14, all strategies besides D and M can be eliminated. In particular, R can be eliminated. However, if we remove R from the game, the remaining game is:

	L	M
U	2, 2	0, 1
D	1, 0	1, 1

In this game, (U, L) is also a Nash equilibrium, and hence U and L can no longer be eliminated, by Proposition 13. ■

This example highlights an interesting issue with respect to using this eliminability criterion as a preprocessing step in the computation of Nash equilibria: it does not suffice to simply throw out eliminated strategies and compute a Nash equilibrium for the remaining game. Rather, we need to use the criterion more carefully: if we know that a strategy is eliminable according to the criterion we can restrict our attention to supports for the player that do not include this strategy.

The example also directly implies that iterated elimination according to the generalized criterion is path-dependent (the choice of which strategy to remove first affects which strategies can/will be removed later). As we discussed in Section 9.1, the same phenomenon occurs with iterated weak dominance. There is a sizeable literature on path (in)dependence for various notions of dominance [Gilboa *et al.*, 1990; Borgers, 1993; Osborne and Rubinstein, 1994; Marx and Swinkels, 1997, 2000; Apt, 2004].

In light of these results, it may appear that there is not much reason to do iterated elimination using the new criterion, because it never increases and sometimes even decreases the set of strategies that we can eliminate. However, we need to keep in mind that Theorem 106, Corollary 30, and Proposition 17 do not pose any restrictions on the sets D_r, E_r, D_c, E_c, and therefore (by Propositions 13 and 14) are effectively results about iteratively removing strategies based on whether they are played in a Nash equilibrium. However, the new criterion is more informative and useful when there are restrictions on the sets D_r, E_r, D_c, E_c. Of particular interest is the restriction $|E_r| + |E_c| \leq k$, because by Corollary 29 this quantity determines the (worst-case) runtime of the mixed integer programming approach that we presented in the previous subsection. Under this restriction, it turns out that iterated elimination can eliminate strategies that single-round elimination cannot.

Proposition 18 *Under a restriction of the form $|E_r| + |E_c| \leq k$, iterated elimination can eliminate strategies that single-round elimination cannot (even when $k = 1$).*

Proof: By Proposition 16, when $k = 1$ the eliminability criterion coincides with strict dominance (and hence iterated application of the criterion coincides with iterated strict dominance). So, consider the following game:

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

Strict dominance cannot eliminate L, but iterated strict dominance (which can remove D first) can eliminate L. ∎

Of course, even under this (or any other) restriction iterated elimination remains sound in the sense of Theorem 106. Therefore, one sensible approach to eliminating strategies is the following. Iteratively apply the eliminability criterion (with whatever restrictions are desired to increase the strength of the argument, or are necessary to make it computationally manageable, such as $|E_r| + |E_c| \leq k$), removing each eliminated strategy, until the process gets stuck. Then, start again with the original game, and take a different path of iterated elimination (which may eliminate strategies that could no longer be eliminated after the first path of elimination, as described in Proposition 17), until the process gets stuck—*etc.* In the end, any strategy that was eliminated in any one of the elimination paths can be considered "eliminated", and this is safe by Theorem 106.[22]

Interestingly, here the analogy with iterated weak dominance breaks down. Because there is no soundness theorem such as Theorem 106 for iterated weak dominance, considering all the strategies that are eliminated in some iterated weak dominance elimination path to be simultaneously "eliminated" can lead to senseless results. Consider for example the following game:

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

U can be eliminated by removing R first, and D can be eliminated by removing M first—but these are the row player's only strategies, so considering both of them to be eliminated makes little sense.

9.4 Summary

A theory of mechanism design for bounded agents cannot rest on game-theoretic solution concepts that are too hard for agents to compute. To assess to what extent this eliminates existing solution concepts from consideration, the first two sections of this chapter were devoted to studying how hard it is to compute solutions according to some of these concepts.

In Section 9.1, we studied computational aspects of dominance and iterated dominance. We showed that checking whether a given strategy is dominated (weakly or strictly) by some mixed strategy can be done in polynomial time using a single linear program solve. We then 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. 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

[22] This procedure is reminiscent of iterative sampling.

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). We also studied dominance and iterated dominance in Bayesian games. 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).

In Section 9.2 we provided a single reduction that demonstrates that 1) it is NP-complete to determine whether Nash equilibria with certain natural properties exist, 2) more significantly, the problems of maximizing certain properties of a Nash equilibrium are inapproximable (unless P=NP), and 3) it is #P-hard to count the Nash equilibria (or connected sets of Nash equilibria). We also showed that determining whether a pure-strategy Bayes-Nash equilibrium exists is NP-complete.

Since these (and other) results suggest that dominance is a more tractable solution concept than (Bayes)-Nash equilibrium, but is often too strict for mechanism design (and other) purposes, one may wonder whether it is possible to strike a compromise between dominance and Nash equilibrium, obtaining intermediate solution concepts that combine good aspects of both. The last section in this chapter, Section 9.3, did precisely that. We defined a generalized eliminability criterion for bimatrix games that considers whether a given strategy is eliminable relative to given dominator & eliminee subsets of the players' strategies. We showed that this definition spans a spectrum of eliminability criteria from strict dominance (when the subsets are as small as possible) to Nash equilibrium (when the subsets are as large as possible). We showed that checking whether a strategy is eliminable according to this criterion is coNP-complete (both when all the sets are as large as possible and when the dominator sets each have size 1). We then gave an alternative definition of the eliminability criterion and showed that it is equivalent using the Minimax Theorem. We showed how this alternative definition can be translated into a mixed integer program of polynomial size with a number of (binary) integer variables equal to the sum of the sizes of the eliminee sets, implying that checking whether a strategy is eliminable according to the criterion can be done in polynomial time if the eliminee sets are small. Finally, we studied using the criterion for iterated elimination of strategies.

The results in this chapter provide an initial step towards building a theory of mechanism design for bounded agents. For such a theory to be complete, it would also require methods for predicting how agents will act in strategic situations where standard game-theoretic solutions are too hard for them to compute. Ideally, these methods would not assume any detailed knowlege of the algorithms available to the agents, but this will undoubtedly be a difficult feat to accomplish. Fortunately, we do not have to wait for the entire theory of mechanism design for bounded agents to be developed before we create some initial techniques for designing such mechanisms nonetheless. (The only downside of not having the general theory is that we will not be able to evaluate how close to optimal these techniques are.) The next chapter provides one such technique, which can in fact also be used to generate mechanisms automatically (albeit in a very different way from that proposed in Chapter 6).