

# Playing Games in Many Possible Worlds

Matt Lepinski  
CSAIL  
Massachusetts Institute of Technology  
Cambridge, MA 02139  
lepinski@theory.lcs.mit.edu

David Liben-Nowell  
Department of Computer Science  
Carleton College  
Northfield, MN 55057  
dlibenno@carleton.edu

Seth Gilbert  
CSAIL  
Massachusetts Institute of Technology  
Cambridge, MA 02139  
sethg@theory.lcs.mit.edu

April Rasala Lehman  
Google, Inc.  
Mountain View, CA 94043  
alehman@google.com

## Abstract

In traditional game theory, players are typically endowed with exogenously given knowledge of the structure of the game—either full omniscient knowledge or partial but fixed information. In real life, however, people are often unaware of the utility of taking a particular action until they perform research into its consequences. In this paper, we model this phenomenon. We imagine a player engaged in a question-and-answer session, asking questions both about his or her own preferences and about the state of reality; thus we call this setting “Socratic” game theory. In a Socratic game, players begin with an *a priori* probability distribution over many *possible worlds*, with a different utility function for each world. Players can make *queries*, at some cost, to learn partial information about which of the possible worlds is the actual world, before choosing an action. We consider two query models: (1) an *unobservable-query* model, in which players learn only the response to their own queries, and (2) an *observable-query* model, in which players also learn which queries their opponents made.

The results in this paper consider cases in which the underlying worlds of a two-player Socratic game are either *constant-sum games* or *strategically zero-sum games*, a class that generalizes constant-sum games to include all games in which the sum of payoffs depends linearly on the interaction between the players. When the underlying worlds are constant sum, we give polynomial-time algorithms to find Nash equilibria in both the observable- and unobservable-query models. When the worlds are strategically zero sum, we give efficient algorithms to find Nash equilibria in unobservable-query Socratic games and correlated equilibria in observable-query Socratic games.

## 1 Introduction

Late October 1960. A smoky room. Democratic Party strategists huddle around a map. How should the Kennedy campaign allocate its remaining advertising budget? Should it focus on, say, California or New York? The Nixon campaign faces the same dilemma. Of course, neither campaign knows the effectiveness of its advertising in each state. Perhaps Californians are susceptible to Nixon’s

advertising, but are unresponsive to Kennedy’s. In light of this uncertainty, the Kennedy campaign may conduct a survey, at some cost, to estimate the effectiveness of its advertising. Moreover, the larger—and more expensive—the survey, the more accurate it will be. Is the cost of a survey worth the information that it provides? How should one balance the cost of acquiring more information against the risk of playing a game with higher uncertainty?

In this paper, we model situations of this type as *Socratic games*. As in traditional game theory, the players in a Socratic game choose actions to maximize their payoffs, but we model players with incomplete information who can make costly queries to reduce their uncertainty about the state of the world before they choose their actions. This approach contrasts with traditional game theory, in which players are usually modeled as having fixed, exogenously given information about the structure of the game and its payoffs. (In traditional games of incomplete and imperfect information, there is information that the players do not have; in Socratic games, unlike in these games, the players have a chance to acquire the missing information, at some cost.) A number of related models have been explored by economists and computer scientists motivated by similar situations, often with a focus on mechanism design and auctions; a sampling of this research includes the work of Larson and Sandholm [41, 42, 43, 44], Parkes [59], Fong [22], Compte and Jehiel [12], Rezende [63], Persico and Matthews [48, 60], Crémer and Khalil [15], Rasmusen [62], and Bergemann and Välimäki [4, 5]. The model of Bergemann and Välimäki is similar in many regards to the one that we explore here; see Section 7 for some discussion.

A Socratic game proceeds as follows. A *real world* is chosen randomly from a set of *possible worlds* according to a common prior distribution. Each player then selects an arbitrary *query* from a set of available costly queries and receives a corresponding piece of information about the real world. Finally each player selects an action and receives a payoff—a function of the players’ selected actions and the identity of the real world—less the cost of the query that he or she made. Compared to traditional game theory, the distinguishing feature of our model is the introduction of explicit costs to the players for learning arbitrary partial information about which of the many possible worlds is the real world.

Our research was initially inspired by recent results in psychology on decision making, but it soon became clear that Socratic game theory is also a general tool for understanding the “exploitation versus exploration” tradeoff, well studied in machine learning, in a strategic multiplayer environment. This tension between the risk arising from uncertainty and the cost of acquiring information is ubiquitous in economics, political science, and beyond.

**Our results.** We consider Socratic games under two models: an *unobservable-query* model where players learn only the response to their own queries and an *observable-query* model where players also learn which queries their opponents made. We give efficient algorithms to find Nash equilibria—i.e., tuples of strategies from which no player has unilateral incentive to deviate—in broad classes of two-player Socratic games in both models. Our first result is an efficient algorithm to find Nash equilibria in unobservable-query Socratic games with *constant-sum* worlds, in which the sum of the players’ payoffs is independent of their actions. Our techniques also yield Nash equilibria in unobservable-query Socratic games with *strategically zero-sum* worlds. Strategically zero-sum games generalize constant-sum games by allowing the sum of the players’ payoffs to depend on individual players’ choices of strategy, but not on any interaction of their choices. Our second result is an efficient algorithm to find Nash equilibria in observable-query Socratic games with constant-sum worlds. Finally, we give an efficient algorithm to find *correlated equilibria*—a weaker

but increasingly well-studied solution concept for games [2, 3, 32, 56, 57]—in observable-query Socratic games with strategically zero-sum worlds.

Like all games, Socratic games can be viewed as a special case of *extensive-form games*, which represent games by trees in which internal nodes represent choices made by chance or by the players, and the leaves represent outcomes that correspond to a vector of payoffs to the players. Algorithmically, the generality of extensive-form games makes them difficult to solve efficiently, and the special cases that are known to be efficiently solvable do not include even simple Socratic games. Every (complete-information) classical game is a trivial Socratic game (with a single possible world and a single trivial query), and efficiently finding Nash equilibria in classical games has been shown to be hard [10, 11, 13, 16, 17, 27, 54, 55]. Therefore we would not expect to find a straightforward polynomial-time algorithm to compute Nash equilibria in general Socratic games. However, it is well known that Nash equilibria can be found efficiently via an LP for two-player constant-sum games [49, 71] (and strategically zero-sum games [51]). A Socratic game is itself a classical game, so one might hope that these results can be applied to Socratic games with constant-sum (or strategically zero-sum) worlds.

We face two major obstacles in extending these classical results to Socratic games. First, a Socratic game with constant-sum worlds is not itself a constant-sum classical game—rather, the resulting classical game is only strategically zero sum. Worse yet, a Socratic game with strategically zero-sum worlds is not itself classically strategically zero sum—indeed, there are no known efficient algorithmic techniques to compute Nash equilibria in the resulting class of classical games. (Exponential-time algorithms like Lemke/Howson, of course, can be used [45].) Thus even when it is easy to find Nash equilibria in each of the worlds of a Socratic game, we require new techniques to solve the Socratic game itself. Second, even when the Socratic game itself is strategically zero sum, the number of possible strategies available to each player is exponential in the natural representation of the game. As a result, the standard linear programs for computing equilibria have an exponential number of variables *and* an exponential number of constraints.

For unobservable-query Socratic games with strategically zero-sum worlds, we address these obstacles by formulating a new LP that uses only polynomially many variables (though still an exponential number of constraints) and then use ellipsoid-based techniques to solve it. For observable-query Socratic games, we handle the exponentiality by decomposing the game into stages, solving the stages separately, and showing how to reassemble the solutions efficiently. To solve the stages, it is necessary to find Nash equilibria in Bayesian strategically zero-sum games, and we give an explicit polynomial-time algorithm to do so.

## 2 Games and Socratic Games

In this section, we review background on game theory and formally introduce Socratic games. We present these models in the context of two-player games, but the multiplayer case is a natural extension. Throughout the paper, boldface variables will be used to denote a pair of variables (e.g.,  $\mathbf{a} = \langle a_I, a_{II} \rangle$ ). Let  $\Pr[x \leftarrow \pi]$  denote the probability that a particular value  $x$  is drawn from the distribution  $\pi$ , and let  $\mathbf{E}_{x \sim \pi}[g(x)]$  denote the expectation of  $g(x)$  when  $x$  is drawn from  $\pi$ .

## 2.1 Background on Game Theory

Consider two players, Player I and Player II, each of whom is attempting to maximize his or her utility (or payoff). A *(two-player) game* is a pair  $\langle \mathbf{A}, \mathbf{u} \rangle$ , where, for  $i \in \{I, II\}$ ,

- $A_i$  is the set of *pure strategies* for Player  $i$ , and  $\mathbf{A} = \langle A_I, A_{II} \rangle$ ; and
- $u_i : \mathbf{A} \rightarrow \mathbb{R}$  is the *utility function* for Player  $i$ , and  $\mathbf{u} = \langle u_I, u_{II} \rangle$ .

We require that  $\mathbf{A}$  and  $\mathbf{u}$  be common knowledge. If each Player  $i$  chooses strategy  $a_i \in A_i$ , then the payoffs to Players I and II are  $u_I(\mathbf{a})$  and  $u_{II}(\mathbf{a})$ , respectively. A game is *constant sum* if, for all  $\mathbf{a} \in \mathbf{A}$ , we have that  $u_I(\mathbf{a}) + u_{II}(\mathbf{a}) = c$  for some fixed  $c$  independent of  $\mathbf{a}$ .

Player  $i$  can also play a *mixed strategy*  $\alpha_i \in \mathcal{A}_i$ , where  $\mathcal{A}_i$  denotes the space of probability measures over the set  $A_i$ . Payoff functions are generalized as  $u_i(\boldsymbol{\alpha}) = u_i(\alpha_I, \alpha_{II}) := \mathbf{E}_{\mathbf{a} \sim \boldsymbol{\alpha}}[u_i(\mathbf{a})] = \sum_{\mathbf{a} \in \mathbf{A}} \boldsymbol{\alpha}(\mathbf{a}) u_i(\mathbf{a})$ , where the quantity  $\boldsymbol{\alpha}(\mathbf{a}) = \alpha_I(a_I) \cdot \alpha_{II}(a_{II})$  denotes the joint probability of the independent events that each Player  $i$  chooses action  $a_i$  from the distribution  $\alpha_i$ . This generalization to mixed strategies is known as *von Neumann/Morgenstern utility* [70], in which players are indifferent between a guaranteed payoff  $x$  and an expected payoff of  $x$ .

A *Nash equilibrium* is a pair  $\boldsymbol{\alpha}$  of mixed strategies so that neither player has an incentive to change his or her strategy unilaterally. Formally, the strategy pair  $\boldsymbol{\alpha}$  is a Nash equilibrium if and only if both  $u_I(\alpha_I, \alpha_{II}) = \max_{\alpha'_I \in \mathcal{A}_I} u_I(\alpha'_I, \alpha_{II})$  and  $u_{II}(\alpha_I, \alpha_{II}) = \max_{\alpha'_{II} \in \mathcal{A}_{II}} u_{II}(\alpha_I, \alpha'_{II})$ ; that is, the strategies  $\alpha_I$  and  $\alpha_{II}$  are *mutual best responses*.

A *correlated equilibrium* is a distribution  $\psi$  over  $\mathbf{A}$  that obeys the following: if  $\mathbf{a} \in \mathbf{A}$  is drawn randomly according to  $\psi$  and Player  $i$  learns  $a_i$ , then no Player  $i$  has incentive to deviate unilaterally from playing  $a_i$ . (A Nash equilibrium is a correlated equilibrium in which  $\psi(\mathbf{a}) = \alpha_I(a_I) \cdot \alpha_{II}(a_{II})$  is a product distribution.) Formally, in a correlated equilibrium, for every  $\mathbf{a} \in \mathbf{A}$  we must have that  $a_I$  is a best response to a randomly chosen  $\hat{a}_{II} \in A_{II}$  drawn according to  $\psi(a_I, \hat{a}_{II})$ , and the analogous condition must hold for Player II.

## 2.2 Socratic Games

In this section, we formally define Socratic games. A *Socratic game* is a 7-tuple  $\langle \mathbf{A}, W, \vec{\mathbf{u}}, S, \mathbf{Q}, p, \delta \rangle$ , where, for  $i \in \{I, II\}$ :

- $A_i$  is, as before, the set of pure strategies for Player  $i$ .
- $W$  is a set of *possible worlds*, one of which is the *real world*  $w_{\text{real}}$ .
- $\vec{u}_i = \{u_i^w : \mathbf{A} \rightarrow \mathbb{R} \mid w \in W\}$  is a set of payoff functions for Player  $i$ , one for each possible world.
- $S$  is a set of *signals*.
- $Q_i$  is a set of available *queries* for Player  $i$ . When Player  $i$  makes query  $q_i : W \rightarrow S$ , he or she receives the signal  $q_i(w_{\text{real}})$ . When Player  $i$  receives signal  $q_i(w_{\text{real}})$  in response to query  $q_i$ , he or she can infer that  $w_{\text{real}} \in \{w : q_i(w) = q_i(w_{\text{real}})\}$ , i.e., the set of possible worlds from which query  $q_i$  cannot distinguish  $w_{\text{real}}$ .
- $p : W \rightarrow [0, 1]$  is a probability distribution over the possible worlds.

- $\delta_i : Q_i \rightarrow \mathbb{R}^{\geq 0}$  gives the *query cost* for each available query for Player  $i$ .

Initially, the world  $w_{\text{real}}$  is chosen according to the probability distribution  $p$ , but the identity of  $w_{\text{real}}$  remains unknown to the players. That is, it is as if the players are playing the game  $\langle \mathbf{A}, \mathbf{u}^{w_{\text{real}}} \rangle$  but do not know  $w_{\text{real}}$ . The players make queries  $\mathbf{q} \in \mathbf{Q}$ , and Player  $i$  receives the signal  $q_i(w_{\text{real}})$ . We consider both *observable* queries and *unobservable* queries. When queries are observable, each player learns *which* query was made by the other player, and the *results* of his or her own query—that is, each Player  $i$  learns  $q_I, q_{II}$ , and  $q_i(w_{\text{real}})$ . For unobservable queries, Player  $i$  learns only  $q_i$  and  $q_i(w_{\text{real}})$ . After learning the results of the queries, the players select strategies  $\mathbf{a} \in \mathbf{A}$  and receive as payoffs  $u_i^{w_{\text{real}}}(\mathbf{a}) - \delta_i(q_i)$ .

In the Socratic game, a pure strategy for Player  $i$  consists of a query  $q_i \in Q_i$  and a *response function* mapping any result of the query  $q_i$  to a strategy  $a_i \in A_i$  to play. A player’s state of knowledge after a query is a point in  $\mathcal{R} := \mathbf{Q} \times S$  or  $\overline{\mathcal{R}}_i := Q_i \times S$  for observable or unobservable queries, respectively. Thus Player  $i$ ’s response function maps  $\mathcal{R}$  or  $\overline{\mathcal{R}}_i$  to  $A_i$ . Note that the number of pure strategies is exponential, as there are exponentially many response functions. A mixed strategy involves both randomly choosing a query  $q_i \in Q_i$  and randomly choosing an action  $a_i \in A_i$  in response to the results of the query. Formally, we will consider a *mixed-strategy-function profile*  $\mathbf{f} = \langle \mathbf{f}^{\text{query}}, \mathbf{f}^{\text{resp}} \rangle$  to have two parts:

- a function  $f_i^{\text{query}} : Q_i \rightarrow [0, 1]$ , where  $f_i^{\text{query}}(q_i)$  is the probability that Player  $i$  makes query  $q_i$ .
- a function  $f_i^{\text{resp}}$  that maps  $\mathcal{R}$  or  $\overline{\mathcal{R}}_i$  to a probability distribution over actions. Player  $i$  chooses an action  $a_i \in A_i$  according to the probability distribution  $f_i^{\text{resp}}(\mathbf{q}, q_i(w))$  for observable queries, and according to  $f_i^{\text{resp}}(q_i, q_i(w))$  for unobservable queries. (With unobservable queries, for example, the probability that Player I plays action  $a_I$  conditioned on making query  $q_I$  in world  $w$  is given by  $\Pr[a_I \leftarrow f_I^{\text{resp}}(q_I, q_I(w))]$ .)

Mixed strategies are typically defined as probability distributions over the pure strategies, but here we represent a mixed strategy by a pair  $\langle \mathbf{f}^{\text{query}}, \mathbf{f}^{\text{resp}} \rangle$ , which is commonly referred to as a “behavioral” strategy in the game-theory literature. As in any game with perfect recall, one can easily map a mixture of pure strategies to a behavioral strategy  $\mathbf{f} = \langle \mathbf{f}^{\text{query}}, \mathbf{f}^{\text{resp}} \rangle$  that induces the same probability of making a particular query  $q_i$  or playing a particular action after making a query  $q_i$  in a particular world. Thus it suffices to consider only this representation of mixed strategies.

For a strategy-function profile  $\mathbf{f}$  for observable queries, the (expected) payoff to Player  $i$  is given by

$$\sum_{\mathbf{q} \in \mathbf{Q}, w \in W, \mathbf{a} \in \mathbf{A}} \left[ \begin{array}{l} f_I^{\text{query}}(q_I) \cdot f_{II}^{\text{query}}(q_{II}) \cdot p(w) \\ \cdot \Pr[a_I \leftarrow f_I^{\text{resp}}(\mathbf{q}, q_I(w))] \\ \cdot \Pr[a_{II} \leftarrow f_{II}^{\text{resp}}(\mathbf{q}, q_{II}(w))] \\ \cdot (u_i^w(\mathbf{a}) - \delta_i(q_i)) \end{array} \right].$$

The payoffs for unobservable queries are analogous, with  $f_j^{\text{resp}}(q_j, q_j(w))$  in place of  $f_j^{\text{resp}}(\mathbf{q}, q_j(w))$ .

### 3 STRATEGICALLY ZERO-SUM Games

We can view a Socratic game  $G$  with constant-sum worlds as an exponentially large classical game, with pure strategies “make query  $q_i$  and respond according to  $f_i$ .” However, this classical game is

not constant sum. The sum of the players' payoffs varies depending upon their strategies, because different queries incur different costs. However, this game still has significant structure: the sum of payoffs varies *only* because of varying query costs. Thus the sum of payoffs does depend on players' choice of strategies, but not on the interaction of their choices—i.e., for fixed functions  $g_I$  and  $g_{II}$ , we have  $u_I(\mathbf{q}, \mathbf{f}) + u_{II}(\mathbf{q}, \mathbf{f}) = g_I(q_I, f_I) + g_{II}(q_{II}, f_{II})$  for all strategies  $\langle \mathbf{q}, \mathbf{f} \rangle$ . Such games are called *strategically zero sum* and were introduced by Moulin and Vial [51], who describe a notion of strategic equivalence and define strategically zero-sum games as those strategically equivalent to zero-sum games. It is interesting to note that two Socratic games with the same queries and strategically equivalent worlds are not necessarily strategically equivalent.

A game  $\langle \mathbf{A}, \mathbf{u} \rangle$  is *strategically zero sum* if there exist labels  $\ell(i, a_i)$  for every Player  $i$  and every pure strategy  $a_i \in A_i$  such that, for all mixed-strategy profiles  $\boldsymbol{\alpha}$ , we have that the sum of the utilities satisfies

$$u_I(\boldsymbol{\alpha}) + u_{II}(\boldsymbol{\alpha}) = \sum_{a_I \in A_I} \alpha_I(a_I) \cdot \ell(I, a_I) + \sum_{a_{II} \in A_{II}} \alpha_{II}(a_{II}) \cdot \ell(II, a_{II}).$$

Note that any constant-sum game is strategically zero sum as well.

It is not immediately obvious that one can efficiently decide if a given game is strategically zero sum. For completeness, we give a characterization of classical strategically zero-sum games in terms of the rank of a simple matrix derived from the game's payoffs, allowing us to efficiently decide if a given game is strategically zero sum and, if it is, to compute the labels  $\ell(i, a_i)$ .

**Theorem 3.1.** *Consider a game  $G = \langle \mathbf{A}, \mathbf{u} \rangle$  with  $A_i = \{a_i^1, \dots, a_i^{n_i}\}$ . Let  $M^G$  be the  $n_I$ -by- $n_{II}$  matrix whose  $\langle i, j \rangle$ th entry  $M_{(i,j)}^G$  satisfies  $\log_2 M_{(i,j)}^G = u_I(a_i^i, a_{II}^j) + u_{II}(a_i^i, a_{II}^j)$ . Then the following are equivalent:*

- (i)  $G$  is strategically zero sum;
- (ii) there exist labels  $\ell(i, a_i)$  for every player  $i \in \{I, II\}$  and every pure strategy  $a_i \in A_i$  such that, for all pure strategies  $\mathbf{a} \in \mathbf{A}$ , we have  $u_I(\mathbf{a}) + u_{II}(\mathbf{a}) = \ell(I, a_I) + \ell(II, a_{II})$ ; and
- (iii)  $\text{rank}(M^G) = 1$ .

*Sketch.* ( $i \Rightarrow ii$ ) is immediate; every pure strategy is a trivially mixed strategy. For ( $ii \Rightarrow iii$ ), let  $\vec{c}_i$  be the  $n$ -element column vector with  $j$ th component  $2^{\ell(i, a_i^j)}$ ; then  $\vec{c}_I \cdot \vec{c}_{II}^T = M^G$ . For ( $iii \Rightarrow i$ ), if  $\text{rank}(M^G) = 1$ , then  $M^G = u \cdot v^T$ . We can prove that  $G$  is strategically zero sum by choosing labels  $\ell(I, a_i^j) := \log_2 u_j$  and  $\ell(II, a_{II}^j) := \log_2 v_j$ .  $\square$

## 4 Socratic Games with Unobservable Queries

We begin with Socratic games with *unobservable* queries, where a player's choice of query is not revealed to her opponent. We give an efficient algorithm to solve unobservable-query Socratic games with strategically zero-sum worlds. Our algorithm is based upon the LP shown in Figure 1, whose feasible points are Nash equilibria for the game. The LP has polynomially many variables but exponentially many constraints. We give an efficient separation oracle for the LP, implying that the ellipsoid method [28, 38] yields an efficient algorithm. This approach extends the techniques of Koller and Megiddo [39] (see also [40]) to solve constant-sum games represented in extensive form. (Recall that their result does not directly apply in our case; even a Socratic game with constant-sum worlds is *not* a constant-sum classical game.)

“Player  $i$  does not prefer ‘make query  $q_i$ , then play according to the function  $f_i$ ’ ”:

$$\forall q_I \in Q_I, f_I : \overline{\mathcal{R}}_I \rightarrow A_I : \quad \rho_I \geq \sum_{w \in W, a_{II} \in A_{II}, q_{II} \in Q_{II}, a_I = f_I(q_I, q_{II}(w))} (p(w) \cdot x_{a_{II}, q_{II}, w}^{II} \cdot [u_I^w(\mathbf{a}) - \delta_I(q_I)]) \quad (\text{I})$$

$$\forall q_{II} \in Q_{II}, f_{II} : \overline{\mathcal{R}}_{II} \rightarrow A_{II} : \quad \rho_{II} \geq \sum_{w \in W, a_I \in A_I, q_I \in Q_I, a_{II} = f_{II}(q_{II}, q_I(w))} (p(w) \cdot x_{a_I, q_I, w}^I \cdot [u_{II}^w(\mathbf{a}) - \delta_{II}(q_{II})]) \quad (\text{II})$$

“Every player’s choices form a probability distribution in every world”:

$$\forall i \in \{I, II\}, w \in W : \quad 1 = \sum_{a_i \in A_i, q_i \in Q_i} x_{a_i, q_i, w}^i \quad (\text{III})$$

$$\forall i \in \{I, II\}, w \in W : \quad 0 \leq x_{a_i, q_i, w}^i \quad (\text{IV})$$

“Queries are independent of the world, and actions depend only on query output”:

$$\forall i \in \{I, II\}, q_i \in Q_i, w \in W, w' \in W \text{ such that } q_i(w) = q_i(w') :$$

$$y_{q_i}^i = \sum_{a_i \in A_i} x_{a_i, q_i, w}^i \quad (\text{V})$$

$$x_{a_i, q_i, w}^i = x_{a_i, q_i, w'}^i \quad (\text{VI})$$

“The payoffs are consistent with the labels  $\ell(i, a_i, w)$ ”:

$$\rho_I + \rho_{II} = \sum_{i \in \{I, II\}} \sum_{w \in W, q_i \in Q_i, a_i \in A_i} (p(w) \cdot x_{a_i, q_i, w}^i \cdot [\ell(i, a_i, w) - \delta_i(q_i)]) \quad (\text{VII})$$

Figure 1: An LP to find Nash equilibria in unobservable-query Socratic games with strategically zero-sum worlds. The input is a Socratic game  $\langle \mathbf{A}, W, \vec{\mathbf{u}}, S, \mathbf{Q}, p, \delta \rangle$  so that world  $w$  is strategically zero sum with labels  $\ell(i, a_i, w)$ . Player  $i$  makes query  $q_i \in Q_i$  with probability  $y_{q_i}^i$  and, when the actual world is  $w \in W$ , makes query  $q_i$  and plays action  $a_i$  with probability  $x_{a_i, q_i, w}^i$ . The expected payoff to Player  $i$  is given by  $\rho_i$ .

**Lemma 4.1.** *Let  $G = \langle \mathbf{A}, W, \bar{\mathbf{u}}, S, \mathbf{Q}, p, \delta \rangle$  be an arbitrary unobservable-query Socratic game with strategically zero-sum worlds. Any feasible point for the LP in Figure 1 can be efficiently mapped to a Nash equilibrium for  $G$ , and any Nash equilibrium for  $G$  can be mapped to a feasible point for the program.*

*Sketch.* We begin with a description of the correspondence between feasible points for the LP and Nash equilibria for  $G$ . First, suppose that strategy profile  $\mathbf{f} = \langle \mathbf{f}^{\text{query}}, \mathbf{f}^{\text{resp}} \rangle$  forms a Nash equilibrium for  $G$ . Then the following setting for the LP variables is feasible:

$$\begin{aligned} y_{q_i}^i &= \mathbf{f}_i^{\text{query}}(q_i) \\ x_{a_i, q_i, w}^i &= \Pr[a_i \leftarrow f_i^{\text{resp}}(q_i, q_i(w))] \cdot y_{q_i}^i \\ \rho_i &= \sum_{w, \mathbf{q} \in \mathbf{Q}, \mathbf{a} \in \mathbf{A}} p(w) \cdot x_{a_i, q_i, w}^{\text{I}} \cdot x_{a_{\text{II}}, q_{\text{II}}, w}^{\text{II}} \cdot [u_i^w(\mathbf{a}) - \delta_i(q_i)]. \end{aligned}$$

(We omit the straightforward calculations that verify feasibility.) Next, suppose  $\langle x_{a_i, q_i, w}^i, y_{q_i}^i, \rho_i \rangle$  is feasible for the LP. Let  $\mathbf{f}$  be the strategy-function profile defined as

$$\begin{aligned} f_i^{\text{query}} &: q_i \mapsto y_{q_i}^i \\ f_i^{\text{resp}}(q_i, q_i(w)) &: a_i \mapsto x_{a_i, q_i, w}^i / y_{q_i}^i. \end{aligned}$$

Verifying that this strategy profile is a Nash equilibrium requires checking that  $f_i^{\text{resp}}(q_i, q_i(w))$  is a well-defined function (from constraint VI), that  $f_i^{\text{query}}$  and  $f_i^{\text{resp}}(q_i, q_i(w))$  are probability distributions (from constraints III and IV), and that each player is playing a best response to his or her opponent's strategy (from constraints I and II). Finally, from constraints I and II, the expected payoff to Player  $i$  is at most  $\rho_i$ . Because the right-hand side of constraint VII is equal to the expected sum of the payoffs from  $\mathbf{f}$  and is at most  $\rho_{\text{I}} + \rho_{\text{II}}$ , the payoffs are correct and imply the lemma.  $\square$

We now give an efficient separation oracle for the LP in Figure 1, thus allowing the ellipsoid method to solve the LP in polynomial time. Recall that a separation oracle is a function that, given a setting for the variables in the LP, either returns “feasible” or returns a particular constraint of the LP that is violated by that setting of the variables. An efficient, correct separation oracle allows us to solve the LP efficiently via the ellipsoid method.

**Lemma 4.2.** *There exists a separation oracle for the LP in Figure 1 that is correct and runs in polynomial time.*

*Proof.* Here is a description of the separation oracle SP. On input  $\langle x_{a_i, q_i, w}^i, y_{q_i}^i, \rho_i \rangle$ :

1. Check each of the constraints (III), (IV), (V), (VI), and (VII). If any one of these constraints is violated, then return it.
2. Define the strategy profile  $\mathbf{f}$  as follows:

$$\begin{aligned} f_i^{\text{query}} &: q_i \mapsto y_{q_i}^i \\ f_i^{\text{resp}}(q_i, q_i(w)) &: a_i \mapsto x_{a_i, q_i, w}^i / y_{q_i}^i \end{aligned}$$



For each query  $q_I$ , we will compute a pure best-response function  $\hat{f}_I^{q_I}$  for Player I to strategy  $f_{II}$  after making query  $q_I$ .

More specifically, given  $f_{II}$  and the result  $q_I(w_{\text{real}})$  of the query  $q_I$ , it is straightforward to compute the probability that, conditioned on the fact that the result of query  $q_I$  is  $q_I(w)$ , the world is  $w$  and Player II will play action  $a_{II} \in A_{II}$ . Therefore, for each query  $q_I$  and response  $q_I(w)$ , Player I can compute the expected utility of each pure response  $a_I$  to the induced mixed strategy over  $A_{II}$  for Player II. Player I can then select the  $a_I$  maximizing this expected payoff.

Let  $\hat{f}_I$  be the response function such that  $\hat{f}_I(q_I, q_I(w)) = \hat{f}_I^{q_I}(q_I(w))$  for every  $q_I \in Q_I$ . Similarly, compute  $\hat{f}_{II}$ .

3. Let  $\hat{\rho}_I^{q_I}$  be the expected payoff to Player I using the strategy “make query  $q_I$  and play response function  $\hat{f}_I$ ” if Player II plays according to  $f_{II}$ .

Let  $\hat{\rho}_I = \max_{q_I \in Q_I} \hat{\rho}_I^{q_I}$  and let  $\hat{q}_I = \arg \max_{q_I \in Q_I} \hat{\rho}_I^{q_I}$ . Similarly, define  $\hat{\rho}_{II}^{q_{II}}$ ,  $\hat{\rho}_{II}$ , and  $\hat{q}_{II}$ .

4. For the  $\hat{f}_i$  and  $\hat{q}_i$  defined in Step 3, return constraint (I- $\hat{q}_I$ - $\hat{f}_I$ ) or (II- $\hat{q}_{II}$ - $\hat{f}_{II}$ ) if either is violated. If both are satisfied, then return “feasible.”

We first note that the separation oracle runs in polynomial time and then prove its correctness. Steps 1 and 4 are clearly polynomial. For Step 2, we have described how to compute the relevant response functions by examining every action of Player I, every world, every query, and every action of Player II. There are only polynomially many queries, worlds, query results, and pure actions, so the running time of Steps 2 and 3 is thus polynomial.

We now sketch the proof that the separation oracle works correctly. The main challenge is to show that if any constraint (I- $q'_I$ - $f'_I$ ) is violated then (I- $\hat{q}_I$ - $\hat{f}_I$ ) is violated in Step 4. First, we observe that, by construction, the function  $\hat{f}_I$  computed in Step 3 must be a best response to Player II playing  $f_{II}$ , no matter what query Player I makes. Therefore the strategy “make query  $\hat{q}_I$ , then play response function  $\hat{f}_I$ ” must be a best response to Player II playing  $f_{II}$ , by definition of  $\hat{q}_I$ . The right-hand side of each constraint (I- $q'_I$ - $f'_I$ ) is equal to the expected payoff that Player I receives when playing the pure strategy “make query  $q'_I$  and then play response function  $f'_I$ ” against Player II’s strategy of  $f_{II}$ . Therefore, because the pure strategy “make query  $\hat{q}_I$  and then play response function  $\hat{f}_I$ ” is a best response to Player II playing  $f_{II}$ , the right-hand side of constraint (I- $\hat{q}_I$ - $\hat{f}_I$ ) is at least as large as the right hand side of any constraint (I- $\hat{q}_I$ - $f'_I$ ). Therefore, if any constraint (I- $q'_I$ - $f'_I$ ) is violated, constraint (I- $\hat{q}_I$ - $\hat{f}_I$ ) is also violated. An analogous argument holds for Player II.  $\square$

These lemmas and the well-known fact that Nash equilibria always exist [52] imply the following theorem:

**Theorem 4.3.** *Nash equilibria can be found in polynomial time for any two-player unobservable-query Socratic game with strategically zero-sum worlds.*  $\square$

## 5 Socratic Games with Observable Queries

In this section, we give efficient algorithms to find (1) a Nash equilibrium for observable-query Socratic games with constant-sum worlds and (2) a correlated equilibrium in the broader class of Socratic games with strategically zero-sum worlds. Recall that a Socratic game  $G = \langle \mathbf{A}, W, \vec{\mathbf{u}}, S, \mathbf{Q}, p, \delta \rangle$  with observable queries proceeds in two stages:

**Stage 1:** The players simultaneously choose queries  $\mathbf{q} \in \mathbf{Q}$ . Player  $i$  receives as output  $q_i$ ,  $q_{\text{II}}$ , and  $q_i(w_{\text{real}})$ .

**Stage 2:** The players simultaneously choose strategies  $\mathbf{a} \in \mathbf{A}$ . The payoff to Player  $i$  is  $u_i^{w_{\text{real}}}(\mathbf{a}) - \delta_i(q_i)$ .

Using backward induction, we first solve Stage 2 and then proceed to the Stage-1 game.

For a query  $\mathbf{q} \in \mathbf{Q}$ , we would like to analyze the Stage-2 “game”  $\hat{G}_{\mathbf{q}}$  resulting from the players making queries  $\mathbf{q}$  in Stage 1. Technically, however,  $\hat{G}_{\mathbf{q}}$  is not actually a game, because at the beginning of Stage 2 the players have different information about the world: Player I knows  $q_I(w_{\text{real}})$ , and Player II knows  $q_{\text{II}}(w_{\text{real}})$ . Fortunately, the situation in which players have asymmetric private knowledge has been well studied in the game-theory literature. A *Bayesian game* is a quadruple  $\langle \mathbf{A}, \mathbf{T}, r, \mathbf{u} \rangle$ , where:

- $A_i$  is the set of *pure strategies* for Player  $i$ .
- $T_i$  is the set of *types* for Player  $i$ .
- $r$  is a probability distribution over  $\mathbf{T}$ ;  $r(\mathbf{t})$  denotes the probability that Player  $i$  has type  $t_i$  for all  $i$ .
- $u_i : \mathbf{A} \times \mathbf{T} \rightarrow \mathbb{R}$  is the payoff function for Player  $i$ . If the players have types  $\mathbf{t}$  and play pure strategies  $\mathbf{a}$ , then  $u_i(\mathbf{a}, \mathbf{t})$  denotes the payoff for Player  $i$ .

Initially, a type  $\mathbf{t}$  is drawn randomly from  $\mathbf{T}$  according to the distribution  $r$ . Player  $i$  learns his type  $t_i$ , but does not learn any other player’s type. Player  $i$  then plays a mixed strategy  $\alpha_i \in \mathcal{A}_i$ —that is, a probability distribution over  $A_i$ —and receives payoff  $u_i(\alpha, \mathbf{t})$ . A *strategy function* is a function  $h_i : T_i \rightarrow \mathcal{A}_i$ ; Player  $i$  plays the mixed strategy  $h_i(t_i) \in \mathcal{A}_i$  when her type is  $t_i$ . A strategy-function profile  $\mathbf{h}$  is a *Bayesian Nash equilibrium* if and only if no Player  $i$  has unilateral incentive to deviate from  $h_i$  if the other players play according to  $\mathbf{h}$ . For a two-player Bayesian game, if  $\alpha = \mathbf{h}(\mathbf{t})$ , then the profile  $\mathbf{h}$  is a Bayesian Nash equilibrium exactly when the following condition and its analogue for Player II hold:  $\mathbf{E}_{\mathbf{t} \sim r}[u_i(\alpha, \mathbf{t})] = \max_{h'_i} \mathbf{E}_{\mathbf{t} \sim r}[u_i(\langle h'_i(t_i), \alpha_{\text{II}} \rangle, \mathbf{t})]$ . These conditions hold if and only if, for all  $t_i \in T_i$  occurring with positive probability, Player  $i$ ’s expected utility conditioned on his type being  $t_i$  is maximized by  $h_i(t_i)$ . A Bayesian game is *constant sum* if for all  $\mathbf{a} \in \mathbf{A}$  and all  $\mathbf{t} \in \mathbf{T}$ , we have  $u_i(\mathbf{a}, \mathbf{t}) + u_{\text{II}}(\mathbf{a}, \mathbf{t}) = c_{\mathbf{t}}$ , for some constant  $c_{\mathbf{t}}$  independent of  $\mathbf{a}$ . A Bayesian game is *strategically zero sum* if the classical game  $\langle \mathbf{A}, \mathbf{u}(\cdot, \mathbf{t}) \rangle$  is strategically zero sum for every  $\mathbf{t} \in \mathbf{T}$ . Whether a Bayesian game is strategically zero sum can be determined as in Theorem 3.1. (For further discussion of Bayesian games, see [25, 31].)

We now formally define the Stage-2 “game” as a Bayesian game. Given a Socratic game  $G = \langle \mathbf{A}, W, \vec{\mathbf{u}}, S, \mathbf{Q}, p, \delta \rangle$  and a query profile  $\mathbf{q} \in \mathbf{Q}$ , we define the Stage-2 Bayesian game  $G_{\text{stage2}}(\mathbf{q}) := \langle \mathbf{A}, \mathbf{T}^{\mathbf{q}}, p^{\text{stage2}}(\mathbf{q}), \mathbf{u}^{\text{stage2}}(\mathbf{q}) \rangle$ , where:

- $A_i$ , the set of pure strategies for Player  $i$ , is the same as in the original Socratic game;
- $T_i^{\mathbf{q}} = \{q_i(w) : w \in W\}$ , the set of types for Player  $i$ , is the set of signals that can result from query  $q_i$ ;
- $p^{\text{stage2}}(\mathbf{q})(\mathbf{t}) = \Pr[\mathbf{q}(w) = \mathbf{t} \mid w \leftarrow p]$ ; and

- $u_i^{\text{stage2}(\mathbf{q})}(\mathbf{a}, \mathbf{t}) = \sum_{w \in W} \Pr[w \leftarrow p \mid \mathbf{q}(w) = \mathbf{t}] \cdot u_i^w(\mathbf{a})$ .

We now define the Stage-1 game in terms of the payoffs for the Stage-2 games. Fix any algorithm  $\text{alg}$  that finds a Bayesian Nash equilibrium  $\mathbf{h}^{\mathbf{q}, \text{alg}} := \text{alg}(G_{\text{stage2}}(\mathbf{q}))$  for each Stage-2 game. Define  $\text{value}_i^{\text{alg}}(G_{\text{stage2}}(\mathbf{q}))$  to be the expected payoff received by Player  $i$  in the Bayesian game  $G_{\text{stage2}}(\mathbf{q})$  if each player plays according to  $\mathbf{h}^{\mathbf{q}, \text{alg}}$ , that is,

$$\begin{aligned} \text{value}_i^{\text{alg}}(G_{\text{stage2}}(\mathbf{q})) \\ := \sum_{w \in W} p(w) \cdot u_i^{\text{stage2}(\mathbf{q})}(\mathbf{h}^{\mathbf{q}, \text{alg}}(\mathbf{q}(w)), \mathbf{q}(w)). \end{aligned}$$

Define the game  $G_{\text{stage1}}^{\text{alg}} := \langle \mathbf{A}^{\text{stage1}}, \mathbf{u}^{\text{stage1}(\text{alg})} \rangle$ , where:

- $\mathbf{A}^{\text{stage1}} := \mathbf{Q}$ , the set of available queries in the Socratic game; and
- $u_i^{\text{stage1}(\text{alg})}(\mathbf{q}) := \text{value}_i^{\text{alg}}(G_{\text{stage2}}(\mathbf{q})) - \delta_i(q_i)$ .

I.e., players choose queries  $\mathbf{q}$  and receive payoffs corresponding to  $\text{value}_i^{\text{alg}}(G_{\text{stage2}}(\mathbf{q}))$ , less query costs.

**Lemma 5.1.** *Consider an observable-query Socratic game  $G = \langle \mathbf{A}, W, \vec{\mathbf{u}}, S, \mathbf{Q}, p, \delta \rangle$ . Let  $G_{\text{stage2}}(\mathbf{q})$  be the Stage-2 games for all  $\mathbf{q} \in \mathbf{Q}$ , let  $\text{alg}$  be an algorithm finding a Bayesian Nash equilibrium in each  $G_{\text{stage2}}(\mathbf{q})$ , and let  $G_{\text{stage1}}^{\text{alg}}$  be the Stage-1 game. Let  $\alpha$  be a Nash equilibrium for  $G_{\text{stage1}}^{\text{alg}}$ , and let  $\mathbf{h}^{\mathbf{q}, \text{alg}} := \text{alg}(G_{\text{stage2}}(\mathbf{q}))$  be a Bayesian Nash equilibrium for each  $G_{\text{stage2}}(\mathbf{q})$ . Then the following strategy profile is a Nash equilibrium for  $G$ :*

- In Stage 1, Player  $i$  makes query  $q_i$  with probability  $\alpha_i(q_i)$ . (That is, set  $\mathbf{f}^{\text{query}}(\mathbf{q}) := \alpha(\mathbf{q})$ .)
- In Stage 2, if  $\mathbf{q}$  is the query in Stage 1 and  $q_i(w_{\text{real}})$  denotes the response to Player  $i$ 's query, then Player  $i$  chooses action  $a_i$  with probability  $h_i^{\mathbf{q}, \text{alg}}(q_i(w_{\text{real}}))$ . (In other words, set  $f_i^{\text{resp}}(\mathbf{q}, q_i(w)) := h_i^{\mathbf{q}, \text{alg}}(q_i(w))$ .)  $\square$

We now find equilibria in the stage games for Socratic games with constant- or strategically zero-sum worlds. We first show that the stage games are well structured in this setting:

**Lemma 5.2.** *Consider an observable-query Socratic game  $G = \langle \mathbf{A}, W, \vec{\mathbf{u}}, S, \mathbf{Q}, p, \delta \rangle$  with constant-sum worlds. Then the Stage-1 game  $G_{\text{stage1}}^{\text{alg}}$  is strategically zero sum for every algorithm  $\text{alg}$ , and every Stage-2 game  $G_{\text{stage2}}(\mathbf{q})$  is Bayesian constant sum. If the worlds of  $G$  are strategically zero sum, then every  $G_{\text{stage2}}(\mathbf{q})$  is Bayesian strategically zero sum.  $\square$*

We now show that we can efficiently compute equilibria for these well-structured stage games.

**Theorem 5.3.** *There exists a polynomial-time algorithm BNE finding Bayesian Nash equilibria in strategically zero-sum Bayesian (and thus classical strategically zero-sum or Bayesian constant-sum) two-player games.*

*Sketch.* Let  $G = \langle \mathbf{A}, \mathbf{T}, r, \mathbf{u} \rangle$  be a strategically zero-sum Bayesian game. Define an unobservable-query Socratic game  $G^*$  with one possible world for each  $\mathbf{t} \in \mathbf{T}$ , one available zero-cost query  $q_i$  for each Player  $i$  so that  $q_i$  reveals  $t_i$ , and all else as in  $G$ . Bayesian Nash equilibria in  $G$  correspond directly to Nash equilibria in  $G^*$ , and the worlds of  $G^*$  are strategically zero sum. Thus by Theorem 4.3 we can compute Nash equilibria for  $G^*$ , and thus we can compute Bayesian Nash equilibria for  $G$ .  $\square$

(LP’s for zero-sum two-player Bayesian games have been previously developed and studied [61].)

**Theorem 5.4.** *We can compute a Nash equilibrium for an arbitrary two-player observable-query Socratic game  $G = \langle \mathbf{A}, W, \vec{\mathbf{u}}, S, \mathbf{Q}, p, \delta \rangle$  with constant-sum worlds in polynomial time.*

*Proof.* Because each world of  $G$  is constant sum, Lemma 5.2 implies that the induced Stage-2 games  $G_{\text{stage2}}(\mathbf{q})$  are all Bayesian constant sum. Thus we can use algorithm BNE to compute a Bayesian Nash equilibrium  $\mathbf{h}^{\mathbf{q}, \text{BNE}} := \text{BNE}(G_{\text{stage2}}(\mathbf{q}))$  for each  $\mathbf{q} \in \mathbf{Q}$ , by Theorem 5.3. Furthermore, again by Lemma 5.2, the induced Stage-1 game  $G_{\text{stage1}}^{\text{BNE}}$  is classical strategically zero sum. Therefore we can again use algorithm BNE to compute a Nash equilibrium  $\alpha := \text{BNE}(G_{\text{stage1}}^{\text{BNE}})$ , again by Theorem 5.3. Therefore, by Lemma 5.1, we can assemble  $\alpha$  and the  $\mathbf{h}^{\mathbf{q}, \text{BNE}}$ ’s into a Nash equilibrium for the Socratic game  $G$ .  $\square$

We would like to extend our results on observable-query Socratic games to Socratic games with strategically zero-sum worlds. While we can still find Nash equilibria in the Stage-2 games, the resulting Stage-1 game is not in general strategically zero sum. Thus, finding Nash equilibria in observable-query Socratic games with strategically zero-sum worlds seems to require substantially new techniques. However, our techniques for decomposing observable-query Socratic games do allow us to find correlated equilibria in this case.

**Lemma 5.5.** *Consider an observable-query Socratic game  $G = \langle \mathbf{A}, W, \vec{\mathbf{u}}, S, \mathbf{Q}, p, \delta \rangle$ . Let  $\text{alg}$  be an arbitrary algorithm that finds a Bayesian Nash equilibrium in each of the derived Stage-2 games  $G_{\text{stage2}}(\mathbf{q})$ , and let  $G_{\text{stage1}}^{\text{alg}}$  be the derived Stage-1 game. Let  $\phi$  be a correlated equilibrium for  $G_{\text{stage1}}^{\text{alg}}$ , and let  $\mathbf{h}^{\mathbf{q}, \text{alg}} := \text{alg}(G_{\text{stage2}}(\mathbf{q}))$  be a Bayesian Nash equilibrium for each  $G_{\text{stage2}}(\mathbf{q})$ . Then the following distribution over pure strategies is a correlated equilibrium for  $G$ :*

$$\psi(\mathbf{q}, \mathbf{f}) := \phi(\mathbf{q}) \prod_{i \in \{I, II\}} \prod_{s \in S} \Pr \left[ f_i(\mathbf{q}, s) \leftarrow h_i^{\mathbf{q}, \text{alg}}(s) \right]. \square$$

Thus to find a correlated equilibrium in an observable-query Socratic game with strategically zero-sum worlds, we need only algorithm BNE from Theorem 5.3 along with an efficient algorithm for finding a correlated equilibrium in a general game. Such an algorithm exists (the definition of correlated equilibria can be directly translated into an LP [3]), and therefore we have the following theorem:

**Theorem 5.6.** *We can provide both efficient oracle access and efficient sampling access to a correlated equilibrium for any observable-query two-player Socratic game with strategically zero-sum worlds.*  $\square$

Because the support of the correlated equilibrium may be exponentially large, providing oracle and sampling access is the natural way to represent the correlated equilibrium.

By Lemma 5.5, we can also compute correlated equilibria in any observable-query Socratic game for which Nash equilibria are computable in the induced  $G_{\text{stage2}}(\mathbf{q})$  games (e.g., when  $G_{\text{stage2}}(\mathbf{q})$  is of constant size).

Another potentially interesting model of queries in Socratic games is what one might call *public queries*, in which both the choice *and* outcome of a player’s query is observable by all players in the game. (This model might be most appropriate in the presence of corporate espionage or media leaks, or in a setting in which the queries—and thus their results—are done in plain view.)

The techniques that we have developed in this section also yield exactly the same results as for observable queries. The proof is actually simpler: with public queries, the players’ payoffs are common knowledge when Stage 2 begins, and thus Stage 2 really is a complete-information game. (There may still be uncertainty about the real world, but all players use the observed signals to infer exactly the same set of possible worlds in which  $w_{\text{real}}$  may lie; thus they are playing a complete-information game against each other.) Thus we have the same results as in Theorems 5.4 and 5.6 more simply, by solving Stage 2 using a (non-Bayesian) Nash-equilibrium finder and solving Stage 1 as before.

Our results for observable queries are weaker than for unobservable: in Socratic games with worlds that are strategically zero sum but not constant sum, we find only a correlated equilibrium in the observable case, whereas we find a Nash equilibrium in the unobservable case. We might hope to extend our unobservable-query techniques to observable queries, but there is no obvious way to do so. The fundamental obstacle is that the LP’s payoff constraint becomes nonlinear if there is any dependence on the probability that the other player made a particular query. This dependence arises with observable queries, suggesting that observable Socratic games with strategically zero-sum worlds may be harder to solve.

## 6 Related Work

Our work was initially motivated by research in the social sciences indicating that real people seem (irrationally) paralyzed when they are presented with additional options. In this section, we briefly review some of these social-science experiments and then discuss technical approaches related to Socratic game theory.

*Prima facie*, a rational agent’s happiness given an added option can only increase. However, recent research has found that more choices tend to decrease happiness: for example, students choosing among extra-credit options are *more* likely to do extra credit if given a small subset of the choices and, moreover, produce higher-quality work [35]. (See also [19].) The psychology literature explores a number of explanations: people may miscalculate their *opportunity cost* by comparing their choice to a “component-wise maximum” of all other options instead of the single best alternative [65], a new option may draw undue attention to aspects of the other options [67], and so on. The present work explores an economic explanation of this phenomenon: *information is not free*. When there are more options, a decision-maker must spend more time to achieve a satisfactory outcome. See, e.g., the work of Skyrms [68] for a philosophical perspective on the role of deliberation in strategic situations. Finally, we note the connection between Socratic games and modal logic [34], a formalism for the logic of possibility and necessity.

The observation that human players typically do not play “rational” strategies has inspired some attempts to model “partially” rational players. The typical model of this so-called *bounded rationality* [36, 64, 66] is to postulate bounds on computational power in computing the consequences of a strategy. The work on bounded rationality [23, 24, 53, 58] differs from the models that we consider here in that instead of putting hard limitations on the computational power of the agents, we instead restrict their *a priori* knowledge of the state of the world, requiring them to spend time (and therefore money/utility) to learn about it.

*Partially observable stochastic games* (POSGs) are a general framework used in AI to model situations of multi-agent planning in an evolving, unknown environment, but the generality of POSGs seems to make them very difficult [6]. Recent work has been done in developing algorithms

for restricted classes of POSGs, most notably classes of cooperative POSGs—e.g., [20, 30]—which are very different from the competitive strategically zero-sum games we address in this paper.

The fundamental question in Socratic games is deciding on the comparative value of making a more costly but more informative query, or concluding the data-gathering phase and picking the best option, given current information. This tradeoff has been explored in a variety of other contexts; a sampling of these contexts includes aggregating results from delay-prone information sources [8], doing approximate reasoning in intelligent systems [72], deciding when to take the current best guess of disease diagnosis from a belief-propagation network and when to let it continue inference [33], among many others.

This issue can also be viewed as another perspective on the general question of *exploration versus exploitation* that arises often in AI: when is it better to actively seek additional information instead of exploiting the knowledge one already has? (See, e.g., [69].) Most of this work differs significantly from our own in that it considers single-agent planning as opposed to the game-theoretic setting. A notable exception is the work of Larson and Sandholm [41, 42, 43, 44] on mechanism design for interacting agents whose computation is costly and limited. They present a model in which players must solve a computationally intractable valuation problem, using costly computation to learn some hidden parameters, and results for auctions and bargaining games in this model.

## 7 Future Directions

Efficiently finding Nash equilibria in Socratic games with non-strategically zero-sum worlds is probably difficult because the existence of such an algorithm for classical games has been shown to be unlikely [10, 11, 13, 16, 17, 27, 54, 55]. There has, however, been some algorithmic success in finding Nash equilibria in restricted classical settings (e.g., [21, 46, 47, 57]); we might hope to extend our results to analogous Socratic games.

An efficient algorithm to find correlated equilibria in general Socratic games seems more attainable. Suppose the players receive recommended queries and responses. The difficulty is that when a player considers a deviation from his recommended query, he already knows his recommended response in each of the Stage-2 games. In a correlated equilibrium, a player’s expected payoff generally depends on his recommended strategy, and thus a player may deviate in Stage 1 so as to land in a Stage-2 game where he has been given a “better than average” recommended response. (Socratic games are “succinct games of superpolynomial type,” so Papadimitriou’s results [56] do not imply correlated equilibria for them.)

Socratic games can be extended to allow players to make *adaptive* queries, choosing subsequent queries based on previous results. Our techniques carry over to  $O(1)$  rounds of unobservable queries, but it would be interesting to compute equilibria in Socratic games with adaptive observable queries or with  $\omega(1)$  rounds of unobservable queries. Special cases of adaptive Socratic games are closely related to single-agent problems like minimum latency [1, 7, 26], determining strategies for using priced information [9, 29, 37], and an online version of minimum test cover [18, 50]. Although there are important technical distinctions between adaptive Socratic games and these problems, approximation techniques from this literature may apply to Socratic games. The question of approximation raises interesting questions even in non-adaptive Socratic games. An  $\varepsilon$ -*approximate Nash equilibrium* is a strategy profile  $\alpha$  so that no player can increase her payoff by an additive  $\varepsilon$  by deviating from  $\alpha$ . Finding approximate Nash equilibria in both adaptive and non-adaptive Socratic games is an interesting direction to pursue.

Another natural extension is the model where query results are stochastic. In this paper, we model a query as deterministically partitioning the possible worlds into subsets that the query cannot distinguish. However, one could instead model a query as probabilistically mapping the set of possible worlds into the set of signals. With this modification, our unobservable-query model becomes equivalent to the model of Bergemann and Välimäki [4, 5], in which the result of a query is a posterior distribution over the worlds. Our techniques allow us to compute equilibria in such a “stochastic-query” model provided that each query is represented as a table that, for each world/signal pair, lists the probability that the query outputs that signal in that world. It is also interesting to consider settings in which the game’s queries are specified by a compact representation of the relevant probability distributions. (For example, one might consider a setting in which the algorithm has only a sampling oracle for the posterior distributions envisioned by Bergemann and Välimäki.) Efficiently finding equilibria in such settings remains an open problem.

Another interesting setting for Socratic games is when the set  $Q$  of available queries is given by  $Q = \mathcal{P}(\Gamma)$ —i.e., each player chooses to make a set  $q \in \mathcal{P}(\Gamma)$  of queries from a specified groundset  $\Gamma$  of queries. Here we take the query cost to be a linear function, so that  $\delta(q) = \sum_{\gamma \in q} \delta(\{\gamma\})$ . Natural groundsets include *comparison queries* (“if my opponent is playing strategy  $a_{II}$ , would I prefer to play  $a_I$  or  $\hat{a}_I$ ?”), *strategy queries* (“what is my vector of payoffs if I play strategy  $a_I$ ?”), and *world-identity queries* (“is the world  $w \in W$  the real world?”). When one can infer a polynomial bound on the number of queries made by a rational player, then our results yield efficient solutions. (For example, we can efficiently solve games in which every groundset element  $\gamma \in \Gamma$  has  $\delta(\{\gamma\}) = \Omega(\overline{M} - \underline{M})$ , where  $\overline{M}$  and  $\underline{M}$  denote the maximum and minimum payoffs to any player in any world.) Conversely, it is NP-hard to compute a Nash equilibrium for such a game when every  $\delta(\{\gamma\}) \leq 1/|W|^2$ , even when the worlds are constant sum and Player II has only a single available strategy. Thus even computing a best response for Player I is hard. (This proof proceeds by reduction from set cover; intuitively, for sufficiently low query costs, Player I must fully identify the actual world through his queries. Selecting a minimum-sized set of these queries is hard.) Computing Player I’s best response can be viewed as maximizing a submodular function, and thus a best response can be  $(1 - 1/e) \approx 0.63$  approximated greedily [14]. An interesting open question is whether this approximate best-response calculation can be leveraged to find an approximate Nash equilibrium.

## 8 Acknowledgements

Part of this work was done while all authors were at MIT CSAIL. We thank Erik Demaine, Natalia Hernandez Gardiol, Claire Monteleoni, Jason Rennie, Madhu Sudan, and Katherine White for helpful comments and discussions.

## References

- [1] Aaron Archer and David P. Williamson. Faster approximation algorithms for the minimum latency problem. In *Proceedings of the Symposium on Discrete Algorithms*, pages 88–96, 2003.
- [2] R. J. Aumann. Subjectivity and correlation in randomized strategies. *J. Mathematical Economics*, 1:67–96, 1974.

- [3] Robert J. Aumann. Correlated equilibrium as an expression of Bayesian rationality. *Econometrica*, 55(1):1–18, January 1987.
- [4] Dick Bergemann and Juuso Välimäki. Information acquisition and efficient mechanism design. *Econometrica*, 70(3):1007–1033, May 2002.
- [5] Dick Bergemann and Juuso Välimäki. Information in mechanism design. Technical Report 1532, Cowles Foundation for Research in Economics, 2005.
- [6] Daniel S. Bernstein, Shlomo Zilberstein, and Neil Immerman. The complexity of decentralized control of Markov Decision Processes. *Mathematics of Operations Research*, pages 819–840, 2002.
- [7] Avrim Blum, Prasad Chalasani, Don Coppersmith, Bill Pulleyblank, Prabhakar Raghavan, and Madhu Sudan. The minimum latency problem. In *Proceedings of the Symposium on the Theory of Computing*, pages 163–171, 1994.
- [8] Andrei Z. Broder and Michael Mitzenmacher. Optimal plans for aggregation. In *Proceedings of the Principles of Distributed Computing*, pages 144–152, 2002.
- [9] Moses Charikar, Ronald Fagin, Venkatesan Guruswami, Jon Kleinberg, Prabhakar Raghavan, and Amit Sahai. Query strategies for priced information. *J. Computer and System Sciences*, 64(4):785–819, June 2002.
- [10] Xi Chen and Xiaotie Deng. 3-NASH is PPAD-complete. In *Electronic Colloquium on Computational Complexity*, 2005.
- [11] Xi Chen and Xiaotie Deng. Settling the complexity of 2-player Nash-equilibrium. In *Electronic Colloquium on Computational Complexity*, 2005.
- [12] Olivier Compte and Philippe Jehiel. Auctions and information acquisition: Sealed-bid or dynamic formats? Technical report, Centre d’Enseignement et de Recherche en Analyse Socio-économique, 2002.
- [13] Vincent Conitzer and Tuomas Sandholm. Complexity results about Nash equilibria. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 765–771, 2003.
- [14] Gerard Cornuejols, Marshall L. Fisher, and George L. Nemhauser. Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. *Management Science*, 23(8), April 1977.
- [15] Jacques Crémer and Fahad Khalil. Gathering information before signing a contract. *American Economic Review*, 82:566–578, 1992.
- [16] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a Nash equilibrium. In *Electronic Colloquium on Computational Complexity*, 2005.
- [17] Konstantinos Daskalakis and Christos H. Papadimitriou. Three-player games are hard. In *Electronic Colloquium on Computational Complexity*, 2005.



- [18] K. M. J. De Bontridder, B. V. Halldórsson, M. M. Halldórsson, C. A. J. Hurkens, J. K. Lenstra, R. Ravi, and L. Stougie. Approximation algorithms for the test cover problem. *Mathematical Programming*, 98(1–3):477–491, September 2003.
- [19] Ap Dijksterhuis, Maarten W. Bos, Loran F. Nordgren, and Rick B. van Baaren. On making the right choice: The deliberation-without-attention effect. *Science*, 311:1005–1007, 17 February 2006.
- [20] Rosemary Emery-Montemerlo, Geoff Gordon, Jeff Schneider, and Sebastian Thrun. Approximate solutions for partially observable stochastic games with common payoffs. In *Autonomous Agents and Multi-Agent Systems*, 2004.
- [21] Alex Fabrikant, Christos Papadimitriou, and Kunal Talwar. The complexity of pure Nash equilibria. In *Proceedings of the Symposium on the Theory of Computing*, 2004.
- [22] Kyna Fong. *Multi-stage Information Acquisition in Auction Design*. Senior thesis, Harvard College, 2003.
- [23] Lance Fortnow and Duke Whang. Optimality and domination in repeated games with bounded players. In *Proceedings of the Symposium on the Theory of Computing*, pages 741–749, 1994.
- [24] Yoav Freund, Michael Kearns, Yishay Mansour, Dana Ron, Ronitt Rubinfeld, and Robert E. Schapire. Efficient algorithms for learning to play repeated games against computationally bounded adversaries. In *Proceedings of the Foundations of Computer Science*, pages 332–341, 1995.
- [25] Drew Fudenberg and Jean Tirole. *Game Theory*. MIT, 1991.
- [26] Michel X. Goemans and Jon Kleinberg. An improved approximation ratio for the minimum latency problem. *Mathematical Programming*, 82:111–124, 1998.
- [27] Paul W. Goldberg and Christos H. Papadimitriou. Reducibility among equilibrium problems. In *Electronic Colloquium on Computational Complexity*, 2005.
- [28] M. Grotschel, L. Lovasz, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:70–89, 1981.
- [29] Anupam Gupta and Amit Kumar. Sorting and selection with structured costs. In *Proceedings of the Foundations of Computer Science*, pages 416–425, 2001.
- [30] Eric A. Hansen, Daniel S. Bernstein, and Shlomo Zilberstein. Dynamic programming for partially observable stochastic games. In *National Conference on Artificial Intelligence (AAAI)*, 2004.
- [31] John C. Harsanyi. Games with incomplete information played by “Bayesian” players. *Management Science*, 14(3,5,7), 1967–1968.
- [32] Sergiu Hart and David Schmeidler. Existence of correlated equilibria. *Mathematics of Operations Research*, 14(1):18–25, 1989.

- [33] Eric Horvitz and Geoffrey Rutledge. Time-dependent utility and action under uncertainty. In *Uncertainty in Artificial Intelligence*, pages 151–158, 1991.
- [34] G. E. Hughes and M. J. Cresswell. *A New Introduction to Modal Logic*. Routledge, 1996.
- [35] Sheena S. Iyengar and Mark R. Lepper. When choice is demotivating: Can one desire too much of a good thing? *J. Personality and Social Psychology*, 79(6):995–1006, 2000.
- [36] Ehud Kalai. Bounded rationality and strategic complexity in repeated games. *Game Theory and Applications*, pages 131–157, 1990.
- [37] Sampath Kannan and Sanjeev Khanna. Selection with monotone comparison costs. In *Proceedings of the Symposium on Discrete Algorithms*, pages 10–17, 2003.
- [38] L.G. Khachiyan. A polynomial algorithm in linear programming. *Doklady Akademiia Nauk SSSR*, 244, 1979.
- [39] Daphne Koller and Nimrod Megiddo. The complexity of two-person zero-sum games in extensive form. *Games and Economic Behavior*, 4:528–552, 1992.
- [40] Daphne Koller, Nimrod Megiddo, and Bernhard von Stengel. Efficient computation of equilibria for extensive two-person games. *Games and Economic Behavior*, 14:247–259, 1996.
- [41] Kate Larson. *Mechanism Design for Computationally Limited Agents*. PhD thesis, CMU, 2004.
- [42] Kate Larson and Tuomas Sandholm. Bargaining with limited computation: Deliberation equilibrium. *Artificial Intelligence*, 132(2):183–217, 2001.
- [43] Kate Larson and Tuomas Sandholm. Costly valuation computation in auctions. In *Proceedings of the Theoretical Aspects of Rationality and Knowledge*, July 2001.
- [44] Kate Larson and Tuomas Sandholm. Strategic deliberation and truthful revelation: An impossibility result. In *Proceedings of the ACM Conference on Electronic Commerce*, May 2004.
- [45] C. E. Lemke and J. T. Howson, Jr. Equilibrium points of bimatrix games. *J. Society for Industrial and Applied Mathematics*, 12, 1964.
- [46] Richard J. Lipton, Evangelos Markakis, and Aranyak Mehta. Playing large games using simple strategies. In *Proceedings of the ACM Conference on Electronic Commerce*, pages 36–41, 2003.
- [47] Michael L. Littman, Michael Kearns, and Satinder Singh. An efficient exact algorithm for singly connected graphical games. In *Proceedings of Neural Information Processing Systems*, 2001.
- [48] Steven A. Matthews and Nicola Persico. Information acquisition and the excess refund puzzle. Technical Report 05-015, Department of Economics, University of Pennsylvania, March 2005.
- [49] Richard D. McKelvey and Andrew McLennan. Computation of equilibria in finite games. In H. Amman, D. A. Kendrick, and J. Rust, editors, *Handbook of Computational Economics*, volume 1, pages 87–142. Elsevier, 1996.

- [50] B.M.E. Moret and H. D. Shapiro. On minimizing a set of tests. *SIAM J. Scientific Statistical Computing*, 6:983–1003, 1985.
- [51] H. Moulin and J.-P. Vial. Strategically zero-sum games: The class of games whose completely mixed equilibria cannot be improved upon. *International J. Game Theory*, 7(3/4), 1978.
- [52] John F. Nash, Jr. Equilibrium points in  $n$ -person games. *Proceedings of the National Academy of Sciences*, 36:48–49, 1950.
- [53] Abraham Neyman. Finitely repeated games with finite automata. *Mathematics of Operations Research*, 23(3):513–552, August 1998.
- [54] Christos Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *J. Computer and System Sciences*, 48:498–532, 1994.
- [55] Christos Papadimitriou. Algorithms, games, and the internet. In *Proceedings of the Symposium on the Theory of Computing*, pages 749–753, 2001.
- [56] Christos H. Papadimitriou. Computing correlated equilibria in multi-player games. In *Proceedings of the Symposium on the Theory of Computing*, 2005.
- [57] Christos H. Papadimitriou and Tim Roughgarden. Computing equilibria in multiplayer games. In *Proceedings of the Symposium on Discrete Algorithms*, 2005.
- [58] Christos H. Papadimitriou and Mihalis Yannakakis. On bounded rationality and computational complexity. In *Proceedings of the Symposium on the Theory of Computing*, pages 726–733, 1994.
- [59] David C. Parkes. Auction design with costly preference elicitation. *Annals of Mathematics and Artificial Intelligence*, 44:269–302, 2005.
- [60] Nicola Persico. Information acquisition in auctions. *Econometrica*, 68(1):135–148, 2000.
- [61] Jean-Pierre Ponsard and Sylvain Sorin. The LP formulation of finite zero-sum games with incomplete information. *International J. Game Theory*, 9(2):99–105, 1980.
- [62] Eric Rasmussen. Strategic implications of uncertainty over one’s own private value in auctions. Technical report, Indiana University, 2005.
- [63] Leonardo Rezende. Mid-auction information acquisition. Technical report, University of Illinois, 2005.
- [64] Ariel Rubinstein. *Modeling Bounded Rationality*. MIT, 1988.
- [65] Barry Schwartz. *The Paradox of Choice: Why More is Less*. Ecco, 2004.
- [66] Herbert Simon. *Models of Bounded Rationality*. MIT, 1982.
- [67] I. Simonson and A. Tversky. Choice in context: Tradeoff contrast and extremeness aversion. *J. Marketing Research*, 29:281–295, 1992.

- [68] Brian Skyrms. Dynamic models of deliberation and the theory of games. In *Proceedings of the Theoretical Aspects of Rationality and Knowledge*, pages 185–200, 1990.
- [69] Richard Sutton and Andrew Barto. *Reinforcement Learning: An Introduction*. MIT, 1998.
- [70] John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton, 1957.
- [71] Bernhard von Stengel. Computing equilibria for two-person games. In R. J. Aumann and S. Hart, editors, *Handbook of Game Theory with Economic Applications*, volume 3, pages 1723–1759. Elsevier, 2002.
- [72] S. Zilberstein and S. Russell. Approximate reasoning using anytime algorithms. In S. Natarajan, editor, *Imprecise and Approximate Computation*. Kluwer, 1995.