3-3-2010

# Computing Equilibria by Incorporating Qualitative Models

Sam Ganzfried
*Carnegie Mellon University*

Tuomas W. Sandholm
*Carnegie Mellon University*

# Computing Equilibria by Incorporating Qualitative Models[1]

**Sam Ganzfried** **Tuomas Sandholm**

March 3, 2010
CMU-CS-10-105

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

[1]This material is based upon work supported by the National Science Foundation under grants IIS-0427858 and IIS-0905390. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. We also acknowledge Intel Corporation and IBM for their machine gifts.

**Keywords:** Game theory, continuous games, games of imperfect information, equilibrium computation, mixed integer programming, computer poker.

## Abstract

We present a new approach for solving large (even infinite) multiplayer games of imperfect information. The key idea behind our approach is that we include additional inputs in the form of qualitative models of equilibrium strategies (how the signal space should be qualitatively partitioned into action regions). In addition, we show that our approach can lead to strong strategies in large finite games that we approximate with infinite games. We prove that our main algorithm is correct even if given a set of qualitative models (satisfying a technical property) of which only some are accurate. We also show how to check the output in settings where all of the models might be wrong (under a weak assumption). Our algorithms can compute equilibria in several classes of games for which no prior algorithms have been developed, and we demonstrate that they run efficiently in practice. In the course of our analysis, we also develop the first mixed-integer programming formulations for computing an epsilon-equilibrium in general multiplayer normal and extensive-form games based on the extension of our initial algorithm to the multiplayer setting, which may be of independent interest. Experiments suggest that our approach of modeling a finite game with an infinite one can outperform the state of the art, abstraction-based approaches on some games. In addition, we demonstrate that our main algorithm can be used to improve play in two-player limit Texas hold'em—the most studied imperfect-information game in computer science—by solving endgames. Finally, we present experimental results on an infinite three-player game which suggest the effectiveness of our algorithm in games with more than two players.

# 1 Introduction

Finding Nash equilibria in games of imperfect information is an important problem. Significant progress has been made in the last few years. In particular, several algorithms have been developed for solving large (finite) two-player zero-sum imperfect-information games (e.g., [11, 22]). On the other hand, relatively little work has been done on developing algorithms for computing equilibria in imperfect-information games that are non-zero sum (with some notable exceptions, e.g., [12]), have more than two players (e.g., [10]), and/or are continuous (i.e., infinite) (e.g., [18]). Such games are significantly harder to solve in the complexity-theoretic sense: two-player zero-sum games can be solved in polynomial time, while two-player general-sum games are PPAD-hard [7], and games with three or more players are FIXP-hard [8].

To make matters worse, many interesting real-world games are so large that computing an equilibrium directly seems hopeless even in the two-player zero-sum case. The standard approach to deal with this is to run an abstraction algorithm on the full game to construct a smaller game that is strategically similar to the original game (e.g., [4]). Then the abstracted game is solved, and its solution is mapped to a strategy profile in the original game. While this approach seems quite natural and promising so far, recent research has shown that significant and surprising pathologies can arise in which a finer-grained abstraction results in strategies that are actually much more exploitable in the full game than the equilibrium strategies of a coarser abstraction [21]. Thus, this abstraction-based approach is not theoretically sound.

We develop a new approach for solving large games. Rather than construct a smaller game via an abstraction algorithm, we propose solving an infinite approximation of the original game, then mapping the equilibrium of the infinite game to a strategy profile in the original game. Perhaps counterintuitively, it is often the case that the infinite approximation can be solved much more easily than the finite game. We show that sometimes very fine abstractions would be needed to match the solution quality of our approach.

Our main algorithmic innovation takes advantage of the fact that in many multiagent settings it is significantly easier to infer qualitative models of the structure of equilibrium strategies than it is to actually compute an equilibrium. For example, in (sequences of) take-it-or-leave-it offers, equilibria involve accepting offers above a certain threshold and rejecting offers below it [14]. Threshold strategies are also common in auctions (e.g., [6]) and in deciding when to make and break partnerships and contracts (e.g., [16]). In poker the hole cards are private signals, and in equilibrium, often the same action is taken in continuous regions of the signal space (e.g., [2]).

We develop an approach for exploiting such qualitative models in equilibrium finding. We study a broad class of imperfect-information games where players are given private signals at the start. We first consider the two-player (general-sum) case in which private signals are drawn independently from finite sets. For this case, we develop an algorithm based on a mixed integer linear feasibility program (MILFP) formulation that provably computes an equilibrium assuming we are given a "correct" qualitative model as input. The size of the program is polynomial in the parameters of the problem and the constraints are very sparse, suggesting that it can be solved quickly in practice. Our experiments confirm that the algorithm runs very fast on a simplified endgame of limit Texas hold'em, leading to a significant performance improvement.

Next, we generalize our algorithm to computing equilibria in the following important exten-

sions: many players, continuous private signal distributions, dependent private signal distributions, and multiple candidate qualitative models that satisfy a certain technical property (some of which can be incorrect). In most of these cases, we present the first algorithm that provably solves the class of games. We also develop new mixed-integer programming based algorithms for computing equilibria in general multiplayer normal and extensive-form games based on the extension of our initial algorithm to the multiplayer setting, which may be of independent interest.

The remainder of the paper is organized as follows. In Section 2, we introduce continuous games and present relevant definitions and results. In Section 3, we present a continuous two-player game that is used to motivate the use of qualitative models and the setting of the remainder of the paper. Section 4 presents the main setting of our paper, and Section 5 introduces parametric (i.e., qualitative) models. Section 6 presents our main algorithm for solving large two-player games given a parametric model, as well as a proof of correctness of the algorithm. Section 7 presents several extensions of our main algorithm. Finally, Section 8 describes our experimental results: in Section 8.1 we present evidence that approximating large finite games with infinite games can outperform abstraction-based approaches; in Section 8.2 we demonstrate that our main algorithm leads to improved play in two-player limit Texas hold'em; and in Section 8.3 we demonstrate the effectiveness of our approach in the multiplayer setting.

# 2 Continuous Games

Continuous games generalize finite strategic-form games to the case of (uncountably) infinite strategy spaces. Many natural games have an uncountable number of actions; for example, games in which strategies correspond to an amount of time, money, or space. One example of a game that has recently been modeled as a continuous game in the AI literature is computational billiards, in which the strategies are vectors of real numbers corresponding to the orientation, location, and velocity at which to hit the ball [3].

**Definition 1.** *A* continuous game *is a tuple* $G = (N, S, U)$ *where*

- $N = \{1, 2, 3, \ldots, n\}$ *is the set of players,*
- $S = (S_1, \ldots, S_n)$ *where each* $S_i$ *is a metric space corresponding to the set of strategies of player* $i$*, and*
- $U = (u_1, \ldots, u_n)$ *where* $u_i : S \rightarrow \mathbb{R}$ *is the utility function of player* $i$*.*

The main result regarding the existence of a Nash equilibrium in continuous games is the following [9]:

**Theorem 1.** *Consider a strategic-form game whose strategy spaces* $S_i$ *are nonempty compact subsets of a metric space. If the payoff functions* $u_i$ *are continuous, there exists a (mixed strategy) Nash equilibrium.*

While this existence result has been around for a long time, there has been very little work on practical algorithms for computing equilibria in continuous games. One interesting class of continuous games for which algorithms have been developed is *separable games* [18]:

**Definition 2.** *A* separable game *is a continuous game with utility functions* $u_i : S \to \mathbb{R}$ *taking the form*

$$u_i(s) = \sum_{j_1=1}^{m_1} \cdots \sum_{j_n=1}^{m_n} a_i^{j_1 \cdots j_n} f_1^{j_1}(s_1) \cdots f_n^{j_n}(s_n),$$

*where* $a_i^{j_1 \cdots j_n} \in \mathbb{R}$ *and the* $f_i^j : S_i \to \mathbb{R}$ *are continuous.*

As we will see, this is a significant restriction on the utility functions, and many interesting continuous games are not separable. Additionally, algorithms for computing approximate equilibria have been developed for several other classes of infinite games, including simulation-based games [20] and graphical tree-games [17].

For a broad class of games, we will show that the equilibrium existence theorem above does not hold directly, but we can nevertheless prove the existence of an equilibrium by incorporating a qualitative equilibrium model. However, we show that these games are not separable, so the prior algorithm does not apply. These are the topics of the next two sections. After that, we will develop new algorithms for solving these games.

# 3 Motivating example

Consider the following simplified poker game [2]. Suppose two players are given private signals, $x_1$ and $x_2$, independently and uniformly at random from [0,1]. Suppose the pot initially has size $\rho$ (one can think of this as both players having put in an ante of $\frac{\rho}{2}$, or that we are at the final betting round—aka final *street*—of a multi-street game). Player 1 is allowed to bet or check. If player 1 checks, the game is over and the player with the *lower* private signal wins the pot (following the convention of [2]). If player 1 bets, then player 2 can call or fold. If player 2 folds, then player 1 wins the pot. If player 2 calls, then whoever has the lower private signal wins $\rho + 1$, while the other player loses 1. This situation can be thought of as an abstraction of the final street of a hand of limit Texas hold'em where raising is not allowed and player 2 has already checked.

It seems natural to define the strategy space $S_1$ of player 1 as the set of functions from $[0, 1]$ to $\{\text{check}, \text{bet}\}$ (i.e., to $\{0, 1\}$), and to define $S_2$ for player 2 as the set of functions from $[0, 1]$ to $\{\text{fold}, \text{call}\}$ (i.e., to $\{0, 1\}$). Let $p_i(a_1, a_2, x_1, x_2)$ denote the payoff of player $i$ when the players play actions $a_i$ and are given private signals $x_i$. Formally, $p_i$ is defined as follows:

$$
\begin{aligned}
p_1(1,0,x_1,x_2) &= \rho \\
p_1(0,a_2,x_1,x_2) &= \begin{cases} 0 & : \quad x_1 > x_2 \\ \rho & : \quad x_1 < x_2 \\ \frac{\rho}{2} & : \quad x_1 = x_2 \end{cases} \\
p_1(1,1,x_1,x_2) &= \begin{cases} -1 & : \quad x_1 > x_2 \\ \rho + 1 & : \quad x_1 < x_2 \\ \frac{\rho}{2} & : \quad x_1 = x_2 \end{cases} \\
p_2(a_1,a_2,x_1,x_2) &= \rho - p_1(a_1,a_2,x_1,x_2)
\end{aligned}
$$

Given this definition of $p_i$, the the utility of player $i$ under the strategy profile $s = (s_1, s_2)$ is defined as

$$u_i(s) = \int_{x_1=0}^{1} \int_{x_2=0}^{1} p_i(s_1(x_1), s_2(x_2), x_1, x_2) dx_2 dx_1.$$

We would like to represent each player's strategy set as a compact metric space, so that we can apply Theorem 1. Unfortunately, the naive representation does not yield compact metric spaces; so, we need to go through a number of transformations to achieve this goal. In particular, by iteratively eliminating dominated strategies, we arrive at a representation where each player's strategy space is isomorphic to a compact subset of a Euclidean space.

In order for $u_i(s)$ to be defined, we must restrict the strategy spaces $S_i$ to consist of only the measurable functions[1]. In addition, if we want to turn $S_i$ into a metric space, we need to define a distance function. A natural distance function to use is the $L_1$ distance function: $d_i(s_i, s_i') = \int_{X_i} |s_i(x_i) - s_i'(x_i)| dx_i$. Note that $(S_i, d_i)$ does not quite define a metric space because the condition $d_i(s,t) = 0$ iff $s = t$ is not satisfied. To turn it into a metric space, we can let $\sim$ be the equivalence relation defined by $s \sim_i s'$ iff $d_i(s, s') = 0$. If we then let $\overline{S_i}$ equal the set of equivalence classes with respect to $\sim_i$, then $(\overline{S_i}, d_i)$ is a metric space.

Unfortunately, the metric space $(\overline{S_i}, d_i)$ is not compact, and we cannot apply Theorem 1 to guarantee the existence of an equilibrium.

**Proposition 1.** *The metric space $(\overline{S_i}, d_i)$ is not compact.*

*Proof.* We prove this by showing that the space is not totally bounded; that is, that we cannot cover the space with a finite number of $\epsilon$-balls for at least one $\epsilon$. Let $\epsilon = 0.5$, and let $\{f_1, \ldots, f_k\}$ be a finite set of elements of $\overline{S_i}$ which is claimed to be a cover. For each $j \in \{1, \ldots, k\}$, let $g_j = \{x : f_j(x) = 0\}$, and let $h_j = \{x : f_j(x) = 1\}$. By the measurability of $f_j$, both $g_j$ and $h_j$ must consist of the union of intervals of $[0,1]$ of the following form: $(a,b), (a,b], [a,b), [a,b]$. Note that we can partition the interval $[0,1]$ by $0 = y_0 < y_1 < \ldots < y_{m-1} < 1 = y_m$ such that $(y_{i'}, y_{i'+1})$ is completely contained in one of the intervals of either $g_j$ or $h_j$ for all $i', j$. Now consider the function $f^*$ defined as follows: for each interval $[y_{i'}, y_{i'+1}]$, $f^*$ equals 0 for the first half of the interval and equals 1 for the second half. Then$f^*$ will agree with each $f_j$ for exactly half of the length of each interval. So $d_i(f^*, f_j) = 0.5 = \epsilon$ for all $j$; thus the set of $\epsilon$-balls centered at $\{f_1, \ldots, f_k\}$ is not a cover and we are done. □

[1]Relevant definitions from measure theory can be found in Appendix A.

Similarly, the space fails to be compact if we use other common distance functions, such as the discrete metric, any $L_p$ metric, or $L_\infty$. So we can not simply use one of those distance functions instead of $L_1$ to get around Proposition 1.

However, the following observations allow us to restrict our attention to a much smaller set of strategies.

**Definition 3.** *Let $S_i$ be the set of pure strategies of player $i$. Then $s_i \in S_i$ is* weakly dominated *for player $i$ if there exists a strategy $s_i^* \in S_i$ such that for all strategy profiles $s_{-i} \in S_{-i}$ for the other players, we have $u_i(s_i^*, s_{-i}) \geq u_i(s_i, s_{-i})$.*

**Definition 4.** *Let $\overline{s_i}$ be an equivalence class of player $i$'s pure strategies with respect to $\sim_i$. Then $\overline{s_i}$ is* weakly dominated *if $s_i$ is weakly dominated for all $s_i \in \overline{s_i}$.*

**Definition 5.** *The equivalence class $\overline{s_i}$ is* undominated *if it is not weakly dominated.*

**Proposition 2.** *The equivalence class of strategies $\overline{s_2} \in \overline{S_2}$ of player 2 is undominated only if it contains a unique strategy of the following form: call if $x_2 \leq x_2^*$ and fold otherwise, for some $x_2^*$.*

*Proof.* First note that any equivalence class that contains a strategy of the desired form can only contain a single such strategy (since otherwise it would contain two strategies that do not have zero distance from each other). Now suppose $\overline{s_2}$ is undominated but does not contain any strategies of the desired form. Let $s_2 \in \overline{s_2}$ be arbitrary. Then there must exist an interval $\alpha_1$ of one of the following forms with $a < b$: $(a, b), (a, b], [a, b), [a, b]$, and an interval $\alpha_2$ of one of the following forms with $c < d$: $(c, d), (c, d], [c, d), [c, d]$, with $b < c$, where $s_2$ calls with all private signals in $\alpha_2$ and folds with all private signals in $\alpha_1$. Now define $s^*$ to be the strategy that calls with all signals in $\alpha_1$, folds with all signals in $\alpha_2$, and agrees with $s_2$ otherwise. It is clear that $s^*$ (weakly)-dominates $s_2$, and we have a contradiction. So all equivalence classes of strategies that are undominated must be of the desired form. □

We can remove all of the strategies from $\overline{S_2}$ that are dominated according to Proposition 2 from our consideration, forming a much smaller strategy space. In addition, we can remove the strategies not satisfying the threshold property given in the proposition from each equivalence class $\overline{s_2} \in \overline{S_2}$, thus turning the equivalence classes into singletons. In the remainder of our discussion, we will let $\overline{S_2}$ denote this smaller set.

We can now iteratively remove many strategies from $S_1$ by the following observation.

**Proposition 3.** *The equivalence class of strategies $\overline{s_1} \in \overline{S_1}$ of player 1 is a best response to some element of $\overline{S_2}$ only if it contains a unique strategy of the following form: bet if $x_1 \leq \underline{x_1^*}$, check if $\underline{x_1^*} \leq x_1 \leq \overline{x_1^*}$ and bet if $\overline{x_1^*} \leq x_1 \leq 1$, for some $\underline{x_1^*} \leq \overline{x_1^*}$[2].*

*Proof.* The uniqueness part follows from the same reasoning as the proof of Proposition 2. Now suppose player 2 is playing the following strategy: call if $x_2 \leq x_2^*$ and fold otherwise. Let $x_1$ denote player 1's private signal.

[2]One can think of $[0, \underline{x_1^*}]$ as player 1's "value betting" range—where he bets hoping to get called by worse hands—and $[\overline{x_1^*}, 1]$ as his "bluffing" range—where he bets hoping to get better hands to fold.

- Case 1: $0 \leq x_1 \leq \frac{x_2^*}{2}$.
  Then the expected payoff of betting is

  $$(1 - x_2^*)\rho + (x_2^* - x_1)(\rho + 1) - x_1 = (1 - x_1)\rho + x_2^* - 2x_1,$$

  and the expected payoff of checking is $(1 - x_1)\rho$. So player 1's best response is to bet.

- Case 2: $\frac{x_2^*}{2} \leq x_1 \leq x_2^*$.
  Then the formulas for the expected payoffs of betting and checking are the same as in Case 1. However, now the expected payoff of checking is larger, so player 1's best response is to check.

- Case 3: $x_2^* \leq x_1 \leq x_2^* \cdot \frac{\rho+1}{\rho}$.
  The expected payoff of betting is

  $$(1 - x_2^*)\rho - x_2^* = \rho - \rho x_2^* - x_2^*,$$

  while the expected payoff of checking is $(1 - x_1)\rho$. So player 1's best response is to check.

- Case 4: $x_2^* \cdot \frac{\rho+1}{\rho} \leq x_1 \leq 1$.
  The formulas for the expected payoffs are the same as in case 3, but now player 1's best response is to bet.

□

After removing the dominated strategies, the new strategy space for player 1 becomes isomorphic to a compact subset of $\mathbb{R}^2$, and the new strategy space for player 2 becomes isomorphic to a compact subset of $\mathbb{R}$. Let $\hat{S}_1$ and $\hat{S}_2$ now refer to these new strategy spaces.

It turns out that the functions $u_i$ are continuous in $s$ for both players using the new strategy spaces, if we define the distance between two strategy profiles $s = (s_1, s_2)$ and $s' = (s_1', s_2')$ as $d(s, s') = d_1(s_1, s_1') + d_2(s_2, s_2')$.

**Proposition 4.** *For both players, the utility functions $u_i$ are continuous in $s$.*

*Proof.* Let $\epsilon > 0$ be arbitrary, and let $\delta = \frac{\epsilon}{2\rho+2}$. Let $s, s' \in S$ be arbitrary, with $s = (s_1, s_2)$ and $s' = (s_1', s_2')$, and suppose that $||s - s'|| \leq \delta$. Note that

$$||s - s'|| = \int_{x=0}^{1} |s_1(x) - s_1'(x)|dx + \int_{x=0}^{1} |s_2(x) - s_2'(x)|dx$$

and $||u_1(s) - u_1(s')||$ equals

$$\left| \int_X \int_Y \left[ u_1((s_1(x), s_2(y)), (x, y)) - u_1((s_1'(x), s_2'(y)), (x, y)) \right] dydx \right|.$$

Note that the maximum possible value of the integrand is $2\rho + 2$. So

$$||u_1(s) - u_1(s')|| \leq$$

$$(2\rho+2)\int_{x=0}^{1}\int_{y=0}^{1} I_1(s_1(x), s_2(y), s_1'(x), s_2'(y))dydx,$$

where $I_1(s_1(x), s_2(y), s_1'(x), s_2'(y))$ equals 0 iff $s_1(x) = s_1'(x)$ and $s_2(y) = s_2'(y)$ and equals 1 otherwise. With this definition, we have

$$I_1(s_1(x), s_2(y), s_1'(x), s_2'(y)) \leq |s_1(x) - s_x'(s)| + |s_2(y) - s_2'(y)|.$$

So we have

$$||u_1(s) - u_1(s')|| \leq$$

$$(2\rho+2)\int_X\int_Y \left[|s_1(x) - s_1'(x)| + |s_2(y) - s_2'(y)|\right] dydx$$

$$= (2\rho+2)\left[\int_X |s_1(x) - s_1'(x)|dx + \int_Y |s_2(y) - s_2'(y)|dy\right]$$

$$= (2\rho+2)||s - s'|| \leq 2(\rho+2)\delta = \epsilon.$$

The continuity of $u_2$ in $s$ follows from the same argument, and we are done. □

It follows from Theorem 1 that the game has a Nash equilibrium using the new strategy spaces $\hat{S}_1$ and $\hat{S}_2$.

Now that the existence of an equilibrium is guaranteed, we must figure out how to compute one. It turns out that the game is not separable, so one cannot apply the algorithm from prior work [18].

**Proposition 5.** *This game is not separable.*

The proof of this result appears in Appendix B.

However, it turns out that we can still solve this game quite easily if we notice that every equilibrium will have $\underline{x_1^*} \leq x_2^* \leq \overline{x_1^*}$. Given this guess of the form of an equilibrium, it is easy to compute an equilibrium by noting that a player must be indifferent between two actions at each threshold. For example, at $x_1^*$ player 1 must be indifferent between betting and checking. His expected payoff of betting is $\overline{(1} - x_2^*)\rho + (x_2^* - \underline{x_1^*})(\rho+1) - \underline{x_1^*}$ and his expected payoff of checking is $\rho(1 - \underline{x_1^*})$. Setting these two expressions equal to each other yields $x_2^* = 2\underline{x_1^*}$. We can create a linear equation similarly for the other two thresholds. Thus we have a system of $n$ linear equations with $n$ unknowns (where $n$ is the total number of thresholds), which can be solved efficiently by matrix inversion.

# 4 Our setting

The main setting of the rest of the paper will be a generalization of the setting of the above example along several dimensions.

**Definition 6.** *A* continuous Bayesian game *is a tuple* $G = (N, X, C, U, F)$ *where*

- $N = \{1, 2, 3, \ldots, n\}$ *is the set of players,*
- $X = (X_1, \ldots, X_n)$ *where each* $X_i$ *is a compact subset of* $\mathbb{R}$ *corresponding to the set of private signals of player* $i$*,*
- $C = (C_1, \ldots, C_n)$ *where each* $C_i$ *is a compact subset of* $\mathbb{R}$ *corresponding to the set of actions of player* $i$*,*
- $U = (u_1, \ldots, u_n)$ *where* $u_i : C \times X \to \mathbb{R}$ *is the measurable utility function of player* $i$*, and*
- $F = (F_1, \ldots, F_n)$ *where* $F_i : X_i \to [0, 1]$ *is the cumulative distribution function (CDF) of the private signal distribution of player* $i$ *(i.e.,* $F_i(x_i) = Pr(X_i \leq x_i)$*).*

The strategy space $S_i$ of player $i$ is the set of all measurable functions from $X_i$ to $C_i$. In this paper we will assume that the sets $C_i$ are finite.

To define mixed strategies in our setting, we will require several mathematical definitions which are given in Appendix A. Let $\Lambda_i$ be the mixed strategy space of player $i$ defined according to Definition 22. Let $\Lambda = \times_i \Lambda_i$. A *(mixed) strategy profile* is $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_n)$, where $\sigma_i \in \Lambda_i$. Recall that $\sigma_i(x_i, c_i)$ denotes the probability that player $i$ will take action $c_i$ given private signal $x_i$.

Then define

$$\hat{u_i}(\boldsymbol{\sigma}, \mathbf{x}) = \sum_{c_1 \in C_1} \cdots \sum_{c_n \in C_n} \left[ u_i(\boldsymbol{c}, \boldsymbol{x}) \prod_{j=1}^{n} \sigma_j(x_j, c_j) \right].$$

Define $u_{i|x_i}(\boldsymbol{\sigma})$, where $\boldsymbol{\sigma} \in \Lambda$, $x_i \in X_i$, as follows:

$$u_{i|x_i}(\boldsymbol{\sigma}) = \int_{X_1} \cdots \int_{X_{i-1}} \int_{X_{i+1}} \cdots \int_{X_n} \hat{u_i}(\boldsymbol{\sigma}, \boldsymbol{x}) f_n(x_n) \ldots f_{i+1}(x_{i+1}) f_{i-1}(x_{i-1}) \ldots f_1(x_1)$$

$$dx_n \ldots dx_{i+1} dx_{i-1} \ldots dx_1,$$

where $f_i(x_i) = \frac{d}{dx_i} F(x_i)$ is the probability density function of $X_i$. This denotes the expected utility of player $i$ given that he has received private signal $x_i$ when strategy profile $\boldsymbol{\sigma}$ is played.

For $i \in N$, $x_i \in X_i$, $\sigma_i \in \Lambda_i$ and $\sigma_{-i} \in \times_{j \neq i} \Lambda_j$, define $u_{i|x_i}(\sigma_i, \sigma_{-i})$ as the expected utility of player $i$ given private signal $x_i$ when he plays $\sigma_i$ and the other players play $\sigma_{-i}$.

According to the definition of Nash equilibrium, player $i$ can play arbitrarily in cases where he has a private signal that has zero measure in the signal distribution $F_i$. Such behavior can result in equilibria that violate our qualitative models (discussed later) even when "equivalent" equilibria exist that satisfy the models. Thus, we define a slightly stronger notion of equilibrium in order to rule out arbitrary behavior in these regions of measure zero. In other words, we require an agent to play rationally even if he gets a private signal that has zero probability. (This strengthening of the equilibrium concept is analogous to *perfect Bayesian equilibrium*, where each agent has to act rationally even in information sets that are reached with zero probability due to the strategies of the players. In our case the reaching with zero probability is due to nature's action, i.e., giving the agents types.)

**Definition 7.** *Strategy profile $\sigma \in \Lambda$ is an* every-private-signal Bayesian (EPSB) equilibrium *of $G$ if for all $i \in N$, for all $x_i \in X_i$, and for all $\tau_i \in \Lambda_i$, we have $u_{i|x_i}(\sigma_i, \sigma_{-i}) \geq u_{i|x_i}(\tau_i, \sigma_{-i})$.*

**Proposition 6.** *Let $G$ be a game of the form given in Definition 6. Then $G$ has an EPSB equilibrium if and only if it has an equilibrium.*

*Proof.* Consider some equilibrium $\sigma$. For $i \in N$, let

$$Z_i = \{x_i \in X_i : u_{i|x_i}(\sigma_i, \sigma_{-i}) \neq \arg\max_{\tau_i \in \Lambda_i} u_{i|x_i}(\tau_i, \sigma_{-i})\}.$$

Since $\sigma$ is an equilibrium, $Z_i$ must have measure zero for all $i$. Now suppose for all $i \in N$ and all $x_i \in Z_i$, player $i$ plays $\sigma_i' \in \arg\max_{\tau_i \in \Lambda_i} u_{i|x_i}(\tau_i, \sigma_{-i})$ and plays $\sigma_i$ otherwise. Call the new profile $\gamma$. Then $\gamma$ and $\sigma$ differ only on a set of measure zero, and thus $\gamma$ is an equilibrium. Since each player is playing a best response given each possible private signal under $\gamma$, $\gamma$ is an EPSB equilibrium.

The other direction is trivial because every EPSB equilibrium is also an equilibrium. □

We now strengthen the notions of best response and dominated strategies analogously. These will be useful when we analyze our algorithms.

**Definition 8.** *Strategy $\sigma_i$ is an EPSB best response for player $i \in N$ to profile $\sigma_{-i}$ if for all $x_i \in X_i$ and for all $\tau_i \in \Lambda_i$, we have $u_{i|x_i}(\sigma_i, \sigma_{-i}) \geq u_{i|x_i}(\tau_i, \sigma_{-i})$.*

**Definition 9.** *Strategy $\sigma_i$ is an EPSB $\epsilon$-best response for player $i \in N$ to profile $\sigma_{-i}$ if for all $x_i \in X_i$ and for all $\tau_i \in \Lambda_i$, we have $u_{i|x_i}(\sigma_i, \sigma_{-i}) \geq u_{i|x_i}(\tau_i, \sigma_{-i}) - \epsilon$.*

**Definition 10.** *Strategy $\sigma_i$ is EPSB-undominated if for all $\tau_i \in \Sigma_i$ there exist $x_i \in X_i$, $\sigma_{-i} \in \Sigma_{-i}$ such that $u_{i|x_i}(\sigma_i, \sigma_{-i}) > u_{i|x_i}(\tau_i, \sigma_{-i})$.*

# 5 Parametric models

In many multiagent settings, it is significantly easier to infer qualitative models of the structure of equilibrium strategies than it is to compute an equilibrium. The introduction gives several examples, including sequences of take-it-or-leave-it offers, certain auctions, and making or breaking partnerships and contracts. In general, we call the values that divide the different strategic regions *thresholds* (e.g., $\underline{x_1^*}, \overline{x_1^*}$, and $x_2^*$ in the example above), and refer to the guess of the structure of an equilibrium defined by these thresholds a *parametric model*. Many additional examples of games that are solved by the procedure used in the above example appear in [2].

**Definition 11.** *A* parametric model *of game $G = (N, X, C, U, F)$ is a tuple $P = (T, Q, \prec)$ where*

- *$T = (T_1, \ldots, T_n)$, where $T_i$ denotes the number of regions for player $i$,*
- *$Q = (Q_1, \ldots, Q_n)$, where $Q_i$ is a sequence $\{q_{ij} : 1 \leq j \leq T_i\}$, where $q_{ij} \in C_i$ denotes the action taken by player $i$ in his $j$'th region of the model (at a boundary the lower region's action is taken), and*

- *$\prec$ is a partial ordering over the set of tuples $(y_{ij}, y_{i'j'})$, where $y_{ij} \prec y_{i'j'}$ if we require that the lower threshold of player $i$'s $j$'th region is less than or equal to the lower threshold of player $i'$'s $j'$'th region.*

We saw in Section 3 that restricting the strategy spaces of a game by forcing all strategies to conform to a specified parametric model can allow us to both guarantee the existence of an equilibrium and to actually compute one when neither of these could be accomplished in the original game by previously known techniques.

# 6 Our main algorithm

In this section we present our algorithm for computing an equilibrium given a parametric model. While parametric models associate a pure action for each interval of signals, this can be problematic when the probability of obtaining individual private signals is nonzero. In this case, our algorithm will actually output mixed strategies.

For now we assume the game is finite, has two players, and a single parametric model is specified. We will extend the algorithm to more general settings along each of these dimensions in Section 7.

## 6.1 Constructing a MILFP

Given a problem instance $G = (N, X, C, U, F)$ and a parametric model $P$, we first construct a mixed integer linear feasibility program (MILFP) that contains both integer and continuous variables. Since $X_i$ is finite for all players, we assume without loss of generality that it is the set of integers from 1 to $\overline{n}$. Let $\{t_i\}$ denote the union of the sets of thresholds for both players under $P$. For each threshold $t_i$, we introduce two real-valued variables, $x_i$ and $y_i$, where $x_i$ corresponds to $F_1(t_i)$ and $y_i$ corresponds to $F_2(t_i)$. For each threshold $t_i$ and each integer $j \in [1, \overline{n}]$, we introduce an indicator (0–1 integer) variable, $z_{i,j}$, such that $z_{i,j} = 1$ implies $j - 1 \leq t_i \leq j$. So, overall we have $2|T| + |T||S|$ variables, where $|T|$ is the number of thresholds in $P$ and $|S|$ is the number of private signals.

**Indifference constraints** As the example in Section 3 demonstrates, we want to obtain indifference between two actions at each threshold. Thus, we have $|T|$ equality constraints where each is of the form $f(x, y) = 0$ where $f$ is linear in the $x_i$ and $y_i$.

**Threshold ordering constraints** In order to guarantee that the solution conforms to the parametric model, we must add inequality constraints corresponding to the partial ordering $\prec$ . If $t_i \prec t_j$, then we add the constraints $x_i \leq x_j$, $y_i \leq y_j$.

**Consistency constraints** Next, we require that for each $i$, $x_i$ and $y_i$ are consistent in the sense that there exists some value for $t_i$ such that $F_1(t_i)$ corresponds to $x_i$ and $F_2(t_i)$ corresponds to $y_i$.

To accomplish this, we include indicator constraints of the following form for each $i, j$:

$$(z_{i,j} = 1) \Rightarrow (F_1(j-1) \leq x_i \leq F_1(j)) \tag{1}$$

$$(z_{i,j} = 1) \Rightarrow (F_2(j-1) \leq y_i \leq F_2(j)), \tag{2}$$

where we define $F_1(-1) = F_2(-1) = 0$. Stated explicitly, we add the following 4 linear inequalities for each $i \in [1, |T|], j \in [0, \overline{n}]$:

$$x_i - F_1(j-1)z_{i,j} \geq 0 \qquad x_i - z_{i,j}(F_1(j) - 1) \leq 1$$

$$y_i - F_2(j-1)z_{i,j} \geq 0 \qquad y_i - z_{i,j}(F_2(j) - 1) \leq 1$$

Finally, we must ensure that for each $i$, $z_{i,j} = 1$ for precisely one $j$ (i.e., $t_i \in [j-1, j]$ for some $j$). We accomplish this by adding the equality constraint $\sum_{j=0}^{\overline{n}} z_{i,j} = 1$ for each $i$.

Thus, there are $O(|T||S|)$ consistency constraints, where $|S|$ is the number of private signals. There are more consistency constraints than constraints of any other type, and thus the MILFP has $O(|T||S|)$ total constraints.

## 6.2 Obtaining mixed strategies from the MILFP solution

Once we obtain the $x_i$ and $y_i$ by solving the MILFP, we must map them into mixed strategies of the game. Suppose player 1 is dealt private signal $z \in [1, \overline{n}]$ and consider the interval $I = [F_1(z-1), F_1(z)]$. Now define the intervals $J_i = [x_{i-1}, x_i]$ where we define $x_{-1} = 0$. Let $O_i$ denote the overlap between sets $I$ and $J_i$. Then player 1 will play the strategy defined by region $i$ with probability $\frac{O_i}{\sum_i O_i}$. The strategy for player 2 is determined similarly, using the $y_i$ and $F_2$.

## 6.3 Algorithm correctness

We are now ready to prove that our algorithm indeed yields an equilibrium.

**Lemma 1.** *Suppose* $0 \leq x_1 \leq \ldots \leq x_n \leq 1$ *and* $0 \leq y_1 \leq \ldots \leq y_n \leq 1$ *and there exists a* $k \in [1, n]$ *s.t.* $x_k \neq y_k$. *Then there exists a* $j \in [1, n]$ *s.t.* $x_j \neq y_j$ *and* $y_{j-1} \leq x_j \leq y_{j+1}$ *(where we define* $x_0 = y_0 = 0$ *and* $x_{n+1} = y_{n+1} = 1$*).*

*Proof.* Let

$$i' = \min_i \text{ s.t. } x_i \neq y_i.$$

- Case 1: $x_{i'} < y_{i'}$
  Suppose $x_{i'} < y_{i'-1}$. Then $x_{i'-1} < y_{i'-1}$ which contradicts the definition of $i'$. So $y_{i'-1} \leq x_{i'} < y_{i'}$ and we are done.

- Case 2: $x_{i'} > y_{i'}$
  Let

$$z(i) = \max_j \text{ s.t. } x_j \leq y_i$$

and let

$$i^* = \min_{i>i'} \text{ s.t. } z(i) \geq i.$$

Clearly $i^*$ exists since $z(n+1) = n+1$.

- Subcase 2.1: $i^* = i' + 1$.
  By assumption we have $x_{i^*-1} > y_{i^*-1}$. Suppose $x_{i^*-1} > y_{i^*}$. By definition of $i^*$ we have $z(i^*) \geq i^*$ and therefore $x_{i^*} \leq y_{i^*}$. Since $x_{i^*-1} \leq x_{i^*}$, we have a contradiction. Thus $y_{i^*-1} < x_{i^*-1} \leq y_{i^*}$ and we are done.
- Subcase 2.2: $i^* > i' + 1$.
  Suppose that $x_{i^*-1} \leq y_{i^*-1}$. By definition of $i^*$ we have $z(i^*-1) < i^*-1$. This implies that $x_{i^*-1} > y_{i^*-1}$ which gives a contradiction. Now suppose that $x_{i^*-1} > y_{i^*}$. We have $z(i^*) \geq i^*$ and therefore $x_{i^*} \leq y_{i^*}$ which gives a contradiction since $x_{i^*-1} \leq x_{i^*}$. So $y_{i^*-1} < x_{i^*-1} \leq y_{i^*}$ and we are done.

□

**Theorem 2.** *Suppose that for all $i \in N$ and for all $\sigma_{-i} \in \Lambda_{-i}$, all pure-strategy EPSB best responses of player $i$ to $\sigma_{-i}$ satisfy the given parametric model. Then our algorithm outputs an equilibrium.*

*Proof.* Suppose our algorithm outputs a non-equilibrium, $s$. Then there exists a player $i$ and an EPSB best response $\sigma_i$ to $s_{-i}$ such that not all of the thresholds of $\sigma_i$ and $s_i$ are equal. Let $\{u_j\}$ denote the thresholds of $\sigma_i$ and $\{v_j\}$ denote the thresholds of $s_i$. By Lemma 1 there exists a $j$ such that $u_j \neq v_j$ and $u_{j-1} \leq v_j \leq u_{j+1}$. Denote by $a_1$ and $a_2$ the two actions that player $i$ is indifferent between at $v_j$. Suppose $a_1$ is played at $v_j$ under $\sigma_i$. Now define the strategy profile $s_i'$ as follows: $s_i'$ is identical to $\sigma_i$ except with private signal $v_j$ action $a_2$ is played. Clearly player $i$ is playing an EPSB best response since $\sigma_i$ is an EPSB best response and he is indifferent between $a_1$ and $a_2$ against $s_{-i}$ at $v_j$. So $s_i'$ is an EPSB best response of player $i$ to $s_{-i}$; however, $s_i'$ violates the parametric model. This contradicts the premise. □

The theorem directly implies the following corollary. In some settings it may be easier to check the premise of the corollary than the premise of the theorem.

**Corollary 1.** *Suppose all EPSB-undominated strategies follow the given parametric model. Then our algorithm outputs an equilibrium.*

# 7 Extensions to more general settings

In this section we describe several important extensions of our approach to more general settings.

## 7.1 Continuous private signal distributions

In this subsection we generalize the approach to continuous private signal distributions. If the CDFs $F_i$ are continuous and piecewise linear, we only need to alter the consistency constraints. Suppose that $\{c_i\}$ denotes the union of the breakpoints of the CDFs of the $F_i$, and suppose $F_i(x) = a_{i,j}x + b_{i,j}$ for $c_{j-1} \leq x \leq c_j$ (where $c_{-1} = 0$).

Recall that for a given threshold $t_k$, we have introduced variables $x_k$ and $y_k$ in the MILFP such that $t_k$ corresponds to $F_1(x_k)$ and $F_2(y_k)$. In the case of continuous CDFs, we actually have $t_k = F_1(x_k) = F_2(y_k)$. To achieve this, we will introduce new variables $t_k$ corresponding to the actual thresholds, and include the following linear constraints for all $k, j$:

$$(z_{k,j} = 1) \rightarrow \left(t_k = \frac{x_k - b_{1,j}}{a_{1,j}}\right)$$

$$(z_{k,j} = 1) \rightarrow \left(t_k = \frac{y_k - b_{2,j}}{a_{2,j}}\right).$$

These constraints replace Constraints 1 and 2 from the discrete case.

Using the technique described in Section 7.3, we can also approximately solve the problem for continuous CDFs that are not piecewise linear by approximating them with piecewise linear functions.

## 7.2 Dependent private signals

Many common real-world situations have dependent private signal distributions. For example, in card games such as Texas hold'em, the cards dealt to each player depend on what was dealt to the other players (if one player is dealt the ace of spades, another player cannot also have it).

We will assume the private signals are given by a joint probability density function (PDF) $h(x, y)$ (rather than independent CDF's $F_1$ and $F_2$ as before). Instead of having $|T|$ variables $x_i$ and $y_i$ corresponding to both players' CDFs at the different thresholds, if the private signals are dependent, we will use $|T|^2$ variables $x_{ij}, y_{ij}$ where $x_{ij}$ corresponds to $Pr(X \leq t_i | Y = t_j)$ and $y_{ij}$ corresponds to $Pr(Y \leq t_j | X = t_i)$. We also need to add $|T|^2|S|^2$ indicator variables (where $|S|$ is the number of private signals) and corresponding consistency constraints using the given joint PDF $h(x, y)$. Thus, the overall size of the CSP is larger by approximately a factor of $|T||S|$.

To construct the new consistency constraints, we define the following functions $q_1$ and $q_2$:

$$q_1(x, y) \equiv Pr(X \leq x | Y = y) = \frac{\int_{-\infty}^{x} h(v, y)dv}{\int_{-\infty}^{\infty} h(v, y)dv}.$$

$$q_2(x, y) \equiv Pr(Y \leq y | X = x) = \frac{\int_{-\infty}^{y} h(x, v)dv}{\int_{-\infty}^{\infty} h(x, v)dv}.$$

We will replace Constraints 1 and 2 from the independent case with the following constraints:

$$(z_{i,j,k,m} = 1) \Rightarrow (q_1(k-1, m) \leq x_{i,j} \leq q_1(k, m))$$

$$(z_{i,j,k,m} = 1) \Rightarrow (q_2(k, m-1) \leq y_{i,j} \leq q_2(k, m)).$$

## 7.3 Many players

In games with more than two players, the indifference constraints are no longer linear functions of the variables (while all other constraints remain linear). With $n$ players the indifference constraints are degree $n-1$ polynomials. Therefore, there is a need to represent products of continuous variables, $x_i x_j$, using linear constraints only since we wish to model the problem as a MILFP.

### 7.3.1 Approximating products of continuous variables using linear constraints

In this subsection we describe how a modeling technique [5] can be applied to approximate the nonlinear functions by piecewise linear functions. First we define two new variables

$$\beta_1 = \frac{1}{2}(x_i + x_j)$$

$$\beta_2 = \frac{1}{2}(x_i - x_j),$$

noting that $\beta_1^2 - \beta_2^2 = x_i x_j$. To approximate $w_1 = \beta_1^2$, we select $k$ breakpoints from the interval [0,1]—in our experiments we will use $q_i = \frac{i-1}{k-1}$, where $k(\epsilon)$ is a function of an input parameter $\epsilon$. Next, we add the constraint

$$\sum_{i=1}^{k} \lambda_{1i} q_i^2 = w_1,$$

where the $\lambda_{1i}$ are continuous variables. Next we add the constraint

$$\sum_{i=1}^{k} \lambda_{1i} q_i = \beta_1.$$

We also add the constraint

$$\sum_{i=1}^{k} \lambda_{1i} = 1,$$

where we also require that at most two adjacent $\lambda'_{1i}s$ are greater than zero (we accomplish this in the standard way of adding a binary indicator variable per segment and the appropriate constraints, called SOS2 constraints). Then if we define the variable $u_{ij}$ to represent the product $x_i x_j$, we just need to add the constraint

$$u_{ij} = w_1 - w_2,$$

where $w_2$ and its additional constraints are defined analogously to $w_1$.

Finally, we replace each indifference equation $f(x) = 0$ with the inequalities $f(x) \leq \frac{\epsilon}{2}$ and $f(x) \geq -\frac{\epsilon}{2}$ where $\epsilon$ is an approximation parameter given as input to the algorithm.

### 7.3.2 Tying the accuracy of the approximation to the accuracy of the equilibrium

Suppose we select $k+1$ breakpoints per piecewise linear curve, with

$$k \geq \sqrt{\frac{(\overline{T}+2)^{(n-1)}M(n-1)}{\epsilon}}, \tag{3}$$

where $\overline{T}$ is the maximal number of thresholds of one of the parametric models for a player, $M$ is the difference between the maximum and minimum possible payoff of the game, $n$ is the number of players, and $\epsilon$ is an approximation parameter given as input to the algorithm.

**Lemma 2.** *Suppose we obtain piecewise linear approximation $\hat{f}(x)$ of $f(x) = x^2$ over $[a,b]$ using $k+1$ breakpoints $q_0, \ldots, q_k$ with $q_i = a + (b-a)\frac{i}{k}$. If $k \geq \frac{b-a}{2\sqrt{\epsilon}}$, then $|\hat{f}(x) - f(x)| \leq \epsilon$ for all $x \in [a,b]$.*

*Proof.* Let $x \in [a,b]$ be arbitrary and suppose $q_i \leq x \leq q_{i+1}$. Then we have

$$\hat{f}(x) = \hat{f}(q_i) + \frac{x - q_i}{q_{i+1} - q_i}(\hat{f}(q_{i+1}) - \hat{f}(q_i)).$$

It is clear that $\hat{f}(x) \geq x^2$, so consider the difference $g(x) = \hat{f}(x) - x^2$. Since $g(q_i) = g(q_{i+1}) = 0$, the derivative of $g$ must be zero at its maximum over the range $[q_i, q_{i+1}]$. Note that

$$g'(x) = \frac{\hat{f}(q_{i+1}) - \hat{f}(q_i)}{q_{i+1} - q_i} - 2x.$$

Setting this equal to 0 yields

$$x^* = \frac{\hat{f}(q_{i+1}) - \hat{f}(q_i)}{2(q_{i+1} - q_i)} = \frac{q_{i+1}^2 - q_i^2}{2(q_{i+1} - q_i)} = \frac{q_{i+1} + q_i}{2}.$$

Thus the maximal value of $g$ is

$$\begin{aligned} g(x^*) &= \hat{f}(q_i) + \frac{x^* - q_i}{q_{i+1} - q_i}(\hat{f}(q_{i+1}) - \hat{f}(q_i)) - (x^*)^2 \\ &= \left(\frac{b-a}{2k}\right)^2 \leq \epsilon. \end{aligned}$$

□

**Theorem 3.** *Suppose that for all $i \in N$ and for all $\sigma_{-i} \in \Lambda_{-i}$, all pure-strategy EPSB $\epsilon$-best responses of player $i$ to $\sigma_{-i}$ satisfy the given parametric model. Furthermore suppose that the number of breakpoints satisfies Equation 3. Then our algorithm outputs an $\epsilon$-equilibrium.*

*Proof.* By Lemma 2 we have

$$|\hat{f}(x) - f(x)| \leq \frac{\epsilon}{4(T+2)^{(n-1)}M(n-1)} \text{ for all } x \in [0,1].$$

Suppose the indifference inequalities have the form $\hat{h}(x) \leq \hat{g}(x) + \frac{\epsilon}{2}$ and $\hat{g}(x) \leq \hat{h}(x) + \frac{\epsilon}{2}$. Note that each term of $g(x)$ and $h(x)$ is a polynomial of degree $n-1$ with coefficients of absolute value at most $M$. Also, note that there are at most $(T+2)^{(n-1)}$ such terms.

So $|\hat{g}(x) - g(x)| \leq \frac{\epsilon}{4}$ and $|\hat{h}(x) - h(x)| \leq \frac{\epsilon}{4}$. So by the triangle inequality,

$$|g(x) - h(x)| \leq |g(x) - \hat{g}(x)| + |\hat{g}(x) - \hat{h}(x)| + |h(x) - \hat{h}(x)| = \epsilon.$$

Suppose our algorithm outputs a non-equilibrium, $s$. Then there exists a player $i$ and an EPSB best response $\sigma_i$ to $s_{-i}$ such that not all of the thresholds of $\sigma_i$ and $s_i$ are equal. Let $\{u_j\}$ denote the thresholds of $\sigma_i$ and $\{v_j\}$ denote the thresholds of $s_i$. By Lemma 1 there exists a $j$ such that $u_j \neq v_j$ and $u_{j-1} \leq v_j \leq u_{j+1}$. Denote by $a_1$ and $a_2$ the two actions that player $i$ is $\epsilon$-indifferent between at $v_j$. Suppose $a_1$ is played at $v_j$ under $\sigma_i$. Now define the strategy profile $s_i'$ as follows: $s_i'$ is identical to $\sigma_i$ except with private signal $v_j$ action $a_2$ is played. Then player $i$ is playing an EPSB $\epsilon$-best response since $\sigma_i$ is an EPSB best response and he is $\epsilon$-indifferent between $a_1$ and $a_2$ against $s_{-i}$ at $v_j$. So $s_i'$ is a pure strategy EPSB $\epsilon$-best response of player $i$ to $s_{-i}$; however, $s_i'$ violates the parametric model. This contradicts the premise. □

For particular games, the number of breakpoints needed to obtain a desired $\epsilon$ can actually be far smaller. For example, if each indifference equation consists of the sum of at most $T^*$ expressions, for $T^* < (\overline{T}+2)^{(n-1)}$, then we can replace $(\overline{T}+2)^{(n-1)}$ with $T^*$ to create a tighter upper bound. Additional constant-factor improvements to Equation 3 will be described in detail in Section 8.3.

Additionally, even though the number of breakpoints in Equation 3 is exponential in the number of players, we can actually model the problem as a MILFP using a polynomial number (in $n$) of constraints and variables (i.e., using a number of constraints and variables that is logarithmic in the number of breakpoints). This is accomplished by a recently published way of modeling piecewise linear functions in a MIP [19]. (It uses a binary rather than unary encoding to refer to the pieces via indicator variables.)

### 7.3.3 New MIP algorithms for computing equilibria in normal and extensive-form games

It is worth noting that the modeling approach of Section 7.3.1 can be used to develop new algorithms for computing an $\epsilon$-equilibrium in general-sum games with two or more players in both normal and extensive form. In particular, the *MIP Nash* algorithm for computing an equilibrium in two-player general-sum normal-form games [15] can be directly extended to a MIP formulation of multiplayer normal-form games which contains some nonlinear constraints (corresponding to the expected utility constraints). If we apply our approach using sufficiently many breakpoints, we can obtain an $\epsilon$-equilibrium for arbitrary $\epsilon$ by approximating the nonlinear constraints by piecewise linear constraints. Additionally, we can represent the equilibrium-computation problem in multiplayer extensive-form games as a MIP if we write out the expected utility constraints separately on

a per-information-set basis. This leads to a new algorithm for computing equilibria in multiplayer extensive-form games, an important class of games for which no algorithms for computing a Nash equilibrium with solution guarantees were known.

## 7.4 Multiple parametric models

Quite often it is prohibitively difficult to come up with one parametric model, $P$, that is correct, but one can construct several parametric models, $P_i$, and know that at least one of them is correct. This is the case for our experiments on Texas hold'em in Section 8.2. This scenario could arise for several reasons; for example, often we can immediately rule out many parametric models because all strategies that satisfy them are dominated. We now generalize our approach to this situation.

We define the notion of model refinement in a natural way:

**Definition 12.** *$P = (T, Q, \prec)$ is a* refinement *of $P' = (T', Q', \prec')$ if for each $i \in N$, $Q'_i$ is a (not necessarily contiguous) subsequence of $Q_i$.*

**Definition 13.** *$P$ is a* US-refinement *of $P'$ if $Q'_i$ corresponds to a unique subsequence of $Q_i$ for each $i$.*

For example, if $N = \{1\}$ and $Q'_1 = \{1, 2\}$ while $Q_1 = \{1,2,3,2\}$, then $P$ is a refinement of $P'$, but is not a US-refinement.

We now generalize our algorithm to the setting where the $P_i$ have a common US-refinement $P'$. We first define an indicator variable $\zeta_i$ corresponding to each model. Next we replace each indifference constraint $f(x) = 0$ corresponding to model $P_i$ by the following two inequalities, where $K$ is a sufficiently large constant: $f(x) - K\zeta_i \geq -K$ and $f(x) + K\zeta_i \leq K$.

Next we add certain inequalities corresponding to the models $P_i$ that differ from $P'$. For simplicity, we will demonstrate these by example. Suppose that, under $P'$, player 1 plays action $a_1$ in his first region, $a_2$ in his second region, and $a_3$ in his third region. Suppose that in $P_1$ he plays $a_1$ in his first region and $a_3$ in his second region (recall that $P'$ is a refinement of $P_1$). Then we must add two constraints that ensure that at the first threshold of $P_1$, both $a_1$ and $a_3$ are (weakly) preferred to $a_2$. In general, whenever actions of $P'$ are omitted by a $P_i$, we must add constraints to the neighboring actions at their intersection ensuring that they are preferred to the omitted actions.

We also replace each order constraint $x_j - x_{j'} \leq 0$ corresponding to model $P_i$ by $x_j - x_{j'} + K\zeta_i \leq K$. Finally, we add the equality $\sum_i \zeta_i = 1$ to ensure that only the constraints corresponding to one of the candidate parametric models are used in the solution.

The following theorem states that our approach is correct even in this setting where there are multiple parametric models, assuming they have a common US-refinement.

**Theorem 4.** *Let there be two players. Let $\{P_i\}$ be a set of parametric models with a common US-refinement. Suppose that for all $i \in N$ and for all $\sigma_{-i} \in \Lambda_{-i}$, all pure-strategy EPSB best responses of player $i$ to $\sigma_{-i}$ satisfy at least one of the $P_i$ (not necessarily the same $P_i$). Then our algorithm outputs an equilibrium.*

*Proof.* Suppose $P'$ is the common US-refinement of the $P_i$. Suppose our algorithm outputs a non-equilibrium, $s$. Then there exists a player $i$ and an EPSB best response $\sigma_i$ to $s_{-i}$ such that not all

of the thresholds of $\sigma_i$ and $s_i$ are equal. Since $P'$ is a refinement of each $P_i$, $\sigma_i$ must satisfy $P'$. Let $\{u_j\}$ denote the thresholds of $\sigma_i$ and $\{v_j\}$ denote the thresholds of $s_i$.

- Case 1: There exists a $v_j$ that differs from all of the $u_j$.
  Then $v_j$ must lie in the interior of some region of $\sigma_i$—suppose action $a_1$ is played in this region. Suppose $v_j$ separates actions $a_2$ and $a_3$. If $a_1 \neq a_2, a_3$, then our algorithm requires that $a_2$ and $a_3$ are both (weakly) preferred to $a_1$ at $v_j$. Suppose that player $i$ plays $\sigma_i$ except with private signal $v_j$ plays $a_2$. Then this must also be a pure-strategy EPSB best response to $s_{-i}$. However, it contradicts the premise of the theorem because no $P_i$ can be a strict refinement of $P'$.

- Case 2: For all $v_j$, there exists a $u_k$ such that $u_k = v_j$.

  - Subcase 2.1: There exists a $v_j$ such that $u_k = v_j$ and the action-regions that $v_j$ separates are not both equal to the action regions that $u_k$ separates.
    Suppose $v_j$ separates $a_1, a_2$ while $u_k$ separates $b_1, b_2$, and that $a_1 \neq b_1, b_2$. Then player $i$ can play $\sigma_i$ except with private signal $u_k$ play $a_1$. This must also be a pure-strategy EPSB best response to $s_{-i}$, since our algorithm ensures that $a_1$ is (weakly) preferred to all other actions at $v_j$. However, it contradicts the premise of the theorem because no $P_i$ can be a strict refinement of $P'$.

  - Subcase 2.2: For all $v_j$, there exists a $u_k$ such that $u_k = v_j$ and $v_j, u_k$ separate the same action-regions.
    Since not all of the thresholds of $\sigma_i$ and $s_i$ are equal, there must exist an additional $u_k$ that does not equal any of the $v_j$. Suppose that $u_k$ lies between $v_j$ and $v'_j$, and suppose that action $a$ is played in this region under $s_i$. By the premise of this subcase, $a$ cannot be the action of either of the regions separated at $u_k$. Also by the premise of this subcase, $a$ must be played both below and above $u_k$. However, this would mean that $\sigma_i$ has two regions where $a$ is played corresponding to the single region of $s_i$, which contradicts the fact that $P'$ is a US-refinement of the $P_i$.

□

We can also obtain a result with an $\epsilon$ guarantee similar to Theorem 3 for the case of more than two players.

It is worth noting that the number of variables and constraints in the new MILFP formulation is still $O(|S||T|)$ (assuming a constant number of parametric models). Alternatively, we could have solved several MILFP's—one for each parametric model. While each MILFP would be smaller than the one we are solving, each would still have $O(|S||T|)$ variables and $O(|S||T|)$ constraints, and thus have the same size asymptotically as our formulation. This alternative formulation is also potentially effective, assuming we have access to several processors to run the threads in parallel.

| $n$ | 50 | 100 | 150 | 200 | 250 |
|---|---|---|---|---|---|
| $v(G_n)$ | $-0.0576$ | $-0.0566$ | $-0.0563$ | $-0.0561$ | $-0.0560$ |
| $\pi(\sigma_n)$ | $-0.0624$ | $-0.0612$ | $-0.0579$ | $-0.0583$ | $-0.0560$ |

Figure 1: Worst-case payoff of playing the projection of the equilibrium of $G_\infty$ ($\pi(\sigma_n)$) versus the value of $G_n$ ($v(G_n)$).

# 8 Experiments

We now present results from several experiments that investigate the practical applicability of our algorithm and extensions, as well as the overall procedure of solving large finite games by approximating them by continuous games.

## 8.1 Approximating large finite games by continuous games

In Section 3 we saw an example of a game with infinite strategy spaces that could be solved by an extremely simple procedure (once we guessed a correct parametric model). If instead the set of private signals were the finite set $\{1, \ldots, n\}$, then it is clear that as $n$ gets large, the running time of computing an exact equilibrium of the game will get arbitrarily large; on the other hand, solving the infinite approximation as $n$ goes to infinity will still take essentially no time at all, and we would expect the solution to the infinite game to be very close to the solution of the finite game. In this section we will consider a similar game and show that very fine-grained abstractions would be needed to match the solution quality of our approach.

Kuhn poker is a simplified version of poker that was one of the first games studied by game theorists [13]. It works as follows. There are two players and a deck containing three cards: 1, 2, and 3. Each player is dealt one card at random, and both players ante \$1. Player 1 acts first and can either check or raise by \$1. If player 1 raises, player 2 can either call—in which case whoever has the higher card wins the \$4 pot—or fold—in which case player 1 wins the entire \$3 pot. If player 1 checks, player 2 can check—in which case whoever has the higher card wins the \$2 pot—or bet. If player 1 checks and player 2 bets, player 1 can call—in which case whoever has the higher card wins the \$4 pot—or fold, in which case player 2 wins the \$3 pot.

Generalized Kuhn poker, $G_n$, has the same rules as Kuhn poker except that the deck contains $n$ cards instead of 3. Define $G_\infty$ to be the same as $G_n$ except the players are both dealt a real number drawn uniformly at random from the unit interval $[0, 1]$. Informally, $G_\infty$ is like the limit as $n$ approaches infinity of $G_n$. It turns out that $G_\infty$ has a relatively simple pure strategy Nash equilibrium that is derived in [2]. It can be computed by solving a system of six linearly independent indifference equations with six unknowns.

Once the infinite game, $G_\infty$, has been solved, its solution can be projected down to a corresponding strategy profile in $G_n$: call this profile $\sigma_n$. We ran experiments for several settings of $n$. We compared the performance of $\sigma_n$ against its nemesis to the value of the game to player 1 (i.e., how an optimal strategy performs): the results are summarized in Figure 1. The payoffs agree to four decimal points when $n = 250$.

Next, we considered different abstractions of $G_{250}$ obtained by grouping consecutive private signals together. For each abstraction, we computed an equilibrium in the corresponding abstracted game, then determined the payoff of player 1's component of that equilibrium against its nemesis in the full game $G_{250}$. As Figure 2 shows, 125 buckets are needed to obtain agreement with the

| # buckets | 2 | 5 | 10 | 25 | 50 | 125 |
|---|---|---|---|---|---|---|
| payoff | $-0.2305$ | $-0.0667$ | $-0.0593$ | $-0.0569$ | $-0.0562$ | $-0.0560$ |

Figure 2: Experiments for $G_{250}$.

value of the game to four decimal places—something that $\sigma_{250}$ accomplishes as we showed above. As $n$ gets even larger, we would expect to require even more buckets in our abstraction to obtain a strategy with exploitability as low as that of $\sigma_n$. Thus we can potentially obtain a given level of exploitability with a much lower runtime with our projection approach, since the computation required by abstraction-based approaches increases dramatically as $n$ increases, while solving $G_\infty$ and projecting its solution down to $G_n$ requires very little computation.

To put these results in perspective, the game tree for two-player limit Texas hold'em has approximately $9.17 \times 10^{17}$ states, while recent solution techniques can compute approximate equilibria for abstractions with up to $10^{10}$ game states (e.g., [11]). Thus the ratio of the number of states in the largest solvable abstraction to the number of states in the full game is approximately $10^{-8}$. On the other hand, we saw in $G_{250}$ that we require at least half of the number of states in the full game in our abstraction to compete with the solution generated by the infinite game (assuming we are restricting ourselves to uniform abstractions). Thus, it is conceivable that one can come up with an infinite approximation of Texas hold'em (or one of its subgames) that results in less exploitable strategies than the current strategies obtained by abstraction-based approaches.

In the next section we conduct an investigation of the feasibility of applying the algorithm and extensions developed in this paper to large real-world games, using Texas hold'em as a benchmark.

## 8.2 Two-player Limit Texas hold'em

We ran our algorithm on a game similar to the river endgame of a hand of limit Texas hold'em. In this game, there is an initial pot of $\rho$, and both players are dealt a private signal from [0,1] according to piecewise linear CDFs $F_1$ and $F_2$. Player 1 acts first and can either check or bet \$1. If player 1 checks, then player 2 can check or bet; if he checks the game is over, and if he bets then player 1 can call or fold. If player 1 bets, then player 2 can fold, call, or raise (by 1). If player 1 bets and player 2 raises, then player 1 can either call or fold. Thus, our game is similar to Game 10 in [2], except that we do not assume uniform private signal distributions.

To obtain a wide range of interesting prior distributions, we decided to use the actual prior distributions generated by a high caliber limit Texas hold'em player. Once the river card is dealt in Texas hold'em, there is no more information to be revealed and the game becomes a 1-street game like the game described above (except that in limit Texas hold'em the private signals are dependent[3], and up to three raises are allowed in each betting round). If we assume that the full

[3]We do not expect the dependence to have a large effect in practice for this game due to the large number of

strategies of both players are known in advance, then when the river card is dealt we can create a distribution over the possible 5-card hand rankings each player could have, given the betting history so far (e.g., the probability he has a royal flush, a full house with 9's over 7's, etc.).

In particular, we obtained the strategies from GS4—a bot that performed competitively in the 2008 AAAI computer poker competition. To test our algorithm, we created a new bot GS4-MIP that plays identically to GS4 on the first three streets, and on the river plays according to our algorithm. Specifically, we assume that both players' hand rankings on the river are distributed assuming they had been following the strategy of GS4 until that point; these determine the private signal distributions.

Given this game model, we developed three different parametric models that we expected equilibria to follow (depending on the private signal distributions at the given hand). This is noteworthy since [2] only considers a single parametric model for their game, and our experiments revealed that if we did not include all three models, our MILFP would sometimes have no solution, demonstrating that all three models are necessary. The models are given in Appendix C. It is easy to see that the first model is a US-refinement of the other two. To solve the MILFP, we used CPLEX's MIP solver on a single machine.

Once we solved this simplified game, we used a very naive mapping to transform it to a strategy in the full 3-raise game[4]. Since this mapping was so simple, we suspect that most of the success of the strategy was due to the solution computed by our algorithm.

We ran GS4-MIP against the top five entrants of the 2008 AAAI computer poker competition, which includes GS4. For each pairing, we used 20,000 duplicate hands to reduce the variance. GS4-MIP performed better against 4 of the 5 competitors than GS4 did. In the match between GS4-MIP and GS4, GS4-MIP won at a rate of 0.018 small bets per hand. This is quite significant since most of the top bots in the competition were separated by just 0.02–0.03 small bets per hand overall, and the only difference between GS4 and GS4-MIP is on the river street, which is reached only on some hands. Additionally, GS4-MIP averaged only 0.49 seconds of computation time per hand on hands that went to the river (and 0.25 seconds of computation per hand overall) even though it was solving a reasonably large MIP at runtime[5] (1,000–2,000 rows and several hundred columns). The actual competition allows an average of 7 seconds per hand, so our algorithm was well within the time limit. Perhaps it is the sparsity of the constraints that enabled CPLEX to solve the problems so quickly, as the majority of the constraints are indicator constraints which only have a few non-zero entries.

It is worth noting that our algorithm is perhaps not the most effective algorithm for solving this particular problem; in the discrete case of actual Texas hold'em, the river subgame can be formulated as a linear program (which can probably be solved faster than our MILFP). On the

possible private signals. In addition, we have developed an efficient extension of our MILFP to deal with the case of dependent private signals (see Section 7.2), which can be used if we expect dependence to have a significant effect.

[4]For example, we assumed player 1 will put in a second raise with hands in the top half of player 2's raise range. We omit a full discussion of our transformation to the 3-raise game, since it is fairly tangential.

[5]The reason we need to solve the MIP at runtime is that we have to solve a different MIP for each betting sequence up until the river and each set of community cards (in the full game, not in the abstract game). Since there is such a large number of such subgames, it is much easier to just solve them quickly at runtime than to solve them all in advance.

other hand, continuous games, two player general-sum games, and multiplayer games cannot be modeled as linear programs while they can be solved with our approach. Furthermore, the results in the previous subsection show that large finite two-player zero-sum games can sometimes be solved more effectively (both according to runtimes and quality of solutions) by approximating them by a continuous game that is easier to solve, than by abstracting the game and solving a smaller finite game.

## 8.3 Multiplayer experiments

To test the extensions of our algorithm to continuous distributions and multiple players, we ran our algorithm on the following simplified three-player poker game. The game has one betting round, and all players are given a private signal in $[0, 1]$. Player 2 initially has \$1 invested in the pot, while player 3 has \$2 invested in the pot (i.e., the small and big blinds). Player 1 is first to act and he can either fold or raise to \$4. If player 1 folds, then player 2 can fold or raise to \$4. If a player is facing a raise in front of him, then he can either call or fold. Thus, this game is an extension of the game in Section 3 to multiple players.

It is easy to see that all EPSB-undominated strategies will have the following form. If a player is first to enter the pot, he will raise with his better hands and fold with his worse hands (and never bluff). If a player is facing a raise ahead of him, he will call with his better hands and fold with his worse hands. Thus given each betting history, the parametric model for each player will just have a single threshold. The full parametric model is shown in Figure 6 in Appendix D.

We ran our algorithm on this game using a variety of continuous piecewise-linear cumulative distribution functions, and obtained rapid convergence to an $\epsilon$-equilibrium for tiny $\epsilon$ for all games on which we experimented. In the remainder of this section, we will describe our results with uniform CDF's (i.e., each player is given his private signal uniformly at random from $[0, 1]$) in detail.

Figure 8 in Appendix D shows our experimentally-obtained values of $\epsilon$, as well as the worst-case theoretical values of $\epsilon$ according to Equation 3. As noted in Section 7.3, the worst-case bound for $k$ as a function of $\epsilon$ is very loose, and we expect to require a much smaller value of $k$ to obtain a given $\epsilon$ in practice. Our results confirm this conjecture on this game. Figure 8(a) shows our experimental values of $\epsilon$ as a function of the number of breakpoints. We obtained an $\epsilon$ of 0.01 using just 5 breakpoints, and observed a rapid decrease of $\epsilon$ to about $10^{-5}$ as we increased the number of breakpoints to 50. Figure 8(b) shows the $\epsilon$ guaranteed according to Equation 3. In sharp contrast, an $\epsilon$ of almost 25 is guaranteed using 5 breakpoints, which is meaningless since the difference between the best and worst-case payoffs of the game is only 12. Even using 50 breakpoints only guarantees an $\epsilon$ of 0.24, which is also essentially meaningless for practical purposes. So our results confirm that we are in fact able to obtain good performance results in practice despite the fact that our worst-case theoretical bound is not very meaningful for such small numbers of breakpoints.

### 8.3.1 Conditional parametric model representation

In some cases, it can be beneficial to use an alternate, but equivalent, representation of parametric models. For example, in this game, rather than use a model for player 3 with three different thresh-

olds, we could instead use three parametric models for player 3—where each one corresponds to a different nonterminal betting sequence of the other two players (e.g., raise/call, raise/fold, or fold/raise)—see Figure 7. Then each of these parametric models would only have a single threshold, thus simplifying the representation of each model (though there are more of them). We call such a representation a *conditional parametric model*, due to the fact that a player's model is conditional on the action sequences of the other players.

It is easy to see that conditional parametric models are equivalent to our standard parametric models both in terms of representation power and size, and that they will create the exact same MILFP. However, they often lead to a simpler visual representation, making them more useful in certain situations. In addition, $\overline{T}$ in Equation 3 (recall this refers to the maximum number of thresholds in a parametric model of any player) can now be replaced by $\hat{T}$ which denotes the maximum number of thresholds in a conditional parametric model of any player. In our example, $\overline{T}$ is three while $\hat{T}$ is only one. This actually gives an exponential improvement with respect to the number of players in the worst-case number of breakpoints needed according to Equation 3 in cases where $\hat{T} < \overline{T}$.

### 8.3.2 Computing best responses

To determine the $\epsilon$'s in the experiments, we need to be able to compute the best response for each player to a given strategy profile of the other players. This is relatively easy if we are sure in advance that for each strategy profile of the other players, there exists a best response that conforms to the given parametric model (e.g., as in the premise of Theorem 2). However, often we are not sure in advance that this is the case, and might only be able to come up with a set of parametric models such that a best response satisfies at least one of them.

In the game considered in this section, it is not the case that every best response satisfies the given parametric model. For example, suppose that if player 1 raises, then players 2 and 3 will call with every hand. Then player 1 will only want to raise some of his hands, and fold his bad hands. Thus his threshold will be below the calling thresholds of the other players, which differs from the equilibrium parametric model given above.

We now present an algorithm for computing a best response in the setting where we are able to construct a set of parametric models for which we can prove that for each player and a given strategy profile of the other players, there exists a best response that satisfies at least one of the models. It is easy to see that our game satisfies this property. First, notice that every EPSB-undominated strategy for each player must satisfy the given threshold structure. However, it is not clear how the thresholds for the different players will relate to each other. But note that given strategies of the other players, there are at most 4 possible parametric models consistent with the threshold structure (i.e., the relevant threshold of the player in question must lie somewhere with respect to the other thresholds).

Our algorithm is the following. For each player, we fix the given strategies $\sigma_{-i}$ of the other players and iterate over all the possible parametric models. Then we compute the best response for the given player $i$ using each fixed model. This can be accomplished by treating the values $x_i = F_i(t_i)$ of player $i$'s CDF evaluated at the thresholds as variable, and writing the formula for the expected profit of player $i$ given $\sigma_{-i}$ in terms of the $x_i$'s. This yields a polynomial function

of degree at most $n$ in terms of the $x_i$'s, where $n$ is the number of players. This constrained optimization problem can be solved efficiently by standard techniques (e.g., using Matlab which presumably uses Newton's method), to determine the expected profit of the best response satisfying the given parametric model. We do this for each model, and take the highest value—call it $\pi^*$. The difference between $\pi^*$ and the expected payoff of player $i$ under $\sigma$ yields $\epsilon_i$—the difference between the payoff of his best response and his actual payoff. We do this for each player, and set $\epsilon$ equal to the maximum of the $\epsilon_i$'s.

This algorithm can be used as an *ex-post* checking procedure even if the premises of Theorem 2 (i.e., every EPSB-best response satisfies the given parametric model) or of Theorem 4 are not satisfied. As long as we can construct a set $S$ of parametric models such that there (provably) exists a best response of each player $i$ to the strategy profile $\sigma_{-i}$ of the other players output by our algorithm that is consistent with a model in $S$, then we have computed an $\epsilon$-equilibrium of the game, where $\epsilon$ is determined by the above procedure. Thus, the results of this section show that our algorithm can still be successful even in cases for which the premises of our theorems are not met. This is important, especially in light of the relatively strong premises of the theorems. Future work could look into relaxing the premise of the theorems, and proving the correctness of our algorithm in a wider range of settings.

### 8.3.3 Algorithm performance

Figure 9 shows the running times of our experiments, as a function of the number of breakpoints used. As shown in the figure, runtimes increased steadily from 0.3 seconds with 5 breakpoints to 8.9 seconds with 50 breakpoints.

Despite a clear positive correlation, the runtimes do not increase monotonically with the number of breakpoints. This is due to the fact that CPLEX is solving a fundamentally different MILFP for each number of breakpoints, and the runtimes of CPLEX's MIP solver are notoriously unpredictable (even on inputs that are seemingly quite similar). We saw a similar deviation from monotonicity of $\epsilon$ as a function of the number of breakpoints in Figure 8(a). Therefore, for large problems one may want to try several different numbers of breakpoints, since even consecutive values can lead to drastically different runtimes and values of $\epsilon$. The optimal number to use will clearly depend on the desired $\epsilon$ and running time limitations. However, one implication of our results is that often far fewer breakpoints are needed to obtain a given $\epsilon$ than one might expect based on our theoretical bound in Equation 3. So a reasonable algorithm to use in practice might be to start by running our algorithm with some small number of breakpoints (such as 5), then increment the number of breakpoints by 1 and repeat until a desired $\epsilon$ or time limit is reached. This procedure could be easily parallelized by running our algorithm with different numbers of breakpoints on different cores, since the computations do not depend on each other.

# 9 Conclusions and future research

We presented a new approach for solving large (even infinite) multiplayer games of imperfect information. The key idea behind our approach is that we include additional inputs in the form of

qualitative models of equilibrium strategies (how the signal space should be qualitatively partitioned into action regions). In addition, we showed that our approach can lead to strong strategies in large finite games that we approximate with infinite games. We proved that our main algorithm is correct even if given a set of qualitative models (with a common US-refinement) of which only some are accurate.

In two player settings, our algorithm finds an exact equilibrium. The solution technique uses a mixed integer linear feasibility program. With more than two players, the models include nonlinear elements, which we approximate with piecewise linear functions. We showed how the accuracy of $\epsilon$-equilibrium depends on the number of those pieces—both with a worst-case theorem and experiments that show that significantly fewer pieces are needed in practice.

For settings where our algorithm outputs a solution but we do not know that even one of the qualitative models is correct, we developed an *ex post* procedure for checking whether the solution is an equilibrium or an $\epsilon$-equilibrium. The *ex post* check works under a significantly weaker assumption than our theorems, namely that we use qualitative models for which we can prove that for each player and the given strategy profile of the other players, there exists a best response that satisfies at least one of the models.

Experiments suggest that approximating a finite game with an infinite one can outperform abstraction-based approaches on some games. We constructed a game in which only a tiny amount of abstraction can be performed while obtaining strategies that are no more exploitable than the equilibrium strategies of our infinite approximation. Thus our approach presents a viable alternative to abstraction-based approaches. This is particularly promising in light of the recently uncovered abstraction pathologies.

We also showed how to extend our algorithm to the cases of more than two players, continuous private signal distributions, and dependent private signal distributions. In most of these cases, we presented the first algorithm that provably solves the class of games. Our experiments show that the algorithm runs efficiently in practice in both two-player and multi-player settings. It leads to a significant performance improvement in two-player limit Texas hold'em poker—the most studied imperfect-information game in computer science—by solving endgames.

While in this paper we inferred the infinite approximations of finite games and the parametric models manually, future research could attempt to develop methods for generating them systematically and automatically. It is also possible that in future research one could prove that our approach works under less restrictive premises than are currently used in the main theorems.

# References


[1] C. D. Aliprantis and K. C. Border. *Infinite Dimensional Analysis: A Hitchhiker's Guide*. Springer-Verlag, 2006.

[2] J. Ankenman and B. Chen. *The Mathematics of Poker*. ConJelCo LLC, 2006.

[3] C. Archibald and Y. Shoham. Modeling billiards games. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, Budapest, Hungary, 2009.

[4] D. Billings, N. Burch, A. Davidson, R. Holte, J. Schaeffer, T. Schauenberg, and D. Szafron. Approximating game-theoretic optimal strategies for full-scale poker. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, 2003.

[5] J. Bisschop. *AIMMS—Optimization Modeling*. Paragon Decision Technology, 2006.

[6] L. Blumrosen, N. Nisan, and I. Segal. Auctions with severely bounded communication. *Journal of Artificial Intelligence Research*, 28:233–266, 2007.

[7] X. Chen and X. Deng. Settling the complexity of 2-player Nash equilibrium. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, 2006.

[8] K. Etessami and M. Yannakakis. On the complexity of Nash equilibria and other fixed points (extended abstract). In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, pages 113–123, 2007.

[9] D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, 1991.

[10] S. Ganzfried and T. Sandholm. Computing equilibria in multiplayer stochastic games of imperfect information. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, 2009.

[11] A. Gilpin, S. Hoda, J. Peña, and T. Sandholm. Gradient-based algorithms for finding Nash equilibria in extensive form games. In *3rd International Workshop on Internet and Network Economics (WINE)*, San Diego, CA, 2007.

[12] D. Koller, N. Megiddo, and B. von Stengel. Efficient computation of equilibria for extensive two-person games. *Games and Economic Behavior*, 14(2):247–259, 1996.

[13] H. W. Kuhn. Simplified two-person poker. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games*, volume 1 of *Annals of Mathematics Studies, 24*, pages 97–103. Princeton University Press, Princeton, New Jersey, 1950.

[14] T. Sandholm and A. Gilpin. Sequences of take-it-or-leave-it offers: Near-optimal auctions without full valuation revelation. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2006.

[15] T. Sandholm, A. Gilpin, and V. Conitzer. Mixed-integer programming methods for finding Nash equilibria. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 495–501, Pittsburgh, PA, 2005. AAAI Press / The MIT Press.

[16] T. Sandholm and V. R. Lesser. Leveled commitment contracts and strategic breach. *Games and Economic Behavior*, 35:212–270, 2001.

[17] S. P. Singh, V. Soni, and M. P. Wellman. Computing approximate Bayes-Nash equilibria in tree-games of incomplete information. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 81–90, New York, NY, 2004. ACM.

[18] N. D. Stein, A. Ozdaglar, and P. A. Parillo. Separable and low-rank continuous games. *International Journal of Game Theory*, 37(4):475–504, 2008.

[19] J. P. Vielma and G. L. Nemhauser. Modeling disjunctive constraints with a logarithmic number of binary variables and constraints. In *Conference on Integer Programming and Combinatorial Optimization (IPCO)*, 2008.

[20] Y. Vorobeychik and M. Wellman. Stochastic search methods for Nash equilibrium approximation in simulation-based games. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, Estoril, Portugal, 2008.

[21] K. Waugh, D. Schnizlein, M. Bowling, and D. Szafron. Abstraction pathologies in extensive games. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2009.

[22] M. Zinkevich, M. Bowling, M. Johanson, and C. Piccione. Regret minimization in games with incomplete information. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2007.

# A Math background

Most of the definitions presented in this section are taken or adapted from [1].

**Definition 14.** *A nonempty family $A$ of subsets of a set $T$ is an* algebra *of sets if it is closed under finite unions and complementation. That is,*

$$G, H \in A \rightarrow \left[G \cup H \in A \text{ and } G^C = T \backslash G \in A\right].$$

*A* $\sigma$-algebra *is an algebra that is also closed under countable unions. That is,*

$$\{G_i\} \subset A \text{ implies } \cup_{i=1}^{\infty} G_i \in A.$$

**Definition 15.** *A* measurable space *is a pair $(T, \Sigma)$, where $T$ is a set and $\Sigma$ is a $\sigma$-algebra of subsets of $T$.*

**Definition 16.** *Let $A_T$ and $A_Y$ be nonempty families of subsets of $T$ and $Y$, respectively. A function $f : T \rightarrow Y$ is* $(A_T, A_Y)$-measurable *if $f^{-1}(A)$ belongs to $A_T$ for each $A$ in $A_Y$. We say that $f$ is* $A_T$-measurable *when $A_Y$ is clearly understood, and we say that $f$ is* measurable *when both $A_Y$ and $A_T$ are clearly understood.*

**Definition 17.** *Let $\Sigma$ be a $\sigma$-algebra over a set $T$. A function $\mu$ from $\Sigma$ to the extended real number line is called a* measure *if it satisfies the following properties:*

*1.*

$$\mu(E) \geq 0 \text{ for all } E \in \Sigma.$$

2.

$$\mu(\emptyset) = 0.$$

3. *For all countable collections* $\{E_i\}$ *of pairwise disjoint sets in* $\Sigma$*:*

$$\mu\left(\cup_i E_i\right) = \sum_i \mu(E_i).$$

**Definition 18.** *A* measure space *is a triplet* $(T, \Sigma, \mu)$*, where* $\Sigma$ *is a* $\sigma$*-algebra of subsets of* $T$ *and* $\mu : \Sigma \to [0, \infty]$ *is a measure. If* $\mu(T) = 1$*, then* $\mu$ *is a* probability measure*.*

**Definition 19.** *Let* $(T, \Sigma)$ *and* $(Y, \Theta)$ *be measurable spaces. A* Markov kernel *is a mapping* $k : T \times \Theta \to [0, 1]$ *satisfying the following two properties.*

1. *For each* $t \in T$*, the set function* $k(t, \cdot) : \Theta \to [0, 1]$ *is a probability measure.*

2. *For each* $\theta \in \Theta$*, the mapping* $k(\cdot, \theta) : T \to [0, 1]$ *is* $\Sigma$*-measurable.*

**Definition 20.** *Let* $(T, \Sigma)$ *be a measurable space and let* $Y$ *be a finite set. A* discrete Markov kernel *is a mapping* $k : T \times Y \to [0, 1]$ *satisfying the following two properties.*

1. *For each* $t \in T$*, the set function* $k(t, \cdot) : Y \to [0, 1]$ *is a probability mass function.*

2. *For each* $y \in Y$*, the mapping* $k(\cdot, y) : T \to [0, 1]$ *is* $\Sigma$*-measurable.*

**Definition 21.** *Let* $\Sigma_i$ *be the set of all intervals of the form* $(a, b), (a, b], [a, b), [a, b]$ *for* $a \leq b$ *that are completely contained in* $X_i$*. (Note that* $X_i$ *is defined in Definition 6). Then* $\Sigma_i$ *is a* $\sigma$*-algebra, and* $(X_i, \Sigma_i)$ *is a measurable space.*

**Definition 22.** *Let* $\Lambda_i$ *denote the set of all discrete Markov kernels from* $(X_i, \Sigma_i)$ *to* $C_i$ *(note that we assume* $C_i$ *is finite). We call* $\Lambda_i$ *the* mixed strategy space *of player* $i$*.*

# B Proof of Proposition 5

Let $\Delta_i$ denote the space of Borel probability measures over $\hat{S}_i$, which we call the set of *mixed strategies* of player $i$. Let $V_i$ denote the space of all finite-valued signed measures on $\hat{S}_i$.

**Definition 23.** *Two measures* $\sigma_i, \tau_i \in V_i$ *are* almost payoff equivalent *if* $u_j(\sigma_i, s_{-i}) = u_j(\tau_i, s_{-i})$ *for all* $j \neq i$ *and all* $s_{-i} \in \hat{S_{-i}}$*.*

Let 0 denote the zero measure in $V_i$, and define

$$Y_i = \{\text{measures almost payoff equivalent to } 0\}.$$

**Definition 24.** *The* rank *of a continuous game is the* $n$*-tuple* $\rho = (\rho_1, \ldots, \rho_n)$ *where* $\rho_i = \frac{dim\ \Delta_i}{Y_i}$*. A game has* finite rank *if* $\rho_i < \infty$ *for all* $i$*.*

**Theorem 5.** *A continuous game is separable iff it has finite rank.*

The preceding definitions and theorem were presented and proved in [18].

**Proposition 5.** *The game considered in Section 3 is not separable.*

*Proof.* Suppose there exists a measure (not equal to 0) for player 2 that is almost payoff equivalent to 0; call this $\tau_2$. (Note that 0 refers to always folding.) Let

$$k = \frac{3\int_0^1 x d\tau_2(x)}{4}.$$

Now let $s_1 \in \hat{S}_1$ be the strategy of always raising in $[k, 1]$ and always folding in $[0, k)$. Then when player 1 has private signal in $[k, 1]$, he will have a negative profit since he will raise and will win less than $\frac{1}{4}$ of the time when he is called. When player 1 has private signal in $[0, k)$, his payoff will be 0 since he will fold. So $u_1(s_1, \tau_2) < 0$. However, if player 2 plays 0, then player 1 will have positive profit in $[k, 1]$ since he will raise and player 2 will fold, and will have zero profit in $[0, k)$, since he folds. So $u_1(s_1, 0) > 0$. Therefore, $u_1(s_1, \tau_2) \neq u_1(s_1, 0)$, and $\tau_2$ is not almost payoff equivalent to 0. So we have a contradiction, and must have $Y_2 = \emptyset$. So $\rho_2 = \infty$, and the game does not have finite rank. Therefore, by Theorem 5, it is not separable. □

# C Parametric models for limit Texas hold'em experiments

In this section we present the parametric models used for our experiments in Section 8.2 on limit Texas hold'em.

The first parametric model, shown in Figure 3, is identical to the model presented in [2] (with the thresholds renamed). For player 1, the action before the hyphen specifies the first action taken in the betting round, and the action after the hyphen specifies the next action taken if the betting gets back to player 1. For example, between thresholds b and c player 1 will check first, and if player 2 bets he will call. A *bluff* denotes a bet with a bad hand (where the player betting is hoping the other player folds). So with a private signal between d and 1 player 1 will bet, and he will fold if player 2 raises. For player 2, the first action listed denotes the action taken when player 1 bets, and the second action (after the slash) denotes the action taken when player 1 checks. For example, between f and g player 2 will call if player 1 bets and check if player 1 checks. The second parametric model, shown in Figure 4, is identical to the first model except that threshold i is shifted above threshold d. In the third parametric model (Figure 5), player 1 only checks when he is first to act (and never bets).

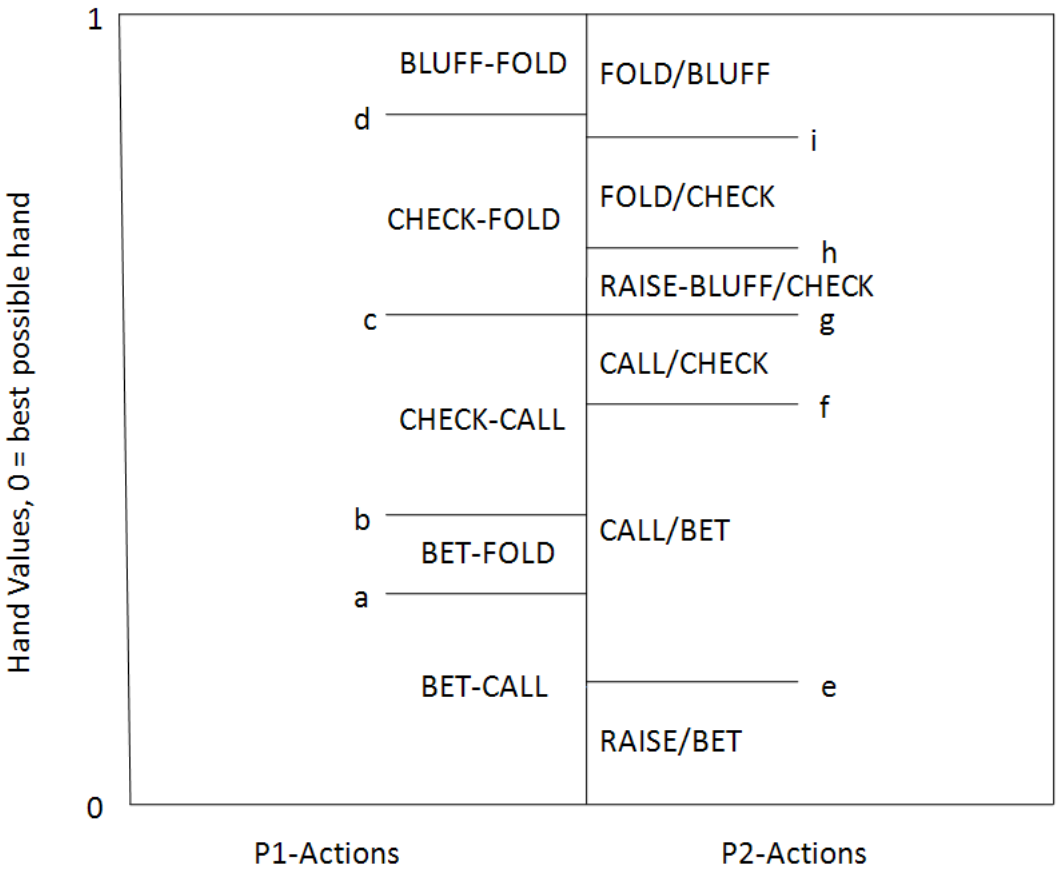



Figure 3: First parametric model.

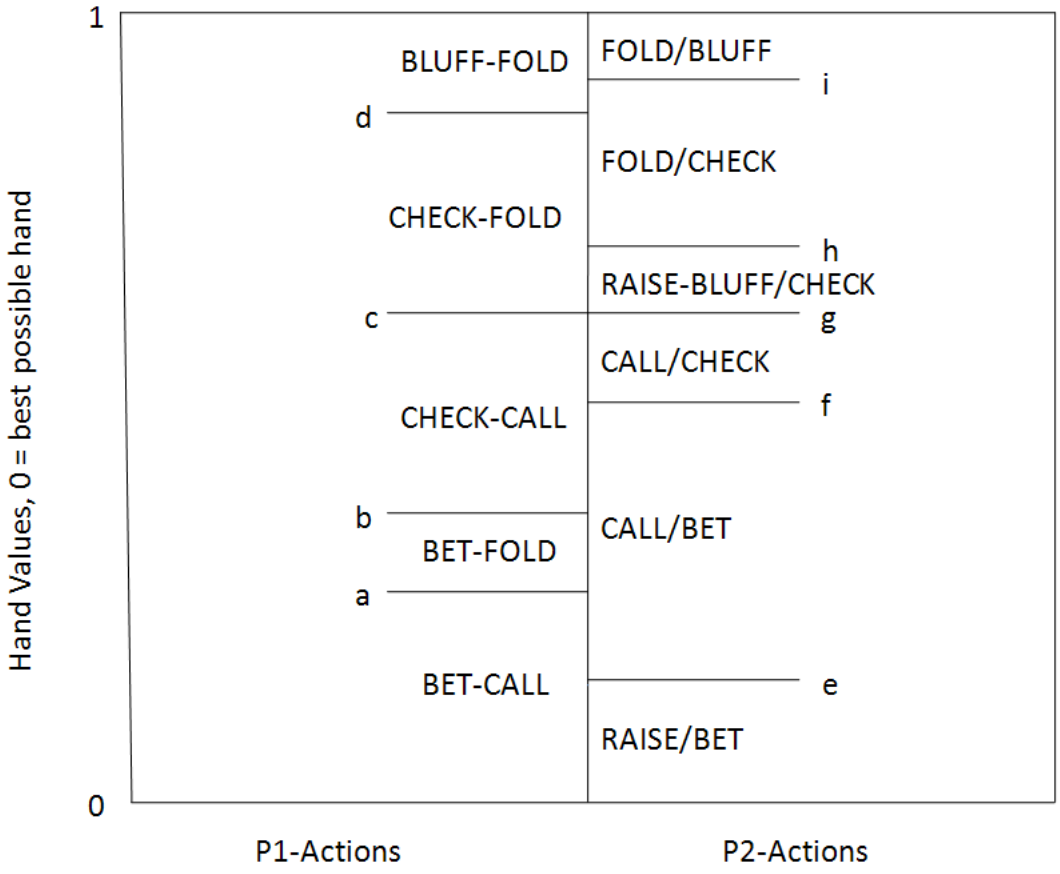



Figure 4: Second parametric model.

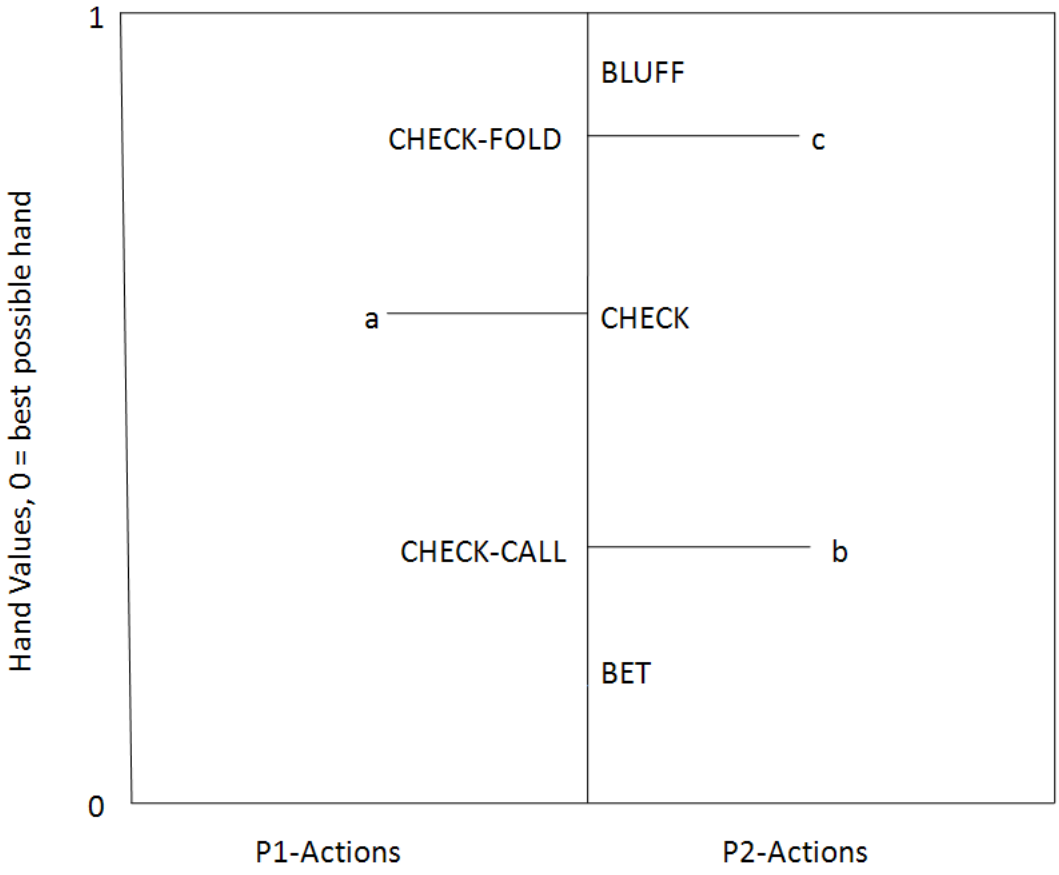



Figure 5: Third parametric model.

# D Figures for multiplayer experiments

In this section we present the parametric models relating to the multiplayer experiments described in Section 8.3. The parametric model is shown in Figure 6. For player 2, the first action listed denotes the action taken when player 1 raises, and the second action (after the slash) denotes the action taken when player 1 folds. For player 3, the first action listed denotes the action taken when player 1 raises and player 2 calls, the second action denotes the action taken when player 1 raises and player 2 folds, and the third action denotes the action taken when player 1 folds and player 2 raises.

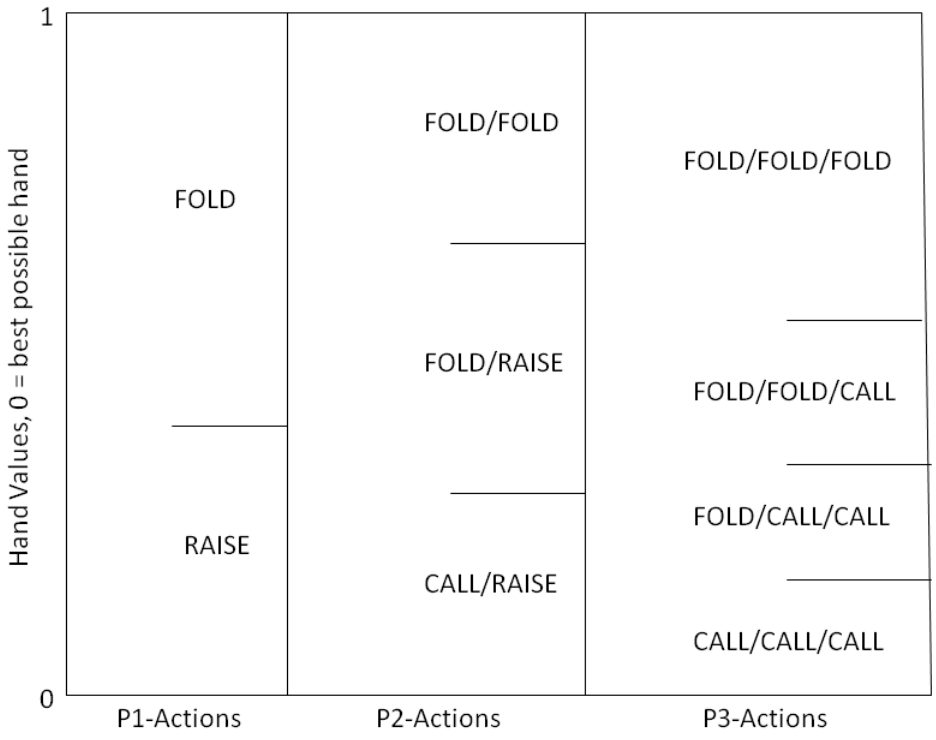



Figure 6: Parametric model for the three-player poker game.

The equivalent *conditional* parametric model representation (as described in Section 8.3.1) is given in Figure 7. The first column denotes player 1's action, the second column denotes player 2's action when player 1 raises, the third column denotes player 2's action when player 1 folds, the fourth column denotes player 3's action when player 1 raises and player 2 calls, the fifth column denotes player 3's action when player 1 raises and player 2 folds, and the sixth column denotes player 3's action when player 1 folds and player 2 raises.

These two representations are equivalent in terms of their expressive power, the representation

sizes of the action spaces, and the MILFP program that gets generated. For example, while only one column corresponds to player 3's action space in Figure 6 and three columns do in Figure 7, the length of the size of player 3's action (e.g., FOLD/FOLD/CALL) is three times larger in Figure 6.

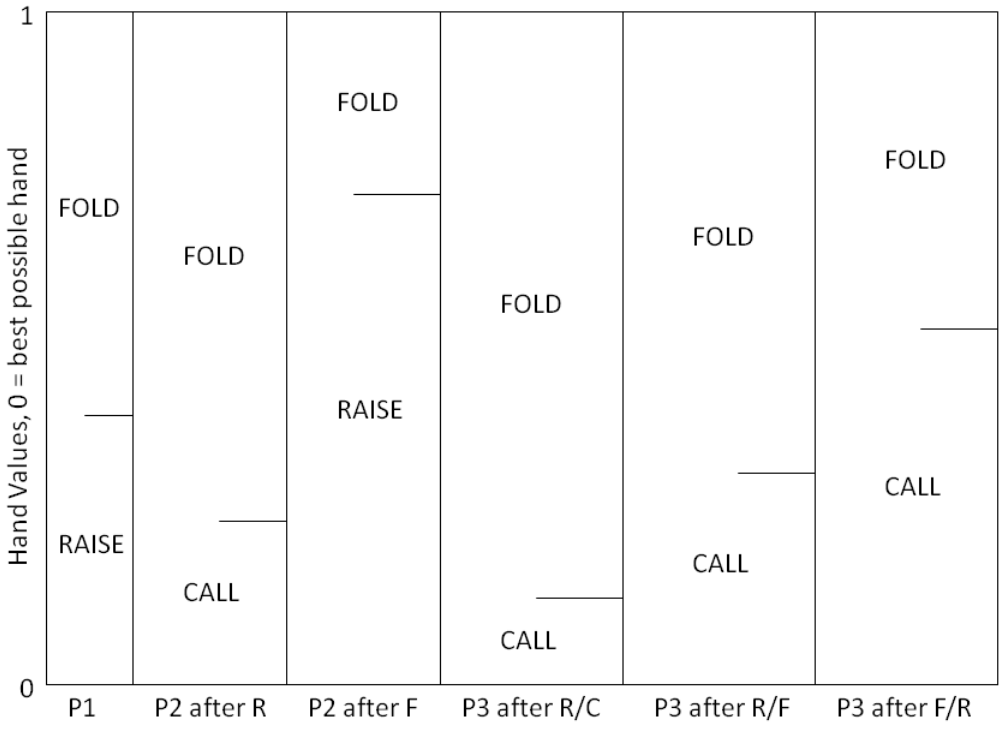
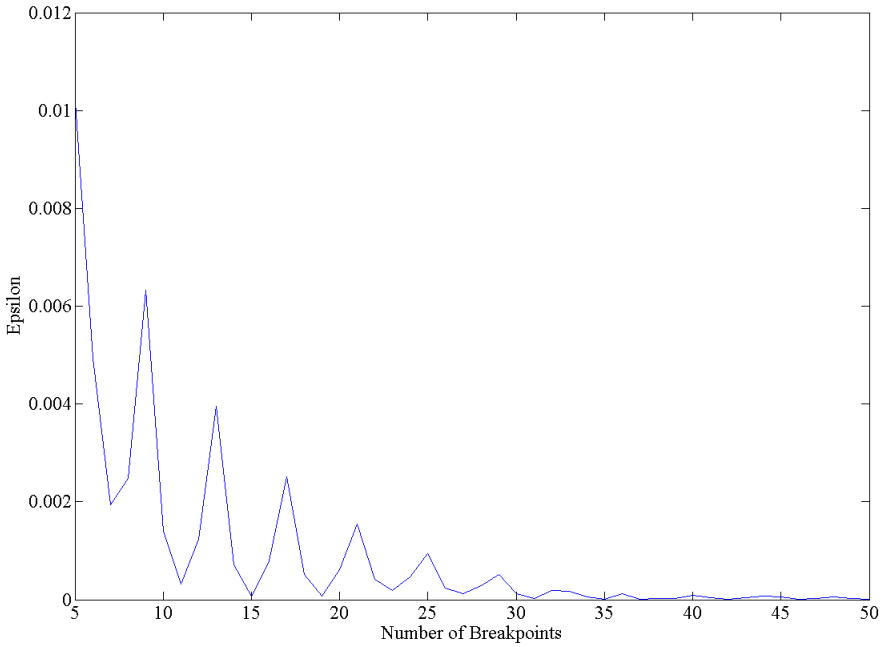
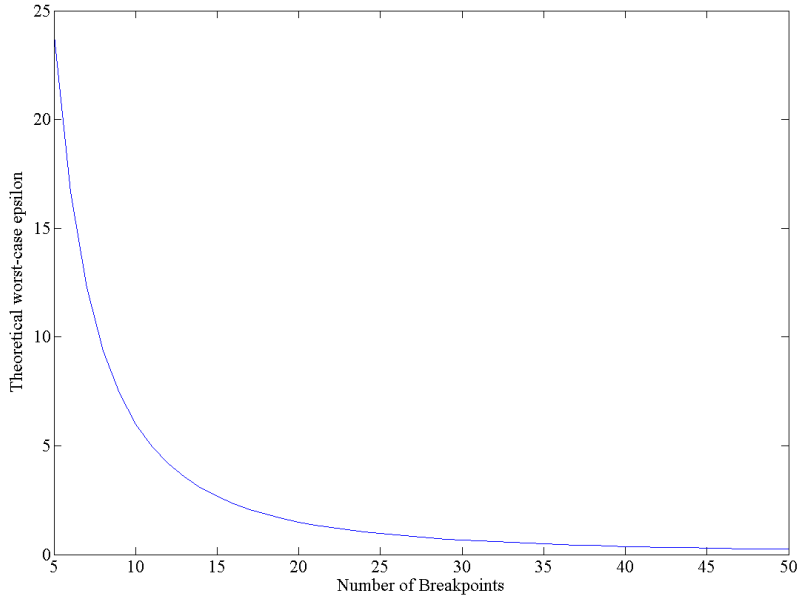



Figure 7: Conditional parametric model representation of the model given in Figure 6.




(a) Empirical solution quality, $\epsilon$, as a function of the number of breakpoints.




(b) Solution quality, $\epsilon$, guaranteed by Equation 3 as a function of the number of breakpoints.

Figure 8: Experimental values of, and the theoretical bound on, $\epsilon$.

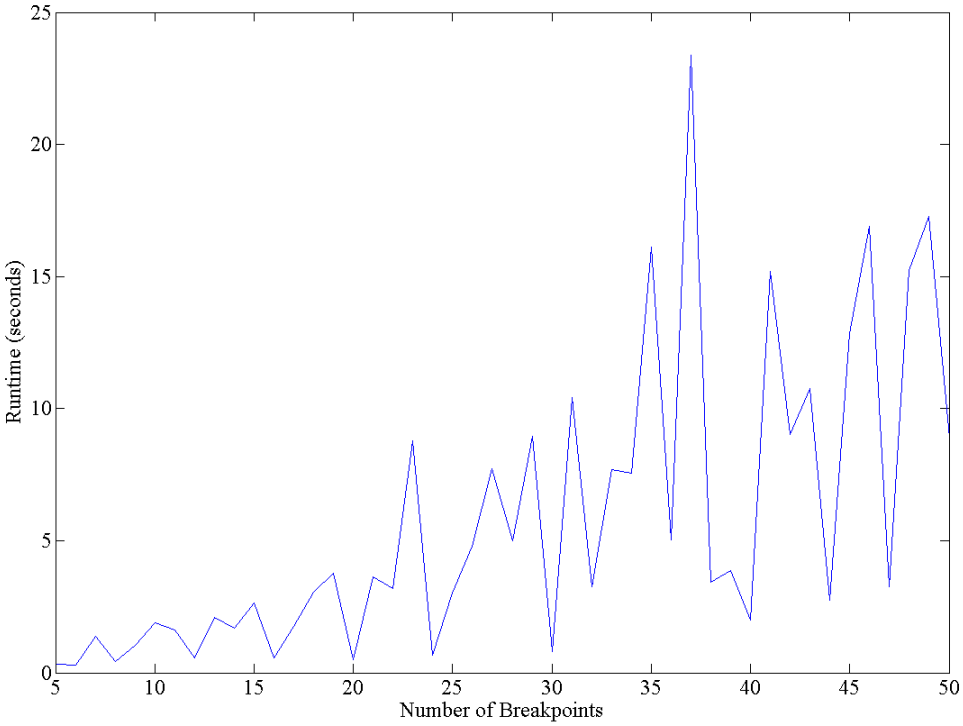



Figure 9: Running time (in seconds) as a function of the number of breakpoints.