Opponent-Model Search in Games with Incomplete Information*

Junkang Li[1,2], Bruno Zanuttini[2], Véronique Ventos[1]
[1]NukkAI, Paris, France
[2]Normandie Univ.; UNICAEN, ENSICAEN, CNRS, GREYC, 14 000 Caen, France
junkang.li@nukk.ai, bruno.zanuttini@unicaen.fr, vventos@nukk.ai

Abstract

Games with incomplete information are games that model situations where players do not have common knowledge about the game they play, e.g. card games such as poker or bridge. Opponent models can be of crucial importance for decision-making in such games. We propose algorithms for computing optimal and/or robust strategies in games with incomplete information, given various types of knowledge about opponent models. As an application, we describe a framework for reasoning about an opponent's reasoning in such games, where opponent models arise naturally.

1 Introduction

Opponent models are models that describe or predict how an opponent reasons or behaves in a game. Such models have been explicitly incorporated into game tree search algorithms (e.g. minimax, $\alpha\beta$ search, MCTS) for games with perfect information (Iida et al. 1993, 1994; Sturtevant and Bowling 2006; Sturtevant, Zinkevich, and Bowling 2006) to find robust strategies against a given opponent model, i.e. strategies that guarantee a given payoff against any strategy deemed possible by the opponent model. The knowledge of opponent models can accelerate the game tree search (e.g. by pruning branches not considered by an opponent) and improve the performance of the strategies obtained (e.g. by exploiting the weakness of an opponent).

In this paper, we extend the idea of opponent-model search to (two-player, zero-sum) games with incomplete information. In such games, players do not share the same information. Notable examples include card games, where players cannot see the set of cards (i.e. hand) of the other players. We present three contributions to this subject. We first propose different ways of taking opponent models into account (which is of interest beyond the setting of incomplete information), and give algorithms for computing the corresponding robust strategies for such games. We then propose a principled method for taking into account a probability that the opponent does not behave according to any of the given models. Finally, we show an application of these models to the recursive modelling of opponents, where a level-k player assumes that their opponent reasons at some level lower than k, and recursively down to level 0.

*A long version with proofs of all claims is available at https://hal.science/hal-04100646.

2 Related Work

When no opponent model is available, one typically aims to be robust against all possible strategies of an opponent. Koller and Megiddo (1992); Koller, Megiddo, and von Stengel (1996); von Stengel (1996) study the complexity of computing maxmin strategies in this case under a variety of settings; in particular, for mixed strategies, they give polynomial-time algorithms based on linear programming for two-player extensive-form games with perfect recall, a more general setting than ours. McMahan, Gordon, and Blum (2003) propose a double-oracle algorithm for computing optimal mixed strategies for Markov decision processes with adversarial cost functions, which can also be regarded as a polynomial-time algorithm for computing the maxmin strategies of a normal-form game. Bosanský et al. (2014) combine these ideas to propose an algorithm for zero-sum extensive-form games with perfect recall, which is efficient when optimal mixed strategies have small supports.

Opponent models can come in diverse forms. Iida et al. (1993, 1994) propose opponent models for games with perfect information, where models are given by the evaluation function and the search depth of the opponent. Sturtevant, Zinkevich, and Bowling (2006) propose opponent models given by opponents' preferences over the outcomes of a game. Rebstock et al. (2019) use opponent models learnt from human games for imperfect information (card) games. A survey of opponent modelling approaches is also provided by Albrecht and Stone (2018). Our work is related to these in the sense that we assume opponent models to be given (called "type-based reasoning" by Albrecht and Stone (2018, Section 4.2)). However, an important stream of work also studies the *learning* of opponent models; we refer the reader to the survey by Nashed and Zilberstein (2022).

An important class of opponent models is that of *recursive* models, where MAX searches a strategy (at level k) assuming that MIN themselves searches a strategy (at level $k-1$) assuming that MAX searches…, etc., down to level 0. Such models have been studied in behavioural game theory (Dhami 2019) to capture human reasoning. For instance, Camerer, Ho, and Chong (2004) propose a cognitive hierarchy model in which an opponent's level is modelled by a Poisson distribution over levels $k-1,\ldots,0$; they also validate this model against empirical data. Wright and Leyton-Brown (2019) assess the relevance of various modelling assumptions for level 0. De Weerd, Verbrugge, and Verheij (2013) assess the efficiency of reasoning with recursive models by simulation. Such recursive models are also used in epistemic game theory to define notions such as common belief in rationality (Perea 2012).

Recursive models are also considered for cooperative planning. In particular, interactive POMDPs are a framework for collaborative decision-making in partially observable environments (Gmytrasiewicz and Doshi 2005; Doshi, Gmytrasiewicz, and Durfee 2020). In this model, a level-k agent optimises their behaviour given a distribution over (partially observed) physical states and over other agents' models at level $k-1$. Interestingly, optimal behaviours at level k can be computed iteratively by solving a sequence of POMDPs, where at each iteration the other agents' model can be considered as part of the environment. For human-agent collaboration, You et al. (2023) propose a recursive model with bounded depth, whereby an agent plans (at level 2) a best response to a mixture of possible strategies for a human, with these human strategies themselves defined (at level 1) by planning assuming that the agent is controlled by the human (at level 0).

3 Formal Setting

Given a finite tree T, we write r for its root, $\mathrm{N}(T)$ for the set of nodes, $\mathrm{C}(n)$ for the set of children of $n \in \mathrm{N}(T)$, and $\mathrm{L}(T)$ for the set of leaves. We consider two-player (MAX and MIN, denoted by $+$ and $-$) and zero-sum[1] extensive-form games in the following form.

Definition 1 (VG). *A* vector game *(VG) is a tuple* $G := \langle T, \mathrm{P}, t, \vec{u}, \vec{\rho}\rangle$*, where* T *is a finite tree,* $\mathrm{P} : \mathrm{N}(T) \setminus \mathrm{L}(T) \to \{+,-\}$ *determines whose turn it is at a node,* $t \in \mathbb{N}$ *is the number of MIN's types,* $\vec{u} : \mathrm{L}(T) \to \mathbb{R}^t$ *gives the utility for MAX depending on MIN's type, and* $\vec{\rho}$ *is a commonly known distribution over MIN's types.*

Example. *A VG with* 5 *types of MIN and the uniform prior (each MIN's type is drawn with probability* $1/5$*) is given in Figure 1, where a square (resp. a circle) denotes a node of MAX (resp. of MIN). If MIN plays* a *(at A) and MAX plays* l *(at B), MAX's payoff is*[2] 11100*, which means MAX's gain is* 1 *if MIN is of one of the first three types, and* 0 *otherwise.*

A VG is a game in which MAX has incomplete information (MAX does not know MIN's type) and MIN has perfect information. It has no chance factor, except the initial drawing of MIN's type. VGs can be obtained by applying the *best-defence model* (Frank and Basin 2001) to general games with incomplete information, and can be used to model card games such as Bridge; under this model, MAX assumes that MIN plays as if MIN knew the dealing of cards.

[1]Our study can be easily extended to more players and general-sum, with the exception of the lexicographic setting, for which the definition of opponent models does not trivially generalise.

[2]When possible, we write vertical vectors compactly inline.

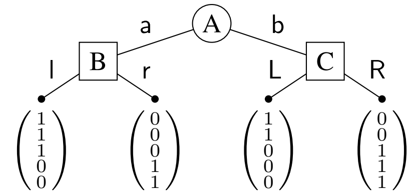


Figure 1: A VG with 5 possible worlds and the uniform prior.

Strategies in a VG A *pure strategy* s_+ of MAX maps every node n of MAX to a child of n; a pure strategy s_- of MIN maps every node n of MIN and each type to a child of n.[3] We write Σ_i^{P} for the set of all pure strategies of i. A *mixed strategy* σ_i of player i is a probability distribution over Σ_i^{P}, written as $p_1 s_i^1 + \cdots + p_k s_i^k$, with the interpretation that i draws s_i^j with probability p_j at the beginning of the game, and then plays s_i^j; we write Σ_i^{M} for the set of all mixed strategies of i. Finally, a *behaviour strategy* π_+ of MAX (resp. π_- of MIN) maps every node n of MAX (resp. every node of MIN and every type) to a probability distribution over $\mathrm{C}(n)$, with the interpretation that the player draws their moves at n according to this distribution, and they do so independently at each of their nodes.

Given a VG $\langle T, \mathrm{P}, t, \vec{u}, \vec{\rho}\rangle$ and a pair of strategies (of any kind) $(\varsigma_+, \varsigma_-)$, we write $\mathcal{U}(\varsigma_+, \varsigma_-)$ for MAX's expected payoff with respect to the probability distribution over $\mathrm{L}(T)$ induced by drawings of MAX's and MIN's strategies/moves (according to ς_+ and ς_-) and MIN's types (according to $\vec{\rho}$).[4]

4 Maxmin Values Without Opponent Models

We first consider the case without opponent models, which is the basis of our algorithms for opponent-model search. In this case, we are interested in computing the *maxmin value*

$$\underline{v}_+ := \max_{\varsigma_+ \in \Sigma_+} \min_{s_- \in \Sigma_-^{\mathrm{P}}} \mathcal{U}(\varsigma_+, s_-),$$

where Σ_+ is Σ_+^{P} (*pure maxmin*) or Σ_+^{M} (*mixed maxmin*), depending on context. Since $\mathcal{U}$ is linear in MIN's mixed strategies, replacing Σ_-^{P} by Σ_-^{M} would not change the value, hence our definition only concerns pure strategies of MIN.

The maxmin value $\underline{v}_+$ is the largest payoff MAX can guarantee by any strategy from Σ_+, no matter how MIN plays. MAX's strategies that achieve $\underline{v}_+$ are called *maxmin strategies*. By definition, the mixed maxmin value is no smaller than the pure maxmin value. As we will see, it is, in general, more difficult to compute the pure maxmin value for VGs than the mixed maxmin value. Still, in some situations, pure maxmin is more desirable or even the only viable solution concept, e.g. when outcomes are only partially ordered, or when mixed strategies are not allowed due to

[3]Hence, MIN can pick different actions at the same node according to their type, which is hidden to MAX.

[4]Concretely, let p_i be the probability over the leaves induced by ς_+ and ς_- when MIN is of type i. Then $\mathcal{U}(\varsigma_+, \varsigma_-)$ is defined to be $\sum_{i=1}^{t} \sum_{l \in \mathrm{L}(T)} p_i(l) u(l)_i \rho_i$, where $u(l)_i$ and ρ_i are the i-th components of the vectors $\vec{u}(l)$ and $\vec{\rho}$.

Algorithm 1: Generic Minimax Algorithm

```
def MiniMax (node n, game ⟨T, P, t, u⃗, ρ⃗⟩):
    if n is a terminal node:
        return eval(n)
    elif P(n) = +:
        return ⋁_{n′∈C(n)} MiniMax (n′)
    else:
        return ⋀_{n′∈C(n)} MiniMax (n′)
```

their probabilistic nature. Hence, we will study algorithms for both notions, with a focus on pure maxmin since algorithms for mixed maxmin only require minor modifications in the presence of opponent models.

Generic Algorithm The maxmin value of games with perfect information is typically computed by the *minimax* algorithm, a generic version of which is shown in Algorithm 1 (the maxmin *strategies* can be computed by bookkeeping).

This algorithm has four parameters, which we use to capture different algorithms in the following sections:

- V is a set of objects called *situational values*;
- eval is an evaluation function which maps each terminal node n to a value $\mathsf{eval}(n) \in V$;
- $\vee, \wedge : V \times V \to V$ are two associative binary operators, referred to as MAX's and MIN's operator, respectively.

With eval as boundary conditions, this algorithm recursively defines a situational value $\mathsf{val}(n)$ for every node n. For an instantiation of this algorithm to compute the maxmin values, one should choose the parameters according to the class of games under consideration such that there is a polynomial-time computable mapping from the situational value of the root $\mathsf{val}(r)$ to the maxmin value of the game.

We write $\mathtt{MiniMax}(V, \mathsf{eval}, \vee, \wedge)$ for its instantiation with the parameters $V, \mathsf{eval}, \vee, \wedge$, and denote by $\mathsf{val}(n)$ the value which it associates to each node n. For example, for games with perfect information (i.e. when $t = 1$ and $\vec{u}(n) \in \mathbb{R}$), it is well-known that $\mathtt{MiniMax}(\mathbb{R}, \vec{u}, \max, \min)$ satisfies $\mathsf{val}(r) = \underline{v}_+$, where r is the root of the tree.

This algorithm has several advantages: returned values for internal nodes are readily interpretable; the algorithm is efficient on memory since the recursion depth is the depth of the game tree, which in general is exponentially smaller than the tree; the search can be combined with other techniques, such as heuristic functions and $\alpha\beta$ pruning (which is possible whenever $(V, \vee, \wedge)$ forms a lattice (Li et al. 2022)), move ordering, Monte Carlo techniques such as MCTS, etc.

Pure Maxmin Frank and Basin (2001) show that the pure maxmin value is NP-hard to compute for VGs. Ginsberg (2001) proposes an exact algorithm which we reframe as follows. Let us write $\mathcal{P}_{<\infty}(\mathbb{R}^t)$ for the set of all finite sets of vectors in $\mathbb{R}^t$. For $f, g \in \mathcal{P}_{<\infty}(\mathbb{R}^t)$, we define $f \sqcap g$ by

$$f \sqcap g \coloneqq \left\{ \left(\min(v_i, v_i') \right)_{1 \le i \le t} \mid \vec{v} \in f, \vec{v}\,' \in g \right\}.$$

Proposition 2. *Let $\langle T, \mathrm{P}, t, \vec{u}, \vec{\rho} \rangle$ be a VG with root r. For $l \in \mathrm{L}(T)$, let $\mathsf{eval}(l) \coloneqq \{\vec{u}(l)\} \in \mathcal{P}_{<\infty}(\mathbb{R}^t)$. Then $\mathtt{MiniMax}(\mathcal{P}_{<\infty}(\mathbb{R}^t), \mathsf{eval}, \cup, \sqcap)$ satisfies*

$$\underline{v}_+ \coloneqq \max_{s_+ \in \Sigma_+^{\mathrm{P}}} \min_{s_- \in \Sigma_-^{\mathrm{P}}} \mathcal{U}(s_+, s_-) = \max_{\vec{v} \in \mathsf{val}(r)} \vec{\rho} \cdot \vec{v}.$$

Example. *For the VG in Figure 1, we get* $\mathsf{val}(B) = \{11100, 00011\}$*;* $\mathsf{val}(C) = \{11000, 00111\}$*;* $\mathsf{val}(A) = \{11000, 00100, 00000, 00011\}$*. This algorithm actually recursively enumerates all pure strategies of MAX: each vector in* $\mathsf{val}(n)$ *implicitly represents one or several strategies of MAX in the subtree rooted at n. At the root A, given the uniform prior over MIN's types, the best vectors are* 11000 *(corresponding to MAX's strategy* (l, L)*, by which MAX chooses* l *at B and* L *at C) and* 00011 *(MAX's strategy* (r, R)*); both achieve the pure maxmin value* $2/5$*.*

Importantly, in a VG, the expected payoff of a strategy in a subtree may depend on that of the strategies in a far away subtree. In our example, l and R are locally optimal with respect to the uniform prior. However, (l, R) is not optimal at the root, since it is MIN who chooses, with perfect information, either a or b as a function of their type. In other words, it is not correct to use the common prior to evaluate strategies locally at nodes B and C: the conditional probabilities of MIN's types at both B and C depend on MIN's strategy and can be different from the prior.[5]

This algorithm can be improved by pruning vectors at each node. If in $\mathsf{val}(n)$, a vector $\vec{v}$ is weakly dominated by another vector $\vec{v}\,'$, then we can discard $\vec{v}$ from $\mathsf{val}(n)$, which amounts to eliminating weakly dominated strategies. For example, if A is an internal node of a larger VG, then 00000 (corresponding to MAX's strategy (r, L)) can be pruned from $\mathsf{val}(A)$, since MAX never does worse by playing, say, the strategy represented by 11000 in the subtree rooted at A. In general, any other kind of reduction of strategies is unsound; contrastingly, we will see that more reductions become sound if opponent models are available.

Mixed Maxmin Contrary to pure maxmin, the mixed maxmin value can be computed in polynomial time with the algorithm proposed by Koller and Megiddo (1992), which relies on two insights: (1) The set of all mixed strategies of MAX can be represented by a system L of linear equalities, with linearly many (in the size of the game tree) variables and equalities; (2) For any threshold v and any mixed strategy σ_+ of MAX represented as a solution to L, it can be verified in linear time whether $\min_{s_- \in \Sigma_-^{\mathrm{P}}} \mathcal{U}(\sigma_+, s_-) \ge v$ holds by computing MIN's best responses to σ_+. This computation serves as a separation oracle, under which a linear program (LP) maximising v with respect to the constraints in L computes the mixed maxmin value.

Example. *In the game in Figure 1, MAX's optimal mixed strategy is the uniform mixture over all 4 pure strategies, which yields an expected payoff of at least $\frac{1}{2}$, the mixed maxmin value (higher than the pure maxmin value $\frac{2}{5}$).*

The above algorithm has been improved by von Stengel (1996); Koller, Megiddo, and von Stengel (1996). However,

[5]This phenomenon, called *non-locality* by Frank and Basin (2001), is the culprit behind the NP-hardness of pure maxmin.

for simplicity, we only show modifications of the initial algorithm for taking opponent models into account. Adapting them to the improved algorithms is straightforward.

5 Opponent-Model Search

We now come to our main contributions, which are algorithms for computing the maxmin value against a given set of opponent's strategies, called *opponent models* (OM):

$$\underline{v}_+ \coloneqq \max_{\varsigma_+ \in \Sigma_+} \min_{\omega_- \in \Sigma_-^{\mathrm{O}}} \mathcal{U}(\varsigma_+, \omega_-),$$

where Σ_+ is the set of all pure or all mixed strategies of MAX and Σ_-^{O} is a given set of OMs.

In general, OMs are models of the opponent's reasoning, which can come in various forms (cf. the section on related work). As a quite general setting, we consider that each OM describes a behaviour strategy of MIN.[6] Algorithmically, we assume that the OMs are given in the input and that each computation of $\omega_-(n, i, n')$ takes constant time, where $\omega_-(n, i, n')$ is the probability that under ω_-, MIN chooses $n' \in \mathrm{C}(n)$ at node n if MIN is of type i.

In this section, we consider situations where MAX is certain that MIN only considers strategies described by these OMs. This assumption will be relaxed later.

Belief States With the knowledge of OMs, MAX can gain information about the actual strategy employed by MIN and MIN's type during the game using Bayesian reasoning. To model this, we define the *non-normalised belief state* (NBS) at node n about OM ω_-, written as $\mathrm{NBS}(n, \omega_-) \in [0, 1]^t$, and defined top-down by: $\mathrm{NBS}(r, \omega_-) \coloneqq \vec{\rho}$; for $n' \in \mathrm{C}(n)$, $\mathrm{NBS}(n', \omega_-) \coloneqq \mathrm{NBS}(n, \omega_-)$ if n is MAX's node, otherwise $\mathrm{NBS}(n', \omega_-)_i \coloneqq \mathrm{NBS}(n, \omega_-)_i \times \omega_-(n, i, n')$. Let us emphasise that we define $\mathrm{NBS}(n, \omega_-)$ to be *non-normalised*; normalising it yields the conditional probability of MIN's types, given that n is reached and MIN plays ω_-.

Single OM When there is only one OM ω_-, MAX has complete knowledge of MIN's strategy. Then the game becomes a single-player game with perfect information (Koller and Megiddo 1992), and the pure/mixed maxmin value reads

$$\underline{v}_+ \coloneqq \max_{s_+ \in \Sigma_+^{\mathrm{P}}} \mathcal{U}(s_+, \omega_-) = \max_{\sigma_+ \in \Sigma_+^{\mathrm{M}}} \mathcal{U}(\sigma_+, \omega_-),$$

where the last equality is due to the linearity of u.

This value can be computed by a depth-first procedure, which recursively computes MAX's best strategies at each of their decision node. More precisely, MIN's decision nodes become chance nodes. Hence, even though MAX does not know MIN's type, they can pick actions to maximise their expected payoff with respect to MIN's type.

Example. *Consider again the game in Figure 1, with ω_- as follows: MIN plays* a *if of type* 1 *or* 2, b *if of type* 4 *or* 5, *and* $\frac{1}{2}\mathsf{a} + \frac{1}{2}\mathsf{b}$ *if of type* 3. *Given ω_- and the uniform prior $\vec{\rho}$, MAX can compute the NBS $(\frac{1}{5}, \frac{1}{5}, \frac{1}{10}, 0, 0)$ at node B. Given this NBS, action* l *yields a higher (non-normalised) payoff of $1/2$ than* r *(with a payoff of 0) at B.*

Similarly, the NBS at C is $(0, 0, \frac{1}{10}, \frac{1}{5}, \frac{1}{5})$ and prescribes action R *(with a payoff of $1/2$). At node A, MAX's payoff is simply the sum of their (non-normalised) payoff at B and C, which yields 1. One can check that 1 is indeed the best MAX can get when playing against MIN under this particular OM; this payoff is obtained by the strategy* (l, R)*, which gives MAX a payoff of 1 independent of MIN's actual type.*

Proposition 3. *Let $\langle T, \mathrm{P}, t, \vec{u}, \vec{\rho}\rangle$ be a VG with root r, and let ω_- be an OM. For $l \in \mathrm{L}(T)$, let $\mathsf{eval}(l) \coloneqq \mathrm{NBS}(l, \omega_-) \cdot \vec{u}(l)$. Then $\mathtt{MiniMax}(\mathbb{R}, \mathsf{eval}, \max, +)$ runs in $\mathcal{O}(t|T|)$ time and satisfies*

$$\underline{v}_+ \coloneqq \max_{s_+ \in \Sigma_+^{\mathrm{P}}} \mathcal{U}(s_+, \omega_-) = \max_{\sigma_+ \in \Sigma_+^{\mathrm{M}}} \mathcal{U}(\sigma_+, \omega_-) = \mathsf{val}(r).$$

This algorithm can be considered as a generalisation of OM search as proposed by Iida et al. (1993), which only considers games with perfect information for which OMs are described by MIN's evaluation functions.

Probabilistic OMs We now consider the case in which MAX has the knowledge of several OMs $\omega_-^1, \dots, \omega_-^m$ of MIN, and a probability distribution $\vec{p} = (p_1, \dots, p_m)$ over them: MIN plays the strategy ω_-^1 with probability p_1, ω_-^2 with probability p_2, etc. In particular, the pure/mixed maxmin value reads

$$\underline{v}_+ = \max_{s_+ \in \Sigma_+^{\mathrm{P}}} \sum_{j=1}^{m} p_j \mathcal{U}(s_+, \omega_-^j) = \max_{\sigma_+ \in \Sigma_+^{\mathrm{M}}} \sum_{j=1}^{m} p_j \mathcal{U}(\sigma_+, \omega_-^j).$$

This setting is not much different from the previous one, due to the linearity of u: these OMs can be merged into one single OM describing the mixed strategy $\omega_- \coloneqq p_1\omega_-^1 + \cdots + p_m\omega_-^m$.[7] In principle, one can traverse the game tree once and compute the behaviour strategy corresponding to ω_-, then run the single-OM algorithm from Proposition 3. However, with the help of NBS, one can avoid explicitly computing and storing the strategy ω_-.

Proposition 4. *Let $\langle T, \mathrm{P}, t, \vec{u}, \vec{\rho}\rangle$ be a VG with root r, and let $\omega_-^1, \dots, \omega_-^m$ be OMs distributed according to $\vec{p} = (p_1, \dots, p_m)$. Let $\mathsf{eval}(l) \coloneqq \sum_{j=1}^{m} p_j \,\mathrm{NBS}(l, \omega_-^j) \cdot \vec{u}(l)$ for $l \in \mathrm{L}(T)$. Then $\mathtt{MiniMax}(\mathbb{R}, \mathsf{eval}, \max, +)$ satisfies $\underline{v}_+ = \mathsf{val}(r)$ and runs in $\mathcal{O}(mt|T|)$ time.*

Lexicographic OMs Let us now consider the case in which MAX holds a lexicographic belief over MIN's OMs $\omega_-^1, \dots, \omega_-^m$: MAX deems that MIN most probably follows ω_-^1; otherwise, with an infinitesimally smaller probability, MIN follows ω_-^2; etc. We define the pure/mixed maxmin value in this case to be the vector of length m

$$\begin{aligned}\overrightarrow{\underline{v}_+} &\coloneqq \operatorname*{lexmax}_{s_+ \in \Sigma_+^{\mathrm{P}}} \left(\mathcal{U}(s_+, \omega_-^1), \dots, \mathcal{U}(s_+, \omega_-^m)\right) \\ &= \operatorname*{lexmax}_{\sigma_+ \in \Sigma_+^{\mathrm{M}}} \left(\mathcal{U}(\sigma_+, \omega_-^1), \dots, \mathcal{U}(\sigma_+, \omega_-^m)\right) \in \mathbb{R}^m,\end{aligned}$$

[6] In VGs, all MIN's mixed strategies can be expressed as behaviour strategies since MIN has perfect information (Kuhn 1953). In addition, given a strategy represented by a mixed strategy or another linear representation (e.g. sequence form (Koller and Megiddo 1992) or evaluation function (Iida et al. 1993)), its equivalent behaviour strategy can be computed in linear time.

[7] We abuse the notation by writing ω_-^i for both the given behaviour strategy and its equivalent mixed strategy.

where lexmax is lexicographic maximum over vectors of length m. In other words, if there is a unique optimal strategy against ω_-^1, then this strategy is chosen; otherwise, ties are broken according to their values against ω_-^2, and so on.

This setting can be regarded as an instance of probabilistic OMs, where the distribution over OMs is $\overrightarrow{p_\varepsilon} = (1, \varepsilon, \varepsilon^2, \ldots, \varepsilon^{m-1})$ with ε an indeterminate interpreted as an infinitesimally small value. However, we also give a direct algorithm below. We write $\mathrm{NBSM}(n)$ for the $m \times t$ matrix $\big(\mathrm{NBS}(n, \omega_-^1)^\intercal, \ldots, \mathrm{NBS}(n, \omega_-^m)^\intercal\big)$, and $+^m$ for component-wise addition of vectors in $\mathbb{R}^m$.

Proposition 5. *Let $\langle T, \mathrm{P}, t, \vec{u}, \vec{\rho}\rangle$ be a VG with root r, and let $\omega_-^1, \ldots, \omega_-^m$ be OMs with a lexicographic interpretation. For $l \in \mathrm{L}(T)$, let $\mathsf{eval}(l) \coloneqq \mathrm{NBSM}(l) \times \vec{u}(l) \in \mathbb{R}^m$. Then $\mathtt{MiniMax}(\mathbb{R}^m, \mathsf{eval}, \mathrm{lexmax}, +^m)$ satisfies $\overrightarrow{\underline{v}_+} = \mathsf{val}(r)$ and runs in $\mathcal{O}(mt|T|)$ time.*

Nondeterministic OMs The last case is when MAX has no probability distribution over MIN's OMs: MIN's strategy is only known to be among $\omega_-^1, \ldots, \omega_-^m$. This situation is similar to planning under adversarial cost functions (McMahan, Gordon, and Blum 2003). The maxmin value is then

$$\underline{v}_+ \coloneqq \max_{\varsigma_+ \in \Sigma_+} \min_{1 \le j \le m} \mathcal{U}(\varsigma_+, \omega_-^j),$$

which, in general, is different depending on whether Σ_+ is Σ_+^{P} or Σ_+^{M}. MIN now has (*a priori*) more agency than in the case of probabilistic OMs, since they can choose from a larger (but still limited) set of strategies.

We first consider pure maxmin. For $f, g \in \mathcal{P}_{<\infty}(\mathbb{R}^m)$, define the following operator:

$$f \oplus^m g \coloneqq \{(v_j + v'_j)_{1 \le j \le m} \mid \vec{v} \in f, \vec{v}\,' \in g\} \subseteq \mathbb{R}^m.$$

Proposition 6. *Let $\langle T, \mathrm{P}, t, \vec{u}, \vec{\rho}\rangle$ be a game with root r, and let $\omega_-^1, \ldots, \omega_-^m$ be OMs with a nondeterministic interpretation. For $l \in \mathrm{L}(T)$, let $\mathsf{eval}(l) \coloneqq \{\mathrm{NBSM}(l) \times \vec{u}(l)\}$. Then $\mathtt{MiniMax}(\mathcal{P}_{<\infty}(\mathbb{R}^m), \mathsf{eval}, \cup, \oplus^m)$ satisfies*

$$\underline{v}_+ \coloneqq \max_{s_+ \in \Sigma_+^{\mathrm{P}}} \min_{1 \le j \le m} \mathcal{U}(s_+, \omega_-^j) = \max_{\vec{v} \in \mathsf{val}(r)} \min_{1 \le j \le m} v_j.$$

This algorithm is exponential time in the worst case; it can indeed be shown that this problem is NP-complete, even if MAX has perfect information (i.e. MIN only has 1 type) and there are only 2 OMs of MIN.

Compared to Proposition 2, the knowledge of OMs transforms MAX's incomplete information about MIN's *type* into their incomplete information about MIN's *strategy*. Situational values are now sets of vectors of length m (instead of t). Each such vector represents a strategy of MAX by its expected payoff against each OM. However, in contrast with probabilistic OMs, we cannot collapse each vector to a real number, since we have no distribution over the OMs. Still, reduction by weak dominance can be used just as for pure maxmin without any opponent model.

From another perspective, this algorithm computes the normal form of the game restricted to MIN's fixed m strategies, which intuitively justifies the correctness of Proposition 6 for pure maxmin.

As for mixed maxmin, one can modify the separation oracle in the LP algorithm of Koller and Megiddo (1992): now the oracle only computes MIN's best responses from the m OMs. This yields a polynomial-time algorithm.

6 Opponent Models with Uncertainty

We now come to our second contribution, about the case in which a set of OMs of MIN is available, but MAX is not certain that MIN will behave as one of them. We focus on the case with a single OM ω_-, which encompasses as well the case of several OMs with a probability distribution or lexicographic interpretation, as discussed in the last section.

We assume that with probability p^∞, which is known to MAX, MIN does not follow ω_-, in which case their behaviour is arbitrary and unpredictable; and with probability $1 - p^\infty$, MIN follows ω_-. Intuitively, p^∞ quantifies MAX's uncertainty about MIN's behaviour; this may arise for instance when the OMs are given by an estimate of MIN's gameplay level.

Formally, we define the following maxmin value:

$$\underline{v}_+ \coloneqq \max_{\varsigma_+ \in \Sigma_+} \Big((1 - p^\infty)\mathcal{U}(\varsigma_+, \omega_-) + p^\infty \min_{s_- \in \Sigma_-^{\mathrm{P}}} \mathcal{U}(\varsigma_+, s_-)\Big),$$

where Σ_+ is either Σ_+^{P} or Σ_+^{M}.

Example. *Consider again Figure 1 and the OM ω_- "MIN plays* a *if of type 1 or 2,* b *if of type 4 or 5, and $\frac{1}{2}\mathsf{a} + \frac{1}{2}\mathsf{b}$ if of type 3". The best strategy of MAX against ω_- is (l, R) with a payoff of 1. However, this strategy does not fare so well if MIN's strategy is not ω_- (or when p^∞ is close to 1): in the worst case, MIN plays* b *if of type 1 or 2, and* a *if of type 4 or 5. Against this strategy, MAX's expected payoff from playing (l, R) is only $1/5$. On the other hand, the pure maxmin strategy (l, L) only has a payoff of $1/2$ against ω_-, and so does the mixed maxmin strategy (which is the uniform strategy); hence neither is optimal when p^∞ is close to 0.*

It is clear from this example that the maxmin value and the optimal strategies depend on the value of p^∞. This demonstrates a conflict between robustness and performance: MAX desires to be cautious and robust against MIN's unpredictable behaviour occurring with probability p^∞, and at the same time to improve their performance by exploiting their knowledge of the OM, which correctly predicts MIN's strategy with probability $1 - p^\infty$.

We now show how to modify algorithms from the last sections to compute the maxmin value.

Mixed Maxmin For the mixed maxmin value, we can use the LP algorithm by Koller and Megiddo (1992) with a minor modification of the separation oracle: given a threshold v and a mixed strategy σ_+ for MAX, the separation oracle should now, apart from computing MAX's payoff v_{BR} with strategy σ_+ against MIN's best response, also compute MAX's payoff against the OM $v_{\mathsf{OM}} = \mathcal{U}(\sigma_+, \omega_-)$, then check whether $(1 - p^\infty)v_{\mathsf{OM}} + p^\infty v_{\mathsf{BR}} \ge v$ holds.

Example. *In the game in Figure 1 with ω_- as above, one can use this algorithm to verify that MAX's optimal strategy is (l, R) for $p^\infty \le 5/8$, otherwise it is the uniform strategy.*

This confirms that when nondeterministic behaviour happens with a small enough probability, it is worth deviating from maxmin strategies in order to exploit the OM.

Pure Maxmin For pure strategies, we build on the algorithm for a single OM (Proposition 3). To cope with non-locality (due to MIN's partially unpredictable behaviour), we use situational values that are finite sets of ordered pairs $\langle s, \vec{v}\rangle$, with $s \in \mathbb{R}$ and $\vec{v} \in \mathbb{R}^t$. We call such a pair an *annotated vector*; it implicitly represents a strategy for MAX for which the payoff against ω_- is s, and the worst payoff against MIN's unpredictable behaviour is given by $\vec{v}$.

We write $\mathcal{P}_{<\infty}(\mathbb{R} \times \mathbb{R}^t)$ for the set of all finite sets of annotated vectors, and for $f, g \in \mathcal{P}_{<\infty}(\mathbb{R} \times \mathbb{R}^t)$, we define $f \oplus^{1,t} g \subseteq \mathbb{R} \times \mathbb{R}^t$ to be the set

$$\{\langle s + s', (\min(v_i, v'_i))_{1\leq i\leq t}\rangle \mid \langle s, \vec{v}\rangle \in f, \langle s', \vec{v}\,'\rangle \in g\}.$$

Proposition 7. *Let $\langle T, \mathrm{P}, t, \vec{u}, \vec{\rho}\rangle$ be a VG with root r, ω_- be an OM, and $p^\infty \in [0, 1]$ a probability that MIN does not follow ω_-. For $l \in \mathrm{L}(T)$, let $\mathsf{eval}(l) := \{\langle \mathrm{NBS}(l, \omega_-) \cdot \vec{u}(l), \vec{u}(l)\rangle\} \in \mathcal{P}_{<\infty}(\mathbb{R} \times \mathbb{R}^t)$. Then $\mathtt{MiniMax}(\mathcal{P}_{<\infty}(\mathbb{R} \times \mathbb{R}^t), \mathsf{eval}, \cup, \oplus^{1,t})$ satisfies*

$$\underline{v}_+ = \max_{\langle s,\vec{v}\rangle\in\mathsf{val}(r)} \big((1 - p^\infty)s + p^\infty(\vec{\rho} \cdot \vec{v})\big).$$

Notice that when combining two annotated vectors at a MIN's node, the scalar part is additive; this reflects the fact that when following the (single) OM, MIN has no agency, just as in the case without uncertainty.

Example. *Using the algorithm above for the game in Figure 1 with the aforementioned OM ω_-, we find out that MAX's optimal strategy is* (l, R) *for $p^\infty \leq 5/7$, otherwise* (l, L) *or* (r, R). *Again, this indicates that it may be worth deviating from maxmin strategies so as to exploit an OM.*

Note that Proposition 7 generalises Proposition 2, which corresponds to the special case $p^\infty = 1$. Interestingly, the case $p^\infty \neq 1$ supports more sound pruning. Indeed, let $n \in \mathrm{N}(T)$ and $\langle s, \vec{v}\rangle, \langle s', \vec{v}\,'\rangle \in \mathsf{val}(n)$. Pruning $\langle s', \vec{v}\,'\rangle$ because of $\langle s, \vec{v}\rangle$ is sound if MAX is never worse-off in the game by choosing $\langle s, \vec{v}\rangle$ instead of $\langle s', \vec{v}\,'\rangle$ at n. Now since scalar parts are summed up, if $s > s'$ holds, then $\langle s, \vec{v}\rangle$ has an advantage $s - s'$ over $\langle s', \vec{v}\,'\rangle$; contrastingly, for the vectorial part, components for which $\vec{v}$ is larger than $\vec{v}\,'$ might be erased by the combination (via component-wise $\min$) of vectors at an ancestor of n, so that the advantage of $\vec{v}$ can be annihilated at the root. Hence, in the worst case, $\vec{v}\,'$ can keep all advantages it has compared to $\vec{v}$, while $\vec{v}$ can lose all its advantages.

To summarise, we can prune $\langle s', \vec{v}\,'\rangle$ when it holds that

$$(1 - p^\infty)(s - s') \geq p^\infty \sum_{1\leq i\leq t} \big(\rho_i \max(v'_i - v_i, 0)\big),$$

which indeed generalises the pruning condition for the algorithm in Proposition 2.

7 Application to Recursive Opponent Models

We now propose an application of the algorithms presented before to the computation of optimal strategies with recursive opponent models. We formulate a quite general setting, where various types of opponent models naturally arise.

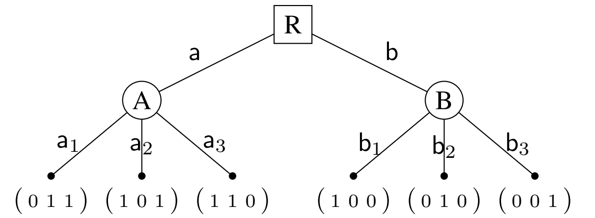


Figure 2: A game with 3 possible types of MAX.

Limitations of the Best-Defence Model In general, in a game with incomplete information, both players have incomplete information, rather than just MAX. As a result, the best-defence model usually gives MIN too much power.

Example. *Consider the game in Figure 2, where MAX has* 3 *types (with the uniform prior) and MIN has only* 1. *Then MIN has incomplete information. If MAX reasons according to the best-defence model, then both actions* a *and* b *have a value of* 0*: MAX of type i reasons that MIN will play* a_i *at node A, and* b_j *at node B for some $j \neq i$. The culprit is that MAX assumes MIN is aware of MAX's type so that MIN can adapt their strategy to MAX's type. However, if MAX realises MIN is unaware of their type, then MAX will prefer* a *since under the uniform common prior over MAX's types,* a *yields an expected payoff of* $2/3$, *compared to* b*'s* $1/3$.

On the other hand, computing maxmin strategies for the original game tree without using the best-defence model is not ideal either, for these strategies fail to exploit any assumption one may have about their adversary, such as that they have limited computational power or reasoning depth, or that they have a predictable behaviour. Such assumptions make sense in particular when playing against humans (Iida et al. 1993; Stahl and Wilson 1995; Dhami 2019).

Proposed Framework We propose a framework that can be considered as a generalisation of the cognitive hierarchy model (Camerer, Ho, and Chong 2004), and as a counterpart of interactive POMDPs (Doshi, Gmytrasiewicz, and Durfee 2020) for competitive games. The idea is to define *level-k* strategies to be the optimal strategies against an adversary of level $k - 1$, and recursively down to level 0. Our framework serves as a compromise between the best-defence model and the full game, and can be used to find better strategies against non-omnipotent and non-omniscient players; in particular, it generalises the best-defence model. Moreover, it can be used to explain real-life human psychological gameplay in games such as Bridge, as we illustrate at the end of this section.

We give a parametrisable definition of (1) how level-0 strategies are defined, (2) how optimal strategies at a given level are aggregated, and (3) how strategies of various levels are aggregated. Let Σ^0_+, Σ^0_- be non-empty sets of strategies of MAX and MIN. Moreover, for $i \in \{+, -\}$, let $\oplus_i : \mathcal{P}(\Sigma_i) \to \Sigma_i$ map any set of (pure or mixed) strategies of player i to a single strategy of player i, and $\mathrm{BR}_i : \Sigma^*_{-i} \to \mathcal{P}(\Sigma_i)$ map any tuple of strategies of player $-i$ to a set of strategies of player i. $\oplus$ will aggregate strategies of a player at a given level, and BR will compute the set of optimal strategies given a tuple of opponent models (one per lower level).[8]

Definition 8 (level-k strategies)**.** *Let Σ_i^0, $\oplus_i$, and BR_i be defined as above for all $i \in \{+,-\}$. The set of* level-0 *strategies for player i is defined to be Σ_i^0. For $k \geq 1$, the set of* level-k strategies *for player i, denoted by Σ_i^k, is defined to be $\mathrm{BR}_i\left(\oplus_{-i}(\Sigma_{-i}^{k-1}), \oplus_{-i}(\Sigma_{-i}^{k-2}), \dots, \oplus_{-i}(\Sigma_{-i}^{0})\right)$.*

In short, the level-k strategies of player i are the best responses (computed by BR_i, the *best-response function*) against an opponent using the strategy $\oplus_{-i}(\Sigma_{-i}^{k'})$ (computed by $\oplus_{-i}$, the *intra-level aggregation*) at each level k' for $0 \leq k' < k$. The level-0 strategies, are given by Σ_i^0, which can come from maxmin strategies under the best-defence model, randomly chosen strategies (McMahan, Gordon, and Blum 2003), modelling assumptions for human players (Wright and Leyton-Brown 2019), etc.

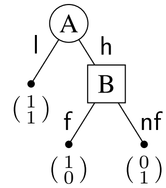
Example. *The Poisson-CH model in Camerer, Ho, and Chong (2004) is captured by choosing Σ_+^0 and Σ_-^0 to be the set of all pure strategies of MAX and MIN, the intra-level aggregation $\oplus$ to map any set of strategies to the uniform mixture of the set, and the best-response function BR to map a tuple of strategies $(\sigma_{-i}^{k-1}, \dots, \sigma_{-i}^{0})$ to the set of all pure best responses to the mixed strategy $p_{k-1}\sigma_{-i}^{k-1} + \dots + p_0\sigma_{-i}^0$, where $p_{k-1}, \dots, p_0$ follow some Poisson distribution.*

Although we do not show it here, this framework is also general enough to encompass many iterative approaches of solving games: iterative best response (also called best response dynamics); fictitious play (Brown 1951; Cloud, Wang, and Kerr 2023); double oracle (McMahan, Gordon, and Blum 2003); max^n or prob-max^n (Sturtevant, Zinkevich, and Bowling 2006); etc.

An interesting choice for intra-level aggregation $\oplus_i : \mathcal{P}(\Sigma_i) \to \Sigma_i$ for all i is the uniform mixture, as in the previous example. With this, many situations can be modelled by using different best-response functions BR for inter-level aggregation, in particular using the algorithms presented in previous sections:

- (Proposition 4) each player i at level k has a subjective distribution (described by $p_{i,k}^{k-1}, p_{i,k}^{k-2}, \dots, p_{i,k}^{0}$) over player $-i$'s reasoning levels, obtained for instance by fitting a model against a population of possible opponents;
- (Setting $p_{i,k}^{k-1} = 1$ for all i and k in the previous model) each player at level k assumes their opponent reasons at level exactly $k-1$;
- (Proposition 5) player i at level k assumes $-i$ to reason at level $k-1$, tie-breaks equivalent strategies by assuming them to reason at level $k-2$, and so on;
- (Proposition 6) player i at level k assumes $-i$ to reason at an unknown level lower than k (then the incomplete information about $-i$'s type becomes one about their level, which is, in general, much smaller);

[8]The framework could be easily adapted to more general functions, e.g. an aggregation of the strategies at the same level into a set or a tuple of strategies. It could also be easily applied to general games, in normal form or extensive form, beyond the two-player and zero-sum assumptions.



k	MAX	MIN 1	MIN 2
0	{nf}	{l, h}	{l, h}
1	{nf}	{h}	{l, h}
2	{f}	{h}	{l, h}
3	{f}	{h}	{h}

Figure 3: Level-k strategies in a VG with 2 types of MIN.

- (Proposition 7) with some probability, opponent $-i$ does not reason at any level lower than k.

Moreover, a straightforward generalisation of this framework allows players in multiplayer games to take into account the incomplete information of their partners, akin to interactive POMDPs.

A Real-Life Example We now give an example application of our formalism, which captures the psychological strategies of a contract bridge deal played in a bridge tournament. We present the abstract version of the game in Figure 3 (left); for the bridge deal itself, see Karpin (1977, p. 266).

In this game, the common prior about MIN's types is given by $p_1 = 0.4$ and $p_2 = 0.6$. For the recursive reasoning, $\oplus$ is given by the uniform mixture, BR is given by the lexicographic model, and the level-0 strategies for both players are their pure maxmin strategies. Figure 3 (right) shows the level-k strategies for MAX, MIN if of type 1, and MIN if of type 2, and for $k = 0, \dots, 3$. For instance, if MIN is of type 2, then their level-1 strategies are l and h.

As it turns out, this recursive reasoning captures perfectly what happened during the bridge deal, where MAX was at level 2 and therefore chose f (rather than the maxmin strategy nf) while MIN, being of type 2, reasoned at level 3 and used strategy h to defeat MAX. Another interesting point is that level-k strategies are not necessarily weakly dominant; hence they incorporate some notion of risk. For instance, MAX's level-2 strategy f performs better than the maxmin strategy nf against MIN of level 1 (it pays 0.7 instead of 0.6), but it performs worse against MIN of level 3 (0.4).

8 Conclusion

We have studied the algorithmic aspects of computing robust strategies in games with incomplete information, when opponent models with various interpretations are given, and proposed a framework for recursive modelling of opponents. Our algorithms are exact, and pave the way for optimisations like $\alpha\beta$ pruning, bounded depth, sampling, etc.

An interesting perspective is to broaden the applicability of our algorithms. For this, we would like to consider other opponent models than explicit behaviour strategies, and the case in which the amount of uncertainty p^∞ is only known up to some precision ε; for this latter point, we conjecture that the algorithms could benefit from the maxmin value being piecewise linear in p^∞. A second perspective is to go beyond the explicit extensive form, and to consider compact representations of games. Finally, an interesting perspective is to give an epistemic logic account of our recursive framework, in the spirit of the notion of rationalisability.

References

Albrecht, S. V.; and Stone, P. 2018. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artif. Intell.*, 258: 66–95.

Bosanský, B.; Kiekintveld, C.; Lisý, V.; and Pechoucek, M. 2014. An Exact Double-Oracle Algorithm for Zero-Sum Extensive-Form Games with Imperfect Information. *J. Artif. Intell. Res.*, 51: 829–866.

Brown, G. W. 1951. Iterative solution of games by fictitious play. *Activity Analysis of Production and Allocation*, 13(1): 374–376.

Camerer, C. F.; Ho, T.-H.; and Chong, J.-K. 2004. A Cognitive Hierarchy Model of Games. *The Quarterly Journal of Economics*, 119(3): 861–898.

Cloud, A.; Wang, A.; and Kerr, W. 2023. Anticipatory Fictitious Play. In *Proc. Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI 2023)*, 73–81. ijcai.org.

de Weerd, H.; Verbrugge, R.; and Verheij, B. 2013. How much does it help to know what she knows you know? An agent-based simulation study. *Artif. Intell.*, 199-200: 67–92.

Dhami, S. 2019. *The Foundations of Behavioral Economic Analysis Volume IV: Behavioral Game Theory*. London, England: Oxford University Press.

Doshi, P.; Gmytrasiewicz, P. J.; and Durfee, E. H. 2020. Recursively modeling other agents for decision making: A research perspective. *Artif. Intell.*, 279.

Frank, I.; and Basin, D. A. 2001. A theoretical and empirical investigation of search in imperfect information games. *Theor. Comput. Sci.*, 252(1-2): 217–256.

Ginsberg, M. L. 2001. GIB: Imperfect Information in a Computationally Challenging Game. *J. Artif. Intell. Res.*, 14: 303–358.

Gmytrasiewicz, P. J.; and Doshi, P. 2005. A Framework for Sequential Planning in Multi-Agent Settings. *J. Artif. Intell. Res.*, 24: 49–79.

Iida, H.; Uiterwijk, J. W. H. M.; van den Herik, H. J.; and Herschberg, I. S. 1993. Potential Applications of Opponent-Model Search, Part 1: The Domain of Applicability. *J. Int. Comput. Games Assoc.*, 16(4): 201–208.

Iida, H.; Uiterwijk, J. W. H. M.; van den Herik, H. J.; and Herschberg, I. S. 1994. Potential Applications of Opponent-Model Search, Part 2: Risks and Strategies. *J. Int. Comput. Games Assoc.*, 17(1): 10–14.

Karpin, F. L. 1977. *Psychological strategy in contract bridge: The techniques of deception and harassment in bidding and play*. Dover Publications.

Koller, D.; and Megiddo, N. 1992. The complexity of two-person zero-sum games in extensive form. *Games and Economic Behavior*, 4(4): 528–552.

Koller, D.; Megiddo, N.; and von Stengel, B. 1996. Efficient Computation of Equilibria for Extensive Two-Person Games. *Games and Economic Behavior*, 14(2): 247–259.

Kuhn, H. W. 1953. *11. Extensive Games and the Problem of Information*, 193–216. Princeton: Princeton University Press. ISBN 9781400881970.

Li, J.; Zanuttini, B.; Cazenave, T.; and Ventos, V. 2022. Generalisation of Alpha-Beta Search for AND-OR Graphs With Partially Ordered Values. In *Proc. Thirty-First International Joint Conference on Artificial Intelligence (IJCAI 2022)*, 4769–4775. ijcai.org.

McMahan, H. B.; Gordon, G. J.; and Blum, A. 2003. Planning in the Presence of Cost Functions Controlled by an Adversary. In Fawcett, T.; and Mishra, N., eds., *Proc. Twentieth International Conference on Machine Learning (ICML 2003)*, 536–543. AAAI Press.

Nashed, S. B.; and Zilberstein, S. 2022. A Survey of Opponent Modeling in Adversarial Domains. *J. Artif. Intell. Res.*, 73: 277–327.

Perea, A. 2012. *Epistemic Game Theory: Reasoning and Choice*. Cambridge University Press.

Rebstock, D.; Solinas, C.; Buro, M.; and Sturtevant, N. R. 2019. Policy Based Inference in Trick-Taking Card Games. In *Proc. IEEE Conference on Games (CoG 2019)*, 1–8. IEEE.

Stahl, D. O.; and Wilson, P. W. 1995. On Players' Models of Other Players: Theory and Experimental Evidence. *Games and Economic Behavior*, 10(1): 218–254.

Sturtevant, N. R.; and Bowling, M. H. 2006. Robust game play against unknown opponents. In *Proc. Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006)*, 713–719. ACM.

Sturtevant, N. R.; Zinkevich, M.; and Bowling, M. H. 2006. Prob-Maxn: Playing N-Player Games with Opponent Models. In *Proc. Twenty-First National Conference on Artificial Intelligence (AAAI 2006)*, 1057–1063. AAAI Press.

von Stengel, B. 1996. Efficient Computation of Behavior Strategies. *Games and Economic Behavior*, 14(2): 220–246.

Wright, J. R.; and Leyton-Brown, K. 2019. Level-0 Models for Predicting Human Behavior in Games. *J. Artif. Intell. Res.*, 64: 357–383.

You, Y.; Thomas, V.; Colas, F.; Alami, R.; and Buffet, O. 2023. Robust Robot Planning for Human-Robot Collaboration. In *Proc. IEEE International Conference on Robotics and Automation (ICRA 2023)*, 9793–9799. IEEE.