The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)

Solving Large Extensive-Form Games with Strategy Constraints

Trevor Davis,[1] **Kevin Waugh,**[2] **Michael Bowling**[2,1]
[1]Department of Computing Science, University of Alberta
[2]DeepMind
trdavis1@ualberta.ca, {waughk, bowlingm}@google.com

Abstract

Extensive-form games are a common model for multiagent interactions with imperfect information. In two-player zero-sum games, the typical solution concept is a Nash equilibrium over the unconstrained strategy set for each player. In many situations, however, we would like to constrain the set of possible strategies. For example, constraints are a natural way to model limited resources, risk mitigation, safety, consistency with past observations of behavior, or other secondary objectives for an agent. In small games, optimal strategies under linear constraints can be found by solving a linear program; however, state-of-the-art algorithms for solving large games cannot handle general constraints. In this work we introduce a generalized form of Counterfactual Regret Minimization that provably finds optimal strategies under any feasible set of convex constraints. We demonstrate the effectiveness of our algorithm for finding strategies that mitigate risk in security games, and for opponent modeling in poker games when given only partial observations of private information.

1 Introduction

Multiagent interactions are often modeled using *extensive-form games* (EFGs), a powerful framework that incoporates sequential actions, hidden information, and stochastic events. Recent research has focused on computing approximately optimal strategies in large extensive-form games, resulting in a solution to heads-up limit Texas Hold'em, a game with approximately 10^{17} states (Bowling et al. 2015), and in two independent super-human computer agents for the much larger heads-up no-limit Texas Hold'em (Moravčík et al. 2017; Brown and Sandholm 2018).

When modeling an interaction with an EFG, for each outcome we must specify the agents' utility, a cardinal measure of the outcome's desirability. Utility is particularly difficult to specify. Take, for example, situations where an agent has multiple objectives to balance: a defender in a security game with the primary objective of protecting a target and a secondary objective of minimizing expected cost, or a robot operating in a dangerous environment with a primary task to complete and a secondary objective of minimizing damage to itself and others. How these objectives combine into a single value, the agent's utility, is ill-specified and error prone.

One approach for handling multiple objectives is to use a linear combination of per-objective utilities. This approach has been used in EFGs to "tilt" poker agents toward taking specific actions (Johanson et al. 2011), and to mix between cost minimization and risk mitigation in sequential security games (Lisý, Davis, and Bowling 2016). However, objectives are typically measured on incommensurable scales. This leads to dubious combinations of weights often selected by trial-and-error.

A second approach is to constrain the agents' strategy spaces directly. For example, rather than minimizing the expected cost, we use a hard constraint that disqualifies high-cost strategies. Using such constraints has been extensively studied in single-agent perfect information settings (Altman 1999) and partial information settings (Isom, Meyn, and Braatz 2008; Santana, Thiébaux, and Williams 2016), as well as in (non-sequential) security games (Brown et al. 2014).

Incorporating strategy constraints when solving EFGs presents a unique challenge. Nash equilibria can be found by solving a linear program (LP) derived using the sequence-form representation (Koller, Megiddo, and von Stengel 1996). This LP is easily modified to incorporate linear strategy constraints; however, LPs do not scale to large games. Specialized algorithms for efficiently solving large games, such as an instantiation of Nesterov's *excessive gap technique* (EGT) (Hoda et al. 2010) as well as *counterfactual regret minimization* (CFR) (Zinkevich et al. 2008) and its variants (Lanctot et al. 2009; Tammelin et al. 2015), cannot integrate arbitrary strategy constraints directly. Currently, the only large-scale approach is restricted to constraints that consider only individual decisions (Farina, Kroer, and Sandholm 2017).

In this work we present the first scalable algorithm for solving EFGs with arbitrary convex strategy constraints. Our algorithm, Constrained CFR, provably converges towards a strategy profile that is minimax optimal under the given constraints. It does this while retaining the $\mathcal{O}(1/\sqrt{T})$ convergence rate of CFR and requiring additional memory proportional to the number of constraints. We demonstrate the empirical effectiveness of Constrained CFR by comparing its solution to that of an LP solver in a security game. We also present a novel constraint-based technique for opponent modeling with partial observations in a small poker game.

2 Background

Formally, an extensive-form game (Osborne and Rubinstein 1994) is a game tree defined by:

- A set of *players* N. This work focuses on games with two players, so $N = \{1, 2\}$.
- A set of *histories* H, the tree's nodes rooted at $\emptyset$. The leafs, $Z \subseteq H$, are *terminal histories*. For any history $h \in H$, we let $h' \sqsubset h$ denote a prefix h' of h, and necessarily $h' \in H$.
- For each $h \in H \setminus Z$, a set of *actions* $A(h)$. For any $a \in A(h)$, $ha \in H$ is a child of h.
- A *player function* $P\colon H \setminus Z \to N \cup \{c\}$ defining the player to act at h. If $P(h) = c$ then *chance* acts according to a known probability distribution $\sigma_c(h) \in \Delta_{|A(h)|}$, where $\Delta_{|A(h)|}$ is the probability simplex of dimension $|A(h)|$.
- A set of *utility functions* $u_i\colon Z \to \mathbb{R}$, for each player. Outcome z has utility $u_i(z)$ for player i. We assume the game is *zero-sum*, i.e., $u_1(z) = -u_2(z)$. Let $u(z) = u_1(z)$.
- For each player $i \in N$, a collection of *information sets* $\mathcal{I}_i$. $\mathcal{I}_i$ partitions H_i, the histories where i acts. Two histories h, h' in an information set $I \in \mathcal{I}_i$ are indistinguishable to i. Necessarily $A(h) = A(h')$, which we denote by $A(I)$. When a player acts they do not observe the history, only the information set it belongs to, which we denote as $I[h]$.

We assume a further requirement on the information sets $\mathcal{I}_i$ called *perfect recall*. It requires that players are never forced to forget information they once observed. Mathematically this means that all indistinguishable histories share the same sequence of past information sets and actions for the actor. Although this may seem like a restrictive assumption, some perfect recall-like condition is needed to guarantee that an EFG can be solved in polynommial time, and all sequential games played by humans exhibit perfect recall.

2.1 Strategies

A *behavioral strategy* for player i maps each information set $I \in \mathcal{I}_i$ to a distribution over actions, $\sigma_i(I) \in \Delta_{|A(I)|}$. The probability assigned to $a \in A(I)$ is $\sigma_i(I, a)$. A *strategy profile*, $\sigma = \{\sigma_1, \sigma_2\}$, specifies a strategy for each player. We label the strategy of the opponent of player i as σ_{-i}. The sets of behavioral strategies and strategy profiles are Σ_i and Σ respectively.

A strategy profile uniquely defines a *reach probability* for any history $h \in H$:

$$\pi^\sigma(h) \coloneqq \prod_{h'a \sqsubseteq h} \sigma_{P(h')}(I[h'], a) \tag{1}$$

This product decomposes into contributions from each player and chance, $\pi_1^{\sigma_1}(h)\pi_2^{\sigma_2}(h)\pi_c(h)$. For a player $i \in N$, we denote the contributions from the opponent and chance as $\pi_{-i}^{\sigma_{-i}}(h)$ so that $\pi^\sigma(h) = \pi_i^{\sigma_i}(h)\pi_{-i}^{\sigma_{-i}}(h)$. By perfect recall we have $\pi_i^{\sigma_i}(h) = \pi_i^{\sigma_i}(h')$ for any h, h' in same information set $I \in \mathcal{I}_i$. We thus also write this probability as $\pi_i^{\sigma_i}(I)$.

Given a strategy profile $\sigma = \{\sigma_1, \sigma_2\}$, the expected utility for player i is given by

$$u_i(\sigma) = u_i(\sigma_1, \sigma_2) \coloneqq \sum_{z \in Z} \pi^\sigma(z) u_i(z). \tag{2}$$

A strategy σ_i is an *ε-best response* to the opponent's strategy σ_{-i} if $u_i(\sigma_i, \sigma_{-i}) + \varepsilon \geq u_i(\sigma_i', \sigma_{-i})$ for any alternative strategy $\sigma_i' \in \Sigma_i$. A strategy profile is an *ε-Nash equilibrium* when each σ_i is a ε-best response to its opponent; such a profile exists for any $\varepsilon \geq 0$. The *exploitability* of a strategy profile is the smallest $\varepsilon = {}^1\!/_2(\varepsilon_1 + \varepsilon_2)$ such that each σ_i is an ε_i-best response. Due to the zero-sum property, the game's Nash equilibria are the saddle-points of the minimax problem

$$\max_{\sigma_1 \in \Sigma_1} \min_{\sigma_2 \in \Sigma_2} u(\sigma_1, \sigma_2) = \min_{\sigma_2 \in \Sigma_2} \max_{\sigma_1 \in \Sigma_1} u(\sigma_1, \sigma_2). \tag{3}$$

A zero-sum EFG can be represented in *sequence form* (von Stengel 1996). The sets of sequence-form strategies for players 1 and 2 are $\mathcal{X}$ and $\mathcal{Y}$ respectively. A sequence-form strategy $\mathbf{x} \in \mathcal{X}$ is a vector indexed by pairs $I \in \mathcal{I}_1, a \in A(I)$. The entry $\mathbf{x}_{(I,a)}$ is the probability of player 1 playing the sequence of actions that reaches I and then playing action a. A special entry, $\mathbf{x}_\emptyset = 1$, represents the empty sequence. Any behavioral strategy $\sigma_1 \in \Sigma_1$ has a corresponding sequence-form strategy $\text{SEQ}(\sigma_1)$ where

$$\text{SEQ}(\sigma_1)_{(I,a)} \coloneqq \pi_1^{\sigma_1}(I)\sigma_1(I, a). \qquad \forall I \in \mathcal{I}_i, a \in A(I)$$

Player i has a unique sequence to reach any history $h \in H$ and, by perfect recall, any information set $I \in \mathcal{I}_i$. Let $\mathbf{x}_h$ and $\mathbf{x}_I$ denote the corresponding entries in $\mathbf{x}$. Thus, we are free to write the expected utility as $u(\mathbf{x}, \mathbf{y}) = \sum_{z \in Z} \pi_c(z)\mathbf{x}_z\mathbf{y}_z u(z)$. This is bilinear, i.e., there exists a *payoff matrix* $\mathbf{A}$ such that $u(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\intercal \mathbf{A} \mathbf{y}$. A consequence of perfect recall and the laws of probability is for $I \in \mathcal{I}_1$ that $\mathbf{x}_I = \sum_{a \in A(I)} \mathbf{x}_{(I,a)}$ and that $\mathbf{x} \geq 0$. These constraints are linear and completely describe the polytope of sequence-form strategies. Using these together, (3) can be expressed as a bilinear saddle point problem over the polytopes $\mathcal{X}$ and $\mathcal{Y}$:

$$\max_{\mathbf{x} \in \mathcal{X}} \min_{\mathbf{y} \in \mathcal{Y}} \mathbf{x}^\intercal \mathbf{A} \mathbf{y} = \min_{\mathbf{y} \in \mathcal{Y}} \max_{\mathbf{x} \in \mathcal{X}} \mathbf{x}^\intercal \mathbf{A} \mathbf{y} \tag{4}$$

For a convex function $f : \mathcal{X} \to \mathbb{R}$, let $\nabla f(\mathbf{x})$ be any element of the subdifferential $\partial f(\mathbf{x})$, and let $\nabla_{(I,a)} f(\mathbf{x})$ be the (I, a) element of this subgradient.

2.2 Counterfactual regret minimization

Counterfactual regret minimization (Zinkevich et al. 2008) is a large-scale equilibrium-finding algorithm that, in self-play, iteratively updates a strategy profile in a fashion that drives its *counterfactual regret* to zero. This regret is defined in terms of *counterfactual values*. The counterfactual value of reaching information set I is the expected payoff under the counterfactual that the acting player attempts to reach it:

$$v(I, \sigma) = \sum_{h \in I} \pi_{-i}^{\sigma_{-i}}(h) \sum_{z \in Z} \pi^\sigma(h, z) u(z) \tag{5}$$

Here $i = P(h)$ for any $h \in I$, and $\pi^\sigma(h, z)$ is the probability of reaching z from h under σ. Let $\sigma_{I \to a}$ be the profile that plays a at I and otherwise plays according to σ. For a series of profiles $\sigma^1, .., \sigma^T$, the *average counterfactual regret* of action a at I is $R^T(I, a) = \frac{1}{T} \sum_{t=1}^T v(I, \sigma_{I \to a}^t) - v(I, \sigma^t)$.

To minimize counterfactual regret, CFR employs *regret matching* (Hart and Mas-Colell 2000). In particular, actions

are chosen in proportion to positive regret, $\sigma^{t+1}(I,a) \propto (R^t(I,a))^+$ where $(x)^+ = \max(x,0)$. It follows that the average strategy profile $\overline{\sigma}^T$, defined by $\overline{\sigma}_i^T(I,a) \propto \sum_{t=1}^T \pi_i^{\sigma^t}(I)\sigma_i^t(I,a)$, is an $\mathcal{O}(1/\sqrt{T})$-Nash equilibrium (Zinkevich et al. 2008). In sequence form, the average is given by $\overline{\mathbf{x}}^{\mathbf{T}} = \frac{1}{T}\sum_{t=1}^T \mathbf{x}^{\mathbf{t}}$.

3 Solving games with strategy constraints

We begin by formally introducing the constrained optimization problem for extensive-form games. We specify convex constraints on the set of sequence-form strategies[1] $\mathcal{X}$ with a set of k convex functions $f_i \colon \mathcal{X} \to \mathbb{R}$ where we require $f_i(\mathbf{x}) \leq 0$ for each $i = 1, ..., k$. We use constraints on the sequence form instead of on the behavioral strategies because reach probabilities and utilities are linear functions of a sequence-form strategy, but not of a behavioral strategy.

The optimization problem can be stated as:

$$\max_{\mathbf{x}\in\mathcal{X}} \min_{\mathbf{y}\in\mathcal{Y}} \mathbf{x}^\intercal \mathbf{A}\mathbf{y} \qquad \text{subject to} \qquad f_i(\mathbf{x}) \leq 0 \qquad \text{for } i = 1, ..., k \tag{6}$$

The first step toward solving this problem is to incorporate the constraints into the objective with Lagrange multipliers. If the problem is feasible (i.e., there exists a feasible $\mathbf{x}_f \in \mathcal{X}$ such that $f_i(\mathbf{x}_f) \leq 0$ for each i), then (6) is equivalent to:

$$\max_{\mathbf{x}\in\mathcal{X}} \min_{\substack{\mathbf{y}\in\mathcal{Y} \\ \boldsymbol{\lambda}\geq\mathbf{0}}} \mathbf{x}^\intercal \mathbf{A}\mathbf{y} - \sum_{i=1}^k \lambda_i f_i(\mathbf{x}) \tag{7}$$

We will now present intuition as to how CFR can be modified to solve (7), before presenting the algorithm and proving its convergence.

3.1 Intuition

CFR can be seen as doing a saddle point optimization on the objective in (3), using the gradients[2] of $g(\mathbf{x},\mathbf{y}) = \mathbf{x}^\intercal\mathbf{A}\mathbf{y}$ given as

$$\nabla_{\mathbf{x}} g(\mathbf{x}^\mathbf{t}, \mathbf{y}^\mathbf{t}) = \mathbf{A}\mathbf{y}^\mathbf{t} \qquad \nabla_{\mathbf{y}} -g(\mathbf{x}^\mathbf{t}, \mathbf{y}^\mathbf{t}) = -(\mathbf{x}^\mathbf{t})^\intercal\mathbf{A}. \tag{8}$$

The intuition behind our modified algorithm is to perform the same updates, but with gradients of the modified utility function

$$h(\mathbf{x},\mathbf{y},\boldsymbol{\lambda}) = \mathbf{x}^\intercal\mathbf{A}\mathbf{y} - \sum_{i=1}^k \lambda_i f_i(\mathbf{x}). \tag{9}$$

The (sub)gradients we use in the modified CFR update are then

$$\begin{aligned} \nabla_{\mathbf{x}} h(\mathbf{x}^\mathbf{t}, \mathbf{y}^\mathbf{t}, \boldsymbol{\lambda}^\mathbf{t}) &= \mathbf{A}\mathbf{y}^\mathbf{t} - \sum_{i=1}^k \lambda_i^t \nabla f_i(\mathbf{x}^\mathbf{t}) \\ \nabla_{\mathbf{y}} -h(\mathbf{x}^\mathbf{t}, \mathbf{y}^\mathbf{t}, \boldsymbol{\lambda}^\mathbf{t}) &= -(\mathbf{x}^\mathbf{t})^\intercal\mathbf{A}. \end{aligned} \tag{10}$$

[1] Without loss of generality, we assume throughout this paper that the constrained player is player 1, i.e. the maximizing player.

[2] For a more complete discussion of the connection between CFR and gradient ascent, see (Waugh and Bagnell 2015).

Note that this leaves the update of the unconstrained player unchanged. In addition, we must update $\boldsymbol{\lambda}^\mathbf{t}$ using the gradients $\nabla_{\boldsymbol{\lambda}} -h(\mathbf{x}^\mathbf{t}, \mathbf{y}^\mathbf{t}, \boldsymbol{\lambda}^\mathbf{t}) = \sum_{i=1}^k f_i(\mathbf{x}^\mathbf{t})\mathbf{e_i}$, which is the k-vector with $f_i(\mathbf{x}^\mathbf{t})$ at index i. This can be done with any gradient method, e.g. simple gradient ascent with the update rule

$$\lambda_i^{t+1} = \max(\lambda_i^t + \alpha^t f_i(\mathbf{x}^\mathbf{t}), 0) \tag{11}$$

for some step size $\alpha^t \propto 1/\sqrt{t}$.

3.2 Constrained counterfactual regret minimization

We give the *Constrained CFR* (CCFR) procedure in Algorithm 1. The constrained player's strategy is updated with the function CCFR and the unconstrained player's strategy is updated with unmodified CFR. In this instantiation $\boldsymbol{\lambda}^\mathbf{t}$ is updated with gradient ascent, though any regret minimizing update can be used. We clamp each λ_i^t to the interval $[0, \beta]$ for reasons discussed in the following section. Together, these updates form a full iteration of CCFR.

Algorithm 1 Constrained CFR

```
1:  function CCFR(σ_i^t, σ_{-i}^t, λ^t)
2:      for I ∈ I_i do                          ▷ in reverse topological order
3:          for a ∈ A(I) do
4:              v^t(I,a) ← Σ_{z∈Z^1[Ia]} π_{-i}^{σ^t_{-i}}(z) u(z)
                           + Σ_{I'∈succ(I,a)} ṽ^t(I')
5:              ṽ^t(I,a) ← v^t(I,a)
                           − Σ_{i=1}^k λ_i^t ∇_{(I,a)} f_i(SEQ(σ_i^t))
6:          end for
7:          ṽ^t(I) ← Σ_{a∈A(I)} σ_i^t(I,a) ṽ^t(I,a)
8:          for a ∈ A(I) do
9:              r̃^t(I,a) ← ṽ^t(I,a) − ṽ^t(I)
10:             R̃^t(I,a) ← R̃^{t−1}(I,a) + r̃^t(I,a)
11:         end for
12:         for a ∈ A(I) do
13:             σ_i^{t+1}(I,a) ← (R̃^t(I,a))^+ / Σ_{b∈A(I)} (R̃^t(I,b))^+
14:         end for
15:     end for
16:     return σ_i^{t+1}
17: end function

18: for t = 1, ..., T do
19:     σ_2^t ← CFR(σ_1^{t−1}, σ_2^{t−1})
20:     for i = 1, ..., k do
21:         λ_i^t ← λ_i^{t−1} + α_t f_i(Ψ(σ_1^{t−1}))
22:         λ_i^t ← CLAMP(λ_i^t, [0, B])
23:     end for
24:     σ_1^t ← CCFR(σ_1^{t−1}, σ_2^t, λ^t)
25:     σ̄_2^t ← ((t−1)/t) σ̄_2^{t−1} + (1/t) σ_2^t
26:     σ̄_1^t ← ((t−1)/t) σ̄_1^{t−1} + (1/t) σ_1^t
27: end for
```

The CCFR update for the constrained player is the same as the CFR update, with the crucial difference of line 5, which

incorporates the second part of the gradient $\nabla_{\mathbf{x}} h$ into the counterfactual value $v^t(I, a)$. The loop beginning on line 2 goes through the constrained player's information sets, walking the tree bottom-up from the leafs. The counterfactual value $v^t(I, a)$ is set on line 4 using the values of terminal states $Z^1[Ia]$ which directly follow from action a at I (this corresponds to the $\mathbf{A}\mathbf{y}^\mathbf{t}$ term of the gradient), as well as the already computed values of successor information sets $succ(I, a)$. Line 7 computes the value of the current information set using the current strategy. Lines 9 and 10 update the stored regrets for each action. Line 13 updates the current strategy with regret matching.

3.3 Theoretical analysis

In order to ensure that the utilities passed to the regret matching update are bounded, we will require $\boldsymbol{\lambda}^\mathbf{t}$ to be bounded from above; in particular, we will choose $\boldsymbol{\lambda}^\mathbf{t} \in [0, \beta]^k$. We can then evaluate the chosen sequence $\boldsymbol{\lambda}^\mathbf{1}, ..., \boldsymbol{\lambda}^\mathbf{T}$ using its regret in comparison to the optimal $\boldsymbol{\lambda}^* \in [0, \beta]^k$:

$$R_\lambda^T(\beta) \coloneqq \max_{\lambda^* \in [0,\beta]^k} \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{k} \big[\lambda_i^* f_i(\text{SEQ}(\sigma_1^t)) - \lambda_i^t f_i(\text{SEQ}(\sigma_1^t))\big]. \tag{12}$$

We can guarantee $R_\lambda^T(\beta) = \mathcal{O}(1/\sqrt{T})$, e.g. by choosing $\boldsymbol{\lambda}^\mathbf{t}$ with projected gradient ascent (Zinkevich 2003).

We now present the theorems which show that CCFR can be used to approximately solve (6). In the following thereoms we assume that $T \in \mathbb{N}$, we have some convex, continuous constraint functions $f_1, ..., f_k$, and we use some regret-minimizing method to select the vectors $\boldsymbol{\lambda}^1, ..., \boldsymbol{\lambda}^T$ each in $[0, \beta]^k$ for some $\beta \geq 0$.

First, we show that the exploitability of the average strategies approaches the optimal value:

Theorem 1. *If CCFR is used to select the sequence of strategies $\sigma_1^1, ..., \sigma_1^T$ and CFR is used to select the sequence of strategies $\sigma_2^1, ..., \sigma_2^T$, then the following holds:*

$$\max_{\substack{\sigma_1^* \in \Sigma_1 \\ \text{s.t. } f_i(\sigma_1^*) \leq 0 \ \forall i}} u(\sigma_1^*, \overline{\sigma}_2^T) - \min_{\sigma_2^* \in \Sigma_2} u(\overline{\sigma}_1^T, \sigma_2^*) \leq \frac{4\,(\Delta_u + k\beta F)\, M \sqrt{|A|}}{\sqrt{T}} + 2R_\lambda^T(\beta) \tag{13}$$

where $\Delta_u = \max_z u(z) - \min_z u(z)$ is the range of possible utilities, $|A|$ is the maximum number of actions at any information set, k is the number of constraints, $F = \max_{\mathbf{x},i} ||\nabla f_i(\mathbf{x})||_1$ is a bound on the subgradients[3], and M is a game-specific constant.

All proofs are given in the supplementary materials. Theorem 1 guarantees that the constrained exploitability of the final CCFR strategy profile converges to the minimum exploitability possible over the set of feasible profiles, at a rate of $\mathcal{O}(1/\sqrt{T})$ (assuming a suitable regret minimizer is used to select $\boldsymbol{\lambda}^t$).

[3]Such a bound must exist as the strategy sets are compact and the constraint functions are continuous.

In order to establish that CCFR approximately solves optimization (6), we must also show that the CCFR strategies converge to being feasible. In the case of arbitrary $\beta \geq 0$:

Theorem 2. *If CCFR is used to select the sequence of strategies $\sigma_1^1, ..., \sigma_1^T$ and CFR is used to select the sequence of strategies $\sigma_2^1, ..., \sigma_2^T$, then the following holds:*

$$f_i(\textit{SEQ}(\overline{\sigma}_1^T)) \leq \frac{R_\lambda^T(\beta)}{\beta} + \frac{(\Delta_u + 2k\beta F)\, M \sqrt{|A|}}{\beta\sqrt{T}} + \frac{\Delta_u}{\beta} \quad \forall i \in \{1, ..., k\} \tag{14}$$

This theorem guarantees that the CCFR strategy converges to the feasible set at a rate of $\mathcal{O}(1/\sqrt{T})$, up to an approximation error of Δ_u/β induced by the bounding of $\boldsymbol{\lambda}^\mathbf{t}$.

We can eliminate the approximation error when β is chosen large enough for some optimal $\boldsymbol{\lambda}^*$ to lie within the bounded set $[0, \beta]^k$. In order to establish the existence of such a $\boldsymbol{\lambda}^*$, we must assume a constraint qualification such as Slater's condition, which requires the existance of a feasible $\mathbf{x}$ which strictly satisfies any nonlinear constraints ($f_i(\mathbf{x}) \leq 0$ for all i and $f_j(\mathbf{x}) < 0$ for all nonlinear f_j). Then there exists a finite $\boldsymbol{\lambda}^*$ which is a solution to optimization (7), which we can use to give the bound:

Theorem 3. *Assume that $f_1, ..., f_k$ satisfy a constraint qualification such as Slater's condition, and define $\boldsymbol{\lambda}^*$ to a finite solution for $\boldsymbol{\lambda}$ in the resulting optimization (7). Then if β is chosen such that $\beta > \lambda_i^*$ for all i, and CCFR and CFR are used to respectively select the strategy sequences $\sigma_1^1, ..., \sigma_1^T$ and $\sigma_2^1, ..., \sigma_2^T$, the following holds:*

$$f_i(\textit{SEQ}(\overline{\sigma}_1^T)) \leq \frac{R_\lambda^T(\beta)}{\beta - \lambda_i^*} + \frac{2\,(\Delta_u + k\beta F)\, M \sqrt{|A|}}{(\beta - \lambda_i^*)\sqrt{T}} \quad \forall i \in \{1, ..., k\} \tag{15}$$

In this case, the CCFR strategy converges fully to the feasible set, at a rate of $\mathcal{O}(1/\sqrt{T})$, given a suitable choice of regret minimizer for $\boldsymbol{\lambda}^t$. We provide an explicit example of such a minimizer in the following corollary:

Corollary 3.1. *If the conditions of Theorem 3 hold and, in addition, the sequence $\boldsymbol{\lambda}^1, ..., \boldsymbol{\lambda}^T$ is chosen using projected gradient descent with constant learning rate $\alpha^t = \beta/(G\sqrt{T})$ where $G = \max_{i,\mathbf{x}} f_i(\mathbf{x})$, then the following hold:*

$$f_i(\textit{SEQ}(\overline{\sigma}_1^T)) \leq \frac{\beta G + 2\,(\Delta_u + k\beta F)\, M \sqrt{|A|}}{(\beta - \lambda_i^*)\sqrt{T}} \quad \forall i \in \{1, ..., k\} \tag{16}$$

$$\max_{\substack{\sigma_1^* \in \Sigma_1 \\ \text{s.t. } f_i(\sigma_1^*) \leq 0 \ \forall i}} u(\sigma_1^*, \overline{\sigma}_2^T) - \min_{\sigma_2^* \in \Sigma_2} u(\overline{\sigma}_1^T, \sigma_2^*) \leq \frac{4\,(\Delta_u + k\beta F)\, M \sqrt{|A|} + 2\beta G}{\sqrt{T}} \tag{17}$$

Proof. This follows from using the projected gradient descent regret bound (Zinkevich 2003) to give $R_\lambda^T(\beta) \leq \frac{\beta^2}{2\alpha^T} + \frac{G^2}{2} \sum_{t=1}^{T} \alpha^t \leq \frac{\beta G}{\sqrt{T}}$. □

Finally, we discuss how to choose β. When there is a minimum acceptable constrant violation, β can be selected with Theorem 2 to guarantee that the violation is no more than the specified value, either asymptotically or after a specified number of iterations T. When no amount of constraint violation is acceptable, β should be chosen such that $\beta \geq \lambda_i^*$ by Theorem 3. If $\boldsymbol{\lambda}^*$ is unknown, CCFR can be run with an arbitrary β for a number of iterations. If the average $\frac{1}{T}\sum_{t=1}^{T}\lambda_i^t$ is close to β, then $\beta \leq \lambda_i^*$, so β is doubled and CCFR run again. Otherwise, it is guaranteed that $\beta > \lambda_i^*$ and CCFR will converge to a solution with no constraint violation.

4 Related Work

To the best of our knowledge, no previous work has proposed a technique for solving either of the optimizations (6) or (7) for general constraints in extensive-form games. Optimization (7) belongs to a general class of saddle point optimizations for which a number of accelerated methods with $\mathcal{O}(1/T)$ convergence have been proposed (Nemirovski 2004; Nesterov 2005b; 2005a; Juditsky, Nemirovski, and Tauvel 2011; Chambolle and Pock 2011). These methods have been applied to unconstrained equilibrium computation in extensive-form games using a family of prox functions initially proposed by Hoda et. al. (Hoda et al. 2010; Kroer et al. 2015; 2017). Like CFR, these algorithms could be extended to solve the optimization (7).

Despite a worse theoretical dependence on T, CFR is preferred to accelerated methods as our base algorithm for a number of practical reasons.

- CFR can be easily modified with a number of different sampling schemes, adapting to sparsity and achieving greatly improved convergence over the deterministic version (Lanctot et al. 2009). Although the stochastic mirror prox algorithm has been used to combine an accelerated update with sampling in extensive-form games, each of its iterations still requires walking each player's full strategy space to compute the prox functions, and it has poor performance in practice (Kroer et al. 2015).
- CFR has good empirical performance in imperfect recall games (Waugh et al. 2009b) and even provably converges to an equilibrium in certain subclasses of well-formed games (Lanctot et al. 2012; Lisý, Davis, and Bowling 2016), which we will make use of in Section 5.1. The prox function used by the accelerated methods is ill-defined in all imperfect recall games.
- CFR theoretically scales better with game size than do the accelerated techniques. The constant M in the bounds of Theorems 1-3 is at worst $|\mathcal{I}|$, and for many games of interest is closer to $|\mathcal{I}|^{1/2}$ (Burch 2017, Section 3.2). The best convergence bound for an accelerated method depends in the worst case on $|\mathcal{I}|^2 2^d$ where d is the depth of the game tree, and is at best $|\mathcal{I}|2^d$ (Kroer et al. 2017).
- The CFR update can be modified to CFR+ to give a guaranteed bound on tracking regret and greatly improve empirical performance (Tammelin et al. 2015). CFR+ has been shown to converge with initial rate faster than $\mathcal{O}(1/T)$ in a variety of games (Burch 2017, Sections 4.3-4.4).

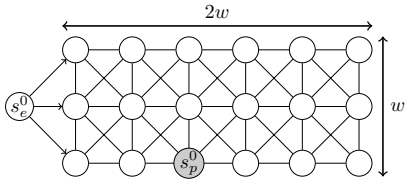


Figure 1: Transit Game

- Finally, CFR is not inherently limited to $\mathcal{O}(1/\sqrt{T})$ worst-case convergence. Regret minimization algorithms can be optimistically modified to give $\mathcal{O}(1/T)$ convergence in self-play (Rakhlin and Sridharan 2013). Such a modification has been applied to CFR (Burch 2017, Section 4.4).

We describe CCFR as an extension of deterministic CFR for ease of exposition. All of the CFR modifications described in this section can be applied to CCFR out-of-the-box.

5 Experimental evaluation

We present two domains for experimental evaluation in this paper. In the first, we use constraints to model a secondary objective when generating strategies in a model security game. In the second domain, we use constraints for opponent modeling in a small poker game. We demonstrate that using constraints for modeling data allows us to learn counter-strategies that approach optimal counter-strategies as the amount of data increases. Unlike previous opponent modeling techniques for poker, we do not require our data to contain full observations of the opponent's private cards for this guarantee to hold.

5.1 Transit game

The *transit game* is a model security game introduced in (Bosansky et al. 2015). With size parameter w, the game is played on an 8-connected grid of size $2w \times w$ (see Figure 1) over $d = 2w + 4$ time steps. One player, the *evader*, wishes to cross the grid from left to right while avoiding the other player, the *patroller*. Actions are movements along the edges of the grid, but each move has a probability 0.1 of failing. The evader receives -1 utils for each time he encounters the patroller, 1 utils when he escapes on reaching the east end of the grid, and -0.02 utils for each time step that passes without escaping. The patroller receives the negative of the evader's utils, making the game zero-sum. The players observe only their own actions and locations.

The patroller has a secondary objective of minimizing the risk that it fails to return to its base (s_p^0 in Figure 1) by the end of the game. In the original formulation, this was modeled using a large utility penalty when the patroller doesn't end the game at its base. For the reasons discussed in the introduction, it is more natural to model this objective as a linear constraint on the patroller's strategy, bounding the maximum probability that it doesn't return to base.

For our experiments, we implemented CCFR on top of the NFGSS-CFR algorithm described in (Lisý, Davis, and Bowling 2016). In the NFGSS framework, each information set is defined by only the current grid state and the time step; history is not remembered. This is a case of imperfect recall,

but our theory still holds as the game is well-formed. The constraint on the patroller is defined as

$$\sum_{s^d,a} \pi^{\sigma_p}(s^d)\sigma(s^d,a) \sum_{s^{d+1} \neq s_p^0} T(s^d,a,s^{d+1}) \leq b_r$$

where s^d, a are state action pairs at time step d, $T(s^d,a,s^{d+1})$ is the probability that s^{d+1} is the next state given that action a is taken from s^d, and b_r is the chosen risk bound. This is a well-defined linear constraint despite imperfect recall, as $\pi^{\sigma_p}(s^d)$ is a linear combination over the sequences that reach s^d. We update the CCFR constraint weights λ using stochastic gradient ascent with constant step size $\alpha^t = 1$, which we found to work well across a variety of game sizes and risk bounds. In practice, we found that bounding $\boldsymbol{\lambda}$ was unnecessary for convergence.

Previous work has shown that double oracle (DO) techniques outperform solving the full game linear program (LP) in the unconstrained transit game (Bosansky et al. 2015; Lisý, Davis, and Bowling 2016). However, an efficient best response oracle exists in the unconstrained setting only because a best response is guaranteed to exist in the space of pure strategies, which can be efficiently searched. Conversely, constrained best responses might exist only in the space of mixed strategies, meaning that the best response computation requires solving an LP of comparable size to the LP for the full game Nash equilibrium. This makes DO methods inappropriate for the general constrained setting, so we omit comparison to DO methods in this work.

Results We first empirically demonstrate that CCFR converges to optimality by comparing its produced strategies with strategies produced by solving the LP representation of the game with the simplex solver in IBM ILOG CPLEX 12.7.1. Figure 2a shows the risk and exploitability for strategies produced by running CCFR for 100,000 iterations on a game of size $w = 8$, with a variety of values for the risk bound b_r. In each case, the computed strategy had risk within 0.001 of the specified bound b_r, and exploitability within 0.001 of the corresponding LP strategy (not shown because the points are indistinguishable). The convergence over time for one particular case, $b_r = 0.1$, is shown in Figure 2b, where the plotted value is the difference in exploitability between the average CCFR strategy and the LP strategy, shown with a log-linear scale. The vertical line shows the time used to compute the LP strategy.

Convergence times for the CPLEX LP and CCFR with risk bound $b_r = 0.1$ are shown on a log scale for a variety of game sizes w in Figure 2c. The time for CCFR is presented for a variety of precisions ϵ, which bounds both the optimality of the final exploitability and the violation of the risk bound. The points for game size $w = 8$ are also shown in Figure 2b. The LP time is calculated with default precision $\epsilon = 10^{-6}$. Changing the precision to a higher value actually results in a slower computation, due to the precision also controlling the size of allowed infeasibility in the Harris ratio test (Klotz and Newman 2013).

Even at $w = 6$, a game which has relatively small strategy sizes of ∼6,000 values, CCFR can give a significant speedup for a small tradeoff in precision. At $w = 8$ and larger, the LP is clearly slower than CCFR even for the relatively modest precision of $\epsilon = 0.001$. For game size $w = 10$, with strategy sizes of ∼25,000 values, the LP is more than an order of magnitude slower than high precision CCFR.

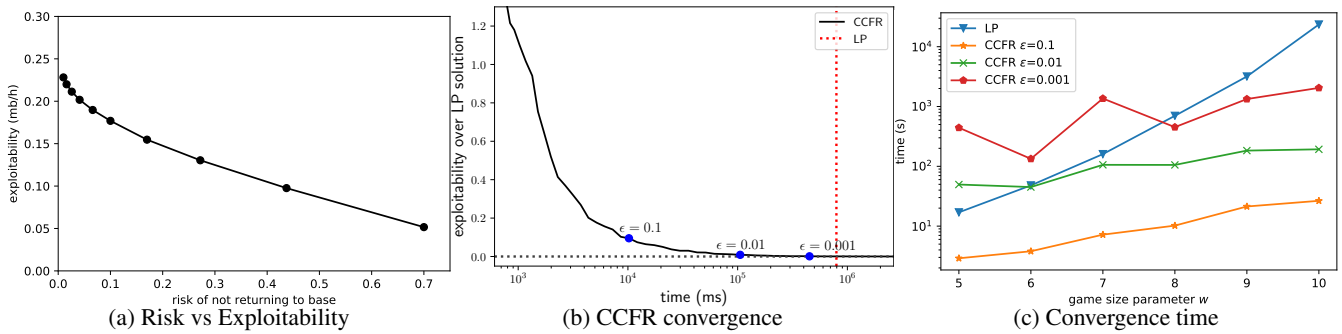
5.2 Opponent modeling in poker

In multi-agent settings, strategy constraints can serve an additional purpose beyond encoding secondary objectives. Often, when creating a strategy for one agent, we have partial information on how the other agent(s) behave. A way to make use of this information is to solve the game with constraints on the other agents' strategies, enforcing that their strategy in the solution is consistent with their observed behavior. As a motivating example, we consider poker games in which we always observe our opponent's actions, but not necessarily the private card(s) that they hold when making the action.

In poker games, if either player takes the *fold* action, the other player automatically wins the game. Because the players' private cards are irrelevant to the game outcome in this case, they are typically not revealed. We thus consider the problem of opponent modeling from observing past games, in which the opponent's *hand* of private card(s) is only revealed when a *showdown* is reached and the player with the better hand wins. Most previous work in opponent modeling has either assumed full observation of private cards after a fold (Johanson, Zinkevich, and Bowling 2008; Johanson and Bowling 2009) or has ignored observations of opponent actions entirely, instead only using observed utilities (Bard et al. 2013). The only previous work which uses these partial observations has no theoretical guarantees on solution quality (Ganzfried and Sandholm 2011).

We first collect data by playing against the opponent with a probe strategy, which is a uniformly random distribution over the non-fold actions. To model the opponent in an unbiased way, we generate two types of sequence-form constraints from this data. First, for each possible sequence of public actions and for each of our own private hands, we build an unbiased confidence interval on the probability that we are dealt the hand and the public sequence occurs. This probability is a weighted sum of opponent sequence probabilities over their possible private cards, and thus the confidence bounds become linear sequence-form constraints. Second, for each terminal history that is a showdown, we build a confidence interval on the probability that the showdown is reached. In combination, these two sets of constraints guarantee that the CCFR strategy converges to a best response to the opponent strategy as the number of observed games increases. A proof of convergence to a best response and full details of the constraints are provided in the supplementary materials.

Infeasible constraints Because we construct each constraint separately, there is no guarantee that the full constraint set is simultaneously feasible. In fact, in our experiments the constraints were typically mildly infeasible. However, this is not a problem for CCFR, which doesn't require feasible constraints to have well-defined updates. In fact, because we bound the Lagrange multipliers, CCFR still theoretically



(a) Risk vs Exploitability (b) CCFR convergence (c) Convergence time

Figure 2: Risk and exploitability of final CCFR strategies with game size $w = 8$ (a), convergence of CCFR exploitability to LP exploitability with $w = 8$ and risk bound $b_r = 0.1$ (b), and dependence of convergence time on game size for LP and CCFR methods (c). All algorithms are deterministic, so times are exact.

converges to a sensible solution, especially when the total infeasibility is small. For more details on how CCFR handles infeasibility, see the supplementary materials.

Results We ran our experiments in Leduc Hold'em (Southey et al. 2005), a small poker game played with a six card deck over two betting rounds. To generate a target strategy profile to model, we solved the "JQ.K/pair.nopair" abstracted version of the game (Waugh et al. 2009a). We then played a probe strategy profile against the target profile to generate constraints as described above, and ran CCFR twice to find each half of a counter-profile that is optimal against the set of constrained profiles. We used gradient ascent with step size $\alpha^t = 1000/\sqrt{t}$ to update the $\boldsymbol{\lambda}$ values, and ran CCFR for 10^6 iterations, which we found to be sufficient for approximate convegence with $\varepsilon < 0.001$.

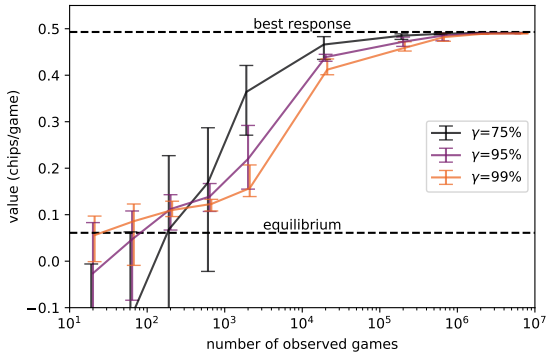


Figure 3: Performance of counter-profiles trained with CCFR against a target profile in Leduc Hold'em. Results are averaged over 10 runs, with minimum and maximum values shown as error bars. Values achieved by a Nash equilibrium profile and a best response profile are shown as horizontal lines for reference.

We evaluate how well the trained counter-profile performs when played against the target profile, and in particular investigate how this performance depends on the number of games we observe to produce the counter-profile, and on the confidence γ used for the confidence interval constraints. Results are shown in Figure 3, with a log-linear scale. With a high confidence $\gamma = 99\%$ (looser constraints), we obtain an expected value that is better than the equilibrium expected value with fewer than 100 observed games on average, and with fewer than 200 observed games consistently. Lower confidence levels (tighter constraints) resulted in more variable performance and poor average value with small numbers of observed games, but also faster learning as the number of observed games increased. For all confidence levels, the expected value converges to the best response value as the number of observed games increases.

6 Conclusion

Strategy constraints are a powerful modeling tool in extensive-form games. Prior to this work, solving games with strategy constraints required solving a linear program, which scaled poorly to many of the very large games of practical interest. We introduced CCFR, the first efficient large-scale algorithm for solving extensive-form games with general strategy constraints. We demonstrated that CCFR is effective at solving sequential security games with bounds on acceptable risk. We also introduced a method of generating strategy constraints from partial observations of poker games, resulting in the first opponent modeling technique that has theoretical guarantees with partial observations. We demonstrated the effectiveness of this technique for opponent modeling in Leduc Hold'em.

7 Acknowledgments

The transit game experiments were implemented with code made publically available by the game theory group of the Artificial Intelligence Center at Czech Technical University in Prague. This research was supported by Alberta Innovates, Alberta Advanced Education, and the Alberta Machine Intelligence Institute (Amii). Computing resources were provided by Compute Canada and Calcul Québec.

References

Altman, E. 1999. *Constrained Markov Decision Processes*. Chapman and Hall/CRC.

Bard, N.; Johanson, M.; Burch, N.; and Bowling, M. 2013. Online implicit agent modeling. In *Proceedings of the Twelth International Conference on Autonomous Agents and Multiagent Systems*.

Bosansky, B.; Jiang, A. X.; Tambe, M.; and Kiekintveld, C. 2015. Combining compact representation and incremental generation in large games with sequential strategies. In *Proceedings of Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Bowling, M.; Burch, N.; Johanson, M.; and Tammelin, O. 2015. Heads-up limit hold'em poker is solved. *Science* 347(6218):145–149.

Brown, N., and Sandholm, T. 2018. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science* 359(6374):418–424.

Brown, M.; An, B.; Kiekintveld, C.; Ordóñez, F.; and Tambe, M. 2014. An extended study on multi-objective security games. *Autonomous Agents and Multi-Agent Systems* 28(1):31–71.

Burch, N. 2017. *Time and Space: Why Imperfect Information Games are Hard*. Ph.D. Dissertation, University of Alberta.

Chambolle, A., and Pock, T. 2011. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision* 40(1):120–145.

Farina, G.; Kroer, C.; and Sandholm, T. 2017. Regret minimization in behaviorally-constrained zero-sum games. In *Proceedings of the 34th International Conference on Machine Learning*.

Ganzfried, S., and Sandholm, T. 2011. Game theory-based opponent modeling in large imperfect-information games. In *Proceedings of the Tenth International Conference on Autonomous Agents and Multiagent Systems*.

Hart, S., and Mas-Colell, A. 2000. A simple adaptive procedure leading to correlated equilibrium. *Econometrica* 5(68):1127–1150.

Hoda, S.; Gilpin, A.; Peña, J.; and Sandholm, T. 2010. Smoothing techniques for computing nash equilibria of sequential games. *Mathematics of Operations Research* 35(2):494–512.

Isom, J. D.; Meyn, S. P.; and Braatz, R. D. 2008. Piecewise linear dynamic programming for constrained POMDPs. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*.

Johanson, M., and Bowling, M. 2009. Data biased robust counter strategies. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*.

Johanson, M.; Bowling, M.; Waugh, K.; and Zinkevich, M. 2011. Accelerating best response calculation in large extensive games. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*.

Johanson, M.; Zinkevich, M.; and Bowling, M. 2008. Computing robust counter-strategies. In *Advances in Neural Information Processing Systems 20*.

Juditsky, A.; Nemirovski, A.; and Tauvel, C. 2011. Solving variational inequalities with stochastic mirror-prox algorithm. *Stochastic Systems* 1(1):17–58.

Klotz, E., and Newman, A. M. 2013. Practical guidelines for solving difficult linear programs. *Surveys in Operations Research and Management Science* 18(1):1–17.

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.

Kroer, C.; Waugh, K.; Kilinç-Karzan, F.; and Sandholm, T. 2015. Faster first-order methods for extensive-form game solving. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*.

Kroer, C.; Waugh, K.; Kilinc-Karzan, F.; and Sandholm, T. 2017. Theoretical and practical advances on smoothing for extensive-form games. In *Proceedings of the Eighteenth ACM Conference on Economics and Computation*.

Lanctot, M.; Waugh, K.; Zinkevich, M.; and Bowling, M. 2009. Monte Carlo sampling for regret minimization in extensive games. In *Advances in Neural Information Processing Systems 22*.

Lanctot, M.; Gibson, R.; Burch, N.; and Bowling, M. 2012. No-regret learning in extensive-form games with imperfect recall. In *Proceedings of the Twenty-Ninth International Conference on Machine Learning*.

Lisý, V.; Davis, T.; and Bowling, M. 2016. Counterfactual regret minimization in security games. In *Proceedings of Thirtieth AAAI Conference on Artificial Intelligence*.

Moravčík, M.; Schmid, M.; Burch, N.; Lisý, V.; Morrill, D.; Bard, N.; Davis, T.; Waugh, K.; Johanson, M.; and Bowling, M. H. 2017. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science* 356 6337:508–513.

Nemirovski, A. 2004. Prox-method with rate of convergence o(1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization* 15(1):229–251.

Nesterov, Y. 2005a. Excessive gap technique in nonsmooth convex minimization. *SIAM Journal on Optimization* 16(1):235–249.

Nesterov, Y. 2005b. Smooth minimization of non-smooth functions. *Mathematical Programming* 103(1):127–152.

Osborne, M. J., and Rubinstein, A. 1994. *A Course in Game Theory*. The MIT Press.

Rakhlin, A., and Sridharan, K. 2013. Online learning with predictable sequences. In *Proceedings of the 26th Annual Conference on Learning Theory*.

Santana, P.; Thiébaux, S.; and Williams, B. 2016. Rao*: an algorithm for chance constrained pomdps. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*.

Southey, F.; Bowling, M.; Larson, B.; Piccione, C.; Burch, N.; Billings, D.; and Rayner, C. 2005. Bayes' bluff: Opponent modelling in poker. In *Proceedings of the Twenty-First Conference on Uncertainty in Artficial Intelligence*.

Tammelin, O.; Burch, N.; Johanson, M.; and Bowling, M. 2015. Solving heads-up limit Texas Hold'em. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*.

von Stengel, B. 1996. Efficient computation of behavior strategies. *Games and Economic Behavior* 14:220–246.

Waugh, K., and Bagnell, J. A. 2015. A unified view of large-scale zero-sum equilibrium computation. In *AAAI Workshop on Computer Poker and Imperfect Information*.

Waugh, K.; Schnizlein, D.; Bowling, M.; and Szafron, D. 2009a. Abstraction pathologies in extensive games. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems*.

Waugh, K.; Zinkevich, M.; Johanson, M.; Kan, M.; Schnizlein, D.; and Bowling, M. 2009b. A practical use of imperfect recall. In *Proceedings of the Eighth Symposium on Abstraction, Reformulation and Approximation*.

Zinkevich, M.; Johanson, M.; Bowling, M.; and Piccione, C. 2008. Regret minimization in games with incomplete information. In *Advances in Neural Information Processing Systems 20*.

Zinkevich, M. 2003. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on Machine Learning*.