Composability of Regret Minimizers

Gabriele Farina and **Christian Kroer** and **Tuomas Sandholm**
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213
{gfarina,ckroer,sandholm}@cs.cmu.edu

Abstract

Regret minimization is a powerful tool for solving large-scale problems; it was recently used in breakthrough results for large-scale extensive-form-game solving. This was achieved by composing simplex regret minimizers into an overall regret-minimization framework for extensive-form-game strategy spaces. In this paper we study the general composability of regret minimizers. We derive a calculus for constructing regret minimizers for complex convex sets that are constructed from convexity-preserving operations on simpler convex sets. In particular, we show that local regret minimizers for the simpler sets can be composed with additional regret minimizers into an aggregate regret minimizer for the complex set. As an application of our framework we show that the CFR framework can be constructed easily from our framework. We also show how to construct a CFR variant for extensive-form games with strategy constraints. Unlike a recently proposed variant of CFR for strategy constraints by Davis, Waugh, and Bowling (2018), the algorithm resulting from our calculus does not depend on any unknown constants and thus avoids binary search.

Introduction

Counterfactual regret minimization (CFR) (Zinkevich et al. 2007), and the newest variant *CFR*$^+$ (Tammelin et al. 2015), have been a central component in several recent milestones in solving imperfect-information *extensive-form games (EFGs)*. Bowling et al. (2015) used CFR$^+$ to near-optimally solve heads-up limit Texas hold'em. Brown and Sandholm (2017) and Moravčík et al. (2017) used CFR variants, along with other scalability techniques, to create AIs that beat professional poker players at the larger game of heads-up no-limit Texas hold'em.

CFR is usually presented as an algorithm for computing a Nash equilibrium in zero-sum EFGs. However, an alternative view is that it is a framework for constructing regret minimizers for the types of action spaces encountered in EFGs, as well as single-agent sequential decision making problems with similarly-structured actions spaces. Viewed from a convex optimization perspective, the class of convex sets CFR applies to are sometimes referred to as *treeplexes* (Hoda et al. 2010; Kroer et al. 2015; 2018). In this view, CFR specifies how a set of regret minimization algorithms for simplexes and linear loss functions can be composed to form a regret minimizer for a treeplex. Farina, Kroer, and Sandholm (2019) take this view further, describing how regret-minimization algorithms can be composed to form regret minimizers for a generalization of treeplexes that allows convex sets and convex losses.

In this paper take a general view on the composability of regret minimizers. We derive a set of rules for how regret minimizers can be constructed for fairly general convex sets via a *calculus* of regret minimization: given regret minimizers for convex sets X_1, X_2 we show how to compose these regret minimizers for various convexity-preserving operations performed on the sets (e.g. intersection, convex hull, Cartesian product), in order to arrive at a regret minimizer for the resulting convex set. This approach draws inspiration from the calculus of convex sets and functions found in books such as Boyd and Vandenberghe (2004). It likewise has parallels to *disciplined convex programming* (Grant, Boyd, and Ye 2006), which emphasizes the solving of convex programs via composition of simple convex functions and sets. This approach has been highly successful in the CVX software package for convex programming (Grant, Boyd, and Ye 2008).

Our approach treats the regret minimizers for individual convex sets as black boxes, and builds a regret minimizer for the resulting convex set by constructing overall solutions from the output of each individual regret minimizer. This is important because it allows freedom in choosing the best regret minimizer for each individual set (from either a practical or theoretical perspective). For example, in practice the *regret matching* (Hart and Mas-Colell 2000) and *regret matching*$^+$ (RM$^+$) (Tammelin et al. 2015) regret minimizers are known to perform better than theoretically-superior regret minimizers such as *Hedge* (Brown, Kroer, and Sandholm 2017), while Hedge may give better theoretical results when trying to prove the convergence rate of a construction through our calculus.

One way to conceptually view our construction is as *regret circuits*: in order to construct a regret minimizer for some convex set $\mathcal{X}$ that consists of convexity-preserving operations on (say) two sets $\mathcal{X}_1, \mathcal{X}_2$ we construct a regret circuit consisting of regret minimizers for $\mathcal{X}_1$ and $\mathcal{X}_2$, along with a sequence of operations that aggregate the results of those circuits in order to form an overall circuit for $\mathcal{X}$. We use this view extensively in the paper, as we show the regret-circuit representation of every operation that we develop.

As an application, we show that the correctness of the CFR algorithm can be showed easily through our calculus, and likewise show that the recent *Constrained CFR* algorithm (Davis, Waugh, and Bowling 2018) can be constructed via our framework.

Regret Minimization

We work inside of the online learning framework called *online convex optimization* (Zinkevich 2003). In this setting, a decision maker repeatedly plays against an unknown environment by making a sequence of decisions $x^1, x^2, \ldots$. As customary, we assume that the set $\mathcal{X} \subseteq \mathbb{R}^n$ of all possible decisions for the decision maker is convex and compact. The outcome of each decision x^t is evaluated as $\ell^t(x^t)$, where ℓ^t is a convex function *unknown* to the decision maker until after the decision is made. Hence, abstractly, a *regret minimizer* is a device that supports two operations:

- it gives a *recommendation* for the next decision $x^{t+1} \in \mathcal{X}$;
- it receives/observes the convex loss function ℓ^t used to "evaluate" decision x^t.

The learning is *online* in the sense that the decision maker/regret minimizer's next decision, x^{t+1}, is based only on the previous decisions $x^1, \ldots, x^t$ and corresponding loss observations $\ell^1, \ldots, \ell^t$.

The quality of the regret minimizer is measured by its *cumulative regret*. Formally, the cumulative regret at time T is defined as

$$R^T_{(\mathcal{X},\mathcal{F})} := \sum_{t=1}^{T} \ell^t(x^t) - \min_{\hat{x}\in\mathcal{X}} \sum_{t=1}^{T} \ell^t(\hat{x}),$$

It measures the difference between the loss cumulated by the sequence of decisions $x^1, \ldots, x^T$ and the loss that would have been cumulated by playing the best time-independent decision $\hat{x}$ in hindsight. A desirable property of a regret minimizer is *Hannan consistency*: the average regret approaches zero, that is, $R^T_{(\mathcal{X},\mathcal{F})}$ grows at a *sublinear* rate in T.

The above discussion can be formalized as follows.

Definition 1 (Regret minimizer). *Let $\mathcal{X}$ be a closed convex set, and let $\mathcal{F}$ be a convex cone in the space of bounded convex functions on $\mathcal{X}$, and such that $\mathcal{F}$ contains the space $\mathcal{L}$ of linear functions. A $(\mathcal{X},\mathcal{F})$-*regret minimizer *is a function that selects the next action $x^{t+1} \in \mathcal{X}$ given the history of actions $x^1, \ldots, x^t$ and observed corresponding loss functions $\ell^1, \ldots, \ell^t \in \mathcal{F}$, so that the cumulative regret $R^T_{(\mathcal{X},\mathcal{F})} \in o(T)$.*

Universality of linear losses

It is known that that a regret minimizer that is powerful enough to handle any (bounded) linear loss is also able to handle any convex function with bounded (sub)gradients (McMahan 2011). In this sense, regret minimizers for linear losses are *universal*.

The crucial insight is in the following observation:

$$\begin{aligned} R^T_{(\mathcal{X},\mathcal{F})} &= \sum_{t=1}^{T} \ell^t(x^t) - \min_{\hat{x}\in\mathcal{X}} \left\{ \sum_{t=1}^{T} \ell^t(\hat{x}) \right\} \\ &\le \sum_{t=1}^{T} \ell^t(x^t) - \min_{\hat{x}\in\mathcal{X}} \left\{ \sum_{t=1}^{T} \ell^t(x^t) + \langle \partial\ell^t(x^t),\, \hat{x} - x^t \rangle \right\} \\ &= \sum_{t=1}^{T} \langle \partial\ell^t(x^t),\, x^t \rangle - \min_{\hat{x}\in\mathcal{X}} \left\{ \sum_{t=1}^{T} \langle \partial\ell^t(x^t),\, \hat{x} \rangle \right\}. \end{aligned} \tag{1}$$

In other words, the regret of a $(\mathcal{X},\mathcal{F})$-regret minimizer is always bounded by the regret of a $(\mathcal{X},\mathcal{L})$-regret minimizer that at each iteration t sees $\langle \partial\ell^t(x^t), \cdot \rangle$ as its (linear) loss function.

The regret circuit corresponding to the above construction is shown in Figure 1. We call diagrams like the one in Figure 1 a *regret circuit*. Throughout this paper, we will use the following conventions when drawing regret circuits:

- the symbol ⧄ is used to denote an operation that constructs or manipulates one or more loss functions;
- the symbol $\otimes$ is used to denote an operation that combines or manipulates one or more recommendations;
- dashed arrows denote recommendations that originate from the previous iteration.

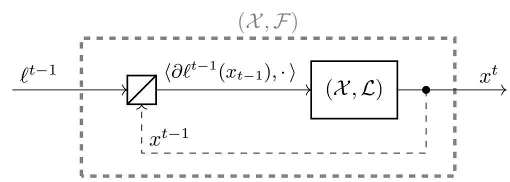

Figure 1: Regret circuit representing the construction of a $(\mathcal{X},\mathcal{F})$-regret minimizer using a $(\mathcal{X},\mathcal{L})$-regret minimizer. Dashed arrows represent recommendations that originate from the previous iteration.

Set shifting

If we have a $(\mathcal{X},\mathcal{L})$-regret minimizer h, but would like to minimize regret over $(\mathcal{Z},\mathcal{L})$ where $\mathcal{Z} = \mathcal{X} + b$ for some vector b, then we can use h directly: since losses are linear subtracting a constant vector from each point has no effect on regret, and so we simply shift $\mathcal{Z}$ to $\mathcal{X}$ via the mapping $x = z - b$.

Due to the above we can always assume without loss of generality that $0 \in \mathcal{X}$; if it is not then we can simply subtract any $x \in \mathcal{X}$ from $\mathcal{X}$ in order to make it so.

Connection to Game Theory and Convex-Concave Saddle Point Problems

Regret minimization is tightly connected to the problem of computing a Nash equilibrium in a zero-sum game. A well-known folk theorem states that the average of a sequence of regret minimizing strategies converges to a Nash equilibrium in a zero-sum game. In this section we explain the relationship to a particular class of games: EFGs. We will later show that our framework allows construction of regret minimizers for EFG strategy spaces, as well as for the more general class of convex-concave saddle-point problems presented below.

In two-player zero-sum extensive-form games with perfect recall, a Nash equilibrium is a solution to the bilinear saddle point problem

$$\min_{x\in\mathcal{X}}\max_{y\in\mathcal{Y}}\ x^\top Ay,$$

where $\mathcal{X}$ and $\mathcal{Y}$ are convex polytopes whose description is provided by the *sequence-form constraints*, and A is a real payoff matrix (von Stengel 1996). Here, we focus on a slightly more general utility structure than is usually considered, and allow an additional *regularization* term for each player. In particular, we assume that we are solving a convex-concave saddle-point problem of the following form:

$$\min_{x\in\mathcal{X}}\max_{y\in\mathcal{Y}}\left\{x^\top Ay + d_1(x) - d_2(y)\right\}, \tag{2}$$

where $\mathcal{X}, \mathcal{Y}$ are allowed to be more general convex sets, and d_1, d_2 are convex functions. This more general formulation allows us to capture applications such as computing a normal-form quantal-response equilibrium (Ling, Fang, and Kolter 2018; Farina, Kroer, and Sandholm 2019), and several types of opponent exploitation. Farina, Kroer, and Sandholm (2019) study opponent exploitation where the goal is to compute a best response, subject to a penalty for moving away from the Nash equilibrium strategy, this is captured in the above by having d_1 or d_2 include a penalty term which penalizes distance to an NE strategy. Farina, Kroer, and Sandholm (2017) study constraints on individual decision points, and Davis, Waugh, and Bowling (2018) study additional constraints on the overall EFG polytopes $\mathcal{X}, \mathcal{Y}$; both these approaches can be captured in our setting by allowing more general $\mathcal{X}, \mathcal{Y}$.

Convex-concave saddle point problems of the form (2) can be solved using online convex optimization. The key idea is to consider the loss functions $\ell_X^t : X \to \mathbb{R}$ and $\ell_Y^t : Y \to \mathbb{R}$, for player 1 and 2 respectively, defined as

$$\ell_X^t : x \mapsto (-Ay^t)x + d_1(x),$$
$$\ell_Y^t : y \mapsto (A^\top x^t)y + d_2(y).$$

With this choice of loss function, the induced regret-minimizing dynamics for the two players lead to a convex-concave saddle-point problem. Specifically, assume the two players play the game T times, accumulating regret after each iteration as in Figure 2. A folk theorem explains the

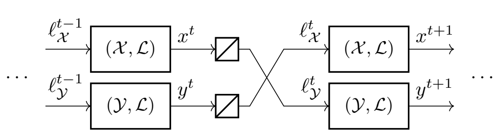

Figure 2: The flow of strategies and losses in regret minimization for games.

tight connection between low-regret strategies and approximate Nash equilibria. We will need a more general variant of that theorem generalized to (2). The convergence criterion we are interested in is the *saddle-point residual (or gap)* ξ of $(\bar{x}, \bar{y})$, defined as

$$\xi = \max_{\hat{y}\in\mathcal{Y}}\{d_1(\bar{x}) - d_2(\hat{y}) + \langle \bar{x}, A\hat{y}\rangle\} - \min_{\hat{x}\in\mathcal{X}}\{d_1(\hat{x}) - d_2(\bar{y}) + \langle \hat{x}, A\bar{y}\rangle\}.$$

The following folk theorem shows that the average of a sequence of regret-minimizing strategies leads to a bounded saddle-point residual (see Farina, Kroer, and Sandholm (2019) for a proof):

Theorem 1. *If the average regret accumulated on $\mathcal{X}$ and $\mathcal{Y}$ by the two sets of strategies $\{x_t\}_{t=1}^T$ and $\{y_t\}_{t=1}^T$ is ϵ_1 and ϵ_2, respectively, then any strategy profile $(\bar{x}, \bar{y})$ such that $\bar{x} = \frac{1}{T}\sum_{t=1}^T x^t, \quad \bar{y} = \frac{1}{T}\sum_{t=1}^T y^t$ has a saddle-point residual bounded by $\epsilon_1 + \epsilon_2$.*

When $d_1 \equiv d_2 \equiv 0$, Theorem 1 asserts that the average strategy profile is a $(\epsilon_1 + \epsilon_2)$-Nash equilibrium.

Simple Operations

As we already observed in the subsection "Universality of linear losses", we can extend any $(\mathcal{X}, \mathcal{L})$-regret minimizer to handle more expressive loss functionals. In fact, once a $(\mathcal{X}, \mathcal{L})$-regret minimizer is known, extending it to a $(\mathcal{X}, \mathcal{F})$-regret minimizer is mostly a mechanical task. For this reason, in the rest of the paper we focus on $(\mathcal{X}, \mathcal{L})$-regret minimizers, and show how these can be constructed by composition of simpler primitives.

Cartesian product

In this section, we show how to combine a $(\mathcal{X}, \mathcal{L})$- and a $(\mathcal{Y}, \mathcal{L})$-regret minimizer to form a $(\mathcal{X}\times\mathcal{Y}, \mathcal{L})$-regret minimizer. As we have already observed, we can assume without loss of generality that $(0, 0) \in (\mathcal{X}, \mathcal{Y})$. Hence, any linear function $\ell : \mathcal{X}\times\mathcal{Y} \to \mathbb{R}$ can then be written as

$$\ell(x, y) = \ell_{\mathcal{X}}(x) + \ell_{\mathcal{Y}}(y)$$

where the linear functions $\ell_{\mathcal{X}} : \mathcal{X} \to \mathbb{R}$ and $\ell_{\mathcal{Y}} : \mathcal{Y} \to \mathbb{R}$ are defined as $\ell_{\mathcal{X}} : x \mapsto \ell(x, 0)$ and $\ell_{\mathcal{Y}} : y \mapsto \ell(0, y)$. It is immediate to verify that indeed

$$\begin{aligned} R^T_{(\mathcal{X}\times\mathcal{Y},\mathcal{L})} &= \left(\sum_{t=1}^T \ell^t_{\mathcal{X}}(x^t) - \min_{\hat{x}\in\mathcal{X}}\left\{\sum_{t=1}^T \ell^t_{\mathcal{X}}(\hat{x})\right\}\right) \\ &\qquad + \left(\sum_{t=1}^T \ell^t_{\mathcal{Y}}(y^t) - \min_{\hat{y}\in\mathcal{Y}}\left\{\sum_{t=1}^T \ell^t_{\mathcal{Y}}(\hat{y})\right\}\right) \\ &= R^T_{(\mathcal{X},\mathcal{L})} + R^T_{(\mathcal{Y},\mathcal{L})}. \end{aligned}$$

In other words, it is possible to minimize regret on $\mathcal{X}\times\mathcal{Y}$ by simply minimizing it on $\mathcal{X}$ and $\mathcal{Y}$ independently and then combining the recommendations.

Affine transformation and Minkowski sum

Let $T : E \to F$ be an affine function between two Euclidean spaces E and F, and let $\mathcal{X} \subseteq E$ be a convex and compact set. We now show how a $(\mathcal{X}, \mathcal{L})$-regret minimizer can be employed to construct a $(T(\mathcal{X}), \mathcal{L})$-regret minimizer.

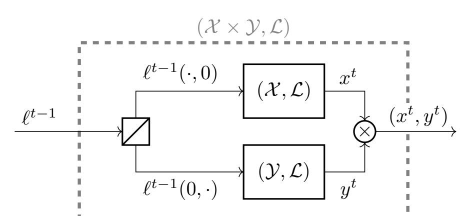

Figure 3: Regret circuit for the Cartesian product $\mathcal{X}\times\mathcal{Y}$.

We start by writing the definition of cumulative regret for a $(T(\mathcal{X}), \mathcal{L})$-regret minimizer:

$$R^T_{(T(\mathcal{X}),\mathcal{L})} = \sum_{t=1}^{T} (\ell^t \circ T)(x^t) - \min_{\hat{x}\in\mathcal{X}} \sum_{t=1}^{T} (\ell^t \circ T)(\hat{x}).$$

Since ℓ^t and T are affine, their composition $\ell^t_T := \ell^t \circ T$ is also affine. Hence, $R^T_{(T(\mathcal{X}),\mathcal{L})}$ is the same regret as a $(\mathcal{X}, \mathcal{L})$-regret minimizer that observes $\ell^t_T(\cdot) - \ell^t_T(0)$ instead of ℓ^t. The construction is summarized by the circuit of Figure 4, where we ignored the constant shifting term $\ell^t_T(0)$. The bound on the new loss functional is $\|\ell^t_T\| \le \|\ell^t\| \cdot \|T\|$.

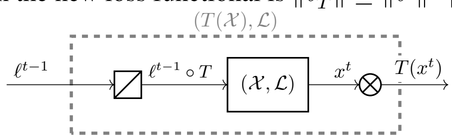

Figure 4: Regret circuit for the image of the affine transformation $T(\mathcal{X})$.

As an application, we use the above construction to form a regret minimizer for the Minkowski sum $\mathcal{X} + \mathcal{Y} := \{x + y : x \in \mathcal{X}, y \in \mathcal{Y}\}$ of two sets. Indeed, note that $\mathcal{X} + \mathcal{Y} = \sigma(\mathcal{X} \times \mathcal{Y})$, where $\sigma : \mathcal{X} \times \mathcal{Y} \ni (x, y) \mapsto x + y$ is a linear transformation. Hence, we can combine the construction in this section together with the construction of the Cartesian product (Figure 3). This results in the circuit of Figure 5.

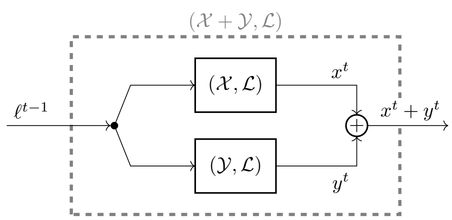

Figure 5: Regret circuit for the Minkowski sum $\mathcal{X} + \mathcal{Y}$.

Convex hull

In this section, we show how to combine a $(\mathcal{X}, \mathcal{L})$- and a $(\mathcal{Y}, \mathcal{L})$-regret minimizer to form a $(\mathrm{co}\{\mathcal{X}, \mathcal{Y}\}, \mathcal{L})$-regret minimizer, where co denotes the convex hull operation,

$$\mathrm{co}\{\mathcal{X}, \mathcal{Y}\} = \{\lambda_1 x + \lambda_2 y \ : \ x \in \mathcal{X}, y \in \mathcal{Y}, (\lambda_1, \lambda_2) \in \Delta^2\},$$

where Δ^2 is the two-dimensional *simplex*

$$\Delta^2 := \{(\lambda_1, \lambda_2) \in \mathbb{R}^+ : \lambda_1 + \lambda_2 = 1\}.$$

Hence, we can think about a $(\mathrm{co}\{\mathcal{X}, \mathcal{Y}\}, \mathcal{L})$-regret minimizer as recommending a triple $(\lambda^t, x^t, y^t) \in \Delta^2 \times \mathcal{X} \times \mathcal{Y}$ at each time point t. Using the linearity of the loss functions,

$$R^T_{(\mathrm{co}\{\mathcal{X},\mathcal{Y}\},\mathcal{L})} = \left(\sum_{t=1}^{T} \lambda^t_1 \ell^t(x^t) + \lambda^t_2 \ell^t(y^t)\right) - \min_{\substack{\hat{\lambda}\in\Delta^2\\ \hat{x}\in\mathcal{X}, \hat{y}\in\mathcal{Y}}} \left\{\hat{\lambda}_1 \sum_{t=1}^{T} \ell^t(\hat{x}) + \hat{\lambda}_2 \sum_{t=1}^{T} \ell^t(\hat{y})\right\}.$$

Now, we make two crucial observations. First, it holds that

$$\min_{\substack{\hat{\lambda}\in\Delta^2\\ \hat{x}\in\mathcal{X}, \hat{y}\in\mathcal{Y}}} \left\{\hat{\lambda}_1 \sum_{t=1}^{T} \ell^t(\hat{x}) + \hat{\lambda}_2 \sum_{t=1}^{T} \ell^t(\hat{y})\right\}$$

$$= \min_{\hat{\lambda}\in\Delta^2} \left\{\hat{\lambda}_1 \min_{\hat{x}\in\mathcal{X}} \left\{\sum_{t=1}^{T} \ell^t(\hat{x})\right\} + \hat{\lambda}_2 \min_{y\in\mathcal{Y}} \left\{\sum_{t=1}^{T} \ell^t(\hat{y})\right\}\right\},$$

since all components of $\hat{\lambda}$ are non-negative. Second, the inner minimization problem over $\mathcal{X}$ is related to the cumulative regret of the $(\mathcal{X}, \mathcal{L})$-regret minimizer as follows:

$$\min_{\hat{x}\in\mathcal{X}} \left\{\sum_{t=1}^{T} \ell^t(\hat{x})\right\} = -R^T_{(\mathcal{X},\mathcal{L})} + \sum_{t=1}^{T} \ell^t(x^t).$$

(An analogous relationship holds for $\mathcal{Y}$.) Combining the two observations, we can write

$$R^T_{(\mathrm{co}\{\mathcal{X},\mathcal{Y}\},\mathcal{L})} = \left(\sum_{t=1}^{T} \lambda^t_1 \ell^t(x^t) + \lambda^t_2 \ell^t(y^t)\right) - \min_{\hat{\lambda}\in\Delta^2} \left\{\left(\sum_{t=1}^{T} \hat{\lambda}_1 \ell^t(x^t) + \hat{\lambda}_2 \ell^t(y^t)\right) - \left(\hat{\lambda}_1 R^T_{(\mathcal{X},\mathcal{L})} + \hat{\lambda}_2 R^T_{(\mathcal{Y},\mathcal{L})}\right)\right\}.$$

Using the fact that $\min(f + g) \ge \min f + \min g$,and introducting the quantity

$$R^T_{(\Delta^2,\mathcal{L})} := \left(\sum_{t=1}^{T} \lambda^t_1 \ell^t(x^t) + \lambda^t_2 \ell^t(y^t)\right) - \min_{\hat{\lambda}\in\Delta^2} \left\{\left(\sum_{t=1}^{T} \hat{\lambda}_1 \ell^t(x^t) + \hat{\lambda}_2 \ell^t(y^t)\right)\right\},$$

we conclude that

$$R^T_{(\mathrm{co}\{\mathcal{X},\mathcal{Y}\},\mathcal{L})} \le R^T_{(\Delta^2,\mathcal{L})} + \max\{R^T_{(\mathcal{X},\mathcal{L})}, R^T_{(\mathcal{Y},\mathcal{L})}\}. \quad (3)$$

The introduced quantity, $R^T_{(\Delta^2,\mathcal{L})}$, is the cumulative regret of a $(\Delta^2, \mathcal{L})$-regret minimizer that, at each time instant t, observes the (linear) loss function

$$\ell^t_\lambda : \Delta^2 \ni (\lambda_1, \lambda_2) \mapsto \lambda_1 \ell^t(x^t) + \lambda_2 \ell^t(y^t). \quad (4)$$

Intuitively, this means that in order to pick "good points" in the convex hull $\mathrm{co}\{\mathcal{X}, \mathcal{Y}\}$, we can let two independent $(\mathcal{X}, \mathcal{L})$- and $(\mathcal{Y}, \mathcal{L})$-regret minimizers pick good recommendations in $\mathcal{X}$ and $\mathcal{Y}$ respectively, and then use a third regret minimizer that decides how to "mix" the recommendations. This way, we break the task of picking the next recommended triple (λ^t, x^t, y^t) into three different subproblems, two of which can be run independently. Equation (3) guarantees that if all three regrets $\{R^T_{(\Delta^2,\mathcal{L})}, R^T_{(\mathcal{X},\mathcal{L})}, R^T_{(\mathcal{Y},\mathcal{L})}\}$ grow sublinearly, then so does $R^T_{(\mathrm{co}\{\mathcal{X},\mathcal{Y}\},\mathcal{L})}$. Figure 6 shows the regret circuit that corresponds to our construction above.

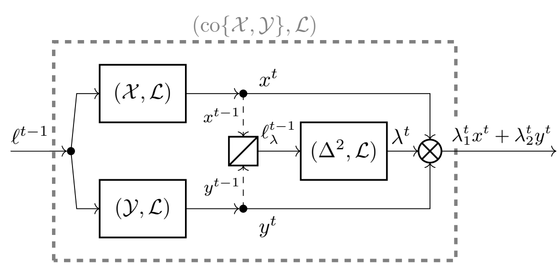

Figure 6: Regret circuit for the convex hull $\mathrm{co}\{\mathcal{X}, \mathcal{Y}\}$. The loss function ℓ^t_λ is defined in Equation (4).

Extending to multiple set Since the convex hull operation is associative, we can handle convex hull $\mathrm{co}\{\mathcal{X}_1, \ldots, \mathcal{X}_n\}$ of a finite number of sets by iteratively adding one set at a time. Equivalently, one can extend the construction shown in Figure 6 to n sets as follows. First, the input loss function ℓ^{t-1} is broadcast to all the $(\mathcal{X}_i, \mathcal{L})$-regret minimizers ($i = 1, \ldots, n$). The loss function ℓ^t_λ is input into a $(\Delta^n, \mathcal{L})$-regret minimizer, where Δ^n is the n-dimensional simplex, and is defined for all time instants t as

$$\ell^t_\lambda : \Delta^n \ni (\lambda_1, \ldots, \lambda_n) \mapsto \lambda_1 x^t_1 + \cdots + \lambda_n x^t_n.$$

Finally, at each time step t, the n recommendations $x^t_1, \ldots, x^t_n$ output by the $(\mathcal{X}_i, \mathcal{L})$-regret minimizers are combined with the recommendation λ^t output by the $(\Delta^n, \mathcal{L})$-regret minimizer to form $\lambda^t_1 x^t_1 + \cdots + \lambda^t_n x^t_n$.

V-polytopes Our construction can be directly applied to construct a $(\mathcal{X}, \mathcal{L})$-regret minimizer for a V-poltope $\mathcal{X} = \mathrm{co}\{v_1, \ldots, v_n\}$ where $v_1, \ldots, v_n$ are n points in a Euclidean space E. Of course, any $(\{v_i\}, \mathcal{L})$-regret minimizer is the constant recommendation v_i. Hence, our construction (Figure 6) simply reduces to a single $(\Delta^n, \mathcal{L})$-regret minimizer that observes the (linear) loss function

$$\ell^t_\lambda : \Delta^n \ni (\lambda_1, \ldots, \lambda_n) \mapsto \lambda_1 \ell^t(v_1) + \cdots + \lambda_n \ell^t(v_n).$$

This observation already appears in Theorem 3 in the work by Farina, Kroer, and Sandholm (2017).

Application: Construction of CFR

We now show that these operations can be used to construct the CFR framework. The first thing to note is that the strategy space of a single player in an EFG is a treeplex, which can be viewed recursively as a series of convex hull and Cartesian product operations[1]. In particular, an information set is viewed as an n-dimensional convex hull (since the sum of probabilities over actions is 1), where each action a at the information set corresponds to a treeplex $\mathcal{X}_a$ representing the set of possible information sets coming after a (in order to perform the convex hull we have to create a new larger representation of $\mathcal{X}_a$ so that the dimension is the same for all a, described in detail below). The Cartesian product operation is used to represent multiple potential information sets being arrived at (for example different hands dealt in a poker game). An example is shown in Figure 7: each information set X_i (except X_0) corresponds to a 2-dimensional convex hull over two treeplexes, one of which is always empty (i.e. a leaf node); each $\otimes$ is a Cartesian product, the top-most $\otimes$ represents the three possible hands that the player may have when making their first decision, the second layer of Cartesian products represent different actions taken by the opponent.

The information-set construction is as follows: let I be the information set under construction, and A_I the set of actions. Each action $a \in A_I$ has some, potentially empty, treeplex $\mathcal{X}_a$ beneath it, let n_a be the dimension of that treeplex. We cannot form a convex hull over $\{X_a\}_{a \in A_I}$ directly, since the sets are not of the same dimension, and we do not wish to average across different strategy spaces. Instead, we create a new convex set $\mathcal{X}'_a \in \mathbb{R}^{|A_I| + \sum_{a \in A_I} n_a}$ for each a; the first $|A_I|$ indices correspond to the actions in A_I, and each $\mathcal{X}_a$ gets its own subset of indices. For each $x \in \mathcal{X}_a$ there is a corresponding $x' \in \mathcal{X}'_a$; x' has a 1 at the index of a, x at the indices corresponding to $\mathcal{X}_a$, and 0 everywhere else. The convex hull is constructed over the set $\{\mathcal{X}'_a\}_a$, which gives

[1] This perspective is also used when constructing distance functions for first-order methods for EFGs (Hoda et al. 2010; Kroer et al. 2015; 2018)

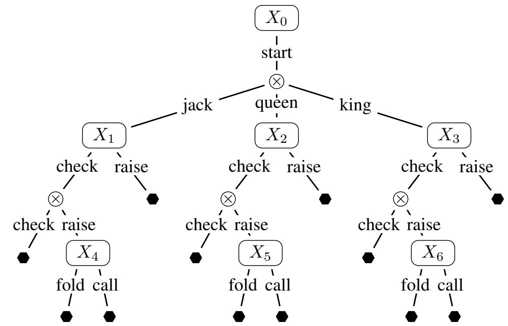

Figure 7: Treeplex for the first player in the game of Kuhn poker. Each X_i represents a convex hull over the treeplexes below, while $\otimes$ denotes the Cartesian product operation.

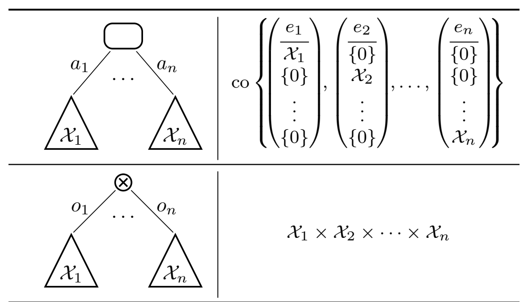

Figure 8: Inductive treeplex construction rule. $e_i \in \mathbb{R}^n$ contains a 1 at index i, and 0 everywhere else.

exactly the treeplex rooted at I. The Cartesian product is easy and can be done over a given set of treeplexes rooted at information sets $I_1, \ldots, I_n$. The inductive construction rule for the treeplex are given in Figure 8.

If we use as our loss function the gradient Ay^t where y^t is the opponent strategy at iteration t, and then apply our expressions for the Cartesian-product and convex-hull regrets inductively then it follows from (4) that the loss associated with each action is exactly the negative counterfactual value. Finally, uniform sampling from the set of treeplex strategies as per Theorem 1 implies the per-information-set averaging used in standard CFR presentations (Zinkevich et al. 2007).

More Complex Operations

Conceptually, all the operations we have studied in the previous sections (that is, convex hulls, Cartesian products, Minkowski sums, ...) take one or more sets and produce a regret minimizer for a larger set. Instead, in this section we deal with operations that curtail the set of recommendations that can be output by our regret minimizer.

Constraint enforcement via Lagrangian relaxation

Suppose that we want to construct a $(\mathcal{X} \cap \{x : g(x) \le 0\}, \mathcal{L})$-regret minimizer, where g is a convex function, but we only dispose of a $(\mathcal{X}, \mathcal{L})$-regret minimizer. One natural idea is to use the latter to approximate the former, by penalizing any choice of $x \in \mathcal{X}$ such that $g(x) > 0$. In particular, it seems natural to introduce the penalized loss function

$$\tilde{\ell}^t : \mathcal{X} \ni x \mapsto \ell^t(x) + \beta^t \max\{0, g(x)\},$$

where β^t is a (large) positive constant that can change over time. This approach is reminiscent of Lagrangian relaxation.

Of course, the loss function $\tilde{\ell}^t$ is not linear, and as such it cannot be handled as is by our $(\mathcal{X}, \mathcal{L})$-regret minimizer. However, as we have observed in the section "Universality of linear losses", the regret induced by $\tilde{\ell}^t$ can be minimized by our $(\mathcal{X}, \mathcal{L})$-regret minimizer if that observes the "linearized" loss function

$$\tilde{\ell}^t_\diamond : \mathcal{X} \ni x \mapsto \ell^t(x) + \beta^t_\diamond \langle \partial g(x^t), x \rangle,$$

where

$$\beta^t_\diamond := \begin{cases} \beta^t & \text{if } g(x^t) > 0 \\ 0 & \text{otherwise.} \end{cases}$$

(This definition of $\beta^t_\diamond$ is a direct consequence of Danskin's theorem.) It is interesting to see what a small cumulative regret guarantees in terms of satisfaction of the constraint $g(x) \le 0$. In particular, let $R^T_{(\mathcal{X},\mathcal{L})}$ be the cumulative regret of our $(\mathcal{X}, \mathcal{L})$-regret minimizer. Then, introducing $\mathcal{X}_g := \mathcal{X} \cap \{x : g(x) \le 0\}$ and $\tau_g := \{t \in \{1, \dots, T\} : g(x^t) > 0\}$,

$$\begin{aligned} R^T_{(\mathcal{X},\mathcal{L})} &\ge \sum_{t=1}^T \ell^t(x^t) + \sum_{t \in \tau_g} \beta^t g(x^t) \\ &\qquad - \min_{\hat{x} \in \mathcal{X}} \left\{ \sum_{t=1}^T \ell^t(\hat{x}) + \left(\sum_{t=1}^T \beta_i \right) \max\{0, g(\hat{x})\} \right\} \\ &\ge \left(\sum_{t=1}^T \ell^t(x^t) - \min_{\hat{x} \in \mathcal{X}_g} \sum_{t=1}^T \ell^t(\hat{x}) \right) + \sum_{t \in \tau_g} \beta^t g(x^t). \quad (5) \end{aligned}$$

where the first inequality is by (1) and the second inequality comes from restricting the domain of the minimization.[2] Thus, if the β^t are sufficiently large, we can guarantee that the average recommendation $\bar{x} := \frac{1}{B}(\beta^1 x^1 + \dots + \beta^T x^T)$ where $B := \beta^1 + \dots + \beta^T$ satisfies

$$\begin{aligned} \max\{0, g\}(\bar{x}) &\le \frac{1}{B} \sum_{t=1}^T \beta^t \max\{0, g\}(x^t) = \frac{1}{B} \sum_{t \in \tau_g} \beta^t g(x^t) \\ &\le \frac{1}{B} \left(R^T_{(\mathcal{X},\mathcal{L})} + \min_{\hat{x} \in \mathcal{X}_g} \left\{ \sum_{t=1}^T \ell^t(\hat{x} - x^t) \right\} \right), \end{aligned}$$

where the first inequality follows by convexity of the max-function $\max\{0, g\}$, while the second inequality follows by (5). In particular, if $B = \sum_{t=1}^T \beta_i \gg TLD$, where L is an upper bound on the norm of any loss function $\ell^{(\cdot)}$ and D is an upper bound on the diameter of $\mathcal{X}_g$, $\max\{0, g(\bar{x})\} \to 0$ as $T \to \infty$. Thus, one practical choice would be to have $\beta^t = \bar{\beta}$ for some constant $\bar{\beta}$. If the number of iterations is known ahead of time then $\bar{\beta}$ can be chosen to guarantee a specific bound on the violation of $g(x) \le 0$. Alternatively, the β^t can be chosen by a regret minimizer which sees the violation as its loss.

We conclude the subsection by pointing out two major drawbacks of this approach for intersections:

1. The recommendation only converges to the domain $\mathcal{X}_g$ *on average*. Thus, formally the construction does *not* show how to construct a $(\mathcal{X}_g, \mathcal{L})$-regret minimizer, but only how to "approximate" one. The next subsection shows how to solve this problem by providing a generic construction for a $(\mathcal{X} \cap \mathcal{Y}, \mathcal{L})$-regret minimizer, a strictly more general task.
2. The construction we just gave requires large penalization factors β^t in order to work properly. However, this means that the norm of $\tilde{\ell}^t_\diamond$ is extremely large, making the task of minimizing the regret $R^T_{(\mathcal{X},\mathcal{L})}$ significantly harder.

[2]At this point, it might seem tempting to recognize in the term in parentheses the cumulative regret of a $(\mathcal{X}_g, \mathcal{L})$-regret minimizer. This would be incorrect, since the recommendations x^t are *not* guaranteed to satisfy $g(x^t) \le 0$.

Intersection with a closed convex set

We consider the problem of constructing a $(\mathcal{X} \cap \mathcal{Y}, \mathcal{L})$-regret minimizer from a $(\mathcal{X}, \mathcal{L})$-regret minimizer, where $\mathcal{Y}$ is a closed convex set. As it turns out, the task is always possible, and can be carried out by letting the $(\mathcal{X}, \mathcal{L})$-regret minimizer give recommendations in $\mathcal{X}$, and then *projecting* them onto the intersection $\mathcal{X} \cap \mathcal{Y}$.

For ease of notation, we will denote the projection $\pi_{\mathcal{X} \cap \mathcal{Y}}(x)$ of a point $x \in \mathcal{X}$ onto $\mathcal{X} \cap \mathcal{Y}$ as $[x]$; it is a well-known fact that such projection always exists and is unique since $\mathcal{X} \cap \mathcal{Y}$ is closed and convex. The cumulative regret of the $(\mathcal{X} \cap \mathcal{Y}, \mathcal{L})$-minimizer is then

$$\begin{aligned} R^T_{(\mathcal{X} \cap \mathcal{Y},\mathcal{L})} &= \sum_{t=1}^T \ell^t([x^t]) - \min_{\hat{x} \in \mathcal{X} \cap \mathcal{Y}} \left\{ \sum_{t=1}^T \ell^t(\hat{x}) \right\} \\ &= \sum_{t=1}^T \ell^t([x^t] - x^t) - \min_{\hat{x} \in \mathcal{X} \cap \mathcal{Y}} \left\{ \sum_{t=1}^T \ell^t(\hat{x} - x^t) \right\}, \end{aligned}$$

where the second equality holds by linearity of ℓ^t. Applying the variational inequality for projections, that is

$$\langle x^t - [x^t], \hat{x} - [x^t] \rangle \le 0 \quad \forall \hat{x} \in \mathcal{X} \cap \mathcal{Y},$$

we find that, provided $\alpha^t \ge 0$ for all t,

$$\begin{aligned} \min_{\hat{x} \in \mathcal{X} \cap \mathcal{Y}} \left\{ \sum_{t=1}^T \ell^t(\hat{x} - x^t) \right\} \ge \min_{\hat{x} \in \mathcal{X} \cap \mathcal{Y}} &\left\{ \sum_{t=1}^T \ell^t(\hat{x} - x^t) \right. \\ &\left. + \sum_{t=1}^T \alpha^t \langle x^t - [x^t], \hat{x} - [x^t] \rangle \right\}. \end{aligned}$$

The role of the α^t coefficients is to penalize choices of x^t that are in $\mathcal{X} \setminus \mathcal{Y}$. In particular, assume that

$$\sum_{t=1}^T \alpha^t \|[x^t] - x^t\|^2 \ge \sum_{t=1}^T \ell_t([x^t] - x^t). \quad (6)$$

Then, we can write

$$\begin{aligned} R^T_{(\mathcal{X} \cap \mathcal{Y},\mathcal{L})} \le \sum_{t=1}^T \alpha^t \|[x^t] - x^t\|^2 - \min_{\hat{x} \in \mathcal{X} \cap \mathcal{Y}} &\left\{ \sum_{t=1}^T \ell^t(\hat{x} - x^t) \right. \\ &\left. + \sum_{t=1}^T \alpha^t \langle x^t - [x^t], \hat{x} - [x^t] \rangle \right\}. \\ \le \left(\sum_{t=1}^T \ell^t(x^t) + \alpha^t \langle x^t - [x^t], x^t \rangle \right)& \\ - \min_{\hat{x} \in \mathcal{X}} &\left\{ \sum_{t=1}^T \ell^t(\hat{x}) + \alpha^t \langle x^t - [x^t], \hat{x} \rangle \right\}, \end{aligned}$$

which is the regret observed by a $(\mathcal{X}, \mathcal{L})$-regret minimizers that at each time instant t observes the linear loss function

$$\tilde{\ell}^t : x \mapsto \ell^t(x) + \alpha^t \langle x^t - [x^t], x \rangle. \quad (7)$$

Hence, as long as condition (6) holds, the regret circuit of Figure 9 is guaranteed to work. On the other hand, condition (6) can be trivially satisfied by the deterministic choice

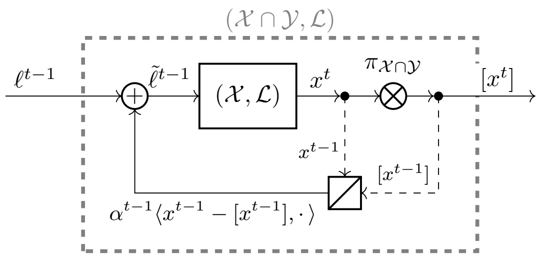

Figure 9: Regret circuit representing the construction of a $(\mathcal{X} \cap \mathcal{Y}, \mathcal{L})$-regret minimizer using a $(\mathcal{X}, \mathcal{L})$-regret minimizer.

$$\alpha^t = \begin{cases} 0 & \text{if } x^t \in \mathcal{X} \cap \mathcal{Y} \\ \max\left\{0, \dfrac{\ell^t([x^t] - x^t)}{\|[x^t] - x^t\|^2}\right\} & \text{otherwise.} \end{cases}$$

The fact that α_t can be arbitrarily large (when x^t and $[x^t]$ are very close) should not worry. Indeed, α^t is only used in $\tilde{\ell}^t$ (Equation (7)) and is always multiplied by a term whose magnitude grows proportionally with the distance between x^t and $[x^t]$. In fact, $\tilde{\ell}^t$ is bounded by

$$\|\tilde{\ell}^t\| \leq \|\ell^t\| + \frac{\ell^t([x^t] - x^t)}{\|[x^t] - x^t\|} \leq \|\ell^t\| + \|\ell^t\| = 2\|\ell^t\|.$$

In other words, our construction dilates the loss function bound by (at most) a factor 2.

Application: CFR with Strategy Constraints

When solving EFGs there may be a need to add additional constraints beyond simply computing feasible strategies. Such needs can arise for several reasons:

- Opponent modeling. If we observe repeated play from an opponent we may wish to constrain our model of their strategy space to reflect such observations. Since observations can in general be consistent with several information sets belonging to the opponent this requires adding constraints that span across information sets.
- Bounding the probability of events. For example in a patroling game we may wish to ensure that a patrol returns to its base at the end of the game with high probability.
- Nash-equilibrium refinement computation. Refinements can be computed, or approximated, via perturbation of the strategy space of each player. For *extensive-form perfect equilibrium* this can be done via lower-bounding the probability of each action at each information set (Farina and Gatti 2017), which can be handled with small modifications to standard CFR or first-order methods (Farina, Kroer, and Sandholm 2017; Kroer, Farina, and Sandholm 2017). However the, arguably superior, *quasi-perfect equilibrium* requires perturbations on the probability of sequences of action (Miltersen and Sørensen 2010), which requires strategy constraints that cross information sets.

All the points above potentially require adding strategy space constraints that span across multiple information sets. Such constraints break the recursive nature of the treeplex, and are thus not easily incorporated into standard regret-minimization or first-order methods for EFG solving. Davis, Waugh, and Bowling (2018) propose a Lagrangian-relaxation approach called *Constrained CFR* (CCFR): each strategy constraint is added to the objective with a Lagrangian multiplier, and a regret minimizer is used to penalize violation of the strategy constraints. The authors prove that if the regret minimizer for the Lagrange multipliers has the optimal Lagrangian multipliers as part of their strategy space then this approach converges to a solution to the strategy-space-constrained game. However, they do not prove a bound on the size of this set, and so in general require repeatedly running CCFR in order to binary search the size of this strategy space.

Two alternative variants of CFR for EFGs with strategy constraints can be obtained by our framework. We already proved that CFR can be constructed from our framework. In order to additionally support strategy constraints we can apply either our method for Lagrangian relaxation of $\mathcal{X}$ and a constraint $g(x) \leq 0$, or we can apply our intersection approach, which requires projection. Our Lagrangian approach yields an algorithm similar to the CCFR algorithm. However, our approach has some major improvements, perhaps most importantly we can pick the size of our Lagrangian multipliers ahead of time, and so do not require any binary search in order to find the correct Lagrangian multipliers. Our approach also supports, but does not require, regret minimization for the Lagrangian multipliers, since we put no constraints on the form of the β^t multipliers. If we are willing to pay the cost of projecting onto $\mathcal{X} \cap \{x : g(x) \leq 0\}$ then a very different regret-minimization approach can be obtained by combining CFR with our intersection operation. This approach has the major advantage that it produces feasible iterates, but projection may not always be desirable for computational reasons.

Conclusion and Future Research

We developed a calculus of regret minimization, which enables the construction of regret minimizers for complex convex sets that can be expressed as a series of convexity-preserving operations on simpler sets. We showed that our calculus can be used to construct the CFR algorithm directly, as well as several variants of the algorithm for the case where we have strategy constraints. However, our regret calculus is much more broadly applicable than just EFGs: it applies to any setting where the decision space can be expressed via the convexity-preserving operations that we support. In the future we plan to investigate novel applications of our regret calculus. One potential application would be online portfolio selection with additional constraints (e.g. capacity constraints across industries); our framework makes it easy to construct such a regret minimizer from any standard online-portfolio-selection algorithm.

We also plan to use our construction for CFR with strategy constraints to test whether the fact that we have explicit bounds on the Lagrangian multipliers leads to practically superior algorithm as compared to Davis, Waugh, and Bowling (2018). We also plan to investigate how much better the projection-based algorithm does in terms of iteration complexity, and whether that makes up for the additional cost of projection.

References

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

[2004] Boyd, S., and Vandenberghe, L. 2004. *Convex Optimization*. Cambridge University Press.

[2017] Brown, N., and Sandholm, T. 2017. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science* eaao1733.

[2017] Brown, N.; Kroer, C.; and Sandholm, T. 2017. Dynamic thresholding and pruning for regret minimization. In *AAAI Conference on Artificial Intelligence (AAAI)*.

[2018] Davis, T.; Waugh, K.; and Bowling, M. 2018. Solving large extensive-form games with strategy constraints. *arXiv preprint arXiv:1809.07893*.

[2017] Farina, G., and Gatti, N. 2017. Extensive-form perfect equilibrium computation in two-player games. In *AAAI Conference on Artificial Intelligence (AAAI)*.

[2017] Farina, G.; Kroer, C.; and Sandholm, T. 2017. Regret minimization in behaviorally-constrained zero-sum games. In *International Conference on Machine Learning (ICML)*.

[2019] Farina, G.; Kroer, C.; and Sandholm, T. 2019. Online convex optimization for sequential decision processes and extensive-form games. In *AAAI Conference on Artificial Intelligence (AAAI)*.

[2006] Grant, M.; Boyd, S.; and Ye, Y. 2006. Disciplined convex programming. In *Global optimization*. Springer. 155–210.

[2008] Grant, M.; Boyd, S.; and Ye, Y. 2008. Cvx: Matlab software for disciplined convex programming.

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

[2010] 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).

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

[2018] Kroer, C.; Waugh, K.; Kılınç-Karzan, F.; and Sandholm, T. 2018. Faster algorithms for extensive-form game solving via improved smoothing functions. *Mathematical Programming* 1–33.

[2017] Kroer, C.; Farina, G.; and Sandholm, T. 2017. Smoothing method for approximate extensive-form perfect equilibrium. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

[2018] Ling, C. K.; Fang, F.; and Kolter, J. Z. 2018. What game are we playing? End-to-end learning in normal and extensive form games. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

[2011] McMahan, B. 2011. Follow-the-regularized-leader and mirror descent: Equivalence theorems and l1 regularization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 525–533.

[2010] Miltersen, P. B., and Sørensen, T. B. 2010. Computing a quasi-perfect equilibrium of a two-player game. *Economic Theory* 42(1).

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

[2015] 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 (IJCAI)*.

[1996] von Stengel, B. 1996. Efficient computation of behavior strategies. *Games and Economic Behavior* 14(2):220–246.

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

[2003] Zinkevich, M. 2003. Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning (ICML)*, 928–936.