Regret Circuits: Composability of Regret Minimizers

Gabriele Farina [1] **Christian Kroer** [2] **Tuomas Sandholm** [1 3 4 5]

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 composite convex sets that are obtained from convexity-preserving operations on simpler convex sets. We show that local regret minimizers for the simpler sets can be combined with additional regret minimizers into an aggregate regret minimizer for the composite set. As one application, we show that the CFR framework can be constructed easily from our framework. We also show ways to include curtailing (constraining) operations into our framework. For one, they enable the construction of CFR generalization for extensive-form games with general convex strategy constraints that can cut across decision points.

1. Introduction

Counterfactual regret minimization (CFR) (Zinkevich et al., 2007), and its newer variants (Lanctot et al., 2009; Brown & Sandholm, 2015a; Tammelin et al., 2015; Brown et al., 2017; Brown & Sandholm, 2017a; 2019), 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 & Sandholm (2017c) used CFR variants, along with other scalability techniques such as real-time endgame solving (Ganzfried & Sandholm, 2015; Burch et al., 2014; Moravcik et al., 2016; Brown & Sandholm, 2017b) and automated action abstraction (Brown & Sandholm, 2014), to create *Libratus*, an AI that beat top human specialist professionals at the larger game of heads-up no-limit Texas hold'em. Moravčík et al. (2017) also used CFR variants and endgame solving to beat professional human players at that game.

CFR and its newer variants are usually presented as algorithms for finding an approximate 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 to which they apply are sometimes referred to as *treeplexes* (Hoda et al., 2010; Kroer et al., 2015; 2018). In this view, those algorithms specify 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 et al. (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. This decomposition into individual optimization problems can be beneficial because it enables the use of 1) different algorithms for different parts of the search space, 2) specialized techniques for different parts of the problem, such as warm starting (Brown & Sandholm, 2014; 2015b; 2016) and pruning (Lanctot et al., 2009; Brown & Sandholm, 2015a; Brown et al., 2017; Brown & Sandholm, 2017a), and 3) approximation of some parts of the space.

In this paper we introduce a general methodology for composing regret minimizers. We derive a set of rules for how regret minimizers can be constructed for composite convex sets via a *calculus* of regret minimization: given regret minimizers for convex sets $\mathcal{X}, \mathcal{Y}$ 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 composite set.

Our approach treats the regret minimizers for individual

[1] Computer Science Department, Carnegie Mellon University, Pittsburgh PA 15213 [2] IEOR Department, Columbia University, New York NY 10027 [3] Strategic Machine, Inc. [4] Strategy Robot, Inc. [5] Optimized Markets, Inc.. Correspondence to: Gabriele Farina <gfarina@cs.cmu.edu>, Christian Kroer <christian.kroer@columbia.edu>, Tuomas Sandholm <sandholm@cs.cmu.edu>.

Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

convex sets as black boxes, and builds a regret minimizer for the resulting composite set by combining the outputs of the individual regret minimizers. 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 & 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 et al., 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; we show the regret-circuit representation of every operation that we develop.

As an application, we show that the correctness and convergence rate of the CFR algorithm can be proven easily through our calculus. We also show that the recent *Constrained CFR* algorithm (Davis et al., 2019) can be constructed via our framework. Our framework enables the construction of two algorithms for that problem. The first is based on Lagrangian relaxation, and only guarantees approximate feasibility of the output strategies. The second is based on projection and guarantees exact feasibility, for the first time in any algorithm that decomposes overall regret into local regrets at decision points.

2. Regret Minimization

We will prove our results in the online learning framework called *online convex optimization* (Zinkevich, 2003) (OCO). In OCO, a decision maker repeatedly interacts with an unknown environment by making a sequence of decisions $\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots$ from a convex and compact set $\mathcal{X} \subseteq \mathbb{R}^n$. After each decision $\boldsymbol{x}^t$, the decision maker faces a convex *loss function* $\ell^t(\boldsymbol{x}^t)$, which is unknown to the decision maker until after the decision is made. So, we are constructing a device that supports two operations: (i) it provides the next decision $\boldsymbol{x}^{t+1} \in \mathcal{X}$ and (ii) it receives/observes the convex loss function ℓ^t used to "evaluate" decision $\boldsymbol{x}^t$. The decision making is *online* in the sense that the next decision, $\boldsymbol{x}^{t+1}$, is based only on the previous decisions $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^t$ and corresponding observed loss functions $\ell^1, \ldots, \ell^t$.

The quality of the device is measured by its *cumulative regret*, which is the difference between the loss cumulated by the sequence of decisions $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^T$ and the loss that would have been cumulated by playing the best-in-hindsight time-independent decision $\hat{\boldsymbol{x}}$. Formally, the cumulative regret up to time T is

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

Above we introduce the new notation of a subscript $(\mathcal{X}, \mathcal{F})$ to be explicit about the domain of the decisions $\{\boldsymbol{x}^t\}$ and the domain of the loss functions $\{\ell^t\}$, respectively. This turns out to be important because we will study composability of devices with different domains.

The device is called a *regret minimizer* if it satisfies the desirable property of *Hannan consistency*: the average regret approaches zero, that is, $R^T_{(\mathcal{X},\mathcal{F})}$ grows *sublinearly* in T. Formally, in our notation, we have the following definition.

Definition 1 ($(\mathcal{X}, \mathcal{F})$-regret minimizer)**.** *Let $\mathcal{X}$ be a convex and compact 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. An* $(\mathcal{X}, \mathcal{F})$-regret minimizer *is a function that selects the next decision $\boldsymbol{x}^{t+1} \in \mathcal{X}$ given the history of decisions $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^t$ and observed loss functions $\ell^1, \ldots, \ell^t \in \mathcal{F}$, so that the cumulative regret $R^T_{(\mathcal{X},\mathcal{F})} = o(T)$.*

2.1. Universality of Linear Loss Functions

Regret minimizers for linear loss functions are in a sense *universal*: one can construct a regret minimizer for convex loss functions from any regret minimizer for linear loss functions (e.g., McMahan (2011)). The crucial insight is that the regret that we are trying to minimize, $R^T_{(\mathcal{X},\mathcal{F})}$, is bounded by the regret of a $(\mathcal{X}, \mathcal{L})$-regret minimizer that, at each time t, observes as its loss function a tangent plane of ℓ^t at the most recent decision $\boldsymbol{x}^t$. Thus we can minimize $R^T_{(\mathcal{X},\mathcal{F})}$ by minimizing $R^T_{(\mathcal{X},\mathcal{L})}$.

Formally, let $\partial\ell^t(\boldsymbol{x}^t)$ be any subgradient of ℓ^t at $\boldsymbol{x}^t$. By convexity of ℓ^t,

$$\ell^t(\hat{\boldsymbol{x}}) \geq \ell^t(\boldsymbol{x}^t) + \langle \partial\ell^t(\boldsymbol{x}^t), \hat{\boldsymbol{x}} - \boldsymbol{x}^t \rangle \quad \forall \hat{\boldsymbol{x}} \in \mathcal{X},$$

and, substituting into (1), we obtain

$$R^T_{(\mathcal{X},\mathcal{F})} \leq \sum_{t=1}^{T} \langle \partial\ell^t(\boldsymbol{x}^t), \boldsymbol{x}^t \rangle - \min_{\hat{x} \in \mathcal{X}} \left\{ \sum_{t=1}^{T} \langle \partial\ell^t(\boldsymbol{x}^t), \hat{\boldsymbol{x}} \rangle \right\}, \tag{2}$$

where the right hand side is $R^T_{(\mathcal{X},\mathcal{L})}$, the regret cumulated by a device that observes the *linear* loss functions $\langle \partial\ell^t(\boldsymbol{x}^t), \cdot \rangle$.[1]

[1] A downside of this approach is that (2) is an inequality, not an equality. When we use a linearization of the loss function at each decision point, we introduce error. This can cause regret to be minimized more slowly than somehow working on the nonlinear loss functions directly. Nevertheless, we obtain a regret minimizer for the original problem.

2.2. Connection to Convex-Concave Saddle-Point Problems and Game Theory

In this subsection we review how regret minimization can be used to compute solutions to *regularized bilinear saddle-point problems*, that is solutions to problems of the form

$$\min_{\boldsymbol{x}\in\mathcal{X}} \max_{\boldsymbol{y}\in\mathcal{Y}} \{\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} + d_1(\boldsymbol{x}) - d_2(\boldsymbol{y})\}, \qquad (3)$$

where $\mathcal{X}, \mathcal{Y}$ are closed convex sets, and d_1, d_2 are convex functions. This general formulation allows us to capture, among other settings, several game-theoretical applications such as computing Nash equilibria in two-player zero-sum games. In that setting, d_1 and d_2 are the constant zero functions, $\mathcal{X}$ and $\mathcal{Y}$ are convex polytopes whose description is provided by the *sequence-form constraints*, and $\boldsymbol{A}$ is a real payoff matrix (von Stengel, 1996).

In order to use regret minimization to solve problems of the form (3), we consider the loss functions $\ell^t_{\mathcal{X}} : \boldsymbol{x} \mapsto (-\boldsymbol{A}\boldsymbol{y}^t)^\top \boldsymbol{x} + d_1(\boldsymbol{x})$, $\ell^t_{\mathcal{Y}} : \boldsymbol{y} \mapsto (\boldsymbol{A}^\top \boldsymbol{x}^t)^\top \boldsymbol{y} + d_2(\boldsymbol{y})$. The error metric that we use is the *saddle-point residual* (or *gap*) ξ of $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$, defined as $\xi(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) := \max_{\hat{\boldsymbol{y}}\in\mathcal{Y}}\{d_1(\bar{\boldsymbol{x}}) - d_2(\hat{\boldsymbol{y}}) + \langle \bar{\boldsymbol{x}}, \boldsymbol{A}\hat{\boldsymbol{y}}\rangle\} - \min_{\hat{\boldsymbol{x}}\in\mathcal{X}}\{d_1(\hat{\boldsymbol{x}}) - d_2(\bar{\boldsymbol{y}}) + \langle \hat{\boldsymbol{x}}, A\bar{\boldsymbol{y}}\rangle\}$. The following folk theorem shows that the average of a sequence of regret-minimizing strategies for the choice of losses above leads to a bounded saddle-point residual (see, for example, Farina et al. (2019) for a proof).

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

When $d_1 \equiv d_2 \equiv 0$ and $\mathcal{X}, \mathcal{Y}$ are the players' sequence-form strategy spaces, Theorem 1 asserts that the average strategy profile produced by the regret minimizers is an $(\epsilon_1 + \epsilon_2)$-Nash equilibrium. Different choices of the regularizing functions d_1 and d_2 can be used to solve for strategies in other game-theoretic applications as well, such as computing a normal-form quantal-response equilibrium (Ling et al., 2018; Farina et al., 2019). Farina et al. (2019) study opponent exploitation where the goal is to compute a best response, subject to a penalty for moving away from a precomputed Nash equilibrium strategy; this is captured by having d_1 or d_2 include a penalty term that penalizes distance from the precomputed strategy. Farina et al. (2017) and Kroer et al. (2017) study constraints on individual decision points, and Davis et al. (2019) study additional constraints on the overall EFG polytopes $\mathcal{X}, \mathcal{Y}$. Regret minimization in those settings requires regret minimizers that can operate on more general domains $\mathcal{X}, \mathcal{Y}$ than the sequence form. In this paper we show how one can construct regret minimizers for any convex domain that can be expressed from simpler domains using convexity-preserving operations.

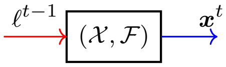
3. Regret Circuits

In this paper, we introduce *regret circuits*. They are composed of independent regret minimizers connected by wires on which the loss functions and decisions can flow. Regret circuits encode how the inputs and outputs of multiple regret minimizers can be combined to achieve a goal, in a divide-and-conquer fashion, and help simplify the design and analysis of regret-minimizing algorithms. Using the constructions that we will present, one can compose different regret circuits and produce increasingly complex circuits.

The regret circuits approach has several advantages that make it appealing when compared to other, more monolithic, approaches. For one, by treating every regret minimizer that appears in a regret circuit as an independent black box, our approach enables one to select the best individual algorithm for each of them. Second, our framework is amenable to pruning or warm-starting techniques in different parts of the circuit, and substituting one or more parts of the circuit with an approximation. Finally, regret circuits can be easily run in distributed and parallel environments.

We will express regret circuits pictorially through block diagrams. We will use the following conventions when drawing regret circuits:

- an $(\mathcal{X}, \mathcal{F})$-regret minimizer is drawn as a box

  where the input (red) arrow represents the loss at a generic time $t-1$, while the output (blue) arrow represents the decision produced at time t;
- the symbol ⧄ is used to denote an operation that constructs or manipulates one or more loss functions;
- the symbol ⊗ is used to denote an operation that combines or manipulates one or more decisions;
- the symbol ⊕ denotes an *adder*, that is a node that outputs the sum of all its inputs;
- dashed arrows denote decisions that originate from the previous iteration.

As an example, consider the construction of Section 2.1, where we showed how one can construct a regret minimizer for generic convex loss functions from any regret minimizer for linear loss functions. Figure 1 shows how that construction can be expressed as a regret circuit.

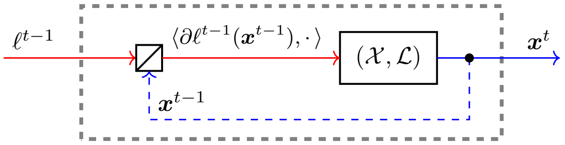

ℓ^{t-1}
$\langle \partial\ell^{t-1}(\boldsymbol{x}^{t-1}), \cdot\rangle$
$(\mathcal{X}, \mathcal{L})$
$\boldsymbol{x}^t$
$\boldsymbol{x}^{t-1}$

Figure 1. Regret circuit representing the construction of an $(\mathcal{X}, \mathcal{F})$-regret minimizer using an $(\mathcal{X}, \mathcal{L})$-regret minimizer.

4. Circuit Construction for Operations that Enlarge or Transform Sets

In this section, we begin the construction of our *calculus of regret minimization*. Given regret minimizers for two closed convex sets $\mathcal{X}$ and $\mathcal{Y}$, we show how to construct a regret minimizer for sets obtained via convexity-preserving operations on $\mathcal{X}$ and $\mathcal{Y}$. In this section, we focus on operations that take one or more sets and produce a regret minimizer for a larger set—this is the case, for instance, of convex hulls, Cartesian products, and Minkowski sums.

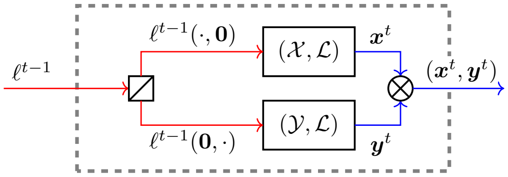
As explained in Section 2.1, one can extend any $(\mathcal{X}, \mathcal{L})$-regret minimizer to handle more expressive loss functionals. Therefore, in the rest of the paper, we focus on $(\mathcal{X}, \mathcal{L})$-regret minimizers.

4.1. Cartesian Product

In this section, we show how to combine an $(\mathcal{X}, \mathcal{L})$- and a $(\mathcal{Y}, \mathcal{L})$-regret minimizer to form an $(\mathcal{X} \times \mathcal{Y}, \mathcal{L})$-regret minimizer. Any linear function $\ell : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ can be written as $\ell(\boldsymbol{x}, \boldsymbol{y}) = \ell_{\mathcal{X}}(\boldsymbol{x}) + \ell_{\mathcal{Y}}(\boldsymbol{y})$ where the linear functions $\ell_{\mathcal{X}} : \mathcal{X} \to \mathbb{R}$ and $\ell_{\mathcal{Y}} : \mathcal{Y} \to \mathbb{R}$ are defined as $\ell_{\mathcal{X}} : \boldsymbol{x} \mapsto \ell(\boldsymbol{x}, \boldsymbol{0})$ and $\ell_{\mathcal{Y}} : \boldsymbol{y} \mapsto \ell(\boldsymbol{0}, \boldsymbol{y})$. It is immediate to verify that

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

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 decisions, as in Figure 2.



ℓ^{t-1}
$\ell^{t-1}(\cdot, \boldsymbol{0})$
$(\mathcal{X}, \mathcal{L})$
$\boldsymbol{x}^t$
$(\boldsymbol{x}^t, \boldsymbol{y}^t)$
$(\mathcal{Y}, \mathcal{L})$
$\ell^{t-1}(\boldsymbol{0}, \cdot)$
$\boldsymbol{y}^t$

Figure 2. Regret circuit for the Cartesian product $\mathcal{X} \times \mathcal{Y}$.

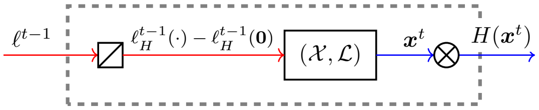
4.2. Affine Transformation and Minkowski Sum

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

Since every $\boldsymbol{y} \in H(\mathcal{X})$ can be written as $\boldsymbol{y} = H(\boldsymbol{x})$ for some $\boldsymbol{x} \in \mathcal{X}$, the cumulative regret for an $(H(\mathcal{X}), \mathcal{L})$-regret minimizer can be expressed as

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

Since ℓ^t and H are affine, their composition $\ell^t_H := \ell^t \circ H$ is also affine. Hence, $R^T_{(H(\mathcal{X}),\mathcal{L})}$ is the same regret as an $(\mathcal{X}, \mathcal{L})$-regret minimizer that observes the linear function $\ell^t_H(\cdot) - \ell^t_H(\boldsymbol{0})$ instead of ℓ^t. The construction is summarized by the circuit in Figure 3.



ℓ^{t-1}
$\ell^{t-1}_H(\cdot) - \ell^{t-1}_H(\boldsymbol{0})$
$(\mathcal{X}, \mathcal{L})$
$\boldsymbol{x}^t$
$H(\boldsymbol{x}^t)$

Figure 3. Regret circuit for the image $H(\mathcal{X})$ of $\mathcal{X}$ under the affine transformation H.

As an application, we use the above construction to form a regret minimizer for the Minkowski sum $\mathcal{X} + \mathcal{Y} := \{\boldsymbol{x} + \boldsymbol{y} : \boldsymbol{x} \in \mathcal{X}, \boldsymbol{y} \in \mathcal{Y}\}$ of two sets. Indeed, note that $\mathcal{X} + \mathcal{Y} = \sigma(\mathcal{X} \times \mathcal{Y})$, where $\sigma : (\boldsymbol{x}, \boldsymbol{y}) \mapsto \boldsymbol{x} + \boldsymbol{y}$ is a linear map. Hence, we can combine the construction in this section together with the construction of the Cartesian product (Figure 2). See Figure 8 in the appendix for the resulting circuit.

4.3. Convex Hull

In this section, we show how to combine an $(\mathcal{X}, \mathcal{L})$- 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, $\mathrm{co}\{\mathcal{X}, \mathcal{Y}\} = \{\lambda_1 \boldsymbol{x} + \lambda_2 \boldsymbol{y} \ : \ \boldsymbol{x} \in \mathcal{X}, \boldsymbol{y} \in \mathcal{Y}, (\lambda_1, \lambda_2) \in \Delta^2\}$, and $\Delta^2 := \{(\lambda_1, \lambda_2) \in \mathbb{R}^2_+ : \lambda_1 + \lambda_2 = 1\}$ is the two-dimensional *simplex*. We can think of a $(\mathrm{co}\{\mathcal{X}, \mathcal{Y}\}, \mathcal{L})$-regret minimizer as picking a triple $(\boldsymbol{\lambda}^t, \boldsymbol{x}^t, \boldsymbol{y}^t) \in \Delta^2 \times \mathcal{X} \times \mathcal{Y}$ at each time point t. Using the linearity of the loss functions,

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

Now, we make two crucial observations. First,

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

since all components of $\hat{\boldsymbol{\lambda}}$ are non-negative. Second, the inner minimization problem over $\mathcal{X}$ is related to the cumulative regret $R^T_{(\mathcal{X},\mathcal{L})}$ of the $(\mathcal{X}, \mathcal{L})$-regret minimizer that observes the loss functions ℓ^t as follows:

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

(An analogous relationship holds for $\mathcal{Y}$.) Hence,

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

Using the fact that $\min(f+g) \geq \min f + \min g$, and introducing the quantity

$$R^T_{(\Delta^2,\mathcal{L})} := \left(\sum_{t=1}^{T} \lambda_1^t \ell^t(\boldsymbol{x}^t) + \lambda_2^t \ell^t(\boldsymbol{y}^t)\right) - \min_{\hat{\boldsymbol{\lambda}}\in\Delta^2}\left\{\left(\sum_{t=1}^{T} \hat{\lambda}_1 \ell^t(\boldsymbol{x}^t) + \hat{\lambda}_2 \ell^t(\boldsymbol{y}^t)\right)\right\},$$

we conclude that

$$R^T_{(\mathrm{co}\{\mathcal{X},\mathcal{Y}\},\mathcal{L})} \leq R^T_{(\Delta^2,\mathcal{L})} + \max\{R^T_{(\mathcal{X},\mathcal{L})}, R^T_{(\mathcal{Y},\mathcal{L})}\}. \quad (4)$$

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

$$\ell^t_\lambda : \Delta^2 \ni (\lambda_1, \lambda_2) \mapsto \lambda_1 \ell^t(\boldsymbol{x}^t) + \lambda_2 \ell^t(\boldsymbol{y}^t). \quad (5)$$

Intuitively, this means that in order to make "good decisions" 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 decisions in $\mathcal{X}$ and $\mathcal{Y}$ respectively, and then use a third regret minimizer that decides how to "mix" the two outputs. This way, we break the task of picking the next recommended triple $(\boldsymbol{\lambda}^t, \boldsymbol{x}^t, \boldsymbol{y}^t)$ into three different subproblems, two of which can be run independently. Equation (4) 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 4 shows the regret circuit that corresponds to our construction above.

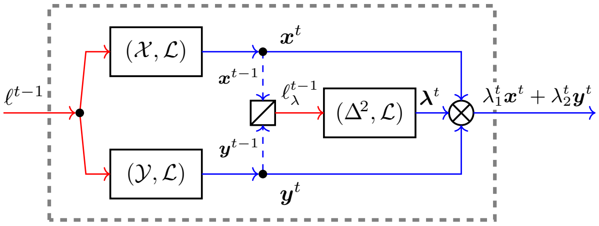


Figure 4. Regret circuit for the convex hull $\mathrm{co}\{\mathcal{X},\mathcal{Y}\}$. The loss function ℓ^t_λ is defined in Equation (5).

Extending to multiple sets The construction shown in Figure 4 can be extended to handle the convex hull $\mathrm{co}\{\mathcal{X}_1,\dots,\mathcal{X}_n\}$ of n sets as follows. First, the input loss function ℓ^{t-1} is fed into all the $(\mathcal{X}_i,\mathcal{L})$-regret minimizers $(i = 1,\dots,n)$. Then, the loss function ℓ^t_λ, defined as

$$\ell^t_\lambda : \Delta^n \ni (\lambda_1,\dots,\lambda_n) \mapsto \lambda_1 \ell(\boldsymbol{x}_1^t) + \cdots + \lambda_n \ell(\boldsymbol{x}_n^t),$$

is input into a $(\Delta^n,\mathcal{L})$-regret minimizer, where Δ^n is the n-dimensional simplex. Finally, at each time instant t, the n decisions $\boldsymbol{x}_1^t,\dots,\boldsymbol{x}_n^t$ output by the $(\mathcal{X}_i,\mathcal{L})$-regret minimizers are combined with the decision $\boldsymbol{\lambda}^t$ output by the $(\Delta^n,\mathcal{L})$-regret minimizer to form $\lambda_1^t\boldsymbol{x}_1^t + \cdots + \lambda_n^t\boldsymbol{x}_n^t$.

V-polytopes Our construction can be directly applied to construct an $(\mathcal{X},\mathcal{L})$-regret minimizer for a V-polytope $\mathcal{X} = \mathrm{co}\{\boldsymbol{v}_1,\dots,\boldsymbol{v}_n\}$ where $\boldsymbol{v}_1,\dots,\boldsymbol{v}_n$ are n points in a Euclidean space $\mathbb{E}$. Of course, any $(\{\boldsymbol{v}_i\},\mathcal{L})$-regret minimizer outputs the constant decision $\boldsymbol{v}_i$. Hence, our construction (Figure 4) reduces to a single $(\Delta^n,\mathcal{L})$-regret minimizer that observes the (linear) loss function

$$\ell^t_\lambda : \Delta^n \ni (\lambda_1,\dots,\lambda_n) \mapsto \lambda_1 \ell^t(\boldsymbol{v}_1) + \cdots + \lambda_n \ell^t(\boldsymbol{v}_n).$$

The observation that a regret minimizer over a simplex can be used to minimize regret over a V-polytope already appeared in Zinkevich (2003), Schuurmans & Zinkevich (2016), and Farina et al. (2017, Theorem 3).

5. Application: Derivation of CFR

We now show that these constructions can be used to construct the CFR framework. The first thing to note is that the strategy space of a single player in an extensive-form game is a treeplex, which can be viewed recursively as a series of convex hull and Cartesian product operations. This perspective is also used when constructing distance functions for first-order methods for EFGs (Hoda et al., 2010; Kroer et al., 2015; 2018). 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 operation, we 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).

Figure 5 shows an example. Each information set X_i (except X_0) corresponds to a 2-dimensional convex hull over two treeplexes, one of which is always empty (that is, 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 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 $\{\mathcal{X}_a\}_{a\in A_I}$ 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 $\boldsymbol{x} \in \mathcal{X}_a$ there is a corresponding $\boldsymbol{x}' \in \mathcal{X}'_a$; $\boldsymbol{x}'$ has a 1 at the index of a, $\boldsymbol{x}$ 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 exactly the treeplex rooted at I. The Cartesian product is easy and can be done over a given set of treeplexes rooted

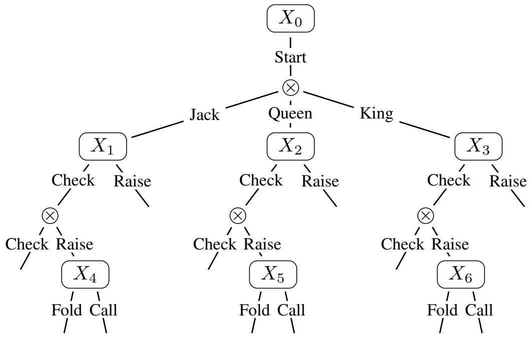


Figure 5. Treeplex for the first player in the game of Kuhn poker. Each X_i represents a convex hull over the treeplexes below, while ⊗ denotes the Cartesian product operation.

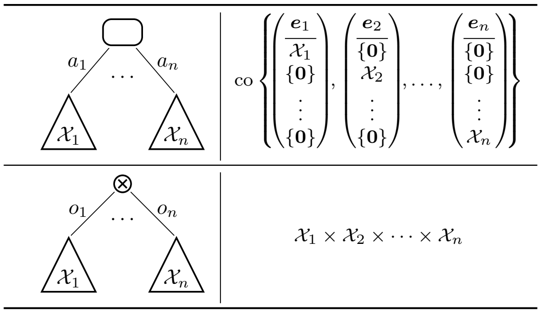

a_1
$\cdots$
a_n
$\mathcal{X}_1$
$\mathcal{X}_n$
$\operatorname{co}\left\{\begin{pmatrix}\boldsymbol{e}_1\\ \hline \mathcal{X}_1\\ \{\boldsymbol{0}\}\\ \vdots\\ \{\boldsymbol{0}\}\end{pmatrix},\begin{pmatrix}\boldsymbol{e}_2\\ \hline \{\boldsymbol{0}\}\\ \mathcal{X}_2\\ \vdots\\ \{\boldsymbol{0}\}\end{pmatrix},\dots,\begin{pmatrix}\boldsymbol{e}_n\\ \hline \{\boldsymbol{0}\}\\ \{\boldsymbol{0}\}\\ \vdots\\ \mathcal{X}_n\end{pmatrix}\right\}$
⊗
o_1
$\cdots$
o_n
$\mathcal{X}_1$
$\mathcal{X}_n$
$\mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_n$

Figure 6. Inductive treeplex construction rules. $\boldsymbol{e}_i \in \mathbb{R}^n$ contains a 1 at index i, and 0 everywhere else.

at information sets $I_1, \dots, I_n$. The inductive construction rules for the treeplex are given in Figure 6. In fact, one can prove that the ℓ_λ loss functions defined in Equation (5) are exactly the counterfactual loss functions defined in the original CFR paper (Zinkevich et al., 2007). If we use as our loss function the gradient $\boldsymbol{A}\boldsymbol{y}^t$ where $\boldsymbol{y}^t$ is the opponent's strategy at iteration t, and then apply our expressions for the Cartesian-product and convex-hull regrets inductively, it follows from (5) that the loss function associated with each action is exactly the negative counterfactual value. Finally, the average treeplex strategy as per Theorem 1 coincides with the per-information-set averaging used in standard CFR expositions (e.g., (Zinkevich et al., 2007)).

6. Circuit Construction for Operations that Constrain Sets

Unlike Section 4, in this section we deal with operations that curtail the set of decisions that can be output by our regret minimizer. Section 6.1 and 6.2 propose two different constructions, and Section 6.3 discusses the merits and drawbacks of the two.

6.1. Constraint Enforcement via Lagrangian Relaxation

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

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

where β^t is a (large) positive constant that can change over time. This approach is reminiscent of Lagrangian relaxation. 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 Section 2.1, the regret induced by $\tilde{\ell}^t$ can be minimized by our $(\mathcal{X}, \mathcal{L})$-regret minimizer if it observes the 'linearized' loss function $\tilde{\ell}^t_\diamond : \mathcal{X} \to \mathbb{R}$ defined as

$$\tilde{\ell}^t_\diamond(\boldsymbol{x}) = \ell^t(\boldsymbol{x}) + \beta^t_\diamond \langle \partial g(\boldsymbol{x}^t), \boldsymbol{x} \rangle,\ \beta^t_\diamond := \begin{cases} \beta^t & \text{if } g(\boldsymbol{x}^t) > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Figure 9 in the appendix shows the regret circuit corresponding to the construction described so far.

In the rest of this subsection we analyze in what sense small cumulative regret implies that the constraint $g(\boldsymbol{x}) \le 0$ is satisfied. Let $R^T_{(\mathcal{X},\mathcal{L})}$ be the cumulative regret of our $(\mathcal{X}, \mathcal{L})$-regret minimizer. Introducing $\mathcal{X}_g := \mathcal{X} \cap \{\boldsymbol{x} : g(\boldsymbol{x}) \le 0\}$ and $\tau_g := \{t \in \{1, \dots, T\} : g(\boldsymbol{x}^t) > 0\}$,

$$\begin{aligned} R^T_{(\mathcal{X},\mathcal{L})} &= \sum_{t=1}^T \tilde{\ell}^t_\diamond(\boldsymbol{x}^t) - \min_{\hat{\boldsymbol{x}} \in \mathcal{X}} \left\{ \sum_{t=1}^T \tilde{\ell}^t_\diamond(\hat{\boldsymbol{x}}) \right\} \\ &\ge \sum_{t=1}^T \ell^t(\boldsymbol{x}^t) + \sum_{t \in \tau_g} \beta^t g(\boldsymbol{x}^t) \\ &\qquad - \min_{\hat{\boldsymbol{x}} \in \mathcal{X}} \left\{ \sum_{t=1}^T \ell^t(\hat{\boldsymbol{x}}) + \left(\sum_{t=1}^T \beta_i \right) \max\{0, g(\hat{\boldsymbol{x}})\} \right\} \\ &\ge \left(\sum_{t=1}^T \ell^t(\boldsymbol{x}^t) - \min_{\hat{\boldsymbol{x}} \in \mathcal{X}_g} \sum_{t=1}^T \ell^t(\hat{\boldsymbol{x}}) \right) + \sum_{t \in \tau_g} \beta^t g(\boldsymbol{x}^t), \quad (6) \end{aligned}$$

where the first inequality is by (2) and the second inequality comes from simply restricting the domain of the minimization from $\mathcal{X}$ to $\mathcal{X}_g$.[2] Thus, if the β^t are sufficiently large, the average decision $\bar{\boldsymbol{x}} := \frac{1}{B}(\beta^1 \boldsymbol{x}^1 + \cdots + \beta^T \boldsymbol{x}^T)$ where $B := \beta^1 + \cdots + \beta^T$ satisfies

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

where the first inequality follows by convexity of $\max\{0, g\}$, and the second inequality follows by (6).

If $\sum_{t=1}^T \beta_i \gg TLD$, where L is an upper bound on the norm of the loss functions $\ell^1, \dots, \ell^T$ and D is an upper bound on the diameter of $\mathcal{X}$, then $\max\{0, g(\bar{\boldsymbol{x}})\} \to 0$ as $T \to \infty$, that

[2] It may tempting to recognize in the term in parentheses in (6) the cumulative regret of an $(\mathcal{X}_g, \mathcal{L})$-regret minimizer. This would be incorrect: the decisions $\boldsymbol{x}^t$ are *not* guaranteed to satisfy $g(\boldsymbol{x}^t) \le 0$.

is, the constraint is satisfied at least by the average in the limit. If L and D are known ahead of time, one practical way to guarantee $\sum_{t=1}^{T}\beta_i \gg TLD$ is to choose $\beta^t = \kappa LD$ where $\kappa > 0$ is a large constant. This guarantees that in the limit, small cumulative regret implies that the average strategy approximately satisfies the constraint and satisfies Hannan consistency. Alternatively, the β^t can be chosen by a regret minimizer which sees the constraint violation $\max\{0, g(\boldsymbol{x}^t)\}$ at time t as its loss function.

6.2. Intersection with a Closed Convex Set

In this subsection we consider constructing an $(\mathcal{X}\cap\mathcal{Y},\mathcal{L})$-regret minimizer from an $(\mathcal{X},\mathcal{L})$-regret minimizer, where $\mathcal{Y}$ is a closed convex set such that $\mathcal{X}\cap\mathcal{Y}\neq\emptyset$. As it turns out, this is always possible, and can be done by letting the $(\mathcal{X},\mathcal{L})$-regret minimizer give decisions in $\mathcal{X}$, and then *projecting* them onto the intersection $\mathcal{X}\cap\mathcal{Y}$.

We will use a *Bregman divergence* $D(\boldsymbol{y}\|\boldsymbol{x}) := d(\boldsymbol{y}) - d(\boldsymbol{x}) - \langle\nabla d(\boldsymbol{x}), \boldsymbol{y}-\boldsymbol{x}\rangle$ as our notion of distance between the points x and y, where the *distance generating function* (DGF) d is μ-strongly convex and β-smooth (that is, d is differentiable and its gradient is Lipschitz continuous with Lipschitz constant β). Our construction makes no further assumptions on d, so the most appropriate DGF can be used for the application at hand. When $d(\boldsymbol{x}) = \|\boldsymbol{x}\|_2^2$ we obtain $D(\boldsymbol{y}\|\boldsymbol{x}) = \|\boldsymbol{y}-\boldsymbol{x}\|_2^2$, so we recover the usual Euclidean distance between $\boldsymbol{x}$ and $\boldsymbol{y}$. In accordance with our generalized notion of distance, we define the projection of a point $\boldsymbol{x}\in\mathcal{X}$ onto $\mathcal{X}\cap\mathcal{Y}$ as $\pi_{\mathcal{X}\cap\mathcal{Y}}(\boldsymbol{x}) = \operatorname{argmin}_{\boldsymbol{y}\in\mathcal{X}\cap\mathcal{Y}} D(\boldsymbol{y}\|\boldsymbol{x})$. For ease of notation, we will denote the projection of $\boldsymbol{x}$ onto $\mathcal{X}\cap\mathcal{Y}$ as $[\boldsymbol{x}]$; since $\mathcal{X}\cap\mathcal{Y}$ is closed and convex, and since $D(\cdot\|\boldsymbol{x})$ is strongly convex, such projection exists and is unique. The cumulative regret of the $(\mathcal{X}\cap\mathcal{Y},\mathcal{L})$-minimizer is

$$
\begin{aligned}
R^T_{(\mathcal{X}\cap\mathcal{Y},\mathcal{L})} &= \sum_{t=1}^{T}\ell^t([\boldsymbol{x}^t]) - \min_{\hat{\boldsymbol{x}}\in\mathcal{X}\cap\mathcal{Y}}\left\{\sum_{t=1}^{T}\ell^t(\hat{\boldsymbol{x}})\right\} \\
&= \sum_{t=1}^{T}\ell^t([\boldsymbol{x}^t]-\boldsymbol{x}^t) - \min_{\hat{\boldsymbol{x}}\in\mathcal{X}\cap\mathcal{Y}}\left\{\sum_{t=1}^{T}\ell^t(\hat{\boldsymbol{x}}-\boldsymbol{x}^t)\right\}, \quad (7)
\end{aligned}
$$

where the second equality holds by linearity of ℓ^t. The first-order optimality condition for the projection problem is

$$
\langle\nabla d(\boldsymbol{x}^t) - \nabla d([\boldsymbol{x}^t]), \hat{\boldsymbol{x}} - [\boldsymbol{x}^t]\rangle \le 0 \quad \forall\hat{\boldsymbol{x}}\in\mathcal{X}\cap\mathcal{Y}.
$$

Consequently, provided $\alpha^t \ge 0$ for all t,

$$
\begin{aligned}
\min_{\hat{\boldsymbol{x}}\in\mathcal{X}\cap\mathcal{Y}}\left\{\sum_{t=1}^{T}\ell^t(\hat{\boldsymbol{x}}-\boldsymbol{x}^t)\right\} \ge \min_{\hat{\boldsymbol{x}}\in\mathcal{X}} &\left\{\sum_{t=1}^{T}\ell^t(\hat{\boldsymbol{x}}-\boldsymbol{x}^t)\right. \\
&\left.+\sum_{t=1}^{T}\alpha^t\langle\nabla d(\boldsymbol{x}^t) - \nabla d([\boldsymbol{x}^t]), \hat{\boldsymbol{x}} - [\boldsymbol{x}^t]\rangle\right\} \quad (8)
\end{aligned}
$$

(note the change in the domain of the minimum between the left- and right-hand side). The role of the α^t coefficients is to penalize choices of $\boldsymbol{x}^t$ that are in $\mathcal{X}\setminus\mathcal{Y}$. In particular, if

$$
\frac{1}{\mu}\sum_{t=1}^{T}\ell_t([\boldsymbol{x}^t]-\boldsymbol{x}^t) \le \sum_{t=1}^{T}\alpha^t\|[\boldsymbol{x}^t]-\boldsymbol{x}^t\|^2, \quad (9)
$$

then, by μ-strong convexity of d, we have

$$
\sum_{t=1}^{T}\ell_t([\boldsymbol{x}^t]-\boldsymbol{x}^t) \le \sum_{t=1}^{T}\alpha^t\langle\nabla d(\boldsymbol{x}^t)-\nabla d([\boldsymbol{x}^t]), \boldsymbol{x}^t-[\boldsymbol{x}^t]\rangle. \quad (10)
$$

Substituting (10) and (8) into Equation (7) we get

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

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

$$
\tilde{\ell}^t : \boldsymbol{x} \mapsto \ell^t(\boldsymbol{x}) + \alpha^t\langle\nabla d(\boldsymbol{x}^t) - \nabla d([\boldsymbol{x}^t]), \boldsymbol{x}\rangle. \quad (11)
$$

Hence, as long as condition (9) holds, the regret circuit of Figure 7 is guaranteed to be Hannan consistent.

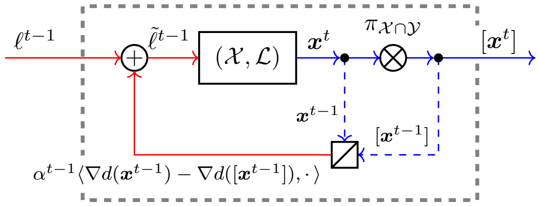

ℓ^{t-1}
$\tilde{\ell}^{t-1}$
$(\mathcal{X},\mathcal{L})$
$\boldsymbol{x}^t$
$\pi_{\mathcal{X}\cap\mathcal{Y}}$
$[\boldsymbol{x}^t]$
$\boldsymbol{x}^{t-1}$
$[\boldsymbol{x}^{t-1}]$
$\alpha^{t-1}\langle\nabla d(\boldsymbol{x}^{t-1}) - \nabla d([\boldsymbol{x}^{t-1}]), \cdot\rangle$

Figure 7. Regret circuit representing the construction of an $(\mathcal{X}\cap\mathcal{Y},\mathcal{L})$-regret minimizer using a $(\mathcal{X},\mathcal{L})$-regret minimizer.

On the other hand, condition (9) can be trivially satisfied by the deterministic choice

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

The fact that α^t can be arbitrarily large (when $\boldsymbol{x}^t$ and $[\boldsymbol{x}^t]$ are very close) is not an issue. Indeed, α^t is only used in $\tilde{\ell}^t$ (Equation 11) and is always multiplied by a term whose magnitude grows proportionally with the distance between $\boldsymbol{x}^t$ and $[\boldsymbol{x}^t]$. In fact, the norm of the functional $\tilde{\ell}^t$ is bounded:

$$
\begin{aligned}
\|\tilde{\ell}^t\| &\le \|\ell^t\| + \left|\frac{\ell^t([\boldsymbol{x}^t]-\boldsymbol{x}^t)}{\mu\|[\boldsymbol{x}^t]-\boldsymbol{x}^t\|^2}\right|\cdot\|\nabla d(\boldsymbol{x}^t) - \nabla d([\boldsymbol{x}^t])\| \\
&\le \|\ell^t\| + \beta\left|\frac{\ell^t([\boldsymbol{x}^t]-\boldsymbol{x}^t)}{\mu\|[\boldsymbol{x}^t]-\boldsymbol{x}^t\|^2}\right|\cdot\|[\boldsymbol{x}^t]-\boldsymbol{x}^t\| \\
&\le \left(1+\frac{\beta}{\mu}\right)\|\ell^t\|,
\end{aligned}
$$

where the second inequality follows by β-smoothness of d. In other words, our construction dilates the loss functions by at most a factor $1+\beta/\mu$. For instance, when $d(\boldsymbol{x}) = \|\boldsymbol{x}\|^2$, this dilation factor is equal to 2.

6.3. Comparison of the Two Constructions

As we pointed out, the decisions in the construction using Lagrangian relaxation only converge to the constrained domain $\mathcal{X}_g$ *on average*. Thus, formally the construction does *not* provide an $(\mathcal{X}_g, \mathcal{L})$-regret minimizer, but only an approximate one. Section 6.2 solves this problem by providing a generic construction for an $(\mathcal{X} \cap \mathcal{Y}, \mathcal{L})$-regret minimizer, a strictly more general task. The price to pay is the need for (generalized) projections, a potentially expensive operation. Thus, the choice of which construction to use reduces to a tradeoff between the computational cost of projecting and the need to have exact versus approximate feasibility with respect to g. The right choice depends on the application at hand. Finally, the construction based on Lagrangian relaxation requires large penalization factors β^t in order to work properly. Therefore, the norm of $\tilde{\ell}^t_\diamond$ can be large, which can complicate the task of minimizing the regret $R^T_{(\mathcal{X},\mathcal{L})}$.

7. Application: Handling Strategy Constraints

When solving EFGs, there may be a need to add additional constraints beyond simply computing feasible strategies:

- Opponent modeling. Upon observing repeated play from an opponent, we may wish to constrain our model of their strategy space to reflect such observations. Since observations can be consistent with several information sets belonging to the opponent, this requires adding constraints that span across information sets.
- Bounding probabilities. For example, in a patrolling 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 by lower-bounding the probability of each action at each information set (Farina & Gatti, 2017), which can be handled with small modifications to standard CFR or first-order methods (Farina et al., 2017; Kroer et al., 2017). However, *quasi-perfect equilibrium* requires perturbations on the probability of sequences of action (Miltersen & Sørensen, 2010), which requires strategy constraints that cross information sets.

All the applications 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 et al. (2019) 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. They prove that if the regret minimizer for the Lagrange multipliers has the optimal Lagrangian multipliers as part of their strategy space, the average output strategy converges to an approximate solution to the constrained game. They also prove a bound on the approximate feasibility of the average output strategy when their algorithm is instantiated with Regret Matching (Hart & Mas-Colell, 2000) as the local regret minimizer at each information set.

At least two alternative variants of CFR for EFGs with strategy constraints can be obtained using our framework. First, we can apply our method for Lagrangian relaxation of $\mathcal{X}$ and a constraint $g(\boldsymbol{x}) \le 0$. Our Lagrangian approach yields as a special case the CCFR algorithm. Our approach supports regret minimization for the Lagrangian multipliers, as was done in CCFR, since we put no constraints on the form of the β^t multipliers. However, our approach is more general in that it also allows instantiation with a fixed choice of multipliers, thus obviating the need for regret minimization. The second alternative is to apply our construction for the intersection of convex sets (Section 6.2), which uses (generalized) projection onto $\mathcal{X} \cap \{\boldsymbol{x} : g(\boldsymbol{x}) \le 0\}$. This leads to a different regret-minimization approach, which has the major advantage that all iterates are feasible, whereas Lagrangian approaches only achieve approximate feasibility. The cost of projection may be nontrivial, and so in general the choice of method depends on the application at hand.

8. Conclusion and Future Research

We developed a calculus of regret minimization, which enables the construction of regret minimizers for composite convex sets that can be inductively 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 of its variants for the more general case where we have strategy constraints. 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., exposure constraints across industries); our framework makes it easy to construct such a regret minimizer from any standard online-portfolio-selection algorithm.

The approach presented in this paper has a large number of potential future applications. For one, it would be interesting to apply our approaches of including additional constraints to the computation of quasi-perfect equilibria. Currently the only solver that is fairly scalable is based on an exact, monolithic, custom approach that uses heavy-weight operations such as matrix inversion, etc. (Farina et al., 2018). Our regret-minimization approach would be the first of its kind for equilibrium refinement that requires constraints that cut across information sets. It obviates the need for the heavy-weight operations, and would still converge to a feasible solution that satisfies an approximate notion of quasi-perfect equilibrium. It would be interesting to study this tradeoff between speed and solution quality.

Acknowledgments

This material is based on work supported by the National Science Foundation under grants IIS-1718457, IIS-1617590, and CCF-1733556, and the ARO under award W911NF-17-1-0082.

References

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

Brown, N. and Sandholm, T. Regret transfer and parameter optimization. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2014.

Brown, N. and Sandholm, T. Regret-based pruning in extensive-form games. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2015a.

Brown, N. and Sandholm, T. Simultaneous abstraction and equilibrium finding in games. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2015b.

Brown, N. and Sandholm, T. Strategy-based warm starting for regret minimization in games. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2016.

Brown, N. and Sandholm, T. Reduced space and faster convergence in imperfect-information games via pruning. In *International Conference on Machine Learning (ICML)*, 2017a.

Brown, N. and Sandholm, T. Safe and nested subgame solving for imperfect-information games. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 689–699, 2017b.

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

Brown, N. and Sandholm, T. Solving imperfect-information games via discounted regret minimization. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

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

Burch, N., Johanson, M., and Bowling, M. Solving imperfect information games using decomposition. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2014.

Davis, T., Waugh, K., and Bowling, M. Solving large extensive-form games with strategy constraints. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

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

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

Farina, G., Gatti, N., and Sandholm, T. Practical exact algorithm for trembling-hand equilibrium refinements in games. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

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

Ganzfried, S. and Sandholm, T. Endgame solving in large imperfect-information games. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2015. Early version in AAAI-13 Workshop on Computer Poker and Incomplete Information.

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

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

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

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

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

Lanctot, M., Waugh, K., Zinkevich, M., and Bowling, M. Monte Carlo sampling for regret minimization in extensive games. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2009.

Ling, C. K., Fang, F., and Kolter, J. Z. 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)*, 2018.

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

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

Moravcik, M., Schmid, M., Ha, K., Hladik, M., and Gaukrodger, S. Refining subgames in large imperfect information games. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2016.

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

Schuurmans, D. and Zinkevich, M. A. Deep learning games. In *Advances in Neural Information Processing Systems*, pp. 1678–1686, 2016.

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

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

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

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