# Conic Blackwell Algorithm: Parameter-Free Convex-Concave Saddle-Point Solving

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

We develop new parameter and scale-free algorithms for solving convex-concave saddle-point problems. Our results are based on a new simple regret minimizer, the Conic Blackwell Algorithm$^+$ (CBA$^+$), which attains $O(1/\sqrt{T})$ average regret. Intuitively, our approach generalizes to other decision sets of interest ideas from the Counterfactual Regret minimization (CFR$^+$) algorithm, which has very strong practical performance for solving sequential games on simplexes. We show how to implement CBA$^+$ for the simplex, $\ell_p$ norm balls, and ellipsoidal confidence regions in the simplex, and we present numerical experiments for solving matrix games and distributionally robust optimization problems. Our empirical results show that CBA$^+$ is a simple algorithm that outperforms state-of-the-art methods on synthetic data and real data instances, without the need for any choice of step sizes or other algorithmic parameters.

## 1 Introduction

We are interested in solving *saddle-point problems* (SPPs) of the form

$$\min_{\boldsymbol{x}\in\mathcal{X}} \max_{\boldsymbol{y}\in\mathcal{Y}} F(\boldsymbol{x}, \boldsymbol{y}), \tag{1}$$

where $\mathcal{X} \subset \mathbb{R}^n, \mathcal{Y} \subset \mathbb{R}^n$ are convex, compact sets, and $F : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is a differentiable convex-concave function. Convex-concave SPPs arise in a number of practical problems. For example, the problem of computing a Nash equilibrium of a zero-sum games can be formulated as a convex-concave SPP, and this is the foundation of most methods for solving sequential zero-sum games [vS96, ZJBP07, TBJB15, KWKKS20]. They also arise in imaging [CP11], $\ell_\infty$-regression [ST18], Markov Decision Processes [Iye05, WKR13, ST18], and in distributionally robust optimization, where the max term represents the distributional uncertainty [ND16, BTHKM15]. In this paper we propose efficient, *parameter-free* algorithms for solving (1) in many settings, i.e., algorithms that do not require any tuning or choices of step sizes.

**Repeated game framework** One way to solve convex-concave SPPs is by viewing the SPP as a repeated game between two players, where each step $t$ consists of one player choosing $\boldsymbol{x}_t \in \mathcal{X}$, the other player choosing $\boldsymbol{y}_t \in \mathcal{Y}$, and then the players observe the payoff $F(\boldsymbol{x}_t, \boldsymbol{y}_t)$. If each player employs a regret-minimization algorithm, then a well-known folk theorem says that the uniform average strategy generated by two regret minimizers repeatedly playing an SPP against each other converges to a solution to the SPP. We will call this the "repeated game framework."

There are already well-known algorithms for instantiating the above repeated game framework for (1). For example, one can employ the *online mirror descent* algorithm, which generates iterates as

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

follows for the first player (and similarly for the second player):

$$\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \langle \eta \nabla_{\boldsymbol{x}} F(\boldsymbol{x}_t, \boldsymbol{y}_t), \boldsymbol{x}\rangle + D(\boldsymbol{x}\|\boldsymbol{x}_t), \tag{2}$$

where $D(\cdot\|\cdot)$ is a Bregman divergence which measures distance between pairs of points and $\eta > 0$ is an appropriate step size. By choosing $D$ appropriately for $\mathcal{X}$, the update step (2) becomes efficient, and one can achieve an overall regret on the order of $O(\sqrt{T})$ after $T$ iterations. However, choosing $\eta$ often requires to know the number of periods $T$ and an upper bound $L$ on the norms of the gradients visited $(\nabla_{\boldsymbol{x}} F(\boldsymbol{x}_t, \boldsymbol{y}_t))_{t\geq 0}$. This is problematic, as 1) the upper bound $L$ may be hard to obtain in many applications, and 2) $L$ may be too conservative in practice. Even adaptive step sizes may be too conservative. This is not just a theoretical issue, as we highlight in our numerical experiments (Section 4) and in the appendices (Appendices E). Similar results and challenges hold for the popular *follow the regularized leader* (FTRL) algorithm (see Appendix E).

**Regret Matching** In this paper, we introduce alternative regret-minimization schemes for instantiating the above framework. Our work is motivated by recent advances on solving large-scale zero-sum sequential games. In the zero-sum sequential game setting, $\mathcal{X}$ and $\mathcal{Y}$ are simplexes, the objective function becomes $F(\boldsymbol{x}, \boldsymbol{y}) = \langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{y}\rangle$, and thus (1) reduces to a *bilinear* SPP. Based on this bilinear SPP formulation, the best practical methods for solving large-scale sequential games use the repeated game framework, where each player minimizes regret via some variant of *counterfactual regret minimization* (CFR, [ZJBP07]). Variants of CFR were used in every recent poker AI challenge, where poker AIs beat human poker players [BBJT15, MSB$^+$17, BS18, BS19]. The CFR framework itself is a decomposition of the overall regret of the bilinear SPP into local regrets at each decision point in a sequential game [FKS19a]. The key to the practical performance of CFR-based algorithms seems to be three ingredients (beyond the CFR decomposition itself): (1) a particular regret minimizer called *regret matching*$^+$ (RM$^+$) [TBJB15] which is employed at each decision point, (2) aggressive iterate averaging schemes that put greater weight on recent iterates (e.g. *linear averaging*, which weights iterate at period $t$ by $2t/T(T+1)$), and (3) an alternation scheme where the updates of the repeated game framework are performed in an asymmetric fashion. The CFR framework itself is specific to sequential bilinear games on simplexes, but these last three ingredients could potentially be generalized to other problems of the form (1). That is the starting point of the present paper.

The most challenging aspect of generalizing the above ingredients is that RM$^+$ is specifically designed for minimizing regret over a simplex. However, many problems of the form (1) have convex sets $\mathcal{X}, \mathcal{Y}$ that are not simplexes, e.g. box constraints or norm-balls for distributionally robust optimization [BTHKM15]. In principle, regret matching arises from a general theory called *Blackwell approachability* [Bla56, HMC00], and similar constructions can be envisioned for other convex sets. However, in practice the literature has only focused on developing concrete implementable instantiations of Blackwell approachability for simplexes. A notable deviation from this is the work of [ABH11], who showed a general reduction between regret minimization over general convex sets and Blackwell approachability. However, their general reduction still does not yield a practically implementable algorithm: among other things, their reduction relies on certain black-box projections that are not always efficient. We show how to implement these necessary projections for the setting where $\mathcal{X}$ and $\mathcal{Y}$ are simplexes, $\ell_p$ balls, and intersections of the $\ell_2$ ball with a hyperplane (with a focus on the case where an $\ell_2$ ball is intersected with a simplex, which arises naturally as confidence regions). This yields an algorithm which we will refer to as the *conic Blackwell algorithm* (CBA), which is similar in spirit to the regret matching algorithm, but crucially generalizes to other decision sets. Motivated by the practical performance of RM$^+$, we construct a variant of CBA which uses a thresholding operation similar to the one employed by RM$^+$. We call this algorithm CBA$^+$.

**Our contributions** We introduce CBA$^+$, a parameter-free algorithm which achieves $O(\sqrt{T})$ regret in the worst case and generalizes the strong performances of RM$^+$ for bilinear, simplex saddle-points solving to other more general settings. A major selling point for CBA$^+$ is that it does not require any step size choices. Instead, the algorithm implicitly adjusts to the structure of the domains and losses by being instantiations of Blackwell's approachability algorithm. After developing the CBA$^+$ algorithm, we then develop analogues of another crucial components for large-scale game solving. In particular, we prove a generalization of the folk theorem for the repeated game framework for solving (1), which allows us to incorporate polynomial averaging schemes such as linear averaging. We then show that CBA$^+$ is compatible with linear averaging on the iterates. This mirrors the case of RM and RM$^+$, where only RM$^+$ is compatible with linear averaging on the iterates. We also show that both

CBA and CBA$^+$ are compatible with polynomial averaging when simultaneously performed on the regrets and the iterates. Combining all these ingredients, we arrive at a new class of algorithms for solving convex-concave SPPs. As long as efficient projection operations can be performed (which we show for several practical domains, including the simplex, $\ell_p$ balls and confidence regions in the simplex), one can apply the repeated game framework on (1), where one can use either CBA or CBA$^+$ as a regret minimizer for $\mathcal{X}$ and $\mathcal{Y}$, use polynomial averaging on the generated iterates, in order to solve (1) at a rate of $O\left(1/\sqrt{T}\right)$.

We highlight the practical efficacy of our algorithmic framework on several domains. First, we solve two-player zero-sum matrix games and extensive form games, where RM$^+$ regret minimizer combined with linear averaging and alternation, and CFR$^+$, lead to very strong practical algorithms [TBJB15]. We find that CBA$^+$ combined with linear averaging and alternation leads to a comparable performance in terms of the iteration complexity, and may even slightly outperform RM$^+$ and CFR$^+$. Second, we apply our approach to a setting where RM$^+$ and CFR$^+$ do not apply: distributionally robust empirical risk minimization (DR-ERM) problems. Across two classes of synthetic problems and four real data sets, we find that our algorithm based on CBA$^+$ performs orders of magnitude better than online mirror descent and FTRL, as well as their optimistic variants, when using their theoretically-correct fixed step sizes. Even when considering adaptive step sizes, or fixed step sizes that are up to $10,000$ larger than those predicted by theory, our CBA$^+$ algorithm performs better, with only a few cases of comparable performance (at step sizes that lead to divergence for some of the other non-parameter free methods). The fast practical performance of our algorithm, combined with its simplicity and the total lack of step sizes or parameters tuning, suggests that it should be seriously considered as a practical approach for solving convex-concave SPPs in various settings.

Finally, we make a brief note on accelerated methods. Our algorithms have a rate of convergence towards a saddle point of $O(1/\sqrt{T})$, similar to OMD and FTRL. In theory, it is possible to obtain a faster $O\left(1/T\right)$ rate of convergence, for example via mirror prox [Nem04] or other primal-dual algorithms [CP16]. However, our experimental results show that CBA$^+$ is faster than optimistic variants of FTRL and OMD, the latter being almost identical to the mirror prox algorithm, and both achieving $O(1/T)$ rate of convergence. A similar conclusion has been drawn in the context of sequential game solving, where the fastest $O(1/\sqrt{T})$ CFR-based algorithms have better practical performance than the theoretically-superior $O\left(1/T\right)$-rate methods [KWKKS20, KFS18].

## 2 Game setup and Blackwell Approchability

As stated in section 1, we will solve (1) using a repeated game framework. The first player chooses strategies from $\mathcal{X}$ in order to minimize the sequence of payoffs in the repeated game, while the second player chooses strategies from $\mathcal{Y}$ in order to maximize payoffs. There are $T$ iterations with indices $t = 1, \ldots, T$. In this framework, each iteration $t$ consists of the following steps:

1. Each player chooses strategies $\boldsymbol{x}_t \in \mathcal{X}, \boldsymbol{y}_t \in \mathcal{Y}$
2. First player observes $\boldsymbol{f}_t = \nabla_{\boldsymbol{x}} F(\boldsymbol{x}_t, \boldsymbol{y}_t)$ and uses $\boldsymbol{f}_t$ when computing the next strategy
3. Second player observes $\boldsymbol{g}_t = \nabla_{\boldsymbol{y}} F(\boldsymbol{x}_t, \boldsymbol{y}_t)$ and uses $\boldsymbol{g}_t$ when computing the next strategy

The goal of each player is to minimize their regret $R_{T,\boldsymbol{x}}, R_{T,\boldsymbol{y}}$ across the $T$ iterations:

$$R_{T,\boldsymbol{x}} = \sum_{t=1}^{T} \langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \min_{\boldsymbol{x} \in \mathcal{X}} \sum_{t=1}^{T} \langle \boldsymbol{f}_t, \boldsymbol{x} \rangle, \quad R_{T,\boldsymbol{y}} = \max_{\boldsymbol{y} \in \mathcal{Y}} \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{y} \rangle - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{y}_t \rangle.$$

The reason this repeated game framework leads to a solution to the SPP problem (1) is the following folk theorem. Relying on $F$ being convex-concave, it connects the regret incurred by each player to the duality gap in (1).

**Theorem 2.1** (Theorem 1, [Kro20]). *Let* $(\bar{\boldsymbol{x}}_T, \bar{\boldsymbol{y}}_T) = \frac{1}{T} \sum_{t=1}^{T} (\boldsymbol{x}_t, \boldsymbol{y}_t)$ *for any* $(\boldsymbol{x}_t)_{t \geq 1}, (\boldsymbol{y}_t)_{t \geq 1}$. *Then*

$$\max_{\boldsymbol{y} \in \mathcal{Y}} F(\bar{\boldsymbol{x}}_T, \boldsymbol{y}) - \min_{\boldsymbol{x} \in \mathcal{X}} F(\boldsymbol{x}, \bar{\boldsymbol{y}}_T) \leq (R_{T,\boldsymbol{x}} + R_{T,\boldsymbol{y}})/T.$$

Therefore, when each player runs a regret minimizer that guarantees regret on the order of $O(\sqrt{T})$, $(\bar{\boldsymbol{x}}_T, \bar{\boldsymbol{y}}_T)_{T\geq 0}$ converges to a solution to (1) at a rate of $O\left(1/\sqrt{T}\right)$. Later we will show a generalization of Theorem 2.1 that will allow us to incorporate more aggressive averaging schemes that put additional weight on the later iterates. Given the repeated game framework, the next question becomes which algorithms to employ in order to minimize regret for each player. As mentioned in Section 1, for zero-sum games, variants of regret matching are used in practice.

**Blackwell Approchability** Regret matching arises from the *Blackwell approchability* framework [Bla56]. In Blackwell approachability, a decision maker repeatedly takes decisions $\boldsymbol{x}_t$ from some convex decision set $\mathcal{X}$ (this set plays the same role as $\mathcal{X}$ or $\mathcal{Y}$ in (1)). After taking decision $\boldsymbol{x}_t$ the player observes a vector-valued affine payoff function $\boldsymbol{u}_t(\boldsymbol{x}) \in \mathbb{R}^n$. The goal for the decision maker is to force the average payoff $\frac{1}{t}\sum_{\tau=1}^{t} \boldsymbol{u}_\tau(\boldsymbol{x}_\tau)$ to approach some convex target $\mathcal{S}$. Blackwell proved that a convex target set $\mathcal{S}$ can be approached if and only if for every halfspace $\mathcal{H} \supseteq \mathcal{S}$, there exists $\boldsymbol{x} \in \mathcal{X}$ such that for every possible payoff function $\boldsymbol{u}(\cdot)$, $\boldsymbol{u}(\boldsymbol{x})$ is guaranteed to lie in $\mathcal{H}$. The action $\boldsymbol{x}$ is said to *force* $\mathcal{H}$. Blackwell's proof is via an algorithm: at iteration $t$, his algorithm projects the average payoff $\bar{\boldsymbol{u}} = \frac{1}{t-1}\sum_{\tau=1}^{t-1} \boldsymbol{u}_\tau(\boldsymbol{x}_\tau)$ onto $\mathcal{S}$, and then the decision maker chooses an action $\boldsymbol{x}_t$ that forces the tangent halfspace to $\mathcal{S}$ generated by the normal $\bar{\boldsymbol{u}} - \pi_{\mathcal{S}}(\bar{\boldsymbol{u}})$, where $\pi_{\mathcal{S}}(\bar{\boldsymbol{u}})$ is the orthogonal projection of $\bar{\boldsymbol{u}}$ onto $\mathcal{S}$. We call this algorithm *Blackwell's algorithm*; it approaches $\mathcal{S}$ at a rate of $O(1/\sqrt{T})$. It is important to note here that Blackwell's algorithm is rather a meta-algorithm than a concrete algorithm. Even within the context of Blackwell's approachability problem, one needs to devise a way to compute the forcing actions needed at each iteration, i.e., to compute $\pi_{\mathcal{S}}(\bar{\boldsymbol{u}})$.

**Details on Regret Matching** Regret matching arises by instantiating Blackwell approachability with the decision space $\mathcal{X}$ equal to the simplex $\Delta(n)$, the target set $\mathcal{S}$ equal to the nonpositive orthant $\mathbb{R}^n_-$, and the vector-valued payoff function $\boldsymbol{u}_t(\boldsymbol{x}_t) = \boldsymbol{f}_t - \langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle \boldsymbol{e}$ equal to the regret associated to each of the $n$ actions (which correspond to the corners of $\Delta(n)$). Here $\boldsymbol{e} \in \mathbb{R}^n$ has one on every component. [HMC00] showed that with this setup, playing each action with probability proportional to its positive regret up to time $t$ satisfies the forcing condition needed in Blackwell's algorithm. Formally, regret matching (RM) keeps a running sum $\boldsymbol{r}_t = \sum_{\tau=1}^{t} (\boldsymbol{f}_\tau - \langle \boldsymbol{f}_\tau, \boldsymbol{x}_\tau\rangle \boldsymbol{e})$, and then action $i$ is played with probability $\boldsymbol{x}_{t+1,i} = [\boldsymbol{r}_{t,i}]^+ / \sum_{i=1}^{n} [\boldsymbol{r}_{t,i}]^+$, where $[\cdot]^+$ denotes thresholding at zero. By Blackwell's approachability theorem, this algorithm converges to zero average regret at a rate of $O(1/\sqrt{T})$. In zero-sum game-solving, it was discovered that a variant of regret matching leads to extremely strong practical performance (but the same theoretical rate of convergence). In regret matching$^+$ (RM$^+$), the running sum is thresholded at zero at every iteration: $\boldsymbol{r}_t = [\boldsymbol{r}_{t-1} + \boldsymbol{f}_t - \langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle \boldsymbol{e}]^+$, and then actions are again played proportional to $\boldsymbol{r}_t$. In the next section, we describe a more general class of regret-minimization algorithms based on Blackwell's algorithm for general set $\mathcal{X}$, introduced in [ABH11].

# 3 Conic Blackwell Algorithm

We present the Conic Blackwell Algorithm Plus (CBA$^+$), a no-regret algorithm which uses a variation of Blackwell's approachability procedure [Bla56] to perform regret minimization on general closed and convex decision sets $\mathcal{X}$. We will assume that losses are coming from a bounded set; in the repeated game framework this is the case if there exists $G_x, G_y$ such that $\|\nabla_x F(\boldsymbol{x}, \boldsymbol{y})\| \leq G_x, \|\nabla_y F(\boldsymbol{x}, \boldsymbol{y})\| \leq G_y$ for all $\boldsymbol{x} \in \mathcal{X}, \boldsymbol{y} \in \mathcal{Y}$. We do not need to know $G_x, G_y$.

CBA$^+$ is best understood as a combination of two steps. The first is the basic CBA algorithm, derived from Blackwell's algorithm, which we describe next. To convert Blackwell's algorithm to a regret minimizer on $\mathcal{X}$, we use the reduction from [ABH11], which considers the conic hull $\mathcal{C} = \text{cone}(\{\kappa\} \times \mathcal{X})$ where $\kappa = \max_{\boldsymbol{x}\in\mathcal{X}} \|\boldsymbol{x}\|_2$. The Blackwell approachability problem is then instantiated with $\mathcal{X}$ as the decision set, target set equal to the polar $\mathcal{C}^\circ = \{\boldsymbol{z} : \langle \boldsymbol{z}, \hat{\boldsymbol{z}}\rangle \leq 0, \forall \hat{\boldsymbol{z}} \in \mathcal{C}\}$ of $\mathcal{C}$, and payoff vectors $(\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle, -\boldsymbol{f}_t)$. The conic Blackwell algorithm (CBA) is implemented by projecting the average payoff vector onto $\mathcal{C}$, calling this projection $\alpha(\kappa, \boldsymbol{x})$ with $\alpha \geq 0$ and $\boldsymbol{x} \in \mathcal{X}$, and playing the action $\boldsymbol{x}$.

The second step in CBA$^+$ is to modify CBA to make it analogous to RM$^+$ rather than to RM. To do this, the algorithm does not keep track of the average payoff vector. Instead, we keep a running aggregation

**Algorithm 1** Conic Blackwell Algorithm Plus ($\mathsf{CBA}^+$)

**Input** A convex, compact set $\mathcal{X} \subset \mathbb{R}^n$.
**Algorithm parameters** Weights $(\omega_\tau)_{\tau \geq 1} \in \mathbb{R}^{\mathbb{N}}$.
**Initialization** $t = 1$, $\boldsymbol{x}_1 \in \mathcal{X}$.
Observe $\boldsymbol{f}_1$ then set $\boldsymbol{u}_1 = (\langle \boldsymbol{f}_1, \boldsymbol{x}_1 \rangle, -\boldsymbol{f}_1) \in \mathbb{R} \times \mathbb{R}^n$.
**for** $t \geq 1$ **do**
  Choose $\boldsymbol{x}_{t+1} = \mathsf{CHOOSEDECISION}_{\mathsf{CBA}^+}(\boldsymbol{u}_t)$.
  Observe the loss $\boldsymbol{f}_{t+1} \in \mathbb{R}^n$.
  Update $\boldsymbol{u}_{t+1} = \mathsf{UPDATEPAYOFF}_{\mathsf{CBA}^+}(\boldsymbol{u}_t, \boldsymbol{x}_{t+1}, \boldsymbol{f}_{t+1}, \omega_{t+1}, \sum_{\tau=1}^{t} \omega_\tau)$.
  Increment $t \leftarrow t + 1$.

of the payoffs, where we always add the newest payoff to the aggregate, and then project the aggregate onto $\mathcal{C}$. More concretely, pseudocode for $\mathsf{CBA}^+$ is given in Algorithm 1. This pseudocode relies on two functions: $\mathsf{CHOOSEDECISION}_{\mathsf{CBA}^+} : \mathbb{R}^{n+1} \to \mathbb{R}^n$, which maps the aggregate payoff vector $\boldsymbol{u}_t$ to a decision, and $\mathsf{UPDATEPAYOFF}_{\mathsf{CBA}^+}$ which controls how we aggregate payoffs. Given an aggregate payoff vector $\boldsymbol{u} = (\tilde{u}, \hat{\boldsymbol{u}}) \in \mathbb{R} \times \mathbb{R}^n$, we have

$$\mathsf{CHOOSEDECISION}_{\mathsf{CBA}^+}(\boldsymbol{u}) = (\kappa/\tilde{u})\hat{\boldsymbol{u}}.$$

If $\tilde{u} = 0$, we just let $\mathsf{CHOOSEDECISION}_{\mathsf{CBA}^+}(\boldsymbol{u}) = (1/n)\boldsymbol{e}$. The function $\mathsf{UPDATEPAYOFF}_{\mathsf{CBA}^+}$ is implemented by adding the most recent payoff to the aggregate payoffs, and then projecting onto $\mathcal{C}$. More formally, it is defined as

$$\mathsf{UPDATEPAYOFF}_{\mathsf{CBA}^+}(\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{f}, \omega, S) = \pi_{\mathcal{C}}\left(\frac{S}{S+\omega}\boldsymbol{u} + \frac{\omega}{S+\omega}\left(\langle \boldsymbol{f}, \boldsymbol{x} \rangle, -\boldsymbol{f}\right)\right),$$

where $\omega$ is the weight assigned to the most recent payoff and $S$ the weight assigned to the previous aggregate payoff $\boldsymbol{u}$. Because of the projection step in $\mathsf{UPDATEPAYOFF}_{\mathsf{CBA}^+}$, we always have $\boldsymbol{u} \in \mathcal{C}$, which in turn guarantees that $\mathsf{CHOOSEDECISION}_{\mathsf{CBA}^+}(\boldsymbol{u}) \in \mathcal{X}$, since $\mathcal{C} = \text{cone}(\{\kappa\} \times \mathcal{X})$. We will see in the next section that $\mathsf{RM}^+$ is related to $\mathsf{CBA}^+$ but replaces the exact projection step $\pi_{\mathcal{C}}(\boldsymbol{u})$ in $\mathsf{UPDATEPAYOFF}_{\mathsf{CBA}^+}$ by a suboptimal solution to the projection problem.

Let us note the difference between $\mathsf{CBA}^+$ and the algorithm introduced in [ABH11], which we have called $\mathsf{CBA}$. $\mathsf{CBA}$ uses different $\mathsf{UPDATEPAYOFF}$ and $\mathsf{CHOOSEDECISION}$ functions. In $\mathsf{CBA}$ the payoff update is defined as

$$\mathsf{UPDATEPAYOFF}_{\mathsf{CBA}}(\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{f}, \omega, S) = \frac{S}{S+\omega}\boldsymbol{u} + \frac{\omega}{S+\omega}\left(\langle \boldsymbol{f}, \boldsymbol{x} \rangle, -\boldsymbol{f}\right).$$

Note in particular the lack of projection as compared to $\mathsf{CBA}^+$, analogous to the difference between $\mathsf{RM}$ and $\mathsf{RM}^+$. The $\mathsf{CHOOSEDECISION}_{\mathsf{CBA}}$ function then requires a projection onto $\mathcal{C}$:

$$\mathsf{CHOOSEDECISION}_{\mathsf{CBA}}(\boldsymbol{u}) = \mathsf{CHOOSEDECISION}_{\mathsf{CBA}^+}\left(\pi_{\mathcal{C}}(\boldsymbol{u})\right).$$

Based upon the analysis in [Bla56], [ABH11] show that $\mathsf{CBA}$ with uniform weights (both on payoffs and decisions) guarantees $O(1/\sqrt{T})$ average regret. The difference between $\mathsf{CBA}^+$ and $\mathsf{CBA}$ is similar to the difference between the $\mathsf{RM}$ and $\mathsf{RM}^+$ algorithms. In practice, $\mathsf{RM}^+$ performs significantly better than $\mathsf{RM}$ for solving matrix games, when combined with *linear averaging* on the decisions (as opposed to the uniform averaging used in Theorem 2.1). In the next theorem, we show that $\mathsf{CBA}^+$ is compatible with linear averaging on decisions only. We present a detailed proof in Appendix A.

**Theorem 3.1.** *Consider* $(\boldsymbol{x}_t)_{t \geq 0}$ *generated by* $\mathsf{CBA}^+$ *with uniform weights:* $\omega_\tau = 1, \forall\, \tau \geq 1$. *Then*

$$\frac{\sum_{t=1}^{T} t\langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \min_{\boldsymbol{x} \in \mathcal{X}} \sum_{t=1}^{T} t\langle \boldsymbol{f}_t, \boldsymbol{x} \rangle}{T(T+1)} = O\left(1/\sqrt{T}\right).$$

Note that in Theorem 3.1, we have *uniform* weights on the sequence of payoffs $(\boldsymbol{u}_t)_{t \geq 0}$, but *linearly increasing* weights on the sequence of decisions. The proof relies on properties specific to $\mathsf{CBA}^+$, and it does not extend to $\mathsf{CBA}$. Numerically it also helps $\mathsf{CBA}^+$ but not $\mathsf{CBA}$. In Appendix A, we show that both $\mathsf{CBA}$ and $\mathsf{CBA}^+$ achieve $O\left(1/\sqrt{T}\right)$ convergence rates when using a weighted average on

*both* the decisions and the payoffs (Theorems A.2-A.3). In practice, using linear averaging only on the decisions, as in Theorem 3.1, performs vastly better than linear averaging on both decisions and payoffs. We present empirical evidence of this in Appendix A.

Some no-regret algorithms attain a stronger $O(1/T)$ average regret guarantee, e.g., Optimistic Online Mirror Descent O-OMD and Optimistic Follow-The-Regularized-Leader O-FTRL, presented in Appendix E. Nonetheless, we show in Section 4 that the empirical performance of CBA$^+$ is better than that of $O(1/T)$ methods. A similar situation occurs for RM$^+$ compared to O-OMD and O-FTRL for solving poker games [FKS19b, KWKKS20].

The following theorem gives the convergence rate of CBA$^+$ for solving saddle-points (1), based on our convergence rate on the regret of each player (Theorem 3.1). The proof is in Appendix B.

**Theorem 3.2.** *Let* $(\bar{\boldsymbol{x}}_T, \bar{\boldsymbol{y}}_T) = 2\sum_{t=1}^{T} t\left(\boldsymbol{x}_t, \boldsymbol{y}_t\right)/(T(T+1))$, *where* $(\boldsymbol{x}_t)_{t\geq 0}, (\boldsymbol{y}_t)_{t\geq 0}$ *are generated by the repeated game framework with* CBA$^+$ *with uniform weights. Then*

$$\max_{\boldsymbol{y}\in\mathcal{Y}} F(\bar{\boldsymbol{x}}_T, \boldsymbol{y}) - \min_{\boldsymbol{x}\in\mathcal{X}} F(\boldsymbol{x}, \bar{\boldsymbol{y}}_T) = O\left(1/\sqrt{T}\right).$$

### 3.1 Efficient implementations of CBA$^+$

To obtain an implementation of CBA$^+$ and CBA, we need to efficiently resolve the functions $\mathsf{CHOOSEDECISION}_{\mathsf{CBA}^+}$ and $\mathsf{UPDATEPAYOFF}_{\mathsf{CBA}^+}$. In particular, we need to compute $\pi_{\mathcal{C}}(\boldsymbol{u})$, the orthogonal projection of $\boldsymbol{u}$ onto the cone $\mathcal{C}$, where $\mathcal{C} = \text{cone}(\{\kappa\} \times \mathcal{X})$:

$$\pi_{\mathcal{C}}(\boldsymbol{u}) \in \arg\min_{\boldsymbol{y}\in\mathcal{C}} \|\boldsymbol{y} - \boldsymbol{u}\|_2^2. \tag{3}$$

Even for CBA this problem must be resolved, since [ABH11] did not study whether (3) can be efficiently solved. It turns out that (3) can be computed in closed-form or quasi closed-form for many decision sets $\mathcal{X}$ of interest. Interestingly, parts of the proofs rely on *Moreau's Decomposition Theorem* [CR13], which states that $\pi_{\mathcal{C}}(\boldsymbol{u})$ can be recovered from $\pi_{\mathcal{C}^\circ}(\boldsymbol{u})$ and vice-versa; we present the detailed complexity results and the proofs in Appendix C.

**Simplex** $\mathcal{X} = \Delta(n)$ is the classical setting used for matrix games and extensive-form games (EFGs) [FKS19a]; $n$ is the number of actions of a player and $\boldsymbol{x} \in \Delta(n)$ represents a randomized strategy. In this case, $\pi_{\mathcal{C}}(\boldsymbol{u})$ can be computed in $O(n\log(n))$. We explain in Appendix C that RM and RM$^+$ are obtained by choosing a suboptimal solution to (3), avoiding the $O(n\log(n))$ sorting operation, whereas CBA and CBA$^+$ choose optimally. In our numerical experiments, we will see that CBA$^+$ slightly outperforms RM$^+$ and CFR$^+$ in terms of iteration count.

**$\ell_p$ balls** This is when $\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^n \mid ; \|\boldsymbol{x}\|_p \leq 1\}$ with $p \geq 1$ or $p = \infty$. This is of interest for instance in distributionally robust optimization [BTHKM15, ND16], $\ell_\infty$ regression [ST18] and saddle-point reformulation of Markov Decision Process [JS20]. For $p = 2$, we can compute $\pi_{\mathcal{C}}(\boldsymbol{u})$ in closed-form, i.e., in $O(n)$ arithmetic operations. For $p \in \{1, \infty\}$, we can compute $\pi_{\mathcal{C}}(\boldsymbol{u})$ in $O(n\log(n))$ arithmetic operations using a sorting algorithm.

**Ellipsoidal confidence region in the simplex** Here, $\mathcal{X}$ is an *ellipsoidal subregion of the simplex*, defined as $\mathcal{X} = \{\boldsymbol{x} \in \Delta(n) \mid \|\boldsymbol{x} - \boldsymbol{x}_0\|_2 \leq \epsilon_x\}$. This type of decision set is widely used because they are associated with confidence regions when estimating a probability distribution from observed data [Iye05, BdHP19]. It can also be used in Bellman update for robust Markov Decision Process [Iye05, WKR13, GGC18]. We also assume that the confidence region is "entirely contained in the simplex": $\{\boldsymbol{x} \in \mathbb{R}^n | \boldsymbol{x}^\top \boldsymbol{e} = 1\} \bigcap \{\boldsymbol{x} \in \mathbb{R}^n \mid \|\boldsymbol{x} - \boldsymbol{x}_0\|_2 \leq \epsilon_x\} \subseteq \Delta(n)$, to avoid degenerate components. In this case, using a change of basis we show that it is possible to compute $\pi_{\mathcal{C}}(\boldsymbol{u})$ in closed-form, i.e., in $O(n)$ arithmetic operations.

## 4 Numerical experiments

In this section we investigate the practical performances of our algorithms on several instances of saddle-point problems. We start by comparing CBA$^+$ with RM$^+$ in the matrix and extensive form

games setting. We then turn to comparing our algorithms on instances from the distributionally robust optimization literature. The code for all experiments is available in the supplemental material.

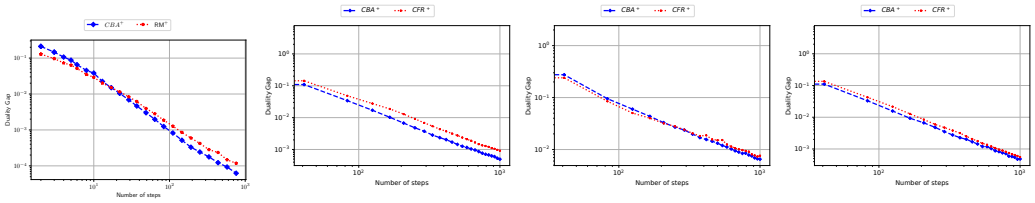
## 4.1 Matrix games on the simplex

Since the motivation for CBA$^+$ is to obtain the strong empirical performances of RM$^+$ and CFR$^+$ on other decision sets than the simplex, we start by checking that CBA$^+$ indeed provide comparable performance on simplex settings. We compare these methods on matrix games

$$\min_{\boldsymbol{x}\in\Delta(n)} \max_{\boldsymbol{y}\in\Delta(m)} \langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{y}\rangle,$$

where $\boldsymbol{A}$ is the matrix of payoff, and on extensive form games (EFGs). EFG can also be written as SPPs with bilinear objective and $\mathcal{X}, \mathcal{Y}$ polytopes encoding the players' space of sequential strategies [vS96]. EFGs can be solved via simplex-based regret minimization by using the counterfactual regret minimization (CFR) framework to decompose regrets into local regrets at each simplex. Explaining CFR is beyond the scope of this work; we point the reader to [ZJBP07] or newer explanations [FKS19c, FKS19a]. We generate 70 synthetic 10-dimensional matrix games with $A_{ij} \sim U[0,1]$ and compare the most efficient algorithms for matrix games with linear averaging: CBA$^+$ and RM$^+$. Figure 1a presents the duality gaps of the current solutions vs. the number of steps. Here, both algorithms use *alternation*, which is a trick that is well-known to improve performance of RM$^+$ [TBJB15], where the repeated game framework is changed such that players take turns updating their strategies, rather than performing these updates simultaneously, see Appendix D for details.[1]

For EFGs, we compare CBA$^+$and CFR$^+$ on many poker AI benchmark instances, including Leduc, Kuhn, search games and sheriff (see [FKS21] for game descriptions). We present our results in Figures 1b-1d. Additional details and experiments are presented in Appendix D. Overall, we see in Figure 1 that CBA$^+$may slightly outperform RM$^+$ and CFR$^+$, two of the strongest algorithms for matrix games and EFGs, which were shown to achieve the best empirical performances compared to a wide range of algorithms, including Hedge and other first-order methods [Kro20, KFS18, FKS19b]. Recall that our goal is to generalize these strong performance to other settings: we present our numerical experiments for solving distributionally robust optimization problems in the next section.




(a) Matrix game. (b) EFG: Leduc, 2 players, 9 ranks. (c) EFG: Battleship, 3 turns, 1 ship. (d) EFG: Leduc, 2 players, 5 ranks.

Figure 1: Comparison of CBA$^+$ with RM$^+$ and CFR$^+$ on matrix games and EFGs.

## 4.2 Distributionally Robust Optimization

**Problem setup** Broadly speaking, DRO attempts to exploit partial knowledge of the statistical properties of the model parameters to obtain risk-averse optimal solutions [RM19]. We focus on the following instance of distributionally robust classification with logistic losses [ND16, BTHKM15]. There are $m$ observed feature-label pairs $(\boldsymbol{a}_i, b_i) \in \mathbb{R}^n \times \{-1, 1\}$, and we want to solve

$$\min_{\boldsymbol{x}\in\mathbb{R}^n, \|\boldsymbol{x}-\boldsymbol{x}_0\|_2\le R} \; \max_{\boldsymbol{y}\in\Delta(m), \|\boldsymbol{y}-\boldsymbol{y}_0\|_2^2\le\lambda} \sum_{i=1}^{m} y_i \ell_i(\boldsymbol{x}), \tag{4}$$

where $\ell_i(\boldsymbol{x}) = \log(1+\exp(-b_i \boldsymbol{a}_i^\top \boldsymbol{x}))$. Note that (4) takes a worst-case approach to put more weight on misclassified observations. The formulation (4) provides some statistical guarantees, e.g., it can be seen as a convex regularization of standard empirical risk minimization instances [DGN21].

[1] We note that RM$^+$ is guaranteed to retain its convergence rate under alternation. In contrast, we leave resolving this property for CBA$^+$ to future work.

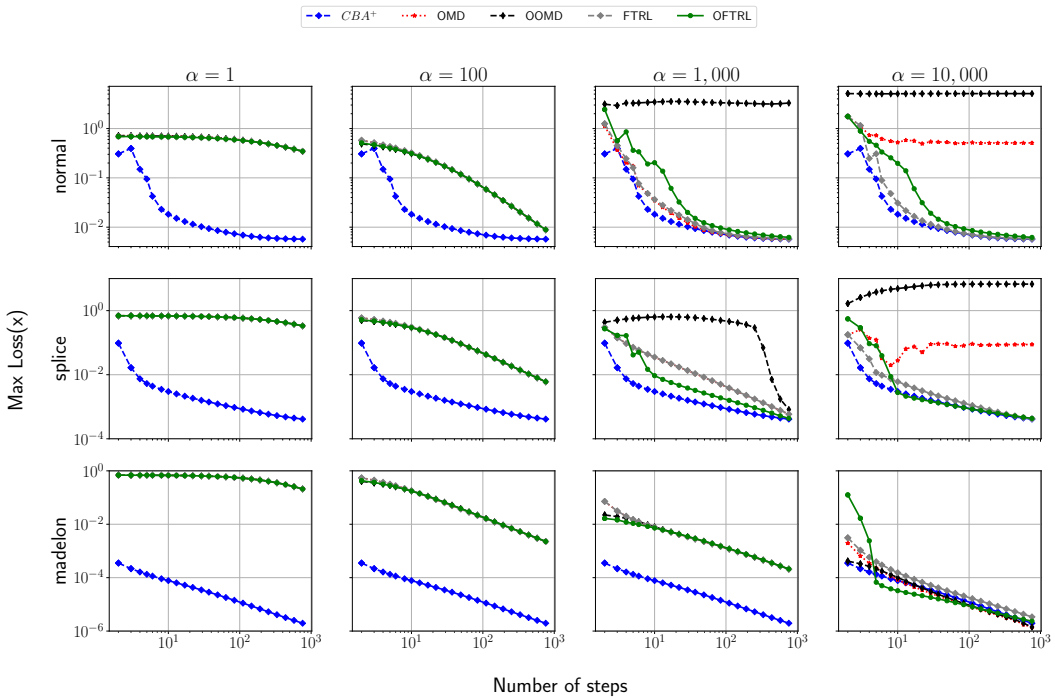



Figure 2: Comparisons of the performances of CBA+ with OMD,FTRL,O-OMD and O-FTRL with fixed step sizes, on synthetic (with *normal* distribution) and real data sets (*splice* and *madelon*).

We compare CBA+ (with linear averaging and alternation) with Online Mirror Descent (OMD), Optimistic OMD (O-OMD), Follow-The-Regularized-Leader (FTRL) and Optimistic FTRL (O-FTRL). We provide a detailed presentation of our implementations of these algorithms in Appendix E. We compare the performances of these algorithms with CBA+ on two synthetic data sets and four real data sets. We use linear averaging on decisions for all algorithms, and parameters $\boldsymbol{x}_0 = \mathbf{0}, R = 10, \boldsymbol{y}_0 = (1, ..., 1)/m, \lambda = 1/2m$ in eq. (4).

**Synthetic and real instances** For the synthetic classification instances, we generate an optimal $\boldsymbol{x}^* \in \mathbb{R}^n$, sample $\boldsymbol{a}_i \sim N(0, \boldsymbol{I})$ for $i \in \{1, ..., m\}$, set labels $b_i = \text{sign}(\boldsymbol{a}_i^\top \boldsymbol{x}^*)$, and then we flip $10\%$ of them. For the real classification instances, we use the following data sets from the libsvm website[2]: *adult*, *australian*, *splice*, *madelon*. Details about the empirical setting, the data sets and additional numerical experiments are presented in Appendix F.

**Choice of step sizes** One of the main motivation for CBA+ is to obtain a *parameter-free* algorithm. All four other algorithms require some step size $\eta$, which require knowing a bound $L$ on the norm of the instantaneous payoffs (see Appendix E.2 for our derivations of this upper bound). This is a major limitation in practice: these bounds may be very conservative, leading to small step sizes. We highlight this by showing the performance of all four algorithms, for various fixed step sizes $\eta = \alpha \times \eta_{\mathsf{th}}$, where $\alpha \in \{1, 100, 1,000, 10,000\}$ is a multiplier and $\eta_{\mathsf{th}}$ is the theoretical step size which guarantees the convergence of the algorithms for each instance. We present the results of our numerical experiments on synthetic and real data sets in Figure 2. Additional simulations with adaptive step sizes $\eta_t = 1/\sqrt{\sum_{\tau=1}^{t-1} \|\boldsymbol{f}_\tau\|_2^2}$ [Ora19] are presented in Figure 3 and in Appendix F.

**Results and discussion** In Figure 2, we present the worst-case loss of the current solution $\bar{\boldsymbol{x}}_T$ in terms of the number of steps $T$. We see that when the step sizes is chosen as the theoretical step sizes guaranteeing the convergence of the non-parameter free algorithms ($\alpha = 1$), CBA+ vastly outperforms all of the algorithms. When we take more aggressive step sizes, the non-parameter-free algorithms become more competitive. For instance, when $\alpha = 1,000$, OMD, FTRL and O-FTRL are competitive with CBA+ for the experiments on synthetic data sets. However, for this same instance and $\alpha = 1,000$, O-OMD diverges, because the step sizes are far greater than the theoretical

[2]https://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/

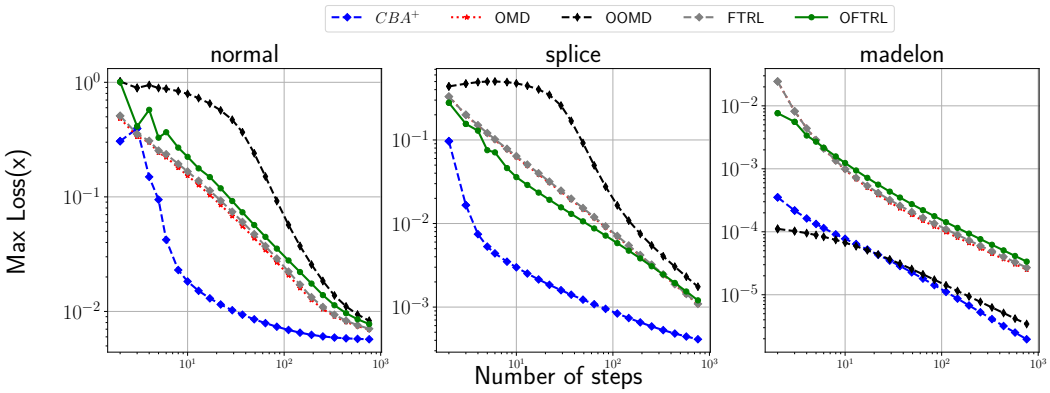



Figure 3: Comparisons of the performances of CBA$^+$, OMD,FTRL,O-OMD and O-FTRL with adaptive step sizes, on synthetic (with *normal* distribution) and real data sets (*splice* and *madelon*).

step sizes guaranteeing convergence. At $\alpha = 10,000$, both OMD and O-OMD diverge. The same type of performances also hold for the *splice* data set. Finally, for the *madelon* data set, the non parameter-free algorithms start to be competitive with CBA$^+$ only when $\alpha = 10,000$. Again, we note that this range of step sizes $\eta$ is completely outside the values $\eta_{\mathsf{th}}$ that guarantee convergence of the algorithms. Similar observations hold for adaptive step sizes (see Figure 3 and Appendix F). The overall poor performances of the optimistic methods (compared to their $O(1/T)$ average regret guarantees) may reflect their sensibility to the choice of the step sizes. Additional experiments in Appendix F with more real EFG and DRO instances show the robustness of the strong performances of CBA$^+$ across additional problem instances.

**Running times compared to CBA$^+$** We conclude this section by discussing the running times of our algorithm. For solving (4), OMD, FTRL, O-OMD, and O-FTRL require binary searches at each period, see Appendix E. However, the functions used in the binary searches themselves require solving an optimization program (an orthogonal projection onto the simplex, see (31)) at each evaluation. Even though computing the orthogonal projection of a vector onto the simplex can be done in $O(n \log(n))$, this results in slower overall running time, compared to CBA$^+$. We acknowledge that the same holds for CBA$^+$ compared to RM$^+$. In particular, CBA$^+$ is slightly slower than RM$^+$, because of the computation of $\pi_{\mathcal{C}}(\boldsymbol{u})$ in $O(n \log(n))$ operations at every iteration. Because we did not aggressively optimize our code for RM$^+$, CBA$^+$, and the other algorithms, we only roughly report here the observed running times. Empirically, we observe that CBA$^+$ is 1x-1.5x slower than RM$^+$. In contrast, CBA$^+$ is 2x-2.5x faster than OMD, FTRL and O-FTRL, and 3x-4x faster than O-OMD, which requires two proximal updates at each iteration.

# 5 Conclusion

We have introduced CBA$^+$, a new algorithm for convex-concave saddle-point solving, that is 1) simple to implement for many practical decision sets, 2) completely parameter-free, and 3) competitive with, or even better than, state-of-the-art approaches for the best choices of parameters, both for matrix games, extensive-form games, and distributionally robust instances. Interesting future directions of research include developing a theoretical understanding of the improvements related to alternation in our setting, designing efficient implementations for other widespread decisions sets (e.g., based on Kullback-Leibler divergence or $f$-divergence), and novel accelerated versions based on strong convex-concavity or optimistim.

**Societal impact** Our work enables faster and simpler computation of solutions of saddle-point problems, which have become widely used for applications in the industry. There is a priori no direct negative societal consequence to this work, since our methods simply return the same solutions as previous algorithms but in a more efficient and simpler way.

# References


[ABH11] Jacob Abernethy, Peter L Bartlett, and Elad Hazan. Blackwell approachability and no-regret learning are equivalent. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 27–46. JMLR Workshop and Conference Proceedings, 2011.

[AHR09] Jacob D Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. 2009.

[BBJT15] Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin. Heads-up limit hold'em poker is solved. *Science*, 347(6218):145–149, 2015.

[BdHP19] Dimitris Bertsimas, Dick den Hertog, and Jean Pauphilet. Probabilistic guarantees in robust optimization. 2019.

[Bla56] David Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6(1):1–8, 1956.

[BMS19] Neil Burch, Matej Moravcik, and Martin Schmid. Revisiting cfr+ and alternating updates. *Journal of Artificial Intelligence Research*, 64:429–443, 2019.

[BS18] Noam Brown and Tuomas Sandholm. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018.

[BS19] Noam Brown and Tuomas Sandholm. Superhuman AI for multiplayer poker. *Science*, 365(6456):885–890, 2019.

[BT03] Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

[BTHKM15] Aharon Ben-Tal, Elad Hazan, Tomer Koren, and Shie Mannor. Oracle-based robust optimization via online learning. *Operations Research*, 63(3):628–638, 2015.

[BTN01] Aharon Ben-Tal and Arkadi Nemirovski. *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*, volume 2. Siam, 2001.

[CP11] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40(1):120–145, 2011.

[CP16] Antonin Chambolle and Thomas Pock. On the ergodic convergence rates of a first-order primal–dual algorithm. *Mathematical Programming*, 159(1-2):253–287, 2016.

[CR13] Patrick L Combettes and Noli N Reyes. Moreau's decomposition in banach spaces. *Mathematical Programming*, 139(1):103–114, 2013.

[CYL+12] Chao-Kai Chiang, Tianbao Yang, Chia-Jung Lee, Mehrdad Mahdavi, Chi-Jen Lu, Rong Jin, and Shenghuo Zhu. Online optimization with gradual variations. In *Conference on Learning Theory*, pages 6–1. JMLR Workshop and Conference Proceedings, 2012.

[DGN21] John C Duchi, Peter W Glynn, and Hongseok Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *Mathematics of Operations Research*, 2021.

[DSSSC08] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the L-1 ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 272–279, 2008.

[EPGMFBV03] Juan José Egozcue, Vera Pawlowsky-Glahn, Glòria Mateu-Figueras, and Carles Barcelo-Vidal. Isometric logratio transformations for compositional data analysis. *Mathematical Geology*, 35(3):279–300, 2003.

[FKS19a] Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Online convex optimization for sequential decision processes and extensive-form games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1917–1925, 2019.

[FKS19b] Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Optimistic regret minimization for extensive-form games via dilated distance-generating functions. In *Advances in Neural Information Processing Systems*, pages 5222–5232, 2019.

[FKS19c] Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Regret circuits: Composability of regret minimizers. In *International Conference on Machine Learning*, pages 1863–1872, 2019.

[FKS21] Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Faster game solving via predictive blackwell approachability: Connecting regret matching and mirror descent. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI, 2021.

[GCK20a] Julien Grand-Clément and Christian Kroer. First-order methods for Wasserstein distributionally robust MDP. *arXiv preprint arXiv:2009.06790*, 2020.

[GCK20b] Julien Grand-Clément and Christian Kroer. Scalable first-order methods for robust mdps. *arXiv preprint arXiv:2005.05434*, 2020.

[GGC18] Vineet Goyal and Julien Grand-Clément. Robust Markov decision process: Beyond rectangularity. *arXiv preprint arXiv:1811.00215*, 2018.

[GKG19] Yuan Gao, Christian Kroer, and Donald Goldfarb. Increasing iterate averaging for solving saddle-point problems. *arXiv preprint arXiv:1903.10646*, 2019.

[HMC00] Sergiu Hart and Andreu Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000.

[Iye05] Garud Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2):257–280, 2005.

[JS20] Yujia Jin and Aaron Sidford. Efficiently solving mdps with stochastic mirror descent. In *International Conference on Machine Learning*, pages 4890–4900. PMLR, 2020.

[KFS18] Christian Kroer, Gabriele Farina, and Tuomas Sandholm. Solving large sequential games with the excessive gap technique. In *Advances in Neural Information Processing Systems*, pages 864–874, 2018.

[Kro20] Christian Kroer. Ieor8100: Economics, ai, and optimization lecture note 5: Computing Nash equilibrium via regret minimization. 2020.

[KWKKS20] Christian Kroer, Kevin Waugh, Fatma Kılınç-Karzan, and Tuomas Sandholm. Faster algorithms for extensive-form game solving via improved smoothing functions. *Mathematical Programming*, pages 1–33, 2020.

[McM11] Brendan McMahan. Follow-the-regularized-leader and mirror descent: Equivalence theorems and l1 regularization. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 525–533. JMLR Workshop and Conference Proceedings, 2011.

[MSB+17] Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisỳ, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356(6337):508–513, 2017.

[ND16] Hongseok Namkoong and John C Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. In *NIPS*, volume 29, pages 2208–2216, 2016.

[Nem04] Arkadi Nemirovski. Prox-method with rate of convergence O(1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

[NY83] Arkadi Nemirovski and David Yudin. *Problem complexity and method efficiency in optimization.* 1983.

[Ora19] Francesco Orabona. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*, 2019.

[RM19] Hamed Rahimian and Sanjay Mehrotra. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*, 2019.

[RS13] Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. In *Conference on Learning Theory*, pages 993–1019. PMLR, 2013.

[ST18] Aaron Sidford and Kevin Tian. Coordinate methods for accelerating l-infinity regression and faster approximate maximum flow. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 922–933. IEEE, 2018.

[TBJB15] Oskari Tammelin, Neil Burch, Michael Johanson, and Michael Bowling. Solving heads-up limit Texas hold'em. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[vS96] Bernhard von Stengel. Efficient computation of behavior strategies. *Games and Economic Behavior*, 14(2):220–246, 1996.

[WKR13] W. Wiesemann, D. Kuhn, and B. Rustem. Robust Markov decision processes. *Operations Research*, 38(1):153–183, 2013.

[ZJBP07] Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. In *Advances in neural information processing systems*, pages 1729–1736, 2007.

# Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] in future works in our conclusion
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] In the supplemental material
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Yes in Appendices F and Appendices D.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] in Appendix D.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] the libsvm license is mentioned in Section 4.
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]