Mathematics of Operations Research

Solving Optimization Problems with Blackwell Approachability

Julien Grand-Clément, Christian Kroer

https://pubsonline.informs.org/journal/moor

MATHEMATICS OF OPERATIONS RESEARCH
Articles in Advance, pp. 1–32
ISSN 0364-765X (print), ISSN 1526-5471 (online)

Solving Optimization Problems with Blackwell Approachability

Julien Grand-Clément,[a,*] **Christian Kroer**[b]

[a] Department of Information Systems and Operations Management, École des Hautes Études Commerciales de Paris (HEC Paris), 78350 Jouy-en-Josas, France; [b] Department of Industrial Engineering and Operations Research, Columbia University, New York, New York 10027
*Corresponding author

Contact: grand-clement@hec.edu, https://orcid.org/0000-0002-2864-8779 (JG-C); christian.kroer@columbia.edu, https://orcid.org/0000-0002-9009-8683 (CK)

Received: August 5, 2022
Revised: March 3, 2023
Accepted: April 7, 2023
Published Online in Articles in Advance: May 16, 2023

MSC2020 Subject Classifications: Primary: 90C47; secondary: 90C25

https://doi.org/10.1287/moor.2023.1376

Copyright: © 2023 INFORMS

Abstract. In this paper, we propose a new algorithm for solving convex-concave saddle-point problems using regret minimization in the repeated game framework. To do so, we introduce the Conic Blackwell Algorithm$^+$ (CBA$^+$), a new parameter- and scale-free regret minimizer for general convex compact sets. CBA$^+$ is based on Blackwell approachability and attains $O(\sqrt{T})$ regret. We show how to efficiently instantiate CBA$^+$ for many decision sets of interest, including the simplex, ℓ_p norm balls, and ellipsoidal confidence regions in the simplex. Based on CBA$^+$, we introduce SP-CBA$^+$, a new parameter-free algorithm for solving convex-concave saddle-point problems achieving a $O(1/\sqrt{T})$ ergodic convergence rate. In our simulations, we demonstrate the wide applicability of SP-CBA$^+$ on several standard saddle-point problems from the optimization and operations research literature, including matrix games, extensive-form games, distributionally robust logistic regression, and Markov decision processes. In each setting, SP-CBA$^+$ achieves state-of-the-art numerical performance and outperforms classical methods, without the need for any choice of step sizes or other algorithmic parameters.

Funding: J. Grand-Clément is supported by the Agence Nationale de la Recherche [Grant 11-LABX-0047] and by Hi! Paris. C. Kroer is supported by the Office of Naval Research [Grant N00014-22-1-2530] and by the National Science Foundation [Grant IIS-2147361].

Keywords: Blackwell approachability • no-regret algorithm • parameter-free algorithm • saddle point

1. Introduction

In this paper, we develop new algorithms for solving the following convex-concave *saddle-point problems* (SPPs):

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

where $\mathcal{X} \subset \mathbb{R}^n, \mathcal{Y} \subset \mathbb{R}^m$ are convex, compact sets and $F : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is a subdifferentiable convex-concave function. The optimization Problem (1) arises in a number of practical problems in optimization, machine learning, and operations research. For example, the problem of computing a Nash equilibrium of a zero-sum game can be formulated as a convex-concave SPP, and this is the foundation of most methods for solving sequential zero-sum games (Kroer et al. [42], Tammelin et al. [61], von Stengel [63], Zinkevich et al. [66]). Other instances include imaging (Chambolle and Pock [18]), ℓ_∞ regression (Sidford and Tian [58]), Markov decision processes (MDPs) and robust MDPs (Iyengar [37], Sidford and Tian [58], Wiesemann et al. [65]), market equilibrium (Kroer et al. [41]), and distributionally robust logistic regression, where the max term represents the distributional uncertainty (Ben-Tal et al. [8], Namkoong and Duchi [47]). We introduce efficient algorithms for solving (1), focusing on *parameter-free* algorithms that do not require choosing, learning, or tuning any step sizes.

1.1. Repeated Game Framework

One way to solve convex-concave SPPs is by viewing the SPP as a repeated game between two players; at each iteration t, one player chooses $x_t \in \mathcal{X}$, the other player chooses $y_t \in \mathcal{Y}$, and then, the players observe $F(x_t, y_t)$. If each player employs a regret-minimization algorithm, then a well-known theorem says that the uniform average of the decisions generated by the players converges to a solution to the SPP (see Theorem 1 in Section 2). We will call this the "repeated game framework." There are already well-known algorithms for instantiating the repeated game framework for (1). For example, one can employ the *online mirror descent* (OMD) algorithm (Nemirovski and Yudin [49]), which generates iterates as follows for the first player (and similarly for the second player):

$$x_{t+1} = \arg\min_{x \in \mathcal{X}} \langle \eta f_t, x \rangle + D(x, x_t), \tag{2}$$

where $\boldsymbol{f}_t \in \partial_x F(\boldsymbol{x}_t, \boldsymbol{y}_t)$ (∂_x denotes the set of subgradients as regards the variable $\boldsymbol{x}$), $\eta > 0$ is an appropriate step size, and D is a *Bregman divergence*, which measures distance between points. Another example of a regret minimizer is follow the regularized leader (FTRL) (Abernethy et al. [2]), which generates updates as follows:

$$\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x} \in \mathcal{X}} \left\langle \eta \sum_{\tau=1}^{t} \boldsymbol{f}_\tau, \boldsymbol{x} \right\rangle + \phi(\boldsymbol{x}), \tag{3}$$

where $\phi : \mathcal{X} \to \mathbb{R}$ is a distance-generating function. The updates (2) and (3) can be computed efficiently for many decision sets $\mathcal{X}$; for instance, when D and ϕ are the squared ℓ_2 norm, (2) and (3) can be computed in $O(n)$ arithmetic operations for ℓ_p balls with $p \in \{1, 2, \infty\}$ and $O(n \log(n))$ operations for the simplex, and when ϕ is the entropy and D is the *Kullback–Leibler divergence*, (2) and (3) can be computed in $O(n)$ operations for the simplex. OMD and FTRL can achieve average regret on the order of $O(1/\sqrt{T})$ after T iterations. For instance, when D and ϕ are the squared ℓ_2 norm, this regret can be achieved by choosing a fixed step size $\eta = \sqrt{2}\Omega/L\sqrt{T}$, where L is an upper bound on the ℓ_2 norms of the subgradients $(\boldsymbol{f}_t)_{t \geq 0}$ and $\Omega = \max\{\|\boldsymbol{x} - \boldsymbol{x}'\|_2 \,|\, \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}\}$. Choosing the step size η is problematic, as it requires choosing in advance the number of iterations T and to know the upper bound L, which may be hard to obtain in many applications or too conservative in practice. This can even be practically infeasible for very large instances because we do not know if the step size will cause a divergence until late in the optimization process. Alternatively, it is possible to choose changing step sizes $\eta_t = \alpha/\sqrt{t}$ for $\alpha > 0$. Still, adequately tuning the parameter α can be time and resource consuming. This is not just a theoretical issue, as we highlight in our numerical experiments (Section 5) and in the appendices (Appendix J).

These issues can be addressed by employing *adaptive* step sizes, which estimate the parameters through the observed subgradients (e.g., AdaHedge for the simplex setting (De Rooij et al. [23]) or AdaFTRL for general compact convex decisions sets (Orabona and Pál [52])). These adaptive variants have not seen practical adoption in large-scale game solving, where variants based on Blackwell approachability are preferred (see the next paragraph). As we show in our experiments, adaptive variants of OMD and FTRL perform much worse than our proposed algorithms. Although these adaptive algorithms are referred to as *parameter free*, this is only true in the sense that they are able to learn the necessary parameters. Our algorithm is parameter free in the stronger sense that there are no parameters that even require learning.

1.2. Blackwell Approachability

In this paper, we use the framework of *Blackwell* approachability (Blackwell [11]) to develop novel parameter-free algorithms for solving the convex-concave saddle-point Problem (1). We refer the reader to Perchet [54] for a survey on Blackwell approachability. In principle, Blackwell approachability arises in the framework of repeated two-player games with vector-valued payoffs; the goal of the first-player is to choose a sequence of decisions $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots,$ such that the *average* of the visited payoffs converges to a known target set $\mathcal{S}$, whereas the second-player wishes to prevent this. Blackwell's celebrated theorem (Blackwell [11]) provides an algorithm for constructing such a sequence of decisions $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots,$ in the case where the target set $\mathcal{S}$ is *half-space forceable* (see details in Section 2).

Blackwell approachability is a very general framework, and the applications are numerous, ranging from stochastic games (Milman [44]); revenue management, market design, and submodular maximization (Niazadeh et al. [51]); calibration (Perchet [53]), and learning in games (Aumann et al. [5]) to fair online learning (Chzhen et al. [21]). In particular, Blackwell approachability can be used as a regret minimizer (Abernethy et al. [1], Blackwell [12]) and provides a no-regret algorithm, with an average regret of $O(1/\sqrt{T})$ after T iterations. Crucially, when applied to online regret minimization, Blackwell approachability can be instantiated without any choices of parameters, and the resulting no-regret algorithm does not use any step sizes; this is in contrast to classical regret minimizers, such as OMD (2) and FTRL (3), which require choosing step sizes.

Despite its appealing properties from a theoretical standpoint, in practice Blackwell approachability is not widely used to solve classical problems from the operations research literature. In fact, to the best of our knowledge, the only practical implementation of Blackwell approachability for solving (1) is for the case of bilinear games on the simplex, where $F(\boldsymbol{x}, \boldsymbol{y}) = \langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{y} \rangle$ for $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ and $\mathcal{X}, \mathcal{Y}$ are simplices. This simplex instantiation is also used for extensive-form games (EFGs) via the aforementioned counterfactual regret minimization (CFR) decomposition (Farina et al. [26], Zinkevich et al. [66]). In the simplex setting, a particular application of Blackwell approachability yields a no-regret algorithm called *regret matching* (RM) (Hart and Mas-Colell [35]). Combining RM with specific weighting, thresholding, and alternating schemes yields an algorithm called *regret matching*$^+$ (RM$^+$) (Tammelin et al. [61]). RM$^+$ has been used in *every* case of solving extremely large-scale EFGs in practice, and in particular, it was used in recent poker artificial intelligence milestones, where poker AIs beat human poker players (Bowling et al. [13], Brown and Sandholm [14], Brown and Sandholm [16], Moravčík et al. [46]). In fact, RM$^+$ routinely outperforms theoretically superior methods, such as optimistic variants of OMD and FTRL (Chiang et al. [20], Rakhlin and Sridharan [57]), which achieve $O(1/T)$ convergence rates in the repeated game framework. Despite its very strong empirical performances, RM$^+$ is only defined when the decision set is the 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) (Ben-Tal et al. [8]). Encouraged by the very strong empirical performance of RM$^+$ and CFR$^+$, we will construct parameter-free algorithms based on Blackwell approachability for solving more general instances of the saddle-point Problem (1).

Our main contributions are as follows.

- *Conic Blackwell Algorithm*$^+$ (CBA$^+$). We start from the general reduction between regret minimization over general convex compact sets and Blackwell approachability (Abernethy et al. [1]). This yields a regret minimizer, which we will refer to as the *conic Blackwell algorithm* (CBA). Motivated by the practical performance of RM$^+$ on simplexes, we construct a variant of CBA, which uses a thresholding operation analogous to the one employed by RM$^+$. We call this regret minimizer CBA$^+$ (Algorithm 1). We show that CBA$^+$ achieves $O(1/\sqrt{T})$ average regret in the worst case, and we show strong guarantees for the tracking regret of CBA$^+$. A major selling point of CBA$^+$ is that it does not require any step size choices; it implicitly adjusts to the structure of the domains and losses by being instantiations of a Blackwell approachability algorithm, which is itself parameter free.
- *Impacts of weights and alternation*. As regret minimizers, we show that both CBA and CBA$^+$ are compatible with increasing weighting schemes that put more weights on more recent decisions and losses (Theorems 2 and 3). Moreover, we show that CBA$^+$ is compatible with different weighting schemes for the decisions and the losses. We then introduce a new algorithm for solving convex-concave saddle-point problems by using CBA$^+$ in a repeated game framework with linear weights on the sequence of decisions and uniform weights on the losses (this is known as *linear averaging* in other algorithms (Tammelin et al. [61])), as well as an alternating update scheme. We call this algorithm SP-CBA$^+$. We quantify the benefits of alternation for solving (1) with SP-CBA$^+$ (Theorem 7) and show the first strict improvement guarantee for using alternation. The method that we develop in our proof for this result is very general, and we adapt it to prove the same strict improvements for combining alternation with RM and RM$^+$. Prior results on RM and RM$^+$ only showed that alternation does not hurt the convergence guarantee (Burch et al. [17]).
- *Efficient implementation of* CBA$^+$. We show how to implement CBA and CBA$^+$ when $\mathcal{X}$ and $\mathcal{Y}$ are simplexes, ℓ_p balls, and intersections of the ℓ_2 ball with a simplex, which arises naturally as a confidence region. More generally, CBA and CBA$^+$ can be implemented when we can efficiently compute orthogonal projections onto the set $\mathcal{X}$ and $\mathcal{Y}$. Note that the general reduction of regret minimization and Blackwell approachability from Abernethy et al. [1] yields CBA but does not yield a practically implementable algorithm, as the authors do not consider which decision sets allow for efficient projections.
- *Practical performance of* SP-CBA$^+$. We study the practical efficacy of our algorithmic framework on several domains. First, we apply SP-CBA$^+$ to two-player zero-sum matrix games, where the objective function is bilinear, and we compare with RM$^+$, as well as with AdaHedge and AdaFTRL, two adaptive first-order algorithms. We then apply SP-CBA$^+$ to EFGs, where the RM$^+$ regret minimizer combined with linear averaging, alternation, and a counterfactual regret (CFR$^+$) minimization scheme leads to state-of-the-art practical algorithms (Gao et al. [29], Kroer et al. [42], Tammelin et al. [61]). For EFGs, we find that SP-CBA$^+$ leads to comparable performance in terms of the iteration complexity, and for some games, it slightly outperforms CFR$^+$. In the simplex setting, we also find that SP-CBA$^+$ outperforms both AdaHedge and AdaFTRL. These results show that SP-CBA$^+$ recovers the strong practical performance of RM$^+$ and CFR$^+$ in the only setting where these two methods apply. Second and more importantly, we show that SP-CBA$^+$ leads to strong practical performance in settings where RM$^+$ and CFR$^+$ do not apply. We consider instances of distributionally robust logistic regression and MDPs. For these two instances of saddle-point problems, we find that SP-CBA$^+$ performs orders of magnitude better than online mirror descent and follow-the-regularized leader, as well as their optimistic variants, when using their theoretically correct fixed step sizes. Even when considering tuned step sizes for the other algorithms, SP-CBA$^+$ performs better, with only a few cases of comparable performance (at step sizes that lead to divergence for some of the other nonparameter-free methods). We also find that SP-CBA$^+$ outperforms a vanilla implementation of CBA combined with the repeated game framework, which highlights the improved practical performance of our algorithm. 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 optimization instances arising naturally in the operations research literature.

Compared with an earlier conference version (Grand-Clément and Kroer [32]), the present paper proves the convergence guarantees of CBA$^+$ with alternation (and of RM and RM$^+$ as a by-product of our novel proofs), a crucial component of its strong empirical performances. It also introduces the tracking regret guarantees for CBA$^+$, presents simpler and more general proofs for the convergence of CBA$^+$, and compares the practical performance of CBA$^+$ on new and important applications.

We conclude our introduction with a brief discussion on the average regret achieved by other methods and resulting convergence to a saddle point. Our algorithm SP-CBA$^+$ has a rate of convergence toward a saddle point of $O(1/\sqrt{T})$, similar to OMD and FTRL. In theory, it is possible to obtain a faster $O(1/T)$ rate of convergence when F is differentiable with Lipschitz gradients: for example, via mirror prox (Nemirovski [48]) or other primal-dual algorithms (Chambolle and Pock [19]). However, our experimental results show that SP-CBA$^+$ is faster than optimistic variants of FTRL and OMD (Syrgkanis et al. [60]), 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 RM$^+$-based algorithms have better practical performance than the theoretically superior $O(1/T)$-rate methods (Kroer et al. [40], Kroer et al. [42]). In a similar vein, using *error-bound conditions*, it is possible to achieve a linear rate (e.g., when solving bilinear saddle-point problems over polyhedral decision sets) by using the extragradient method (Tseng [62]) or optimistic gradient descent-ascent (Wei et al. [64]). However, these linear rates rely on unknown constants and may not be indicative of practical performance.

2. Repeated Game Framework and Blackwell Approachability

We will solve (1) using a repeated game framework. For any $\boldsymbol{x} \in \mathcal{X}, \boldsymbol{y} \in \mathcal{Y}$, we assume that F is subdifferentiable in $\boldsymbol{x}$ and superdifferentiable in $\boldsymbol{y}$; we define $\partial_x F(\boldsymbol{x},\boldsymbol{y})$ to be the set of subgradients of F with respect to $\boldsymbol{x}$ at some $(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{X} \times \mathcal{Y}$:

$$\partial_x F(\boldsymbol{x},\boldsymbol{y}) = \{\boldsymbol{f} \in \mathbb{R}^n \,|\, F(\boldsymbol{z},\boldsymbol{y}) \geq F(\boldsymbol{x},\boldsymbol{y}) + \langle \boldsymbol{f}, \boldsymbol{z}-\boldsymbol{x}\rangle,\ \forall\, \boldsymbol{z} \in \mathcal{X}\},$$

and we define $\partial_y F(\boldsymbol{x},\boldsymbol{y}) \subset \mathbb{R}^m$ as the set of supergradients of F with respect to $\boldsymbol{y}$ at $(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{X} \times \mathcal{Y}$:

$$\partial_y F(\boldsymbol{x},\boldsymbol{y}) = \{\boldsymbol{g} \in \mathbb{R}^m \,|\, F(\boldsymbol{x},\boldsymbol{z}) \leq F(\boldsymbol{x},\boldsymbol{y}) + \langle \boldsymbol{g}, \boldsymbol{z}-\boldsymbol{y}\rangle,\ \forall\, \boldsymbol{z} \in \mathcal{Y}\}.$$

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 decisions $\boldsymbol{x}_t \in \mathcal{X}, \boldsymbol{y}_t \in \mathcal{Y}$.
2. The first player observes $\boldsymbol{f}_t \in \partial_x F(\boldsymbol{x}_t,\boldsymbol{y}_t)$ and uses $\boldsymbol{f}_t$ when computing the next decision.
3. The second player observes $\boldsymbol{g}_t \in \partial_y F(\boldsymbol{x}_t,\boldsymbol{y}_t)$ and uses $\boldsymbol{g}_t$ when computing the next decision.

In the repeated game framework described, the goal of each player is to minimize their regret $R_{T,x}, R_{T,y}$ across the T iterations:

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

The reason this repeated game framework leads to a solution to the SPP Problem (1) is the following well-known theorem (e.g., Kroer [39, theorem 1]). Relying on F being convex-concave and subdifferentiable, it connects the regret incurred by each player to the duality gap in (1).

Theorem 1. *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,*

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

Therefore, when each player uses 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 $O(1/\sqrt{T})$. Later, we will show a generalization of Theorem 1 that will allow us to incorporate increasing averaging schemes that put additional weight on the later iterates and use alternating updates (Theorem 6). 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 matrix games and EFGs, variants of Blackwell approachability are used in practice.

2.1. Blackwell Approachability

In Blackwell approachability, a decision maker repeatedly takes decisions $\boldsymbol{x}_t$ from some 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 $\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{u}_t(\boldsymbol{x}_t)$ to approach some convex target set $\mathcal{S}$. 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 $\bar{\boldsymbol{u}}_t = \frac{1}{t-1}\sum_{\tau=1}^{t-1} \boldsymbol{u}_\tau(\boldsymbol{x}_\tau)$ onto $\mathcal{S}$, and then, the decision maker chooses an action $\boldsymbol{x}_t$ that forces the tangent half-space to $\mathcal{S}$ generated by the normal vector $\bar{\boldsymbol{u}}_t - \pi_{\mathcal{S}}(\bar{\boldsymbol{u}}_t)$, where $\pi_{\mathcal{S}}(\bar{\boldsymbol{u}}_t)$ is the orthogonal projection of $\bar{\boldsymbol{u}}_t$ onto $\mathcal{S}$. We call this algorithm *Blackwell's algorithm*; it approaches $\mathcal{S}$ at a rate of $O(1/\sqrt{T})$ (Blackwell [11]). In particular, for $d(\bar{\boldsymbol{u}}_T, \mathcal{S})$ defined as $d(\bar{\boldsymbol{u}}_T, \mathcal{S}) = \min\{\|\bar{\boldsymbol{u}}_T - \boldsymbol{z}\|_2 \,|\, \boldsymbol{z} \in \mathcal{S}\}$, we have $d(\bar{\boldsymbol{u}}_T, \mathcal{S}) = O(1/\sqrt{T})$. Blackwell's algorithm is really a meta-algorithm, rather than a concrete algorithm. Even within the context of the Blackwell approachability problem, one needs to devise a way to compute the forcing actions needed at each iteration (i.e., to compute $\pi_{\mathcal{S}}(\bar{\boldsymbol{u}})$). To the best of our knowledge, prior to this paper, the only practical implementation of Blackwell approachability for solving (1) is on the simplex for solving bilinear saddle-point problems and extensive-form games, which leads to RM and RM$^+$.

2.2. Details on Regret Matching

Let $\Delta(n)$ be the n-dimensional probability simplex. RM arises by instantiating Blackwell approachability with the decision space $\mathcal{X}$ equal to $\Delta(n)$, 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 with each of the n actions (which correspond to the vertices of $\Delta(n)$). Here, $\boldsymbol{e} = (1, \ldots, 1) \in \mathbb{R}^n$. Hart and Mas-Colell [35] 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 $x_{t+1,i} = [r_{t,i}]^+ / \sum_{i=1}^{n} [r_{t,i}]^+$, 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: $\boldsymbol{r}_t = [\boldsymbol{r}_{t-1} + \boldsymbol{f}_t - \langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle \boldsymbol{e}]^+$; then, actions are again played proportional to $\boldsymbol{r}_t$. In the next section, we describe a framework by Abernethy et al. [1] for using Blackwell's algorithm to construct regret minimizers for more general convex sets $\mathcal{X}$; this will lead to the CBA algorithm, from which we will construct CBA$^+$.

3. Conic Blackwell Algorithm

3.1. Our Algorithm

In this section, we introduce our main regret minimizer, CBA$^+$, which uses a variation of Blackwell's approachability procedure (Blackwell [11]) to perform regret minimization on a general convex compact decision set $\mathcal{X}$. We assume that the sequences $(\boldsymbol{f}_t)_{t\geq 1}$ for the first player and $(\boldsymbol{g}_t)_{t \geq 1}$ for the second player are such that $\|\boldsymbol{f}_t\|_2 \leq L_x, \|\boldsymbol{g}_t\|_2 \leq L_y, \forall t \geq 1,$ for some $L_x > 0, L_y > 0$ (possibly unknown). In the repeated game framework where we use regret minimization to solve a saddle-point Problem (1), this occurs for instance if

$$\forall (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{Y}, \exists\, \boldsymbol{f} \in \partial_x F(\boldsymbol{x}, \boldsymbol{y}), \exists\, \boldsymbol{g} \in \partial_y F(\boldsymbol{x}, \boldsymbol{y}), \|\boldsymbol{f}\| \leq L_x, \|\boldsymbol{g}\| \leq L_y, \tag{4}$$

and the oracle returning a subgradient $\boldsymbol{f}_t \in \partial_x F(\boldsymbol{x}_t, \boldsymbol{y}_t)$ and a supergradient $\boldsymbol{g}_t \in \partial_y F(\boldsymbol{x}_t, \boldsymbol{y}_t)$ ensures that $\|\boldsymbol{f}_t\| \leq L_x, \|\boldsymbol{g}_t\| \leq L_y$ always holds. We will simply write L for L_x or L_y when we focus on the regret of a single player. We will also use the notation $\kappa = \max_{\boldsymbol{x} \in \mathcal{X}} \|\boldsymbol{x}\|_2$ (recall that $\mathcal{X}$ is compact). 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 Abernethy et al. [1], which considers the conic hull $\mathcal{C} = \text{cone}(\{\kappa\} \times \mathcal{X}) \subset \mathbb{R}^{n+1}$. The Blackwell approachability problem is then instantiated with $\mathcal{X}$ as the decision set, the 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 *instantaneous payoff vectors* $\boldsymbol{v} = (\langle \boldsymbol{f}, \boldsymbol{x} \rangle / \kappa, \ -\boldsymbol{f}) \in \mathbb{R}^{n+1}$. The conic Blackwell algorithm (CBA) is implemented by computing the projection $\pi_{\mathcal{C}}(\boldsymbol{u})$ of the *aggregate payoff vector* $\boldsymbol{u}$ onto $\mathcal{C}$, noting that the projection can be written as $\alpha(\kappa, \boldsymbol{x})$ where $\alpha \geq 0$ is a scalar and $\boldsymbol{x} \in \mathcal{X}$, and playing the decision $\boldsymbol{x}$. The second step in CBA$^+$ is to replace the aggregate payoff vector $\boldsymbol{u}$ with a running *projected* aggregate payoff vector, where we always add the instantaneous payoff vector to the aggregate and then project the aggregate onto $\mathcal{C}$.

More concretely, pseudocode for CBA$^+$ is given in Algorithm 1. This pseudocode relies on two functions: $\text{CHOOSEDECISION}_{\text{CBA}^+} : \mathbb{R}^{n+1} \to \mathbb{R}^n$, which maps the aggregate payoff vector $\boldsymbol{u}_t$ to a decision in $\mathcal{X}$, and $\text{UPDATEPAYOFF}_{\text{CBA}^+}$, which controls how we aggregate payoffs. Given a vector $\boldsymbol{u} = (\tilde{u}, \hat{\boldsymbol{u}}) \in \mathbb{R} \times \mathbb{R}^n$, representing a (projected) aggregate payoff $\boldsymbol{u} \in \mathcal{C}$, we define

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

If $\tilde{u} = 0$, we just let $\text{CHOOSEDECISION}_{\text{CBA}^+}(\boldsymbol{u}) = \boldsymbol{x}_0$ for some arbitrary $\boldsymbol{x}_0 \in \mathcal{X}$. The function $\text{UPDATEPAYOFF}_{\text{CBA}^+}$ is implemented by adding the instantaneous payoff vector to the aggregate payoffs and then projecting onto $\mathcal{C}$. More formally, it is defined as

$$\text{UPDATEPAYOFF}_{\text{CBA}^+}(\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{f}, \omega) = \pi_{\mathcal{C}}(\boldsymbol{u} + \omega(\langle \boldsymbol{f}, \boldsymbol{x} \rangle / \kappa, \ -\boldsymbol{f})),$$

where ω is the weight assigned to the instantaneous payoff vector. Because of the projection step in $\mathsf{UPDATEPAYOFF}_{\mathsf{CBA}^+}$, we always have $\boldsymbol{u} \in \mathcal{C}$, which in turn, guarantees that $\mathsf{CHOOSEDECISION}_{\mathsf{CBA}^+}(\boldsymbol{u}) \in \mathcal{X}$ because $\mathcal{C} = \text{cone}(\{\kappa\} \times \mathcal{X})$.

Algorithm 1 (CBA$^+$)

```
Input A convex, compact set $\mathcal{X} \subset \mathbb{R}^n$, $\kappa = \max\{\|x\|_2 \mid x \in \mathcal{X}\}$.
Algorithm parameters Weights $(\omega_t)_{t\geq 1} \in \mathbb{R}^{\mathbb{N}}$.
Initialization $t = 1, x_1 \in \mathcal{X}$.
Observe $f_1$ then set $u_1 = \omega_1(\langle f_1, x_1\rangle/\kappa, -f_1) \in \mathbb{R} \times \mathbb{R}^n$.
for $t \geq 1$ do
    Choose $x_{t+1} = \mathsf{CHOOSEDECISION}_{\mathsf{CBA}^+}(u_t)$.
    Observe the loss $f_{t+1} \in \mathbb{R}^n$.
    Update $u_{t+1} = \mathsf{UPDATEPAYOFF}_{\mathsf{CBA}^+}(u_t, x_{t+1}, f_{t+1}, \omega_{t+1})$.
end for
```

Let us give some intuition on the effect of the projection onto $\mathcal{C}$. For a geometric intuition, it is easier to visualize the dynamics in $\mathbb{R}^2$. Figure 1 illustrates the projection step $\pi_{\mathcal{C}}(\cdot)$ of CBA$^+$. At a high level, from $\boldsymbol{u}_t$ to $\boldsymbol{u}_{t+1}$, an instantaneous payoff vector

$$\boldsymbol{v}_{t+1} = (\langle \boldsymbol{f}_{t+1}, \boldsymbol{x}_{t+1}\rangle/\kappa, -\boldsymbol{f}_{t+1})$$

is first reweighted by ω_{t+1} and added to $\boldsymbol{u}_t$, and then, the resulting vector $\boldsymbol{u}_t^+ = \boldsymbol{u}_t + \omega_{t+1}\boldsymbol{v}_{t+1}$ is projected onto $\mathcal{C}$ to obtain $\boldsymbol{u}_{t+1}$. The projection $\pi_{\mathcal{C}}(\cdot)$ moves the vector $\boldsymbol{u}_t^+$ along the edges of the cone $\mathcal{C}^\circ$, preserving the orthogonal distance d to $\mathcal{C}^\circ$. Intuitively, from a game-theoretic perspective in the case $\mathcal{C} = \mathbb{R}^2_+$, the projection eliminates the negative components of the aggregate payoff, meaning that CBA$^+$ does not remember "negative regrets."

Let us also note the difference between CBA$^+$ and the algorithm introduced in Abernethy et al. [1], which we have called CBA. CBA uses different UPDATEPAYOFF and CHOOSEDECISION functions. In CBA, the aggregate payoff update is defined as

$$\mathsf{UPDATEPAYOFF}_{\mathsf{CBA}}(\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{f}, \omega) = \boldsymbol{u} + \omega(\langle \boldsymbol{f}, \boldsymbol{x}\rangle/\kappa, -\boldsymbol{f}).$$

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

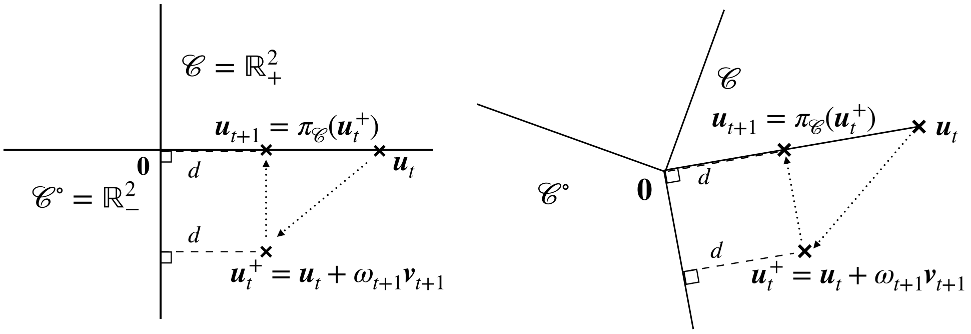
$$\mathsf{CHOOSEDECISION}_{\mathsf{CBA}}(\boldsymbol{u}) = \mathsf{CHOOSEDECISION}_{\mathsf{CBA}^+}(\pi_{\mathcal{C}}(\boldsymbol{u})).$$

Based upon the analysis in Blackwell [11], Abernethy et al. [1] show that CBA with uniform weights (both on payoffs and decisions) guarantees $O(1/\sqrt{T})$ average regret.

3.2. Regret Bounds for CBA and CBA$^+$

In this section, we investigate the theoretical performance guarantees of CBA and CBA$^+$ when we vary the weights on decisions and payoffs. This is motivated by practical performance, where it has been observed in several other settings that increasing weights usually perform better (Brown and Sandholm [15], Gao et al. [29], Tammelin et al. [61]) and that *alternating* update schemes are helpful (Kroer [39], Tammelin et al. [61]). First, we show that CBA and CBA$^+$ are both compatible with varying weights $(\omega_t)_{t\geq 1}$ when those weights are used on both decisions and payoffs. Second, we show that CBA$^+$ is compatible with weights $(\omega_t)_{t\geq 1}$ on payoffs and weights $(\theta_t)_{t\geq 1}$ on decisions, possibly with $\omega_t \neq \theta_t$.

Figure 1. Illustration of $\pi_{\mathcal{C}}(\cdot)$ for $\mathcal{C} = \mathbb{R}^2_+$ (left panel) and $\mathcal{C}$ any cone in $\mathbb{R}^2$ (right panel).


$\mathscr{C} = \mathbb{R}^2_+$
$\boldsymbol{u}_{t+1} = \pi_{\mathscr{C}}(\boldsymbol{u}_t^+)$
$\boldsymbol{0}$
d
$\boldsymbol{u}_t$
$\mathscr{C}^\circ = \mathbb{R}^2_-$
d
$\boldsymbol{u}_t^+ = \boldsymbol{u}_t + \omega_{t+1}\boldsymbol{v}_{t+1}$
$\mathscr{C}$
$\boldsymbol{u}_{t+1} = \pi_{\mathscr{C}}(\boldsymbol{u}_t^+)$
$\boldsymbol{u}_t$
$\mathscr{C}^\circ$
$\boldsymbol{0}$
d
d
$\boldsymbol{u}_t^+ = \boldsymbol{u}_t + \omega_{t+1}\boldsymbol{v}_{t+1}$

We start with the following theorem, which shows that CBA with weights on both decisions and payoffs is a no-regret algorithm. This generalizes the result of Abernethy et al. [1], which shows that CBA works for uniform weights.

Theorem 2. *Let $(\boldsymbol{x}_t)_{t\geq 1}$ be the sequence of decisions generated by* CBA *with weights $(\omega_t)_{t\geq 1}$ on the instantaneous payoff vectors, and let $S_t = \sum_{\tau=1}^{t} \omega_\tau$ for any $t \geq 1$. Then,*

$$\sum_{t=1}^{T} \omega_t \langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \min_{\boldsymbol{x}\in\mathcal{X}} \sum_{t=1}^{T} \omega_t \langle \boldsymbol{f}_t, \boldsymbol{x} \rangle \leq \sqrt{2}\kappa \cdot d(\boldsymbol{u}_T, \mathcal{C}^{\circ}).$$

Additionally,

$$d(\boldsymbol{u}_T, \mathcal{C}^{\circ}) \leq \sqrt{2}L \cdot \sqrt{\sum_{t=1}^{T} \omega_t^2}.$$

Overall, the average regret is such that

$$\frac{\sum_{t=1}^{T} \omega_t \langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \min_{\boldsymbol{x}\in\mathcal{X}} \sum_{t=1}^{T} \omega_t \langle \boldsymbol{f}_t, \boldsymbol{x} \rangle}{S_T} \leq 2\kappa L \frac{\sqrt{\sum_{t=1}^{T} \omega_t^2}}{\sum_{t=1}^{T} \omega_t}.$$

The proof of Theorem 2 uses the following facts from conic optimization. We provide proofs of all statements in Lemma 1 in Appendix A.

Lemma 1. *Let $\mathcal{C} \subset \mathbb{R}^{n+1}$ be a closed convex cone and $\mathcal{C}^{\circ}$ its polar.*

1. *(Moreau's decomposition (Combettes and Reyes [22])) If $\boldsymbol{u} \in \mathbb{R}^{n+1}$, then $\boldsymbol{u} - \pi_{\mathcal{C}^{\circ}}(\boldsymbol{u}) = \pi_{\mathcal{C}}(\boldsymbol{u}), \langle \boldsymbol{u} - \pi_{\mathcal{C}^{\circ}}(\boldsymbol{u}), \pi_{\mathcal{C}^{\circ}}(\boldsymbol{u}) \rangle = 0$, and $\|\boldsymbol{u} - \pi_{\mathcal{C}^{\circ}}(\boldsymbol{u})\|_2 \leq \|\boldsymbol{u}\|_2$.*
2. *(Abernethy et al. [1, lemma 13]) If $\boldsymbol{u} \in \mathbb{R}^{n+1}$, then $d(\boldsymbol{u}, \mathcal{C}) = \max_{\boldsymbol{w}\in\mathcal{C}^{\circ}\cap B_2(1)} \langle \boldsymbol{u}, \boldsymbol{w} \rangle$, where $B_2(1) = \{\boldsymbol{w} \in \mathbb{R}^{n+1} \,|\, \|\boldsymbol{w}\|_2 \leq 1\}$.*
3. *If $\boldsymbol{u} \in \mathcal{C}$, then $d(\boldsymbol{u}, \mathcal{C}^{\circ}) = \|\boldsymbol{u}\|_2$.*
4. *Assume that $\mathcal{C} = \text{cone}(\{\kappa\} \times \mathcal{X})$ with $\mathcal{X} \subset \mathbb{R}^n$ convex compact and $\kappa = \max_{\boldsymbol{x}\in\mathcal{X}} \|\boldsymbol{x}\|_2$. Then, $\mathcal{C}^{\circ}$ is a closed convex cone. Additionally, if $\boldsymbol{u} \in \mathcal{C}$, we have $-\boldsymbol{u} \in \mathcal{C}^{\circ}$.*
5. *Let us write $\leq_{\mathcal{C}^{\circ}}$ for the ordering induced by $\mathcal{C}^{\circ} : \boldsymbol{x} \leq_{\mathcal{C}^{\circ}} \boldsymbol{y} \iff \boldsymbol{y} - \boldsymbol{x} \in \mathcal{C}^{\circ}$. Then,*

$$\boldsymbol{x} \leq_{\mathcal{C}^{\circ}} \boldsymbol{y}, \boldsymbol{x}' \leq_{\mathcal{C}^{\circ}} \boldsymbol{y}' \Rightarrow \boldsymbol{x} + \boldsymbol{x}' \leq_{\mathcal{C}^{\circ}} \boldsymbol{y} + \boldsymbol{y}', \qquad \forall\, \boldsymbol{x}, \boldsymbol{x}', \boldsymbol{y}, \boldsymbol{y}' \in \mathbb{R}^{n+1}, \tag{5}$$

$$\boldsymbol{x} + \boldsymbol{x}' \leq_{\mathcal{C}^{\circ}} \boldsymbol{y} \Rightarrow \boldsymbol{x} \leq_{\mathcal{C}^{\circ}} \boldsymbol{y}, \qquad \forall\, \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{n+1},\ \forall\, \boldsymbol{x}' \in \mathcal{C}^{\circ}. \tag{6}$$

6. *Assume that $\boldsymbol{x} \leq_{\mathcal{C}^{\circ}} \boldsymbol{y}$ for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{n+1}$. Then, $d(\boldsymbol{y}, \mathcal{C}^{\circ}) \leq \|\boldsymbol{x}\|_2$.*

We are now ready to prove Theorem 2.

Proof of Theorem 2. The proof proceeds in two steps. We first have

$$d(\boldsymbol{u}_T, \mathcal{C}^{\circ}) = \max_{\boldsymbol{w}\in\text{cone}(\{\kappa\}\times\mathcal{X})\cap B_2(1)} \left\langle \sum_{t=1}^{T} \omega_t \boldsymbol{v}_t, \boldsymbol{w} \right\rangle \tag{7}$$

$$\geq \max_{\boldsymbol{x}\in\mathcal{X}} \left\langle \sum_{t=1}^{T} \omega_t \boldsymbol{v}_t, \frac{(\kappa, \boldsymbol{x})}{\|(\kappa, \boldsymbol{x})\|_2} \right\rangle \tag{8}$$

$$= \max_{\boldsymbol{x}\in\mathcal{X}} \frac{\sum_{t=1}^{T} \omega_t \langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^{T} \omega_t \langle \boldsymbol{f}_t, \boldsymbol{x} \rangle}{\|(\kappa, \boldsymbol{x})\|_2}, \tag{9}$$

where (7) follows from statement 2 in Lemma 1. For (8), we note that for $\boldsymbol{w}$ attaining the arg max in the right-hand side of (7), we must have $\|\boldsymbol{w}\|_2 = 1$ or $\boldsymbol{w} = \boldsymbol{0}$; we obtain (8) by dropping the second case. Equality (9) follows from CBA maintaining $\boldsymbol{u}_t = \left(\sum_{\tau=1}^{t} \omega_\tau \frac{\langle \boldsymbol{f}_\tau, \boldsymbol{x}_\tau \rangle}{\kappa}, -\sum_{\tau=1}^{t} \omega_\tau \boldsymbol{f}_\tau\right)$, $\forall t \geq 1$. Because $\|(\kappa, \boldsymbol{x})\|_2 \leq \sqrt{2}\kappa$, we conclude that

$$\sqrt{2}\kappa \cdot d(\boldsymbol{u}_T, \mathcal{C}^{\circ}) \geq \sum_{t=1}^{T} \omega_t \langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \min_{\boldsymbol{x}\in\mathcal{X}} \sum_{t=1}^{T} \omega_t \langle \boldsymbol{f}_t, \boldsymbol{x} \rangle.$$

We now prove that $d(\boldsymbol{u}_T, \mathcal{C}^\circ) \leq \sqrt{2} L \sqrt{\sum_{\tau=1}^{T} \omega_\tau^2}$. We have

$$d(\boldsymbol{u}_{t+1}, \mathcal{C}^\circ)^2 = \min_{z \in \mathcal{C}^\circ} \|\boldsymbol{u}_{t+1} - \boldsymbol{z}\|_2^2 \leq \|\boldsymbol{u}_{t+1} - \pi_{\mathcal{C}^\circ}(\boldsymbol{u}_t)\|_2^2 \leq \|\boldsymbol{u}_t + \omega_{t+1}\boldsymbol{v}_{t+1} - \pi_{\mathcal{C}^\circ}(\boldsymbol{u}_t)\|_2^2.$$

This shows that

$$\begin{aligned} d(\boldsymbol{u}_{t+1}, \mathcal{C}^\circ)^2 &\leq \|\boldsymbol{u}_t - \pi_{\mathcal{C}^\circ}(\boldsymbol{u}_t)\|_2^2 + \omega_{t+1}^2 \|\boldsymbol{v}_{t+1}\|_2^2 + 2\omega_{t+1}\langle \boldsymbol{u}_t - \pi_{\mathcal{C}^\circ}(\boldsymbol{u}_t), \boldsymbol{v}_{t+1}\rangle \\ &\leq \|\boldsymbol{u}_t - \pi_{\mathcal{C}^\circ}(\boldsymbol{u}_t)\|_2^2 + \omega_{t+1}^2 \|\boldsymbol{v}_{t+1}\|_2^2, \end{aligned} \tag{10}$$

where (10) follows from

$$\langle \boldsymbol{u}_t - \pi_{\mathcal{C}^\circ}(\boldsymbol{u}_t), \boldsymbol{v}_{t+1}\rangle = 0. \tag{11}$$

This is one of the crucial components of Blackwell's approachability framework; the current decision is chosen to force the next instantaneous payoff vector to lie in the hyperplane generated by projecting the aggregate payoff onto the target set. To see this, first note that $\boldsymbol{u}_t - \pi_{\mathcal{C}^\circ}(\boldsymbol{u}_t) = \pi_{\mathcal{C}}(\boldsymbol{u}_t)$. Let us write $\boldsymbol{\pi} = (\tilde{\pi}, \hat{\boldsymbol{\pi}}) = \pi_{\mathcal{C}}(\boldsymbol{u}_t)$. Note that by definition, $\boldsymbol{x}_{t+1} = (\kappa/\tilde{\pi})\hat{\boldsymbol{\pi}}$, and $\boldsymbol{v}_{t+1} = (\langle \boldsymbol{f}_{t+1}, \boldsymbol{x}_{t+1}\rangle/\kappa, -\boldsymbol{f}_{t+1})$. Therefore,

$$\begin{aligned} \langle \boldsymbol{u}_t - \pi_{\mathcal{C}^\circ}(\boldsymbol{u}_t), \boldsymbol{v}_{t+1}\rangle &= \langle \boldsymbol{\pi}, \boldsymbol{v}_{t+1}\rangle \\ &= \langle (\tilde{\pi}, \hat{\boldsymbol{\pi}}), (\langle \boldsymbol{f}_{t+1}, \boldsymbol{x}_{t+1}\rangle/\kappa, -\boldsymbol{f}_{t+1})\rangle \\ &= \langle (\tilde{\pi}, \hat{\boldsymbol{\pi}}), (\langle \boldsymbol{f}_{t+1}, (\kappa/\tilde{\pi})\hat{\boldsymbol{\pi}}\rangle/\kappa, -\boldsymbol{f}_{t+1})\rangle \\ &= \langle \hat{\boldsymbol{\pi}}, \boldsymbol{f}_{t+1}\rangle - \langle \hat{\boldsymbol{\pi}}, \boldsymbol{f}_{t+1}\rangle \\ &= 0. \end{aligned}$$

Next, recall that $d(\boldsymbol{u}_t, \mathcal{C}^\circ)^2 = \|\boldsymbol{u}_t - \pi_{\mathcal{C}^\circ}(\boldsymbol{u}_t)\|_2^2$. Therefore, we have shown that

$$d(\boldsymbol{u}_{t+1}, \mathcal{C}^\circ)^2 \leq d(\boldsymbol{u}_t, \mathcal{C}^\circ)^2 + \omega_{t+1}^2 \|\boldsymbol{v}_{t+1}\|_2^2.$$

Applying the previous inequality inductively and telescoping, we obtain

$$d(\boldsymbol{u}_t, \mathcal{C}^\circ)^2 \leq \sum_{\tau=1}^{t} \omega_\tau^2 \|\boldsymbol{v}_\tau\|_2^2 \leq 2L^2 \cdot \sum_{\tau=1}^{t} \omega_\tau^2,$$

where the last inequality follows from the definition of $\boldsymbol{v}_t$ and L. □

In the next theorem, we show a result that may seem surprising; CBA$^+$ allows us to use two separate and different weighting schemes for the decisions in the regret definition and the aggregate payoffs. This result is important because as we show in Appendix I, using linear averaging for the decisions and uniform weights for the instantaneous payoffs results in dramatically faster empirical convergence for CBA$^+$. This is analogous to RM$^+$ in the simplex case; using linear averaging on decisions but constant weights on the instantaneous payoffs is vastly superior numerically (Brown and Sandholm [15], Tammelin et al. [61]). Both for CBA$^+$ and RM$^+$, this requires combination with alternation, which we study in Section 3.4.

Theorem 3. *Let $(\boldsymbol{x}_t)_{t\geq 1}$ be the sequence of decisions generated by* CBA$^+$ *with weights $(\omega_t)_{t\geq 1}$ on the instantaneous payoff vectors. Let $(\theta_t)_{t\geq 1}$ be the weights on the decisions and $S_T = \sum_{t=1}^{T} \theta_t$. Assume that $\frac{\theta_{t+1}}{\theta_t} \geq \frac{\omega_{t+1}}{\omega_t}$, $\forall t \geq 1$. Then,*

$$\frac{\sum_{t=1}^{T} \theta_t \langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \min_{\boldsymbol{x}\in\mathcal{X}} \sum_{t=1}^{T} \theta_t \langle \boldsymbol{f}_t, \boldsymbol{x}\rangle}{S_T} \leq 2\kappa L \frac{\theta_T}{\omega_T} \frac{\sqrt{\sum_{t=1}^{T} \omega_t^2}}{\sum_{t=1}^{T} \theta_t}.$$

Our proof heavily relies on the sequence of aggregate payoffs belonging to the cone $\mathcal{C}$ at every iteration ($\boldsymbol{u}_t \in \mathcal{C}$, $\forall t \geq 1$), and for this reason, it does not extend to CBA. We also note that the use of conic optimization somewhat simplifies the argument compared with the proof that RM$^+$ is compatible with polynomial averaging on decisions and uniform weights on payoffs.

Proof of Theorem 3. Recall that $\boldsymbol{v}_t = (\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle/\kappa, -\boldsymbol{f}_t)$. By construction and following the same argument as for the proof of Theorem 2, we have

$$\sum_{t=1}^{T} \theta_t \langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \min_{\boldsymbol{x}\in\mathcal{X}} \sum_{t=1}^{T} \theta_t \langle \boldsymbol{f}_t, \boldsymbol{x}\rangle \leq \sqrt{2}\kappa \cdot d\left(\sum_{t=1}^{T} \theta_t \boldsymbol{v}_t, \mathcal{C}^\circ\right). \tag{12}$$

Additionally, we always have

$$\omega_{t+1}\boldsymbol{v}_{t+1} \geq_{\mathcal{C}^\circ} \boldsymbol{u}_{t+1} - \boldsymbol{u}_t. \tag{13}$$

This is because

$$\begin{aligned}\omega_{t+1}\boldsymbol{v}_{t+1} - \boldsymbol{u}_{t+1} + \boldsymbol{u}_t &= \boldsymbol{u}_t + \omega_{t+1}\boldsymbol{v}_{t+1} - \boldsymbol{u}_{t+1}\\ &= \boldsymbol{u}_t + \omega_{t+1}\boldsymbol{v}_{t+1} - \pi_{\mathcal{C}}(\boldsymbol{u}_t + \omega_{t+1}\boldsymbol{v}_{t+1})\\ &= \pi_{\mathcal{C}^\circ}(\boldsymbol{u}_t + \omega_{t+1}\boldsymbol{v}_{t+1}) \in \mathcal{C}^\circ,\end{aligned}$$

where the second equality follows from the definition of $\boldsymbol{u}_{t+1}$ and the third equality follows from Moreau's decomposition. Therefore, multiplying (13) by θ_{t+1} and dividing by ω_{t+1}, we obtain

$$\theta_{t+1}\boldsymbol{v}_{t+1} \geq_{\mathcal{C}^\circ} \frac{\theta_{t+1}}{\omega_{t+1}}(\boldsymbol{u}_{t+1} - \boldsymbol{u}_t).$$

Reformulating the right-hand side, we obtain

$$\theta_{t+1}\boldsymbol{v}_{t+1} \geq_{\mathcal{C}^\circ} \frac{\theta_{t+1}}{\omega_{t+1}}\boldsymbol{u}_{t+1} - \frac{\theta_t}{\omega_t}\boldsymbol{u}_t - \left(\frac{\theta_{t+1}}{\omega_{t+1}} - \frac{\theta_t}{\omega_t}\right)\boldsymbol{u}_t.$$

Note that $\left(\frac{\theta_{t+1}}{\omega_{t+1}} - \frac{\theta_t}{\omega_t}\right)\boldsymbol{u}_t \in \mathcal{C}$ because $\frac{\theta_{t+1}}{\omega_{t+1}} - \frac{\theta_t}{\omega_t} \geq 0$ and $\boldsymbol{u}_t \in \mathcal{C}$. Statement 4 in Lemma 1 shows that if $\boldsymbol{u} \in \mathcal{C}$, then $-\boldsymbol{u} \in \mathcal{C}^\circ$. Therefore, $-\left(\frac{\theta_{t+1}}{\omega_{t+1}} - \frac{\theta_t}{\omega_t}\right)\boldsymbol{u}_t \in \mathcal{C}^\circ$. Now, by applying (6) in statement 5 of Lemma 1, we have that

$$\theta_{t+1}\boldsymbol{v}_{t+1} \geq_{\mathcal{C}^\circ} \frac{\theta_{t+1}}{\omega_{t+1}}\boldsymbol{u}_{t+1} - \frac{\theta_t}{\omega_t}\boldsymbol{u}_t - \left(\frac{\theta_{t+1}}{\omega_{t+1}} - \frac{\theta_t}{\omega_t}\right)\boldsymbol{u}_t \Rightarrow \theta_{t+1}\boldsymbol{v}_{t+1} \geq_{\mathcal{C}^\circ} \frac{\theta_{t+1}}{\omega_{t+1}}\boldsymbol{u}_{t+1} - \frac{\theta_t}{\omega_t}\boldsymbol{u}_t.$$

Summing up the previous inequalities from $t = 1$ to $t = T - 1$ and using $\boldsymbol{u}_1 = \boldsymbol{v}_1$, we obtain $\sum_{t=1}^{T} \theta_t\boldsymbol{v}_t \geq_{\mathcal{C}^\circ} \frac{\theta_T}{\omega_T}\boldsymbol{u}_T$. Now, statement 6 shows that

$$d\left(\sum_{t=1}^{T}\theta_t\boldsymbol{v}_t, \mathcal{C}^\circ\right) \leq \left\|\frac{\theta_T}{\omega_T}\boldsymbol{u}_T\right\|_2. \tag{14}$$

By construction, $\boldsymbol{u}_T$ is the sequence of aggregate payoffs generated by CBA$^+$ with weights $(\omega_t)_{t\geq 1}$. We now show that

$$d(\boldsymbol{u}_T, \mathcal{C}^\circ) = \|\boldsymbol{u}_T\|_2 \leq \sqrt{2}L\sqrt{\sum_{t=1}^{T}\omega_t^2}.$$

We have, from statement 1 in Lemma 1,

$$\|\boldsymbol{u}_{t+1}\|_2^2 = \|\pi_{\mathcal{C}}(\boldsymbol{u}_t + \omega_{t+1}\boldsymbol{v}_{t+1})\|_2^2 \leq \|\boldsymbol{u}_t + \omega_{t+1}\boldsymbol{v}_{t+1}\|_2^2.$$

Therefore,

$$\|\boldsymbol{u}_{t+1}\|_2^2 \leq \|\boldsymbol{u}_t\|_2^2 + \omega_{t+1}^2\|\boldsymbol{v}_{t+1}\|_2^2 + 2\omega_{t+1}\langle\boldsymbol{u}_t, \boldsymbol{v}_{t+1}\rangle.$$

By construction and for the same reason as for (11), $\langle\boldsymbol{u}_t, \boldsymbol{v}_{t+1}\rangle = 0$. Therefore, we have the inequality

$$\|\boldsymbol{u}_{t+1}\|_2^2 \leq \|\boldsymbol{u}_t\|_2^2 + \omega_{t+1}^2\|\boldsymbol{v}_{t+1}\|_2^2.$$

By telescoping this inequality, we obtain $\|\boldsymbol{u}_t\|_2^2 \leq \sum_{\tau=1}^{t}\omega_\tau^2\|\boldsymbol{v}_\tau\|_2^2$. By definition of L, we conclude that

$$\|\boldsymbol{u}_T\|_2 \leq \sqrt{2}L\sqrt{\sum_{t=1}^{T}\omega_t^2}. \tag{15}$$

From (14), $d(\sum_{t=1}^{T}\theta_t\boldsymbol{v}_t, \mathcal{C}^\circ) \leq \sqrt{2}L\frac{\theta_T}{\omega_T}\sqrt{\sum_{t=1}^{T}\omega_t^2}$, which together with (12), concludes the proof of Theorem 3. □

Remark 1. We could have defined more general versions of CBA and CBA$^+$, parametrized by a positive scalar $\lambda > 0$, by defining the cone $\mathcal{C} = \text{cone}(\{\lambda\} \times \mathcal{X}) \subset \mathbb{R}^{n+1}$ and the instantaneous payoffs as $\boldsymbol{v}_t = \left(\frac{\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle}{\lambda}, -\boldsymbol{f}_t\right)$. The algorithms introduced in this section correspond to the choice $\lambda = \kappa$ with $\kappa = \max_{\boldsymbol{x}\in\mathcal{X}}\|\boldsymbol{x}\|_2$. In Appendix B, we provide the regret guarantees for these more general CBA and CBA$^+$ algorithms based on the value of $\lambda > 0$. We then show that choosing $\lambda = \kappa$ minimizes our regret bounds, simplifies the exposition, and performs well empirically.

So far, we have studied the classical notion of regret where we compare the performance of the regret minimizer against the best *stationary* decision in hindsight. We now show that CBA$^+$ provides a stronger *tracking regret* guarantee. In tracking regret, a constant K is given, and we measure the performance against the best sequence from the set of all sequences $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_T \in \mathcal{X}$ that change at most $K-1$ times (Herbster and Warmuth [36]). The standard notion of regret corresponds to $K=1$. Analogously to the case of RM$^+$ and RM, we have that CBA$^+$ provides a tracking regret guarantee, whereas CBA does not. We provide the detailed proof in Appendix C.

Theorem 4. *Let $(\boldsymbol{x}_t)_{t\geq 1}$ be the sequence of decision generated by* CBA$^+$ *with weights $(\omega_t)_{t\geq 1}$ on the instantaneous payoff vectors. Let $(\theta_t)_{t\geq 1}$ be the weights on the decisions and $S_T = \sum_{t=1}^T \theta_t$. Assume that $\frac{\theta_{t+1}}{\theta_t} \geq \frac{\omega_{t+1}}{\omega_t}$, $\forall t \geq 1$. For any $K \in \mathbb{N}$, let $\sigma_K \subset \mathcal{X}^T$ be the set of sequences of T elements of $\mathcal{X}$ that change at most $K-1$ times:*

$$\sigma_K = \{(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_T) \in \mathcal{X}^T \,|\, Card(\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_T\}) \leq K\}.$$

Then, we have

$$\frac{\sum_{t=1}^T \theta_t \langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \min_{\boldsymbol{z}\in\sigma_K} \sum_{t=1}^T \theta_t \langle \boldsymbol{f}_t, \boldsymbol{z}_t\rangle}{S_T} \leq 2\kappa L K \frac{\theta_T}{\omega_T} \frac{\sqrt{\sum_{t=1}^T \omega_t^2}}{\sum_{t=1}^T \theta_t}.$$

3.3. Convergence Bounds for Saddle-Point Problems

In this section, we show how the regret bounds from the previous section translate into convergence rates for solving convex-concave saddle-point problems in the repeated game framework. In particular, the following theorem gives the convergence rate of CBA$^+$ and CBA for solving saddle-point problems of the form (1) based on our bounds on the regret of each player under various weighting schemes. The proof is in Appendix D.

Theorem 5. *Let $L = \max\{L_x, L_y\}$ defined in (4) and $\kappa = \max\{\max\{\|\boldsymbol{x}\|_2, \|\boldsymbol{y}\|_2\} \,|\, \boldsymbol{x} \in \mathcal{X}, \boldsymbol{y} \in \mathcal{Y}\}$.*

1. *Let $p \in \mathbb{N}$. Let $(\bar{\boldsymbol{x}}_T, \bar{\boldsymbol{y}}_T) = \sum_{t=1}^T \omega_t(\boldsymbol{x}_t, \boldsymbol{y}_t)/S_T$, where $(\boldsymbol{x}_t)_{t\geq 1}, (\boldsymbol{y}_t)_{t\geq 1}$ are generated by the repeated game framework with* CBA *with weights $(\omega_\tau)_{t\geq 1}$ on both decisions and payoffs and $S_T = \sum_{t=1}^T \omega_t$. Assume that $\omega_t = t^p$, $\forall t \geq 1$. Then,*

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

2. *Let $p, q \in \mathbb{N}$ with $q \geq p$. Let $(\bar{\boldsymbol{x}}_T, \bar{\boldsymbol{y}}_T) = \sum_{t=1}^T \theta_t(\boldsymbol{x}_t, \boldsymbol{y}_t)/S_T$, where $(\boldsymbol{x}_t)_{t\geq 1}, (\boldsymbol{y}_t)_{t\geq 1}$ are generated by the repeated game framework with* CBA$^+$ *with payoff weights $(\omega_t)_{t\geq 1}$, decision weights $(\theta_t)_{t\geq 1}$, and $S_T = \sum_{t=1}^T \theta_t$. Assume that $\theta_t = t^q, \omega_t = t^p$, $\forall t \geq 1$. Then,*

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

Let us compare our bounds with the regret bounds of classical first-order methods (FOMs). We consider $p, q = 0$. CBA and CBA$^+$ achieve $O(\kappa L/\sqrt{T})$ average regret, whereas OMD (Nemirovski and Yudin [49]) and FTRL (Abernethy et al. [2]) achieve $O(\Omega L/\sqrt{T})$ average regret, where $\Omega = \max\{\|\boldsymbol{x} - \boldsymbol{x}'\|_2 \,|\, \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}\}$. We can always recenter $\mathcal{X}$ to contain $\boldsymbol{0}$, in which case the bounds for OMD/FTRL and CBA$^+$ are equivalent because $\kappa \leq \Omega \leq 2\kappa$. The bound on the average regret for *optimistic* online mirror descent (OOMD) (Chiang et al. [20]) and *optimistic* follow the regularized leader (OFTRL) (Rakhlin and Sridharan [57]) is $O(\Omega^2 L/T)$ in the repeated game framework, a priori better than the bound for CBA$^+$ as regards the number of iterations T. Nonetheless, we will see in Section 5 that the empirical performance of CBA$^+$ is better than that of $O(1/T)$ methods. A similar situation occurs for RM$^+$ compared with OOMD and OFTRL for solving extensive-form games, such as poker (Farina et al. [27]).

3.4. Improved Convergence Bounds Using Alternation

Alternation is a simple variation of the repeated game framework from Section 2. Alternation is known to lead to significant empirical speedups for RM$^+$ and CFR$^+$ (Tammelin et al. [61]), and we observe in our simulations (Appendix I) that this holds for CBA$^+$ as well. In the repeated game framework with alternation, at iteration t, the second player is provided with the decision $\boldsymbol{x}_t$ of the first player for iteration t. Because alternation is defined the same way for both CBA and CBA$^+$, we omit the subscripts in CHOOSEDECISION and UPDATEPAYOFF. In particular, at iteration t of the repeated game framework with alternation, the players choose $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ as follows.

1. Both players start with aggregate payoffs $\boldsymbol{u}^x_{t-1}, \boldsymbol{u}^y_{t-1}$.
2. The first player chooses a decision $\boldsymbol{x}_t$ based on $\boldsymbol{u}^x_{t-1}$: $\boldsymbol{x}_t = \text{CHOOSEDECISION}(\boldsymbol{u}^x_{t-1})$.
3. For $\boldsymbol{g}_{t-1} = \partial_y F(\boldsymbol{x}_t, \boldsymbol{y}_{t-1})$, the second player updates its aggregate payoff:
$$\boldsymbol{u}^y_t = \text{UPDATEPAYOFF}(\boldsymbol{u}^y_{t-1}, \boldsymbol{y}_{t-1}, \boldsymbol{g}_{t-1}, \omega_t).$$
4. The second player chooses a decision $\boldsymbol{y}_t$ based on $\boldsymbol{u}^y_t$: $\boldsymbol{y}_t = \text{CHOOSEDECISION}(\boldsymbol{u}^y_t)$.
5. For $\boldsymbol{f}_t = \partial_x F(\boldsymbol{x}_t, \boldsymbol{y}_t)$, the first player updates its aggregate payoff:
$$\boldsymbol{u}^x_t = \text{UPDATEPAYOFF}(\boldsymbol{u}^x_{t-1}, \boldsymbol{x}_t, \boldsymbol{f}_t, \omega_t).$$

Recall that we use the repeated game framework to solve (1) because we can bound the duality gap by the sum of the average regrets of each player using Theorem 1. It is known that in the repeated game framework with alternation, it is possible to construct decisions such that Theorem 1 fails to hold because of the mismatch in the sequences of decisions of the players (Farina et al. [26]). That said, it was later shown that a modified version of Theorem 1 holds (Burch et al. [17]). Here, we state a more general version of that result, which was first shown in a set of lecture notes (Kroer [39]). In particular, the following bound holds on the duality gap. For the sake of completeness, we provide the proof in Appendix E.

Theorem 6. *Consider some weights* $(\theta_t)_{t\geq 1}$ *and* $S_T = \sum_{t=1}^T \theta_{t+1}$. *Let* $(\bar{\boldsymbol{x}}_T, \bar{\boldsymbol{y}}_T) = \sum_{t=1}^T \theta_{t+1}(\boldsymbol{x}_{t+1}, \boldsymbol{y}_t)/S_T$, *where* $(\boldsymbol{x}_t)_{t\geq 1}$, $(\boldsymbol{y}_t)_{t\geq 1}$ *are generated by the repeated game framework with alternation. Then,*

$$\begin{aligned}\max_{\boldsymbol{y}\in\mathcal{Y}} F(\bar{\boldsymbol{x}}_T, \boldsymbol{y}) - \min_{\boldsymbol{x}\in\mathcal{X}} F(\boldsymbol{x}, \bar{\boldsymbol{y}}_T) \leq \frac{1}{S_T}&\left(\max_{\boldsymbol{y}\in\mathcal{Y}} \sum_{t=1}^T \theta_{t+1}\langle \boldsymbol{g}_t, \boldsymbol{y}\rangle - \sum_{t=1}^T \theta_{t+1}\langle \boldsymbol{g}_t, \boldsymbol{y}_t\rangle\right)\\ &+\frac{1}{S_T}\left(\sum_{t=1}^T \theta_{t+1}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \min_{\boldsymbol{x}\in\mathcal{X}} \sum_{t=1}^T \theta_{t+1}\langle \boldsymbol{f}_t, \boldsymbol{x}\rangle\right)\\ &+\frac{1}{S_T}\left(\sum_{t=1}^T \theta_{t+1}(F(\boldsymbol{x}_{t+1}, \boldsymbol{y}_t) - F(\boldsymbol{x}_t, \boldsymbol{y}_t))\right).\end{aligned}$$

From Theorem 6, we see that alternation guarantees convergence to a solution of (1) if

$$\sum_{t=1}^T \theta_{t+1}(F(\boldsymbol{x}_{t+1}, \boldsymbol{y}_t) - F(\boldsymbol{x}_t, \boldsymbol{y}_t)) \leq 0. \tag{16}$$

In the framework of RM and RM$^+$, we have $\mathcal{X} = \Delta(n), \mathcal{Y} = \Delta(m)$, and the objective function is bilinear. In this case, it is shown in Burch et al. [17] that (16) holds. In particular, for any $t \in [T]$, it holds that $F(\boldsymbol{x}_{t+1}, \boldsymbol{y}_t) - F(\boldsymbol{x}_t, \boldsymbol{y}_t) \leq 0$. We provide the following stronger result for CBA$^+$ in the case of an objective function F that is linear in one of the two variables, with any convex compact decision sets $\mathcal{X}$ and $\mathcal{Y}$. The proof is presented in Appendix F.

Theorem 7. *Assume that* $(\boldsymbol{x}, \boldsymbol{y}) \longmapsto F(\boldsymbol{x}, \boldsymbol{y})$ *is linear in* $\boldsymbol{x}$.

1. *In the framework of Theorem 6, suppose that* $(\boldsymbol{x}_t)_{t\geq 1}, (\boldsymbol{y}_t)_{t\geq 1}$ *are generated by* CBA$^+$ *with weights* $(\omega_t)_{t\geq 1}$ *on the payoffs. Let* $t \geq 1$. *If* $\boldsymbol{u}^x_t = \boldsymbol{0}$, *then* $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t$ *and* $F(\boldsymbol{x}_{t+1}, \boldsymbol{y}_t) - F(\boldsymbol{x}_t, \boldsymbol{y}_t) = 0$. *Otherwise,*
$$F(\boldsymbol{x}_{t+1}, \boldsymbol{y}_t) - F(\boldsymbol{x}_t, \boldsymbol{y}_t) \leq -\frac{\kappa}{\omega_t \cdot \|\boldsymbol{u}^x_t\|_\infty}\|\boldsymbol{u}^x_t - \boldsymbol{u}^x_{t-1}\|_2^2.$$
2. *In the framework of Theorem 6, suppose that* $(\boldsymbol{x}_t)_{t\geq 1}, (\boldsymbol{y}_t)_{t\geq 1}$ *are generated by* CBA *with weights* $(\omega_t)_{t\geq 1}$ *on the payoffs. Let* $t \geq 1$. *If* $\pi_{\mathcal{C}}(\boldsymbol{u}^x_t) = \boldsymbol{0}$, *then* $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t$ *and* $F(\boldsymbol{x}_{t+1}, \boldsymbol{y}_t) - F(\boldsymbol{x}_t, \boldsymbol{y}_t) = 0$. *Otherwise,*
$$F(\boldsymbol{x}_{t+1}, \boldsymbol{y}_t) - F(\boldsymbol{x}_t, \boldsymbol{y}_t) \leq -\frac{\kappa}{\omega_t \cdot \|\pi_{\mathcal{C}}(\boldsymbol{u}^x_t)\|_\infty}\|\pi_{\mathcal{C}}(\boldsymbol{u}^x_t) - \pi_{\mathcal{C}}(\boldsymbol{u}^x_{t-1})\|_2^2.$$

Note that our results in Theorem 7 for CBA and CBA$^+$ improve upon the analogous results for RM and RM$^+$ (Burch et al. [17]) because Theorem 7 guarantees a strict improvement from alternation, where Burch et al. [17] only show that "alternation does not hurt" (i.e., Burch et al. [17] only show that (16) holds). Second, their result is for the case of a bilinear objective function, whereas we only require linearity in one of the variables. In fact, our results in Theorem 7 also extend to RM and RM$^+$, which provides the first explanation for the strong performances of RM and RM$^+$ combined with alternation. For the sake of conciseness, we present the details on RM and RM$^+$ in Appendix G. Finally, we would like to note that our assumption that the objective function is linear in one of the decision variable is satisfied for many important decision problems (e.g., Markov decision processes, distributionally robust logistic regression, and matrix games), as we will see in our simulations in Section 5.

4. Efficient Implementations of CBA

The main bottleneck of both CBA$^+$ and CBA is to efficiently compute $\pi_{\mathcal{C}}(\boldsymbol{u})$, the orthogonal projection of a vector $\boldsymbol{u}$ on the cone $\mathcal{C} = \text{cone}(\{\kappa\} \times \mathcal{X})$:

$$\pi_{\mathcal{C}}(\boldsymbol{u}) \in \arg\min_{\boldsymbol{y} \in \mathcal{C}} \|\boldsymbol{y} - \boldsymbol{u}\|_2^2. \tag{17}$$

Note that this issue is not discussed in Abernethy et al. [1], where the authors do not provide an efficient implementation of CBA. In this section, we show how to efficiently solve (17) for many important decision sets $\mathcal{X}$. One of the critical components of our proofs is *Moreau's decomposition theorem* (Combettes and Reyes [22]) (statement 1 in Lemma 1), which states that $\pi_{\mathcal{C}}(\boldsymbol{u})$ can be recovered from $\pi_{\mathcal{C}^\circ}(\boldsymbol{u})$ and vice versa because for any convex cone $\mathcal{C}$, we have $\pi_{\mathcal{C}}(\boldsymbol{u}) + \pi_{\mathcal{C}^\circ}(\boldsymbol{u}) = \boldsymbol{u}$. All the proofs for this section are presented in Appendix H.

4.1. Simplex

Assume that $\mathcal{X} = \Delta(n)$. This setting is standard for matrix games, where n is the number of actions of a player and $\boldsymbol{x} \in \Delta(n)$ represents a randomized decision. It is also used for extensive-form games because CFR decomposes regret minimization over the tree-like decision space into a set of local regret minimizations over simplexes (Zinkevich et al. [66]). When $\mathcal{X} = \Delta(n)$, we show that $\pi_{\mathcal{C}^\circ}(\boldsymbol{u})$ can be computed in $O(n\log(n))$ using a sorting trick similar to that for the standard simplex projection, and therefore, $\pi_{\mathcal{C}}(\boldsymbol{u})$ can be computed in $O(n\log(n))$ using Moreau's decomposition. In particular, we provide the following closed-form expression for the polar cone $\mathcal{C}^\circ$.

Lemma 2. *Let $\mathcal{C} = cone(\{1\} \times \Delta(n))$. Then, $\mathcal{C}^\circ = \{(\tilde{y}, \hat{\boldsymbol{y}}) \in \mathbb{R}^{n+1} \,|\, \max_{i\in[n]} \hat{y}_i \leq -\tilde{y}\}$.*

Therefore, $\pi_{\mathcal{C}^\circ}(\boldsymbol{u})$ is a solution to $\min\{(\tilde{y} - \tilde{u})^2 + \|\hat{\boldsymbol{y}} - \hat{\boldsymbol{u}}\|_2^2 \,|\, (\tilde{y}, \hat{\boldsymbol{y}}) \in \mathbb{R}^{n+1}, \max_{i\in[n]} \hat{y}_i \leq -\tilde{y}\}$.

Proposition 1. *Let $\mathcal{X} = \Delta(n)$. An optimal solution $\pi_{\mathcal{C}^\circ}(\boldsymbol{u})$ can be computed in $O(n\log(n))$ arithmetic operations. Therefore, $\pi_{\mathcal{C}}(\boldsymbol{u})$ can be computed in $O(n\log(n))$ arithmetic operations.*

4.2. ℓ_p Balls

For $p \geq 1$ and $p = \infty$, we consider the ℓ_p balls $\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^n \,|\, \|\boldsymbol{x}\|_p \leq 1\}$. This type of decision set appears in many problems in optimization, including robust optimization (Ben-Tal et al. [8]), distributionally robust logistic regression (Namkoong and Duchi [47]), ℓ_∞ regression (Sidford and Tian [58]), and saddle-point reformulation of Markov decision processes (Jin and Sidford [38]). We first reformulate the cones $\mathcal{C}$ and $\mathcal{C}^\circ$. Recall that $\kappa = \max\{\|\boldsymbol{x}\|_2 \,|\, \boldsymbol{x} \in \mathcal{X}\}$.

Lemma 3. *Let $\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^n \,|\, \|\boldsymbol{x}\|_p \leq 1\}$, with $p \geq 1$ or $p = \infty$. Let $q \in \mathbb{R} \cup \{+\infty\}$ be such that $1/p + 1/q = 1$. Then, $\mathcal{C} = \{(\tilde{y}, \boldsymbol{y}) \in \mathbb{R} \times \mathbb{R}^n \,|\, \|\boldsymbol{y}\|_p \leq \tilde{y}/\kappa\}, \mathcal{C}^\circ = \{(\tilde{y}, \boldsymbol{y}) \in \mathbb{R} \times \mathbb{R}^n \,|\, \|\boldsymbol{y}\|_q \leq -\kappa\tilde{y}\}$.*

Based on Lemma 3, we can prove the following propositions.

Proposition 2. *Let $\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^n \,|\, \|\boldsymbol{x}\|_p \leq 1\}$ for $p \in \{1, \infty\}$. Then, $\pi_{\mathcal{C}}(\boldsymbol{u})$ can be computed in $O(n\log(n))$ operations.*

Proposition 3. *Let $\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^n \,|\, \|\boldsymbol{x}\|_2 \leq 1\}$. Then, $\pi_{\mathcal{C}}(\boldsymbol{u})$ can be computed in $O(n)$ operations.*

4.3. Ellipsoidal Confidence Region in the Simplex

Here, $\mathcal{X}$ is an *ellipsoidal subregion of the simplex* defined as $\mathcal{X} = \{\boldsymbol{x} \in \Delta(n) \,|\, \|\boldsymbol{x} - \boldsymbol{x}_0\|_2 \leq \epsilon_x\}$. This type of decision set is widely used because it is associated with confidence regions when estimating a probability distribution from observed data (Bertsimas et al. [9], Iyengar [37]). It can also be used in the Bellman update for robust Markov decision processes (Goyal and Grand-Clément [30], Iyengar [37], Wiesemann et al. [65]). We assume that the confidence region is "entirely contained in the simplex": $\{\boldsymbol{x} \in \mathbb{R}^n \,|\, \boldsymbol{x}^\top \boldsymbol{e} = 1\} \cap \{\boldsymbol{x} \in \mathbb{R}^n \,|\, \|\boldsymbol{x} - \boldsymbol{x}_0\|_2 \leq \epsilon_x\} \subseteq \Delta(n)$ 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).

Proposition 4. *Let $\mathcal{X} = \{\boldsymbol{x} \in \Delta(n) \,|\, \|\boldsymbol{x} - \boldsymbol{x}_0\|_2 \leq \epsilon_x\}$, and assume that $\{\boldsymbol{x} \in \mathbb{R}^n \,|\, \boldsymbol{x}^\top \boldsymbol{e} = 1\} \cap \{\boldsymbol{x} \in \mathbb{R}^n \,|\, \|\boldsymbol{x} - \boldsymbol{x}_0\|_2 \leq \epsilon_x\} \subseteq \Delta(n)$. Then, $\pi_{\mathcal{C}}(\boldsymbol{u})$ can be computed in $O(n)$ arithmetic operations.*

5. Numerical Experiments

In this section, we compare the performances of SP-CBA$^+$ on real and synthetic instances of classical saddle-point problems from the operations research and optimization literature. We focus on bilinear matrix games, extensive-form games, distributionally robust logistic regression, and MDPs. Recall that we have defined SP-CBA$^+$ by combining the repeated game framework from Section 2 with CBA$^+$ as a regret minimizer, with uniform weights on the payoffs, linear weights on the decisions, and the alternating updates from Section 3.4. We justify in Appendix I that this leads to the strongest empirical performances for SP-CBA$^+$. We first examine the performance of SP-CBA$^+$ on matrix and extensive-form games, where RM$^+$ and CFR$^+$ are already known to perform extremely well empirically, and the goal of these experiments is to see whether SP-CBA$^+$ retains that very strong empirical performance. The experiments on distributionally robust logistic regression and MDPs then show the performance on new domains where no Blackwell-based algorithms were known prior to this paper. To illustrate the superior performance of CBA$^+$ compared with the vanilla CBA algorithm, in all instances we also run a vanilla version of SP-CBA, which corresponds to combining the repeated game framework from Section 2 with CBA as a regret minimizer.

5.1. Matrix Games

Matrix games have a bilinear objective function and simplexes as decision sets:

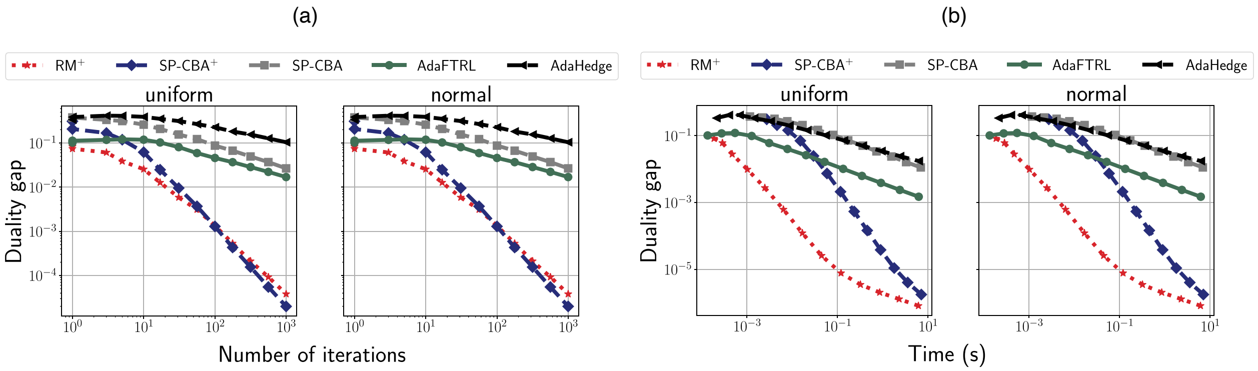
$$\min_{x\in\Delta(n)} \max_{y\in\Delta(m)} \langle x, Ay\rangle, \tag{18}$$

where $A\in\mathbb{R}^{n\times m}$ is the matrix of payoffs of the game. We can view (18) as a zero-sum game between the first player and the second player, where the coefficient $A_{ij}\in\mathbb{R}$ represents payoff obtained by the second player when the first player chooses action i and the second player chooses action j.

5.1.1. Experimental Setup. We generate 100 synthetic matrices A of size $\mathbb{R}^{n\times m}$ with $(n,m)=(100,50)$. Similarly, as in Chambolle and Pock [19] and Nesterov [50], for the coefficients of A we consider a uniform distribution in $[0,1]$ or a normal distribution of mean 0 and variance 1. We compare SP-CBA$^+$ with SP-CBA and with RM$^+$, which is known to achieve the best empirical performance compared with a wide range of algorithms, including Hedge and other first-order methods (Farina et al. [27], Kroer [39], Kroer et al. [40]). We also compare with two other scale-free and parameter-free no-regret algorithms, AdaHedge (De Rooij et al. [23]) and AdaFTRL (Orabona and Pál [52]), with the ℓ_2 norm as the Bregman divergence. Similarly as for SP-CBA$^+$, for RM$^+$ we use the repeated game framework with alternation, along with linear averaging on the decisions and uniform averaging on the payoffs. In Figure 2, we compare the performance of the five algorithms (SP-CBA$^+$, SP-CBA, RM$^+$, AdaHedge, and AdaFTRL) for solving (18). In Figure 2(a), we let the five algorithms run for $T=1{,}000$ iterations, and we show the duality gap of the current running average as a function of the number of iterations. This shows the progress made by the algorithms toward solving (18) at each iteration. In Figure 2(b), we run the five algorithms for time max $=10$ seconds, and we show the duality gap as a function of the time of computation. We average all the results over 50 randomly generated instances. Note that both axes are in logarithmic scale.

5.1.2. Results and Discussion. When we compare the duality gap as a function of the number of iterations (Figure 2(a)), we note that SP-CBA$^+$ performs on par with RM$^+$, and both algorithms vastly outperform SP-CBA,

Figure 2. (Color online) Comparison of SP-CBA$^+$, SP-CBA, RM$^+$, AdaHedge, and AdaFTRL on instances of matrix games. We compare the duality gap with respect to (a) the number of iterations or (b) the computation time. The coefficients of A are chosen randomly, with a uniform distribution or a normal distribution.



AdaHedge, and AdaFTRL. However, each iteration of SP-CBA$^+$ on the simplex requires solving $O(n\log(n))$ arithmetic operations (see Section 4.1), whereas each iteration of RM$^+$ can be performed in $O(n)$ operations. Therefore, when we compare the duality gap as a function of the computation time (Figure 2(b)), we note that RM$^+$ outperforms SP-CBA$^+$, even though after roughly 10 seconds of computation, the performances of SP-CBA$^+$ and RM$^+$ are equivalent.

5.2. Extensive-Form Games

EFGs (von Stengel [63]) are used to model sequential games with imperfect information. For example, they were used for superhuman poker AIs in games such as Texas hold'em (Brown and Sandholm [14], Brown and Sandholm [16], Tammelin et al. [61]). EFGs can be written as saddle-point problems, with a bilinear objective functions and polytopes $\mathcal{X},\mathcal{Y}$ encoding the players' decision spaces. Based on the CFR framework (Zinkevich et al. [66]), EFGs can be solved via decomposition into simplex-based regret minimization problems.

5.2.1. Experimental Setup. For solving EFGs, we combine the CFR decomposition with CBA$^+$ as a regret minimizer on the simplex. For the sake of simplicity, we will still call the resulting algorithm SP-CBA$^+$ (because we use alternation and linear averaging on the decisions), even though the algorithm relies on the CFR decomposition for EFGs (which is not necessary for solving the other saddle-point instances from Sections 5.1, 5.3, and 5.4). We compare SP-CBA$^+$ with CFR$^+$ (Bowling et al. [13]), the algorithm with the strongest empirical performance for solving EFGs. We also compare SP-CBA$^+$ with SP-CBA. We compare these algorithms on two Leduc poker benchmark instances (Leduc 2 players (2 pl.) and 3 ranks or 5 ranks (rks)): a search game and sheriff; we refer to Farina et al. [28] for details about the instances. Similarly, as in Section 5.1, we compare the performance as both a function of computation time and the number of iterations in the repeated game framework. We run the algorithms for time max = 100 seconds and $T = 1{,}500$ iterations; note that we choose time max and T larger for EFGs than for matrix games because the EFG instances are much larger than the matrix games from Section 5.1.

5.2.2. Results and Discussion. If we only consider the duality gap as a function of the number of iterations (Figure 3), SP-CBA$^+$ performs on par with CFR$^+$ and significantly outperforms CFR$^+$ on some EFGs instances. SP-CBA is always the slowest algorithm. However, when we consider the progress made by each algorithm during time max = 100 seconds (Figure 4), CFR$^+$ enjoys better numerical performances than SP-CBA$^+$ and SP-CBA. This is because the updates are closed form in CFR$^+$, whereas each update of SP-CBA$^+$ (or SP-CBA) requires us to solve an equation, a situation similar to that for matrix games over the simplex (Section 5.2). It is interesting to note that for EFGs, the difference in per-iteration computation time has a bigger impact than for matrix games; it is possible that this is because of our python-based implementation of SP-CBA$^+$ compared with the C-based implementation of CFR$^+$. Better implementations of SP-CBA$^+$ for EFGs could potentially lead to better results. To conclude this section, we note that CFR$^+$ enjoys the best empirical performances for solving EFGs, and it is not concerning that SP-CBA$^+$ cannot outperform CFR$^+$ on EFGs (in terms of computation time). We will see in

Figure 3. (Color online) Comparison of SP-CBA$^+$, SP-CBA, and CFR$^+$ for solving extensive-form games, as regards the number of iterations.

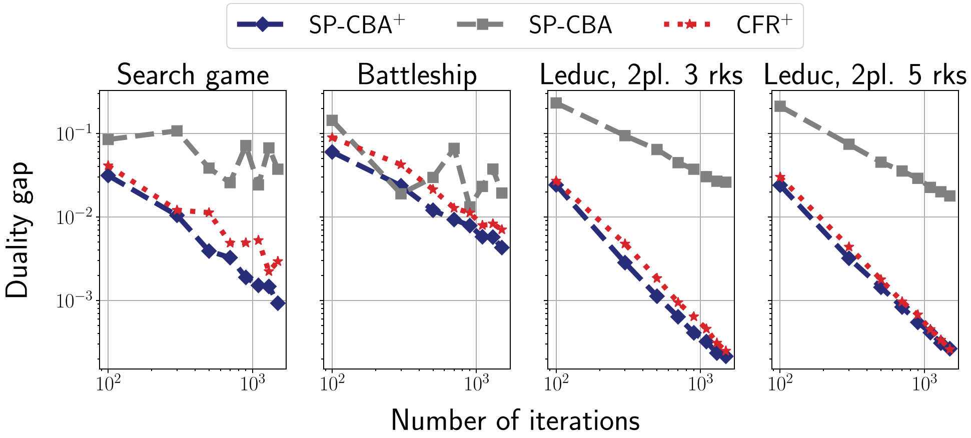
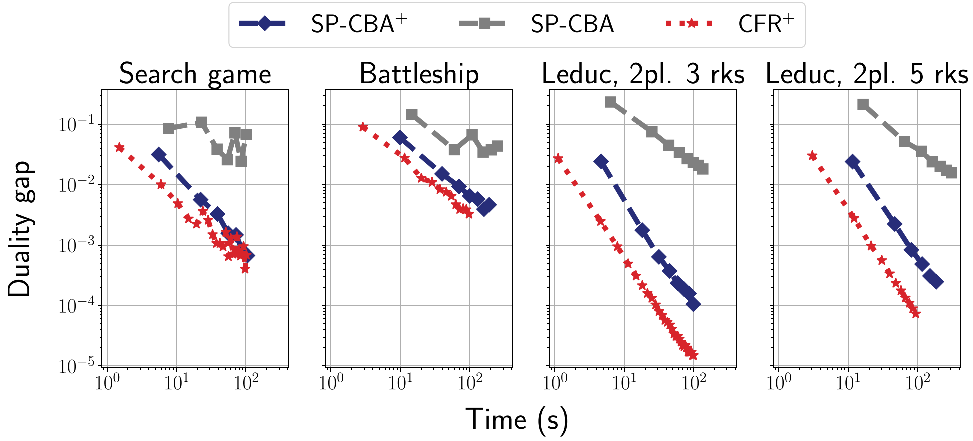


Figure 4. (Color online) Comparison of SP-CBA$^+$, SP-CBA, and CFR$^+$ for solving extensive-form games, as regards the computation time.



the next section how SP-CBA$^+$ carries over these very strong empirical results to instances where CFR$^+$ does not apply and where SP-CBA$^+$ can be implemented more efficiently.

5.3. Distributionally Robust Logistic Regression

Distributionally robust optimization exploits knowledge of the statistical properties of the model parameters to obtain risk-averse optimal solutions (Rahimian and Mehrotra [56]). We focus on the following instance of distributionally robust logistic regression (Ben-Tal et al. [8], Namkoong and Duchi [47]). There are m observed feature-label pairs $(\boldsymbol{a}_i, b_i) \in \mathbb{R}^n \times \{-1, 1\}$, and we solve

$$\min_{x\in\mathbb{R}^n, \|x-x_0\|_2 \le \epsilon_x} \max_{y\in\Delta(m), \|y-y_0\|_2\le\epsilon_y} \sum_{i=1}^{m} y_i \ell_i(x) + \frac{\mu}{2}\|x\|_2^2, \tag{19}$$

where $\ell_i(\boldsymbol{x}) = \log(1 + \exp(-b_i \boldsymbol{a}_i^\top \boldsymbol{x})), \mu \ge 0$.

5.3.1. Experimental Setup. We compare SP-CBA$^+$ with four classical FOMs: OMD, OOMD, FTRL, and OFTRL. We provide a detailed presentation of our implementations of these algorithms and our experimental setting in Appendix J; we use the ℓ_2 norm as the Bregman divergence. We also compare the performance of SP-CBA$^+$ with SP-CBA. We compare the performances of these algorithms with SP-CBA$^+$ on two synthetic data sets and two real data sets. We use parameters $\boldsymbol{x}_0 = (1, \ldots, 1)/n, \epsilon_x = 10, \boldsymbol{y}_0 = (1, \ldots, 1)/m, \epsilon_y = 1/2m, \mu = 0.1$ in (19), and we initialize all algorithms at $\boldsymbol{x}_0, \boldsymbol{y}_0$. For the synthetic classification instances, we generate a vector $\boldsymbol{x}^* \in \mathbb{R}^n$; we sample some vectors $\boldsymbol{a}_i \in \mathbb{R}^n$ at random for $i \in \{1, \ldots, m\}$ and set labels $b_i = \operatorname{sign}(\boldsymbol{a}_i^\top \boldsymbol{x}^*)$, and then, we flip 10% of the labels. We consider two types of synthetic instances: one where a_{ij} is sampled from a uniform distribution in $[0, 1]$ and one where a_{ij} is sampled from a normal distribution with mean 0 and variance 1. For the real classification instances, we use the *Australian* and *splice* data sets from the libsvm data sets library from https://www.csie.ntu.edu.tw/?cjlin/libsvmtools/datasets/. For the synthetic instances, we choose $(m, n) = (50, 100)$; for the *Australian* data set, we have $(m, n) = (690, 14)$, and for the *splice* data set, we have $(m, n) = (1,000, 60)$.

One of the main motivations for SP-CBA$^+$ is to obtain a *parameter-free* algorithm. In contrast, the other FOMs considered in this section require choosing step sizes η_t at every iteration t. This is a major limitation in practice; if the step sizes are too small, the iterates may be very conservative, whereas the algorithms may diverge with very large step sizes. We will compare the performances of the FOMs for both the fixed, theoretically correct step sizes and the tuned step sizes. The computation of the theoretically correct step sizes is presented in Appendix J.3. To tune the FOMs, we run them for the first 10 iterations, with step size $\eta_t = \alpha/\sqrt{t+1}$ for OMD and FTRL and step size $\eta_t = \alpha$ for OOMD and OFTRL, and we search for the best $\alpha \in \{0.01, 0.1, 1, 10, 100\}$. We then choose the value of α that lead to the smallest duality gap after 10 iterations and use this value for the remaining $T = 1{,}000$ iterations. Note that the tuning time and iterations (where the first 10 iterations are repeated with various values of α) are counted in the total computation time and number of iterations of the FOMs. We acknowledge that this tuning method is only one possibility and that the multiplicative factor α could be chosen in many different ways. However, any other tuning framework would still be resource demanding and uncertain. In contrast, SP-CBA$^+$ does not require any tuning, and as we will see, it outperforms even the tuned FOMs. Finally, on the y axis, we only report the worst-case loss of the current average $\bar{x}_T$; we do not report the duality gap because for a fixed value of $\boldsymbol{y}$, computing the optimal $\boldsymbol{x}$ requires solving a (regularized) nominal logistic regression, which would be computationally intensive to do at every iteration.

5.3.2. Proximal Updates for the First-Order Methods. Note that in (19), SP-CBA$^+$ is instantiated on an ℓ_2 ball (for the first player) and the intersection of an ℓ_2 ball and the simplex (for the second player). As shown in Sections 4.2 and 4.3, this leads to closed-form updates for SP-CBA$^+$ and SP-CBA at every iteration. In contrast, OMD, FTRL, OOMD, and OFTRL require binary searches for the decision of the second player at each iteration; see Appendix J. The functions used in the binary searches themselves require solving an optimization program (an orthogonal projection onto the simplex) at each evaluation. Even though computing the orthogonal projection of a vector onto the simplex of size m can be done in $O(m\log(m))$, this results in slower overall running time compared with SP-CBA$^+$ and SP-CBA with closed-form updates at each iteration. The situation is even worse for OOMD, which requires two proximal updates at each iteration.

5.3.3. Results and Discussion. In Figure 5, we show the progress of all algorithms toward solving (19) as a function of the number of iterations when the theoretical step sizes are used for the FOMs. We notice that all FOMs are progressing very slowly toward an optimal solution. This is because the theoretical step sizes are very small, relying on upper bounds on the Lipschitz constants of the objective function of (19). In contrast, SP-CBA$^+$ quickly converges to an optimal solution, even though we see in Figure 5 that during the first few iterations for the *uniform* instance, SP-CBA$^+$ (and SP-CBA) may increase the objective function. SP-CBA performs almost on par with SP-CBA$^+$ after the first few iterations. In Figure 6, we tune the FOMs for the first 10 iterations before running them (with the tuned step sizes). We note that depending on the data sets, the tuned FOMs may perform very well (e.g., OMD for the *uniform* instance, all FOMS for the *normal* instance, OOMD for the *Australian* instance) but may also fail to converge to an optimal solution, even after very good performances during the first iterations (e.g., OFTRL for the *Australian* instance). This is because the convergence guarantees of the FOMs may fail to hold for large choices of the multiplicative factor α. In Figures 7 and 8, we present the same experiments but where we record the computation time on the x axis. The per-iteration computation time of SP-CBA$^+$ is shorter than for the FOMs because SP-CBA$^+$ has closed-form updates in this setting, and we observe in Figures 7 and 8 that SP-CBA$^+$ outperforms the FOMs.

5.4. Markov Decision Processes

MDPs are used as a modeling tool for sequential decision-making problems (Puterman [55]) and have found applications in game learning (Mnih et al. [45]) and healthcare (Alagoz et al. [3], Grand-Clément et al. [34], Steimle and Denton [59]). In a finite MDP, the set of states is $[n]$, and there are A actions. For each state-action pair (s,a), there is an associated instantaneous reward r_{sa} as well as a distribution $\boldsymbol{P}_{sa} \in \Delta(n)$ over the possible next states in $[n]$. We write $r_\infty = \max_{s,a} r_{sa}$, and we assume, without loss of generality, that $r_{sa} \geq 0$, $\forall (s,a) \in [n] \times [A]$. Given a discount factor $\lambda \in (0,1)$ and an

Figure 5. (Color online) Comparisons of SP-CBA$^+$ and SP-CBA with FOMs with theoretical choices of step sizes to solve (19), with respect to the number of iterations.

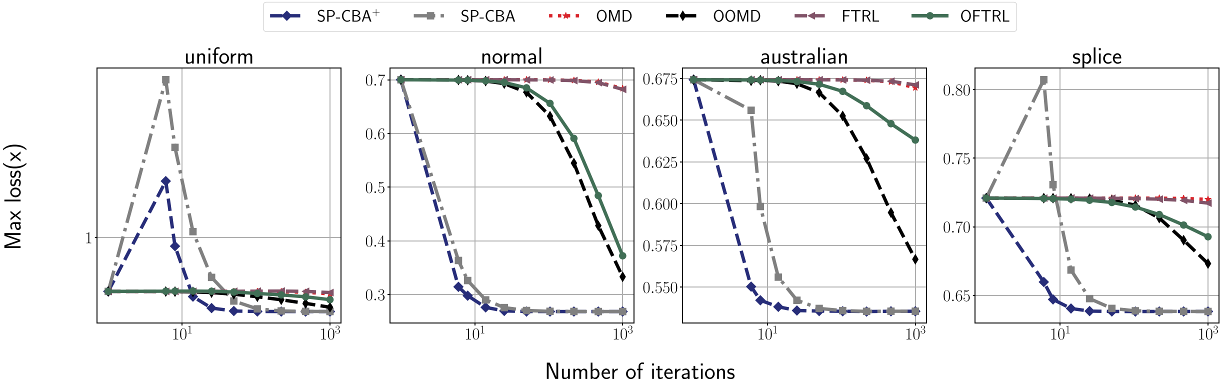
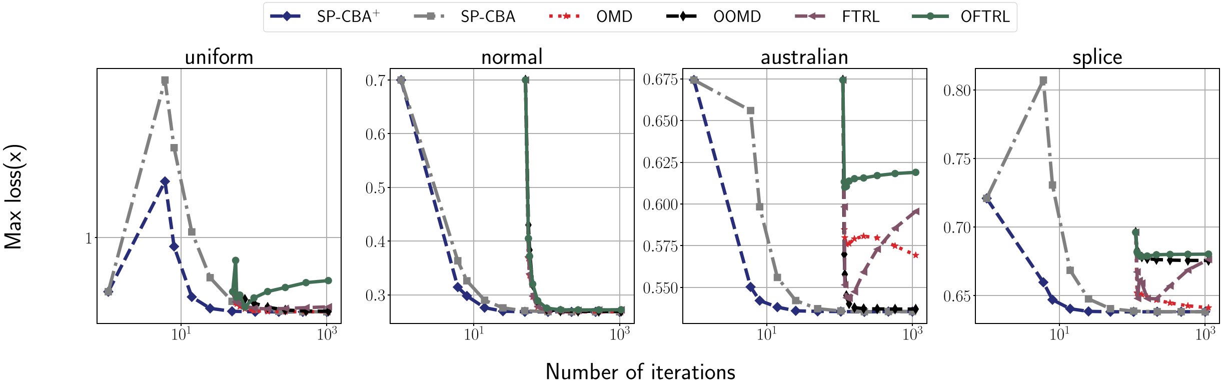


Figure 6. (Color online) Comparisons of SP-CBA$^+$ and SP-CBA with FOMs with tuned choices of step sizes to solve (19), with respect to the number of iterations.



initial probability distribution $\boldsymbol{p}_0 \in \Delta(n)$, the goal of the decision making is to maximize the infinite-horizon discounted cumulated reward. This leads to the following linear programming formulation (Puterman [55]) $\min\{(1-\lambda)\boldsymbol{p}_0^\top \boldsymbol{v} \mid v_s \geq r_{sa} + \lambda \boldsymbol{P}_{sa}^\top \boldsymbol{v},\ \forall (s,a) \in [n] \times [A]\}$, which can be rewritten as (Jin and Sidford [38])

$$\min_{\boldsymbol{v}\in\mathbb{R}^n, \|\boldsymbol{v}\|_2\leq\sqrt{n}r_\infty/(1-\lambda)} \max_{\mu\in\Delta(n\times A)} (1-\lambda)\boldsymbol{p}_0^\top \boldsymbol{v} + \sum_{s=1}^{n}\sum_{a=1}^{A} \mu_{sa}(r_{sa} + \lambda \boldsymbol{P}_{sa}^\top \boldsymbol{v} - v_s), \tag{20}$$

where $\|\boldsymbol{v}\|_2 \leq \sqrt{n}r_\infty/(1-\lambda)$ is a valid constraint for the optimal solution $\boldsymbol{v}^* \in \mathbb{R}^n$ because $\boldsymbol{v}^*$ satisfies $0 \leq v_s^* \leq r_\infty/(1-\lambda),\ \forall s \in [n]$.

5.4.1. Experimental Setup. We test the performances of SP-CBA$^+$ for solving (20) on random generalized average reward nonstationary environment test bench (garnet) MDPs (Archibald et al. [4], Bhatnagar et al. [10]), a class of random MDP instances widely used for benchmarking sequential decision-making algorithms. Garnet MDPs are parametrized by a branching factor n_b, which represents the proportion of reachable next states from each state-action pair (s, a). We choose $S = 100, A = 50, n_b = 50\%, \lambda = 0.95$. We average the performances of our algorithm over 10 random instances of garnet MDPs, where the reward parameters are drawn at random uniformly in $[0, 10]$. We compare SP-CBA$^+$ with the same first-order methods as in the previous section, OMD, FTRL, and their optimistic variants, with the same tuning method. We also compare SP-CBA$^+$ with SP-CBA. The computation of the upper bounds L_v and L_μ for each player is detailed in Appendix K. We acknowledge that at the scale of the instances considered in this paper, MDPs can be solved efficiently using policy iteration. This algorithm is specialized to solving MDPs and differs greatly from SP-CBA$^+$, which is based on the repeated game framework; for this reason, we compare SP-CBA$^+$ with first-order methods that

Figure 7. (Color online) Comparisons of SP-CBA$^+$ and SP-CBA with FOMs with theoretical choices of step sizes to solve (19), with respect to the computation time.

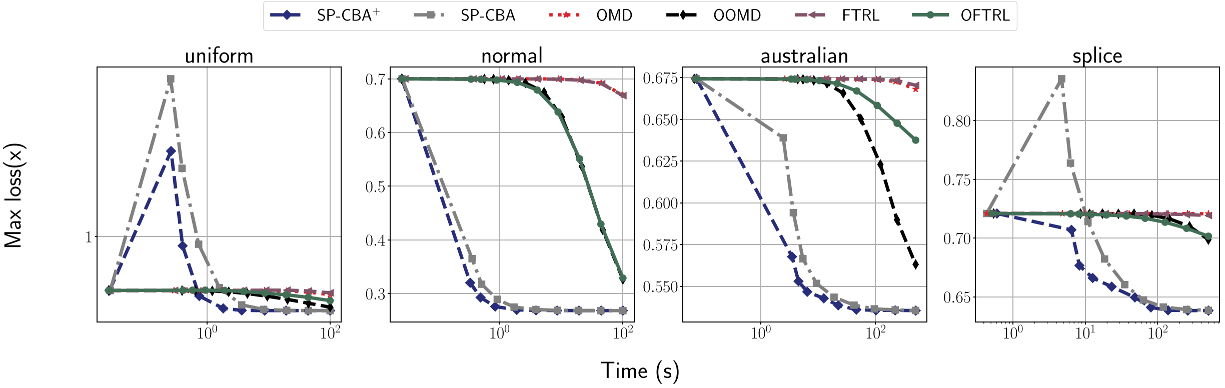
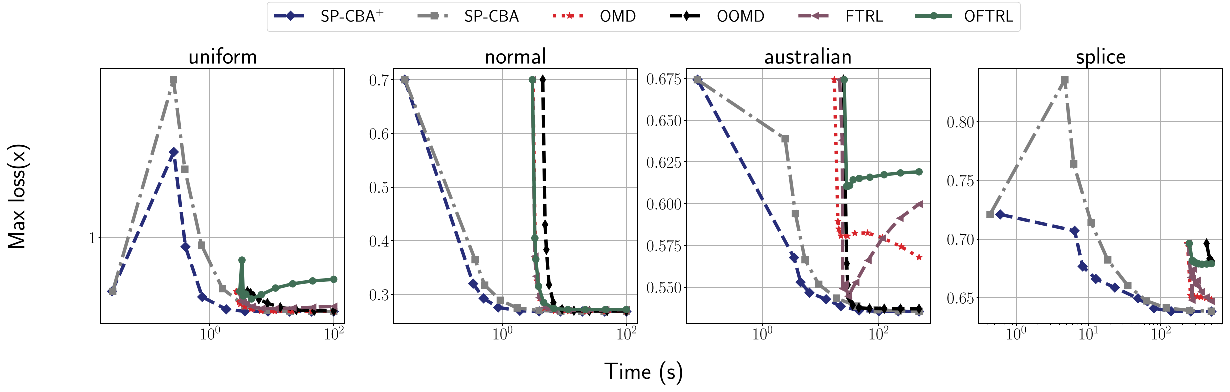


Figure 8. (Color online) Comparisons of SP-CBA$^+$ and SP-CBA with FOMs with tuned choices of step sizes to solve (19), with respect to the computation time.



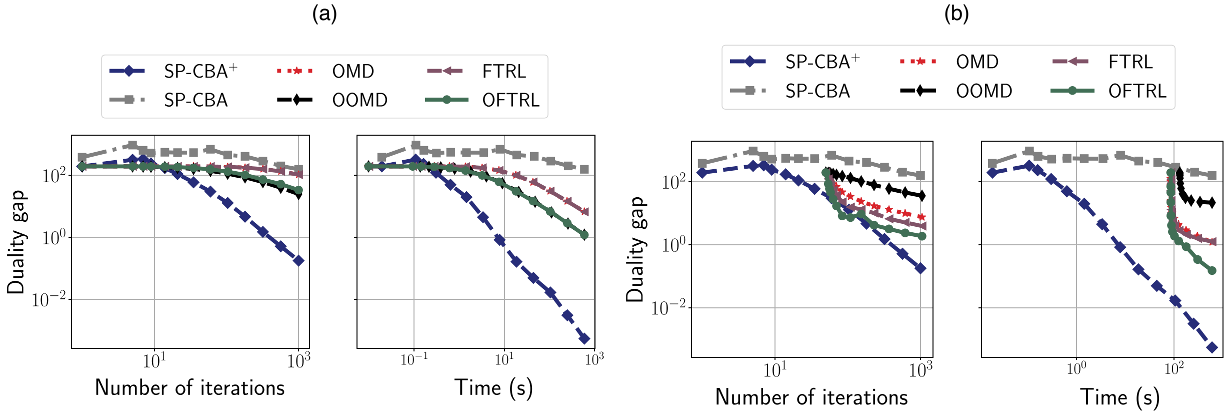
are widely applicable and that have been developed for larger MDP instances (e.g., online mirror descent for MDPs) (Jin and Sidford [38]).

5.4.2. Results and Discussion. SP-CBA$^+$ outperforms OMD and FTRL as well as the optimistic variants, even after they are tuned. Choosing the best step sizes after observing the first 10 iterations may even lead to algorithms that choose step sizes that are too large and algorithms that fail to converge, such as OOMD in Figures 9(b), and tuning the FOMs requires a lot of computation time. In contrast, SP-CBA$^+$ does not need to be tuned, and all the computation time in SP-CBA$^+$ is used to make progress toward solving (20). SP-CBA does not outperform the FOMs.

6. Conclusion

We have proposed SP-CBA$^+$, an algorithm based on Blackwell approachability for solving classical instances of saddle-point optimization. Our algorithm is (1) simple to implement for many practical decision sets, (2) completely parameter free and does not attempt to learn any step sizes, and (3) competitive with or even better than state-of-the-art approaches with both theoretical and tuned parameters. Interesting future directions of research include designing efficient implementations for other widespread decision sets (e.g., based on ϕ divergence), extending SP-CBA$^+$ to unbounded decision sets, and developing accelerated versions based on strong convex-concavity or optimism.

Figure 9. (Color online) Comparisons of SP-CBA$^+$ and SP-CBA with FOMs to solve (19) and (20), with respect to the number of iterations and to the computation time. The theoretical choices of step sizes are used in panel (a), and the tuned step sizes are used in panel (b).



Appendix A. Proof of Lemma 1

Proof of Lemma 1.

1. The fact that $\boldsymbol{u} - \pi_{\mathcal{C}^\circ}(\boldsymbol{u}) = \pi_{\mathcal{C}}(\boldsymbol{u}) \in \mathcal{C}, \langle \boldsymbol{u} - \pi_{\mathcal{C}^\circ}(\boldsymbol{u}), \pi_{\mathcal{C}^\circ}(\boldsymbol{u})\rangle = 0$ follows from Moreau's decomposition theorem (Combettes and Reyes [22]). The fact that $\|\boldsymbol{u} - \pi_{\mathcal{C}^\circ}(\boldsymbol{u})\|_2 \leq \|\boldsymbol{u}\|_2$ is a straightforward consequence of $\langle \boldsymbol{u} - \pi_{\mathcal{C}^\circ}(\boldsymbol{u}), \pi_{\mathcal{C}^\circ}(\boldsymbol{u})\rangle = 0$.

2. We follow the lines of Abernethy et al. [1, lemma 13]. For any $\boldsymbol{w} \in \mathcal{C}^\circ \cap B_2(1)$, we have

$$\langle \boldsymbol{u}, \boldsymbol{w}\rangle \leq \langle \boldsymbol{u} - \pi_{\mathcal{C}}(\boldsymbol{u}), \boldsymbol{w}\rangle \leq \|\boldsymbol{w}\|_2 \|\boldsymbol{u} - \pi_{\mathcal{C}}(\boldsymbol{u})\|_2 \leq \|\boldsymbol{u} - \pi_{\mathcal{C}}(\boldsymbol{u})\|_2.$$

Conversely, because $(\boldsymbol{u} - \pi_{\mathcal{C}}(\boldsymbol{u}))/\|\boldsymbol{u} - \pi_{\mathcal{C}}(\boldsymbol{u})\|_2 \in \mathcal{C}^\circ$, we have

$$\max_{\boldsymbol{w} \in \mathcal{C}^\circ \cap B_2(1)} \langle \boldsymbol{u}, \boldsymbol{w}\rangle \geq \|\boldsymbol{u} - \pi_{\mathcal{C}}(\boldsymbol{u})\|_2.$$

This shows that

$$\max_{\boldsymbol{w} \in \mathcal{C}^\circ \cap B_2(1)} \langle \boldsymbol{u}, \boldsymbol{w}\rangle = \|\boldsymbol{u} - \pi_{\mathcal{C}}(\boldsymbol{u})\|_2 = d(\boldsymbol{u}, \mathcal{C}).$$

3. For any $\boldsymbol{u} \in \mathbb{R}^{n+1}$, by definition we have $d(\boldsymbol{u}, \mathcal{C}^\circ) = \|\boldsymbol{u} - \pi_{\mathcal{C}^\circ}(\boldsymbol{u})\|_2$. Now, if $\boldsymbol{u} \in \mathcal{C}$, we have $\pi_{\mathcal{C}^\circ}(\boldsymbol{u}) = 0$ so $d(\boldsymbol{u}, \mathcal{C}^\circ) = \|\boldsymbol{u}\|_2$.

4. Let $\boldsymbol{u} \in \mathcal{C}$. Then, $\boldsymbol{u} = \alpha(\kappa, \boldsymbol{x})$ for $\alpha \geq 0, \boldsymbol{x} \in \mathcal{X}$. We will show that $-\boldsymbol{u} \in \mathcal{C}^\circ$. We have

$$\begin{aligned} -\boldsymbol{u} \in \mathcal{C}^\circ &\iff \langle -\boldsymbol{u}, \boldsymbol{u}'\rangle \leq 0, \ \forall \boldsymbol{u}' \in \mathcal{C} \\ &\iff \langle -\alpha(\kappa, \boldsymbol{x}), \alpha'(\kappa, \boldsymbol{x}')\rangle \leq 0, \ \forall \alpha' \geq 0, \ \forall \boldsymbol{x}' \in \mathcal{X} \\ &\iff \kappa^2 + \langle \boldsymbol{x}, \boldsymbol{x}'\rangle \geq 0, \ \forall \boldsymbol{x}' \in \mathcal{X} \\ &\iff -\langle \boldsymbol{x}, \boldsymbol{x}'\rangle \leq \kappa^2, \ \forall \boldsymbol{x}' \in \mathcal{X}, \end{aligned}$$

and $-\langle \boldsymbol{x}, \boldsymbol{x}'\rangle \leq \kappa^2$ is true by Cauchy–Schwartz inequality and the definition of $\kappa = \max_{\boldsymbol{x} \in \mathcal{X}} \|\boldsymbol{x}\|_2$.

5. We start by proving (5). Let $\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{y}, \boldsymbol{y}' \in \mathbb{R}^{n+1}$, and assume that $\boldsymbol{x} \leq_{\mathcal{C}^\circ} \boldsymbol{y}, \boldsymbol{x}' \leq_{\mathcal{C}^\circ} \boldsymbol{y}'$. Then, $\boldsymbol{y} - \boldsymbol{x} \in \mathcal{C}^\circ, \boldsymbol{y}' - \boldsymbol{x}' \in \mathcal{C}^\circ$. Because $\mathcal{C}^\circ$ is a convex set and a cone, we have $2 \cdot \left(\frac{\boldsymbol{y}-\boldsymbol{x}}{2} + \frac{\boldsymbol{y}'-\boldsymbol{x}'}{2}\right) \in \mathcal{C}^\circ$. Therefore, $\boldsymbol{y} + \boldsymbol{y}' - \boldsymbol{x} - \boldsymbol{x}' \in \mathcal{C}^\circ$ (i.e., $\boldsymbol{x} + \boldsymbol{x}' \leq_{\mathcal{C}^\circ} \boldsymbol{y} + \boldsymbol{y}'$).

We now prove (6). Let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{n+1}, \boldsymbol{x}' \in \mathcal{C}^\circ$, and assume that $\boldsymbol{x} + \boldsymbol{x}' \leq_{\mathcal{C}^\circ} \boldsymbol{y}$. Then, by definition, $\boldsymbol{y} - \boldsymbol{x} - \boldsymbol{x}' \in \mathcal{C}^\circ$. Additionally, $\boldsymbol{x}' \in \mathcal{C}^\circ$ by assumption. Because $\mathcal{C}^\circ$ is convex and is a cone, $2 \cdot \left(\frac{\boldsymbol{y}-\boldsymbol{x}-\boldsymbol{x}'}{2} + \frac{\boldsymbol{x}'}{2}\right) \in \mathcal{C}^\circ$ (i.e., $\boldsymbol{y} - \boldsymbol{x} \in \mathcal{C}^\circ$). Therefore, $\boldsymbol{x} \leq_{\mathcal{C}^\circ} \boldsymbol{y}$.

6. Let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{n+1}$ such that $\boldsymbol{x} \leq_{\mathcal{C}^\circ} \boldsymbol{y}$. Then, $\boldsymbol{y} - \boldsymbol{x} \in \mathcal{C}^\circ$. We have $d(\boldsymbol{y}, \mathcal{C}^\circ) = \min_{\boldsymbol{z} \in \mathcal{C}^\circ} \|\boldsymbol{y} - \boldsymbol{z}\|_2 \leq \|\boldsymbol{y} - (\boldsymbol{y} - \boldsymbol{x})\|_2 = \|\boldsymbol{x}\|_2$. □

Appendix B. Choice of κ for CBA and CBA$^+$

In this appendix, we introduce two extensions of CBA and CBA$^+$ parametrized by a positive scalar λ, and we compare their regret guarantees with the algorithms introduced in Section 3.1. Let us call CBA$_\lambda$ and CBA$^+{}_\lambda$ two new versions of CBA and CBA$^+$, where $\mathcal{C} = \text{cone}(\{\lambda\} \times \mathcal{X})$ and the instantaneous payoffs are defined as $\boldsymbol{v}_t = \left(\frac{\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle}{\lambda}, -\boldsymbol{f}_t\right)$.

B.1. Guarantee for CBA$_\lambda$

Following the steps of the proof of Theorem 2, we obtain the following regret guarantee for CBA$_\lambda$.

Theorem B.1. *Let $(\boldsymbol{x}_t)_{t\geq 1}$ be the sequence of decision generated by* CBA$_\lambda$ *with weights $(\omega_t)_{t\geq 1}$ on the instantaneous payoff vectors, and let $S_t = \sum_{\tau=1}^t \omega_\tau$ for any $t \geq 1$. Then,*

$$\frac{\sum_{t=1}^T \omega_t \langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \min_{\boldsymbol{x} \in \mathcal{X}} \sum_{t=1}^T \omega_t \langle \boldsymbol{f}_t, \boldsymbol{x}\rangle}{S_T} \leq \frac{\kappa^2 + \lambda^2}{\lambda} L \frac{\sqrt{\sum_{t=1}^T \omega_t^2}}{\sum_{t=1}^T \omega_t}.$$

Choosing $\lambda = \kappa$ recovers the guarantees of Theorem 2. We note that choosing $\lambda = \kappa$ minimizes the term $\lambda \longmapsto \frac{\kappa^2+\lambda^2}{\lambda}$ appearing in the upper bound on the regret in Theorem B.1.

B.2. Theoretical Guarantee for CBA$^+{}_\lambda$

We now turn to analyzing the performances of CBA$^+{}_\lambda$. In the case $\frac{\theta_{t+1}}{\theta_t} \geq \frac{\omega_{t+1}}{\omega_t}$ (i.e., when we choose potentially different weights $(\theta_t)_{t\geq 1}$ for the decisions and $(\omega_t)_{t\geq 1}$ for the payoffs), to obtain a regret guarantee for CBA$^+{}_\lambda$, we need the following assumption.

Assumption B.1.

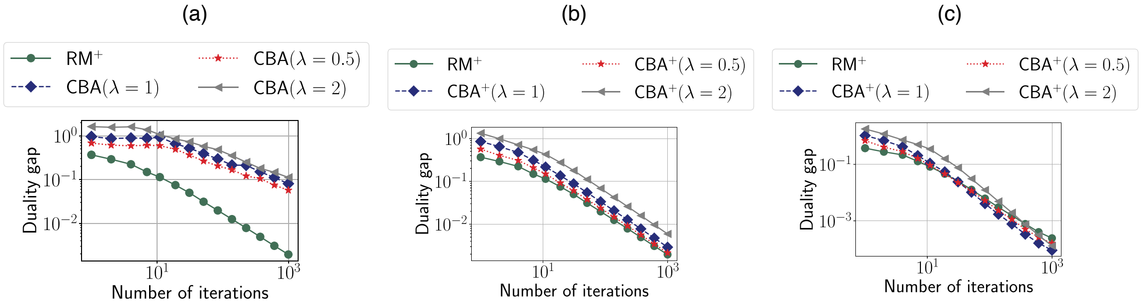
$$\langle \boldsymbol{x}, \boldsymbol{x}'\rangle + \lambda^2 \geq 0, \ \forall \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}. \tag{B.1}$$

Under this assumption, we obtain the following regret guarantees for CBA$^+{}_\lambda$ by following the same steps as the proof of Theorem 3.

Theorem B.2. *Let Assumption B.1 hold. Let $(\boldsymbol{x}_t)_{t\geq 1}$ be the sequence of decision generated by* CBA$^+$ *with weights $(\omega_t)_{t\geq 1}$ on the instantaneous payoff vectors. Let $(\theta_t)_{t\geq 1}$ be the weights on the decisions and $S_T = \sum_{t=1}^T \theta_t$. Assume that $\frac{\theta_{t+1}}{\theta_t} \geq \frac{\omega_{t+1}}{\omega_t}$, $\forall t \geq 1$. Then,*

$$\frac{\sum_{t=1}^T \theta_t \langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \min_{\boldsymbol{x} \in \mathcal{X}} \sum_{t=1}^T \theta_t \langle \boldsymbol{f}_t, \boldsymbol{x}\rangle}{S_T} \leq \frac{\kappa^2 + \lambda^2}{\lambda} L \frac{\theta_T}{\omega_T} \frac{\sqrt{\sum_{t=1}^T \omega_t^2}}{\sum_{t=1}^T \theta_t}.$$

Figure B.1. (Color online) Comparison of the empirical performances of CBA$_\lambda$ and CBA$^+{}_\lambda$ for various values of λ and various weighting schemes. (a) CBA, $\omega_t = 1$. (b) CBA$^+$, $\omega_t = 1, \theta_t = 1$. (c) CBA$^+$, $\omega_t = 1, \theta_t = t$.



Let us explain the role of Assumption B.1. The proof of Theorem 3 requires statement 4 in Lemma 1 to show that $-\left(\frac{\theta_{t+1}}{\omega_{t+1}} - \frac{\theta_t}{\omega_t}\right)\boldsymbol{u}_t \in \mathcal{C}^\circ$ using the fact that $\boldsymbol{u} \in \mathcal{C} \Rightarrow -\boldsymbol{u} \in \mathcal{C}^\circ$. Note that for $\mathcal{C} = \text{cone}(\{\lambda\} \times \mathcal{X})$ and $\boldsymbol{u} = \alpha(\lambda, \boldsymbol{x})$ for $\alpha > 0$ and $\boldsymbol{x} \in \mathcal{X}$, we have

$$-\boldsymbol{u} \in \mathcal{C}^\circ \Longleftrightarrow \lambda^2 + \langle \boldsymbol{x}, \boldsymbol{x}' \rangle \geq 0, \ \forall \boldsymbol{x}' \in \mathcal{X}.$$

Note that with $\lambda = \kappa$, Cauchy–Schwarz inequality directly yields that Assumption B.1 holds and therefore, that $\boldsymbol{u} \in \mathcal{C} \Rightarrow -\boldsymbol{u} \in \mathcal{C}^\circ$, which we use in our proof of Theorem 3. However, for $\lambda < \kappa$, Assumption B.1 may fail to hold, and we are not able to guarantee the convergence of CBA$^+{}_\lambda$ with different weights $(\theta_t)_{t \geq 1}$ on the decisions and $(\omega_t)_{t \geq 1}$ on the payoffs, which empirically perform better than using the same weights on both the decisions and the payoffs, as we show in Figure B.1, in Appendix I, and in our numerical experiments in Section 5.

Note that when $\theta_t = \omega_t$, $\forall t \geq 1$, we simply have $-\left(\frac{\theta_{t+1}}{\omega_{t+1}} - \frac{\theta_t}{\omega_t}\right)\boldsymbol{u}_t = \boldsymbol{0}$, and because $\boldsymbol{0} \in \mathcal{C}^\circ$, we obtain that $-\left(\frac{\theta_{t+1}}{\omega_{t+1}} - \frac{\theta_t}{\omega_t}\right)\boldsymbol{u}_t \in \mathcal{C}^\circ$ without any assumption. This yields the following theorem for the convergence guarantees of CBA$^+{}_\lambda$ when using the same weights on both the decisions and payoffs.

Theorem B.3. *Let $(\boldsymbol{x}_t)_{t \geq 1}$ be the sequence of decisions generated by* CBA$^+{}_\lambda$ *with weights $(\omega_t)_{t \geq 1}$ on both the decisions and the instantaneous payoff vectors, and let $S_t = \sum_{\tau=1}^t \omega_\tau$ for any $t \geq 1$. Then,*

$$\frac{\sum_{t=1}^T \omega_t \langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \min_{\boldsymbol{x} \in \mathcal{X}} \sum_{t=1}^T \omega_t \langle \boldsymbol{f}_t, \boldsymbol{x} \rangle}{S_T} \leq \frac{\kappa^2 + \lambda^2}{\lambda} L \frac{\sqrt{\sum_{t=1}^T \omega_t^2}}{\sum_{t=1}^T \omega_t}.$$

B.3. Empirical Comparison

We illustrate in Figure B.1 the performances of CBA$_\lambda$ and CBA$^+{}_\lambda$ for various values of $\lambda > 0$; in the case of small matrix games $(n, m) = (30, 30)$, the coefficients of $\boldsymbol{A}$ are drawn at random with a normal distribution, and CBA$_\lambda$ and CBA$^+{}_\lambda$ are used to solve $\min_{\boldsymbol{x} \in \Delta(n)} \max_{\boldsymbol{y} \in \Delta(m)} \langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{y} \rangle$. Recall that $\kappa = 1$ in this setup. We average the performances over 10 random instances. We present experiments for $\lambda \in \{0.5, 1, 2\}$ for CBA$_\lambda$ with uniform weights (Figure B.1(a)), CBA$^+{}_\lambda$ with uniform weights on both the decisions and the payoffs and alternation (Figure B.1(b)), and CBA$^+{}_\lambda$ with uniform weights on the payoffs and linear weights on the decisions and alternation (Figure B.1(c)). As a reference, we also show the performances of RM$^+$ with linear averaging and alternation. We note that the choice of $\lambda = \kappa = 1$ performs well compared with the other choices of λ, with $\lambda = 0.5$ being slightly better than the choice $\lambda = 1$ in the early iterations and all choices of $\{0.5, 1, 2\}$ having comparable performances after 10^3 iterations of the repeated game framework.

Appendix C. Proof of Theorem 4

Proof of Theorem 4. Let $\boldsymbol{z}^\star \in \sigma_K$ be the sequence attaining the minimum in

$$\min_{\boldsymbol{z} \in \sigma_K} \sum_{t=1}^T \theta_t \langle \boldsymbol{f}_t, \boldsymbol{z}_t \rangle.$$

We first partition the set $\{1, \ldots, T\}$ into subintervals where $\boldsymbol{z}^\star$ is constant. In particular, we can partition $\{1, \ldots, T\}$ into K intervals $\mathcal{I}_1, \ldots, \mathcal{I}_K$, such that

$$z_t^\star = z_{t'}^\star, \ \forall t, t' \in \mathcal{I}_\ell, \ \forall \ell \in [K].$$

With this notation, we have

$$\begin{aligned} \sum_{t=1}^T \theta_t \langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \min_{\boldsymbol{z} \in \sigma_K} \sum_{t=1}^T \theta_t \langle \boldsymbol{f}_t, \boldsymbol{z}_t \rangle &= \sum_{\ell=1}^K \sum_{t \in \mathcal{I}_\ell} \theta_t \langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \sum_{t \in \mathcal{I}_\ell} \theta_t \langle \boldsymbol{f}_t, \boldsymbol{z}_\ell^\star \rangle \\ &= \sum_{\ell=1}^K \sum_{t \in \mathcal{I}_\ell} \theta_t \langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \min_{\boldsymbol{z} \in \mathcal{X}} \sum_{t \in \mathcal{I}_\ell} \theta_t \langle \boldsymbol{f}_t, \boldsymbol{z} \rangle \end{aligned}$$

because each $z^\star_\ell$ attains $\min_{z\in\mathcal{X}}\sum_{t\in\mathcal{I}_\ell}\theta_t\langle \boldsymbol{f}_t, \boldsymbol{z}\rangle$ (otherwise, $\boldsymbol{z}^\star$ would not be optimal). Similarly as in the proof of Theorem 3, we obtain, for all $\ell\in[K]$,

$$\sum_{t\in\mathcal{I}_\ell}\theta_t\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \min_{z\in\mathcal{X}}\sum_{t\in\mathcal{I}_\ell}\theta_t\langle \boldsymbol{f}_t, \boldsymbol{z}\rangle \le \sqrt{2}\kappa d\left(\sum_{t\in\mathcal{I}_\ell}\theta_t\boldsymbol{v}_t, \mathcal{C}^\circ\right).$$

Let us call $\mathsf{top}(\ell)$ and $\mathsf{down}(\ell)$ the largest and smallest integers in each interval $\mathcal{I}_\ell$, respectively. The same proof as for Theorem 3 shows that

$$\sum_{t\in\mathcal{I}_\ell}\theta_t\boldsymbol{v}_t = \sum_{t=\mathsf{down}(\ell)}^{\mathsf{top}(\ell)}\theta_t\boldsymbol{v}_t \ge_{\mathcal{C}^\circ} \frac{\theta_{\mathsf{top}(\ell)}}{\omega_{\mathsf{top}(\ell)}}\boldsymbol{u}_{\mathsf{top}(\ell)} - \frac{\theta_{\mathsf{down}(\ell)}}{\omega_{\mathsf{down}(\ell)}}\boldsymbol{u}_{\mathsf{down}(\ell)},$$

and from (6) in statement 5 in Lemma 1, we conclude that

$$\sum_{t\in\mathcal{I}_\ell}\theta_t\boldsymbol{v}_t \ge_{\mathcal{C}^\circ} \frac{\theta_{\mathsf{top}(\ell)}}{\omega_{\mathsf{top}(\ell)}}\boldsymbol{u}_{\mathsf{top}(\ell)}.$$

Overall, from statement 6 in Lemma 1, we obtain that for each $\ell\in[K]$,

$$d\left(\sum_{t\in\mathcal{I}_\ell}\theta_t\boldsymbol{v}_t, \mathcal{C}^\circ\right) \le \frac{\theta_{\mathsf{top}(\ell)}}{\omega_{\mathsf{top}(\ell)}}\|\boldsymbol{u}_{\mathsf{top}(\ell)}\|_2.$$

By assumption, $\frac{\theta_{\mathsf{top}(\ell)}}{\omega_{\mathsf{top}(\ell)}} \le \frac{\theta_T}{\omega_T}$, and following (15), we have

$$\|\boldsymbol{u}_{\mathsf{top}(\ell)}\|_2 \le \sqrt{2}L\sqrt{\sum_{t=1}^{\mathsf{top}(\ell)}\omega_t^2} \le \sqrt{2}L\sqrt{\sum_{t=1}^{T}\omega_t^2}.$$

Because this holds for any $\ell\in[K]$, we conclude that

$$\frac{\sum_{t=1}^T\theta_t\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \min_{z\in\sigma_K}\sum_{t=1}^T\theta_t\langle \boldsymbol{f}_t, \boldsymbol{z}_t\rangle}{S_T} \le 2\kappa LK\frac{\theta_T}{\omega_T}\frac{\sqrt{\sum_{t=1}^T\omega_t^2}}{\sum_{t=1}^T\theta_t}. \quad \square$$

Appendix D. Proof of Theorem 5

Proof of Theorem 5. We prove the theorem for each part separately.

1. Let

$$\bar{\boldsymbol{x}}_T = \frac{1}{S_T}\sum_{t=1}^T\omega_t\boldsymbol{x}_t, \bar{\boldsymbol{y}}_T = \frac{1}{S_T}\sum_{t=1}^T\omega_t\boldsymbol{y}_t.$$

Because F is convex-concave, we first have

$$\max_{y\in\mathcal{Y}}F(\bar{\boldsymbol{x}}_T, \boldsymbol{y}) - \min_{x\in\mathcal{X}}F(\boldsymbol{x}, \bar{\boldsymbol{y}}_T) \le \frac{1}{S_T}\left(\max_{y\in\mathcal{Y}}\sum_{t=1}^T\omega_tF(\boldsymbol{x}_t, \boldsymbol{y}) - \min_{x\in\mathcal{X}}\sum_{t=1}^T\omega_tF(\boldsymbol{x}, \boldsymbol{y}_t)\right).$$

Now,

$$\begin{aligned}\max_{y\in\mathcal{Y}}\sum_{t=1}^T\omega_tF(\boldsymbol{x}_t, \boldsymbol{y}) - \min_{x\in\mathcal{X}}\sum_{t=1}^T\omega_tF(\boldsymbol{x}, \boldsymbol{y}_t) &= \max_{y\in\mathcal{Y}}\sum_{t=1}^T\omega_tF(\boldsymbol{x}_t, \boldsymbol{y}) - \sum_{t=1}^T\omega_tF(\boldsymbol{x}_t, \boldsymbol{y}_t)\\ &\quad + \sum_{t=1}^T\omega_tF(\boldsymbol{x}_t, \boldsymbol{y}_t) - \min_{x\in\mathcal{X}}\sum_{t=1}^T\omega_tF(\boldsymbol{x}, \boldsymbol{y}_t).\end{aligned}$$

Now, because F is convex-concave, we can upper bound each pair of terms using the subgradient inequality:

$$\max_{y\in\mathcal{Y}}\sum_{t=1}^T\omega_tF(\boldsymbol{x}_t, \boldsymbol{y}) - \sum_{t=1}^T\omega_tF(\boldsymbol{x}_t, \boldsymbol{y}_t) \le \max_{y\in\mathcal{Y}}\omega_t\sum_{t=1}^T\langle \boldsymbol{g}_t, \boldsymbol{y}\rangle - \sum_{t=1}^T\omega_t\langle \boldsymbol{g}_t, \boldsymbol{y}_t\rangle,$$

$$\sum_{t=1}^T\omega_tF(\boldsymbol{x}_t, \boldsymbol{y}_t) - \min_{x\in\mathcal{X}}\sum_{t=1}^T\omega_tF(\boldsymbol{x}, \boldsymbol{y}_t) \le \sum_{t=1}^T\omega_t\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \min_{x\in\mathcal{X}}\sum_{t=1}^T\omega_t\langle \boldsymbol{f}_t, \boldsymbol{x}\rangle,$$

where $\boldsymbol{f}_t \in \partial_x F(\boldsymbol{x}_t, \boldsymbol{y}_t), \boldsymbol{g}_t \in \partial_y F(\boldsymbol{x}_t, \boldsymbol{y}_t)$ (recall the repeated game framework presented at the beginning of Section 2). We recognize the right-hand side as the regrets in the repeated game framework. For CBA with weights on both payoffs and decisions (Theorem 2), we have shown that

$$\begin{aligned}
\frac{1}{S_T}\max_{y\in\mathcal{Y}}\sum_{t=1}^{T}\omega_t\langle \boldsymbol{g}_t, \boldsymbol{y}\rangle - \sum_{t=1}^{T}\langle \omega_t \boldsymbol{g}_t, \boldsymbol{y}_t\rangle &= O\left(\kappa L\frac{\sqrt{\sum_{t=1}^{T}\omega_t^2}}{\sum_{t=1}^{T}\omega_t}\right),\\
\frac{1}{S_T}\sum_{t=1}^{T}\omega_t\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \min_{x\in\mathcal{X}}\sum_{t=1}^{T}\omega_t\langle \boldsymbol{f}_t, \boldsymbol{x}\rangle &= O\left(\kappa L\frac{\sqrt{\sum_{t=1}^{T}\omega_t^2}}{\sum_{t=1}^{T}\omega_t}\right).
\end{aligned}$$

Recall that $\omega_t = t^p$. Because $t \longmapsto t^p$ is an increasing function, we have

$$\int_0^k t^p dt \le \sum_{t=1}^{k} t^p \le \int_0^{k+1} t^p dt.$$

Therefore, we can conclude that

$$\begin{aligned}
\sum_{t=1}^{T}\omega_t^2 &= O\left(\frac{1}{p+1}T^{2p+1}\right),\\
\frac{1}{p+1}T^{p+1} &\le \sum_{t=1}^{T}\omega_t.
\end{aligned}$$

Overall, we obtain that

$$O\left(\kappa L\frac{\sqrt{\sum_{t=1}^{T}\omega_t^2}}{\sum_{t=1}^{T}\omega_t}\right) = O\left(\frac{\kappa L\sqrt{p+1}}{\sqrt{T}}\right).$$

2. This proof is mostly similar to the first part. We have

$$\begin{aligned}
\theta_T &= T^q,\\
\frac{T^{q+1}}{q+1} &\le \sum_{t=1}^{T}\theta_t,\\
\sqrt{\sum_{t=1}^{T}\omega_t^2} &= O\left(\frac{1}{\sqrt{p+1}}T^{p+1/2}\right),\\
\omega_T &= T^p.
\end{aligned}$$

Combining all this, we obtain that an upper bound of

$$O\left(\kappa L\frac{(q+1)T^q}{T^{q+1}}\frac{T^{p+1/2}}{\sqrt{p+1}T^p}\right)$$

is equal to $O\left(\frac{\kappa L(q+1)}{\sqrt{p+1}\sqrt{T}}\right)$. □

Appendix E. Proof for Theorem 6

Proof of Theorem 6. The proof of Theorem 6 is similar to the proof of Theorem 5 presented in Appendix D. Let

$$\bar{\boldsymbol{x}}_T = \frac{1}{S_T}\sum_{t=1}^{T}\theta_{t+1}\boldsymbol{x}_{t+1}, \bar{\boldsymbol{y}}_T = \frac{1}{S_T}\sum_{t=1}^{T}\theta_{t+1}\boldsymbol{y}_t.$$

Because F is convex-concave, we first have

$$\max_{y\in\mathcal{Y}} F(\bar{\boldsymbol{x}}_T, \boldsymbol{y}) - \min_{x\in\mathcal{X}} F(\boldsymbol{x}, \bar{\boldsymbol{y}}_T) \le \frac{1}{S_T}\left(\max_{y\in\mathcal{Y}}\sum_{t=1}^{T}\theta_{t+1}F(\boldsymbol{x}_{t+1}, \boldsymbol{y}) - \min_{x\in\mathcal{X}}\sum_{t=1}^{T}\theta_{t+1}F(\boldsymbol{x}, \boldsymbol{y}_t)\right).$$

Now, we can rewrite

$$\max_{y\in\mathcal{Y}}\sum_{t=1}^{T}\theta_{t+1}F(x_{t+1},y)-\min_{x\in\mathcal{X}}\sum_{t=1}^{T}\theta_{t+1}F(x,y_t)$$

as

$$\begin{aligned}&\max_{y\in\mathcal{Y}}\sum_{t=1}^{T}\theta_{t+1}F(x_{t+1},y)-\sum_{t=1}^{T}\theta_{t+1}F(x_{t+1},y_t)\\&+\sum_{t=1}^{T}\theta_{t+1}F(x_{t+1},y_t)-\sum_{t=1}^{T}\theta_{t+1}F(x_t,y_t)\\&+\sum_{t=1}^{T}\theta_{t+1}F(x_t,y_t)-\min_{x\in\mathcal{X}}\sum_{t=1}^{T}\theta_{t+1}F(x,y_t).\end{aligned}$$

Now, because F is convex-concave, we can use the following upper bound:

$$\max_{y\in\mathcal{Y}}\sum_{t=1}^{T}\theta_{t+1}F(x_{t+1},y)-\sum_{t=1}^{T}\theta_{t+1}F(x_{t+1},y_t)\le\max_{y\in\mathcal{Y}}\theta_{t+1}\sum_{t=1}^{T}\langle g_t,y\rangle-\sum_{t=1}^{T}\theta_{t+1}\langle g_t,y_t\rangle,$$
$$\sum_{t=1}^{T}\theta_{t+1}F(x_t,y_t)-\min_{x\in\mathcal{X}}\sum_{t=1}^{T}\theta_{t+1}F(x,y_t)\le\sum_{t=1}^{T}\theta_{t+1}\langle f_t,x_t\rangle-\min_{x\in\mathcal{X}}\sum_{t=1}^{T}\theta_{t+1}\langle f_t,x\rangle,$$

where $f_t\in\partial_xF(x_t,y_t),g_t\in\partial_yF(x_{t+1},y_t)$. This concludes the proof of Theorem 6. □

Appendix F. Proof of Theorem 7

We start with the following lemma. It shows that once a nondegenerate update has been chosen ($u_t\neq\mathbf{0}$ for CBA$^+$ and $\pi_{\mathcal{C}}(u_t)\neq\mathbf{0}$ for CBA), all the future updates are also nondegenerate.

Lemma F.1.

1. *Let $(u_t)_{t\ge1}\in(\mathbb{R}^{n+1})^{\mathbb{N}}$ be the sequence of aggregate payoffs generated by* CBA *with weights $(\omega_t)_{t\ge1}$ on the instantaneous payoff vectors. Let $t\ge1$. If $\pi_{\mathcal{C}}(u_t)\neq\mathbf{0}$, then for all $t'\ge t$, we also have $\pi_{\mathcal{C}}(u_{t'})\neq\mathbf{0}$.*
2. *Let $(u_t)_{t\ge1}\in(\mathbb{R}^{n+1})^{\mathbb{N}}$ be the sequence of aggregate payoffs generated by* CBA$^+$ *with weights $(\omega_t)_{t\ge1}$ on the instantaneous payoff vectors. Let $t\ge1$. If $u_t\neq\mathbf{0}$, then for all $t'\ge t$, we also have $u_{t'}\neq\mathbf{0}$.*

Proof of Lemma F.1.

1. Assume that $\pi_{\mathcal{C}}(u_t)\neq\mathbf{0}$. Let $\boldsymbol{\pi}_t=(\tilde{\pi}_t,\hat{\boldsymbol{\pi}}_t)$ such that $\boldsymbol{\pi}_t=\pi_{\mathcal{C}}(u_t)$. In this case, we can define $x_{t+1}=(\kappa/\tilde{\pi}_t)\hat{\boldsymbol{\pi}}_t$. By definition of the updates in CBA, we have

$$u_{t+1}=u_t+\omega_{t+1}v_{t+1},$$

for $v_{t+1}=\left(\frac{\langle f_{t+1},x_{t+1}\rangle}{\kappa},-f_{t+1}\right)$. We will show that $u_{t+1}\notin\mathcal{C}^{\circ}$. By definition,

$$u_{t+1}\notin\mathcal{C}^{\circ}\iff\exists z\in\mathcal{C},\langle z,u_{t+1}\rangle>0.$$

If we take $z=\boldsymbol{\pi}_t$, we have

$$\langle\boldsymbol{\pi}_t,u_{t+1}\rangle=\langle\boldsymbol{\pi}_t,u_t+\omega_{t+1}v_{t+1}\rangle=\langle\boldsymbol{\pi}_t,u_t\rangle$$

because by definition of x_{t+1}, we have $\langle\boldsymbol{\pi}_t,v_{t+1}\rangle=0$. Now,

$$\langle\boldsymbol{\pi}_t,u_t\rangle=\langle\pi_{\mathcal{C}}(u_t),\pi_{\mathcal{C}}(u_t)+\pi_{\mathcal{C}^{\circ}}(u_t)\rangle=\langle\pi_{\mathcal{C}}(u_t),\pi_{\mathcal{C}}(u_t)\rangle=\|\pi_{\mathcal{C}}(u_t)\|_2^2>0.$$

This shows that $u_{t+1}\notin\mathcal{C}^{\circ}$. Because $u_{t+1}=\pi_{\mathcal{C}}(u_{t+1})+\pi_{\mathcal{C}^{\circ}}(u_{t+1})$, this also shows that $\pi_{\mathcal{C}}(u_{t+1})\neq\mathbf{0}$. By induction, we have shown that $\pi_{\mathcal{C}}(u_t)\neq\mathbf{0}\Rightarrow\pi_{\mathcal{C}}(u_{t'})\neq\mathbf{0}$, $\forall t'\ge t$.

2. The proof is very similar to the proof of the first statement. Suppose that $u_t\neq\mathbf{0}$. In this case, we can define $x_{t+1}=(\kappa/\tilde{u}_t)\hat{u}_t$. Note that by definition of the updates in CBA$^+$, we have

$$u_{t+1}=\pi_{\mathcal{C}}(u_t+\omega_{t+1}v_{t+1}).$$

We will show that

$$u_t+\omega_{t+1}v_{t+1}\notin\mathcal{C}^{\circ}.$$

By definition of $\mathcal{C}^{\circ}$,

$$\boldsymbol{u}_t + \omega_{t+1}\boldsymbol{v}_{t+1} \notin \mathcal{C} \iff \exists \boldsymbol{z} \in \mathcal{C}, \langle \boldsymbol{z}, \boldsymbol{u}_t + \boldsymbol{v}_{t+1} \rangle > 0.$$

For $\boldsymbol{z} = \boldsymbol{u}_t$, we obtain

$$\langle \boldsymbol{u}_t, \boldsymbol{u}_t + \omega_{t+1}\boldsymbol{v}_{t+1} \rangle = \langle \boldsymbol{u}_t, \boldsymbol{u}_t \rangle = \|\boldsymbol{u}_t\|_2^2 > 0,$$

where

$$\langle \boldsymbol{u}_t, \boldsymbol{v}_{t+1} \rangle = 0$$

follows from the choice of $\boldsymbol{x}_{t+1}$ as in Blackwell approachability framework (see (11) in the proof of Theorem 2 for more details). Therefore, for any $t \geq 1$, we have $\boldsymbol{u}_t \neq \boldsymbol{0} \Rightarrow \boldsymbol{u}_{t+1} \neq \boldsymbol{0}$. This concludes the proof of Lemma F.1 by induction. □

We are now ready to prove Theorem 7.

Proof of Theorem 7. Assume that $(\boldsymbol{x}, \boldsymbol{y}) \longmapsto F(\boldsymbol{x}, \boldsymbol{y})$ is linear in $\boldsymbol{x}$.

1. We start with the case $\boldsymbol{u}_t^x = \boldsymbol{0}$. In this case, by definition of $\boldsymbol{x}_{t+1}$ as $\boldsymbol{x}_{t+1} = \text{CHOOSEDECISION}_{\text{CBA}^+}(\boldsymbol{u}_t^x)$, we have $\boldsymbol{x}_{t+1} = \boldsymbol{x}_0$. From Lemma F.1, we know that $\boldsymbol{u}_{t-1}^x = \boldsymbol{0}$ (i.e., $\boldsymbol{x}_t = \boldsymbol{x}_0$). This shows that $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t$ and therefore, that $F(\boldsymbol{x}_t, \boldsymbol{y}_t) - F(\boldsymbol{x}_{t+1}, \boldsymbol{y}_t) = 0$.

We now consider the case $\boldsymbol{u}_t^x \neq \boldsymbol{0}$. We want to prove that

$$F(\boldsymbol{x}_t, \boldsymbol{y}_t) \geq F(\boldsymbol{x}_{t+1}, \boldsymbol{y}_t) + \frac{\kappa}{\omega_t \|\boldsymbol{u}_t\|_\infty} \|\boldsymbol{u}_t - \boldsymbol{u}_{t-1}\|_2^2. \tag{F.1}$$

Let $t \geq 1$. Recall that

$$\begin{aligned} \boldsymbol{x}_t &= \text{CHOOSEDECISION}_{\text{CBA}^+}(\boldsymbol{u}_{t-1}), \\ \boldsymbol{x}_{t+1} &= \text{CHOOSEDECISION}_{\text{CBA}^+}(\boldsymbol{u}_t). \end{aligned}$$

We consider the following two cases.

Case F.1. $\boldsymbol{u}_t = \boldsymbol{0}$. From Lemma F.1, we must have $\boldsymbol{u}_{t-1} = \boldsymbol{0}$, in which case $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t = \boldsymbol{x}_0$ (the default value for the decisions of the first player) so that (F.1) holds because every term is 0, with the convention that $0/0 = 0$ (in case $\boldsymbol{u}_{t+1} = \boldsymbol{0}$).

Case F.2. $\boldsymbol{u}_t \neq \boldsymbol{0}$. We start from

$$\boldsymbol{u}_t = \pi_{\mathcal{C}}(\boldsymbol{u}_{t-1} + \omega_t \boldsymbol{v}_t)$$

with $\boldsymbol{v}_t = \left(\frac{\langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle}{\kappa}, -\boldsymbol{f}_t\right)$. The optimality condition for the projection on $\mathcal{C}$ shows that

$$\langle \boldsymbol{u}_t - \boldsymbol{u}_{t-1} - \omega_t \boldsymbol{v}_t, \boldsymbol{u}_t - \boldsymbol{z} \rangle \leq 0, \ \forall \boldsymbol{z} \in \mathcal{C}.$$

We can apply this with $\boldsymbol{z} = \boldsymbol{u}_{t-1}$ to obtain

$$\langle \boldsymbol{u}_t - \boldsymbol{u}_{t-1} - \omega_t \boldsymbol{v}_t, \boldsymbol{u}_t - \boldsymbol{u}_{t-1} \rangle \leq 0.$$

This shows that

$$\|\boldsymbol{u}_t - \boldsymbol{u}_{t-1}\|_2^2 \leq \langle \omega_t \boldsymbol{v}_t, \boldsymbol{u}_t - \boldsymbol{u}_{t-1} \rangle.$$

Recall that by definition of $\boldsymbol{x}_t$ and $\boldsymbol{v}_t$, we have

$$\langle \boldsymbol{v}_t, \boldsymbol{u}_{t-1} \rangle = 0.$$

Recall that $\boldsymbol{u}_t = \alpha_{t+1}(\kappa, \boldsymbol{x}_{t+1})$, with $\alpha_{t+1} > 0$ because $\boldsymbol{u}_t \neq \boldsymbol{0}$. This implies that

$$\begin{aligned} \langle \omega_t \boldsymbol{v}_t, \boldsymbol{u}_t - \boldsymbol{u}_{t-1} \rangle &= \langle \omega_t \boldsymbol{v}_t, \boldsymbol{u}_t \rangle \\ &= \omega_t \left\langle \left(\frac{\langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle}{\kappa}, -\boldsymbol{f}_t \right), \alpha_{t+1}(\kappa, \boldsymbol{x}_{t+1}) \right\rangle \\ &= \omega_t \alpha_{t+1} (\langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \langle \boldsymbol{f}_t, \boldsymbol{x}_{t+1} \rangle). \end{aligned}$$

Overall, we have obtained

$$\langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle \geq \langle \boldsymbol{f}_t, \boldsymbol{x}_{t+1} \rangle + \frac{1}{\omega_t \alpha_{t+1}} \|\boldsymbol{u}_t - \boldsymbol{u}_{t-1}\|_2^2.$$

Recall that by definition, $\boldsymbol{u}_t = \alpha_{t+1}(\kappa, \boldsymbol{x}_{t+1})$, with $\kappa = \max\{\|\boldsymbol{x}\|_2 \,|\, \boldsymbol{x} \in \mathcal{X}\}$. Therefore,

$$\|\boldsymbol{u}_t\|_\infty = \alpha_{t+1} \max\{\kappa, \|\boldsymbol{x}_{t+1}\|_\infty\} = \alpha_{t+1}\kappa,$$

where the last inequality follows from $\|x_{t+1}\|_\infty \leq \|x_{t+1}\|_2 \leq \kappa$. Overall, we have shown that

$$\langle f_t, x_t\rangle \geq \langle f_t, x_{t+1}\rangle + \frac{\kappa}{\omega_t \|u_t\|_\infty}\|u_t - u_{t-1}\|_2^2.$$

Recall that in the repeated game framework with alternation, we have $f_t = \partial_x F(x_t, y_t)$. For an objective function that is linear in x, we obtain

$$\langle f_t, x_{t+1}\rangle = F(x_{t+1}, y_t),$$
$$\langle f_t, x_t\rangle = F(x_t, y_t).$$

In this case, we have shown that

$$F(x_t, y_t) \geq F(x_{t+1}, y_t) + \frac{\kappa}{\omega_t \|u_t\|_\infty}\|u_t - u_{t-1}\|_2^2.$$

This concludes the proof of the first statement of Theorem 7.

2. The proof is identical to the first claim of this theorem. For the sake of conciseness, we omit it in this paper. □

Appendix G. Alternation for RM and RM$^+$

In this appendix, we prove that alternation *strictly* improves the convergence guarantees of RM and RM$^+$ in the repeated game framework. In particular, we show that our results from Theorem 7 for CBA and CBA$^+$ extend to RM and RM$^+$. We first give some more context on RM and RM$^+$.

The RM algorithm (Hart and Mas-Colell [35]) maintains a sequence of *aggregate payoffs* $(r_t)_{t\geq 0}$ and is presented in Algorithm G.1. Here, we write $[r]^+$ for the vector $(\max\{0, r_i\})_{i\in[n]}$. The RM$^+$ algorithm is a simple variation of RM, where the aggregate payoffs are thresholded at every iteration (Tammelin et al. [61]). In particular, RM$^+$ only keeps track of the nonnegative components of the aggregate payoffs to compute a decision. This is analogous to the projection step of CBA$^+$. We present RM$^+$ in Algorithm G.2.

Algorithm G.1 (RM)

```
Algorithm parameters Weights $(\omega_t)_{t\geq 0}$
Initialization $t = 1, x_1 \in \Delta(n)$.
Observe $f_1$ then set $r_1 = \omega_1(\langle f_1, x_1\rangle e - f_1) \in \mathbb{R}^n$.
for $t \geq 1$ do
    If $[r_t]^+ \neq 0$ then $x_{t+1} = [r_t]^+/\|[r_t]^+\|_1$ else $x_{t+1} = x_0$.
    Observe the loss $f_{t+1} \in \mathbb{R}^n$.
    Update $r_{t+1} = r_t + \omega_{t+1}(\langle f_{t+1}, x_{t+1}\rangle e - f_{t+1})$.
end for
```

Algorithm G.2 (RM$^+$)

```
Algorithm parameters Weights $(\omega_t)_{t\geq 0}$
Initialization $t = 1, x_1 \in \Delta(n)$.
Observe $f_1$ then set $q_1 = [\omega_1(\langle f_1, x_1\rangle e - f_1)]^+ \in \mathbb{R}^n$.
for $t \geq 1$ do
    If $q_t \neq 0$ then $x_{t+1} = q_t/\|q_t\|_1$ else $x_{t+1} = x_0$.
    Observe the loss $f_{t+1} \in \mathbb{R}^n$.
    Update $q_{t+1} = [q_t + \omega_{t+1}(\langle f_{t+1}, x_{t+1}\rangle e - f_{t+1})]^+$.
end for
```

We have the following theorem for RM and RM$^+$, which provides the first result showing a strict benefit to using alternation in the RM and RM$^+$ setting; previously, it was only known that alternation does not hurt the theoretical rate (Burch et al. [17]). We omit the proof as it is similar to our proof of Theorem 7 for CBA and CBA$^+$.

Theorem G.1. *Assume that $\mathcal{X} = \Delta(n), \mathcal{Y} = \Delta(m)$ and that $(x, y) \longmapsto F(x, y)$ is linear in x.*

1. *In the framework of Theorem 6, suppose that $(x_t)_{t\geq 1}, (y_t)_{t\geq 1}$ are generated by* RM$^+$ *with weights $(\omega_t)_{t\geq 1}$ on the payoffs. We have, for $t \geq 1$,*

$$F(x_{t+1}, y_t) - F(x_t, y_t) \leq -\frac{\|q_t^x - q_{t-1}^x\|_2^2}{\omega_t \cdot \|q_t^x\|_1}. \tag{G.1}$$

2. *In the framework of Theorem 6, suppose that $(x_t)_{t\geq 1}, (y_t)_{t\geq 1}$ are generated by* RM *with weights $(\omega_t)_{t\geq 1}$ on the payoffs. We have, for $t \geq 1$,*

$$F(x_{t+1}, y_t) - F(x_t, y_t) \leq -\frac{\|[r_t^x]^+ - [r_{t-1}^x]^+\|_2^2}{\omega_t \cdot \|[r_t^x]^+\|_2}. \tag{G.2}$$

Appendix H. Proofs for the Efficient Projections of Section 4

H.1. Proofs for the Simplex

Proof of Lemma 2. For $\mathcal{X} = \Delta(n)$, we can choose $\kappa = \max\{\|\boldsymbol{x}\|_2 \,|\, \boldsymbol{x} \in \mathcal{X}\} = 1$. Therefore, $\mathcal{C} = \{\alpha(1, \boldsymbol{x}) \,|\, \boldsymbol{x} \in \Delta(n), \alpha \geq 0\}$. For $\boldsymbol{y} = (\tilde{y}, \hat{\boldsymbol{y}}) \in \mathbb{R}^{n+1}$, we have

$$\begin{aligned} \boldsymbol{y} \in \mathcal{C}^\circ &\iff \langle \boldsymbol{y}, \boldsymbol{z} \rangle \leq 0, \ \forall \boldsymbol{z} \in \mathcal{C} \\ &\iff \langle (\tilde{y}, \hat{\boldsymbol{y}}), \alpha(1, \boldsymbol{x}) \rangle \leq 0, \ \forall \boldsymbol{x} \in \Delta(n), \ \forall \alpha \geq 0 \\ &\iff \tilde{y} + \langle \hat{\boldsymbol{y}}, \boldsymbol{x} \rangle \leq 0, \ \forall \boldsymbol{x} \in \Delta(n) \\ &\iff \max_{\boldsymbol{x} \in \Delta(n)} \langle \hat{\boldsymbol{y}}, \boldsymbol{x} \rangle \leq -\tilde{y} \\ &\iff \max_{i=1,\ldots,n} \hat{y}_i \leq -\tilde{y}. \quad \square \end{aligned}$$

Proof of Proposition 1. Let us fix $\tilde{y} \in \mathbb{R}$, and let us first solve

$$\begin{aligned} \min \ & \|\hat{\boldsymbol{y}} - \hat{\boldsymbol{u}}\|_2^2 \\ & \hat{\boldsymbol{y}} \in \mathbb{R}^n, \\ & \max_{i \in [n]} \hat{y}_i \leq -\tilde{y}. \end{aligned} \tag{H.1}$$

This is essentially the projection of $\hat{\boldsymbol{u}}$ on $(-\infty, -\tilde{y}]^n$. So, a solution to (H.1) is $\hat{y}_i(\tilde{y}) = \min\{-\tilde{y}, \hat{u}_i\}$, $\forall i = 1, \ldots, n$. Note that in this case, we have $\hat{\boldsymbol{u}} - \hat{\boldsymbol{y}}(\tilde{y}) = (\hat{\boldsymbol{u}} + \tilde{y}\boldsymbol{e})^+$. So, overall the orthogonal projection on $\mathcal{C}^\circ$ boils down to the optimization of the function $\phi : \mathbb{R} \longmapsto \mathbb{R}_+$ such that

$$\phi : \tilde{y} \longmapsto (\tilde{y} - \tilde{u})^2 + \|(\hat{\boldsymbol{u}} + \tilde{y}\boldsymbol{e})^+\|_2^2. \tag{H.2}$$

In principle, we could use binary search with a doubling trick to compute a ϵ minimizer of the convex function ϕ in $O(\log(\epsilon^{-1}))$ calls to ϕ. However, it is possible to find a minimizer $\tilde{y}^*$ of ϕ using the following remark.

By construction, we know that $\boldsymbol{u} - \pi_{\mathcal{C}^\circ}(\boldsymbol{u}) \in \mathcal{C}$. Here, $\mathcal{C} = \text{cone}(\{1\} \times \Delta(n))$, and $\boldsymbol{u} - \pi_{\mathcal{C}^\circ}(\boldsymbol{u}) = (\tilde{u} - \tilde{y}^*, (\hat{\boldsymbol{u}} + \tilde{y}^*\boldsymbol{e})^+)$. We first check if $\tilde{u} = \tilde{y}^*$. This is the case if and only if $\boldsymbol{u} - \pi_{\mathcal{C}^\circ}(\boldsymbol{u}) = \boldsymbol{0}$ (i.e., if and only if $\boldsymbol{u} \in \mathcal{C}^\circ$), which is straightforward to check using Lemma 2. Now, if $\tilde{u} \neq \tilde{y}^*$, we must have $\tilde{u} > \tilde{y}^*$ by definition of $\mathcal{C}$. This also implies that

$$\frac{(\hat{\boldsymbol{u}} + \tilde{y}^*\boldsymbol{e})^+}{\tilde{u} - \tilde{y}^*} \in \Delta(n),$$

which in turn, implies that

$$\tilde{y}^* + \sum_{i=1}^n \max\{\hat{u}_i + \tilde{y}^*, 0\} = \tilde{u}. \tag{H.3}$$

We can use (H.3) to efficiently compute $\tilde{y}^*$ without using any binary search. In particular, we can sort the coefficients of $\hat{\boldsymbol{u}}$ in $O(n\log(n))$ arithmetic operations and use (H.3) to find $\tilde{y}^*$. □

H.2. Proofs for ℓ_p Balls

Proof of Lemma 3. Let us write $B_p(1) = \{\boldsymbol{z} \in \mathbb{R}^n \,|\, \|\boldsymbol{z}\|_p \leq 1\}$. Here, we consider $\mathcal{X} = B_p(1)$. Recall that $\kappa = \max\{\|\boldsymbol{x}\|_2 \,|\, \boldsymbol{x} \in \mathcal{X}\}$. Therefore, by definition, $\mathcal{C} = \{\alpha(\kappa, \boldsymbol{x}) \,|\, \boldsymbol{x} \in B_p(1), \alpha \geq 0\}$.

We first provide the reformulation for $\mathcal{C}$. Let $\boldsymbol{y} = (\tilde{y}, \hat{\boldsymbol{y}}) \in \mathcal{C}$. Then, $\tilde{y} = \alpha\kappa, \hat{\boldsymbol{y}} = \alpha\boldsymbol{x}$ with $\alpha \geq 0$ and with $\boldsymbol{x}$ such that $\|\boldsymbol{x}\|_p \leq 1$. For $\alpha > 0$, we have $\|\boldsymbol{x}\|_p \leq 1 \iff \|\alpha\boldsymbol{x}\|_p \leq \alpha \iff \|\hat{\boldsymbol{y}}\|_p \leq \tilde{y}/\kappa$.

We now provide the reformulation for $\mathcal{C}^\circ$. Note that for $\boldsymbol{y} = (\tilde{y}, \hat{\boldsymbol{y}}) \in \mathbb{R}^{n+1}$, we have

$$\begin{aligned} \boldsymbol{y} \in \mathcal{C}^\circ &\iff \langle \boldsymbol{y}, \boldsymbol{z} \rangle \leq 0, \ \forall \boldsymbol{z} \in \mathcal{C} \\ &\iff \langle (\tilde{y}, \hat{\boldsymbol{y}}), \alpha(\kappa, \boldsymbol{x}) \rangle \leq 0, \ \forall \boldsymbol{x} \in B_p(1), \ \forall \alpha \geq 0 \\ &\iff \kappa\tilde{y} + \langle \hat{\boldsymbol{y}}, \boldsymbol{x} \rangle \leq 0, \ \forall \boldsymbol{x} \in B_p(1), \\ &\iff \max_{\boldsymbol{x} \in B_p(1),} \langle \hat{\boldsymbol{y}}, \boldsymbol{x} \rangle \leq -\kappa\tilde{y} \\ &\iff \|\hat{\boldsymbol{y}}\|_q \leq -\kappa\tilde{y}, \end{aligned}$$

and $\|\cdot\|_q$ is the dual norm of $\|\cdot\|_p$. □

Proof of Proposition 2. For $p=1$, we have $\|\cdot\|_q = \|\cdot\|_\infty$, and we can choose $\kappa=1$. Let us compute the projection of $(\tilde{u},\hat{\boldsymbol{u}})$ on $\mathcal{C}^\circ$ using the reformulation of Lemma 3:

$$\begin{aligned} &\min\,(\tilde{y}-\tilde{u})^2 + \|\hat{\boldsymbol{y}}-\hat{\boldsymbol{u}}\|_2^2 \\ &\quad \tilde{y}\in\mathbb{R}, \hat{\boldsymbol{y}}\in\mathbb{R}^n, \\ &\quad \|\hat{\boldsymbol{y}}\|_\infty \le -\tilde{y}. \end{aligned} \tag{H.4}$$

For a fixed $\tilde{y}\in\mathbb{R}$, we want to compute $\min\{\|\hat{\boldsymbol{y}}-\hat{\boldsymbol{u}}\|_2^2 \,|\, \hat{\boldsymbol{y}}\in\mathbb{R}^n, \|\hat{\boldsymbol{y}}\|_\infty \le -\tilde{y}\}$. This projection can be computed in closed form as $\hat{\boldsymbol{y}}^*(\tilde{y}) = \min\{-\tilde{y}, \max\{\tilde{y},\hat{\boldsymbol{u}}\}\}$ because this is simply the orthogonal projection of $\hat{\boldsymbol{u}}$ onto the ℓ_∞ ball of radius $-\tilde{y}$. Let us call $\phi:\mathbb{R}\longmapsto\mathbb{R}$ such that

$$\phi(\tilde{y}) = (\tilde{y}-\tilde{u})^2 + \|\hat{\boldsymbol{y}}^*(\tilde{y})-\hat{\boldsymbol{u}}\|_2^2.$$

Note that $\hat{\boldsymbol{y}}^*(\tilde{y})-\hat{\boldsymbol{u}} = (\hat{\boldsymbol{u}}+\tilde{y}\boldsymbol{e})^+$, so we have

$$\phi:\tilde{y}\longmapsto(\tilde{y}-\tilde{u})^2 + \|(\hat{\boldsymbol{u}}+\tilde{y}\boldsymbol{e})^+\|_2^2.$$

Assume that we have ordered the coefficients of $\hat{\boldsymbol{u}}\in\mathbb{R}^n$ in decreasing order. This can be done in $O(n\log(n))$ arithmetic operations. Then, on each of the $n+1$ intervals $\mathcal{I}_1 = (-\infty,-\hat{u}_1), \mathcal{I}_2 = (-\hat{u}_1,-\hat{u}_2),\ldots,\mathcal{I}_{n+1} = (-\hat{u}_n,+\infty)$, the map ϕ is a second-order polynomial in $\tilde{y}$, with a nonnegative coefficient in front of $\tilde{y}^2$. Therefore, for each $i\in[n+1]$, we can find a closed-form expression for the minimum ϕ_i^* of ϕ on $\mathcal{I}_i$ and the scalar $\tilde{y}_i^*$ attaining this minimum. We can then simply search for a global minimum of ϕ among the scalars

$$\{\tilde{y}_i^* \,|\, i\in[n+1]\}\cup\{-\hat{u}_i \,|\, i\in[n]\}.$$

Once we have found $\tilde{y}^*$, the minimizer of ϕ, we obtain the solution of $\pi_{\mathcal{C}^\circ}(\boldsymbol{u})$ as $\pi_{\mathcal{C}^\circ}(\boldsymbol{u}) = (\tilde{y}^*, \hat{\boldsymbol{y}}^*(\tilde{y}))$, and we can recover $\pi_{\mathcal{C}}(\boldsymbol{u})$ from $\pi_{\mathcal{C}}(\boldsymbol{u}) = \boldsymbol{u} - \pi_{\mathcal{C}^\circ}(\boldsymbol{u})$.

Let us now focus on the case $p=\infty$. We know that $\|\cdot\|_1$ and $\|\cdot\|_\infty$ are dual norms to each other. Therefore, from Lemma 3, it is as computationally demanding to compute orthogonal projections onto $\mathcal{C}^\circ$ (when $p=1$) and onto $\mathcal{C}$ (when $p=\infty$). Therefore, the method described in the first part of this proof for computing $\pi_{\mathcal{C}^\circ}(\boldsymbol{u})$ for $p=1$ can be applied for computing $\pi_{\mathcal{C}}(\boldsymbol{u})$ in the case $p=\infty$. □

Proof of Proposition 3. First, we check if $\boldsymbol{u}\in\mathcal{C}$ (i.e., we check if $\|\hat{\boldsymbol{u}}\|_2\le\tilde{u}$). If this is the case, then $\pi_{\mathcal{C}}(\boldsymbol{u}) = \boldsymbol{u}$. Second, we check if $\boldsymbol{u}\in\mathcal{C}^\circ$ (i.e., we check if $\|\hat{\boldsymbol{u}}\|_2\le-\tilde{u}$). If this is the case, then $\pi_{\mathcal{C}}(\boldsymbol{u}) = \boldsymbol{0}$. Else, we have $\|\hat{\boldsymbol{u}}\|_2 > |\tilde{u}|$, and we can provide a closed-form solution to $\pi_{\mathcal{C}}(\boldsymbol{u})$. Let us fix $\tilde{y}\in\mathbb{R}$, and define $\hat{\boldsymbol{y}}^*(\tilde{y})$ as the vector attaining the minimum in $\min\{\|\hat{\boldsymbol{y}}-\hat{\boldsymbol{u}}\|_2^2 \,|\, \hat{\boldsymbol{y}}\in\mathbb{R}^n, \|\hat{\boldsymbol{y}}\|_2\le\tilde{y}\}$. With this notation, we want to find the minimum of $\phi:\mathbb{R}\longmapsto\mathbb{R}$ defined as

$$\phi(\tilde{y}) = (\tilde{y}-\tilde{u})^2 + \|\hat{\boldsymbol{y}}^*(\tilde{y})-\hat{\boldsymbol{u}}\|_2^2.$$

If $\tilde{y}\ge\|\hat{\boldsymbol{u}}\|_2$, then $\hat{\boldsymbol{y}}^*(\tilde{y}) = \hat{\boldsymbol{u}}$. This shows that the minimum of ϕ on $[\|\hat{\boldsymbol{u}}\|_2,+\infty)$ is attained at $\tilde{y}_1 = \|\hat{\boldsymbol{u}}\|_2$ at a value of $\phi(\tilde{y}_1)) = (\|\hat{\boldsymbol{u}}\|_2-\tilde{u})^2$. When $\tilde{y}\in[0,\|\hat{\boldsymbol{u}}\|_2]$, we have $\hat{\boldsymbol{y}}^*(\tilde{y}) = (\tilde{y}/\|\hat{\boldsymbol{u}}\|_2)\hat{\boldsymbol{u}}$. Note that here, $\tilde{y}\longmapsto\hat{\boldsymbol{y}}^*(\tilde{y})$ is differentiable. Therefore, $\phi:\tilde{y}\longmapsto(\tilde{y}-\tilde{u})^2+\|\hat{\boldsymbol{y}}^*(\tilde{y})-\hat{\boldsymbol{u}}\|_2^2$ is also differentiable. The first-order optimality conditions yield a closed-form solution for the minimum of ϕ on $[0,\|\hat{\boldsymbol{u}}\|_2]$, with $\tilde{y}_2 = \frac{\tilde{u}+\|\hat{\boldsymbol{u}}\|_2}{2}$. For this value of $\tilde{y}_2$, we obtain $\phi(\tilde{y}_2) = (1/2)(\|\hat{\boldsymbol{u}}\|_2-\tilde{u})^2$. Therefore, the global minimum of ϕ on $[0,+\infty)$ is attained at $\tilde{y}_2$, yielding

$$\pi_{\mathcal{C}}(\boldsymbol{u}) = \left(\frac{\tilde{u}+\|\hat{\boldsymbol{u}}\|_2}{2}, \frac{\tilde{u}+\|\hat{\boldsymbol{u}}\|_2}{2}\frac{\hat{\boldsymbol{u}}}{\|\hat{\boldsymbol{u}}\|_2}\right). \quad \square$$

H.3. Proofs for Confidence Regions in the Simplex

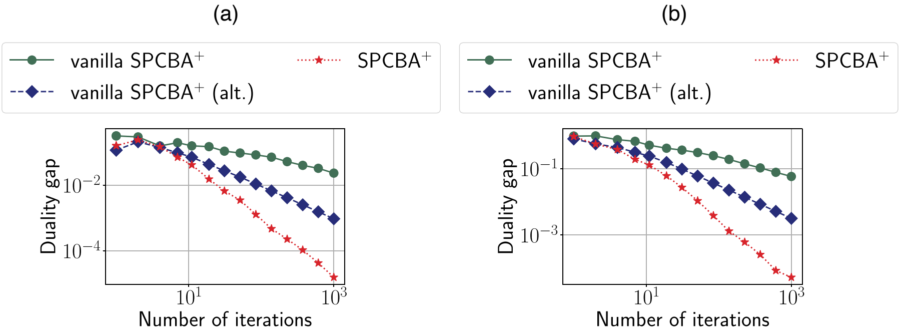
Proof of Proposition 4. We can write $\mathcal{X} = \boldsymbol{x}_0 + \epsilon\tilde{B}$, where $\tilde{B} = \{\boldsymbol{z}\in\mathbb{R}^n \,|\, \boldsymbol{z}^\top\boldsymbol{e} = 0, \|\boldsymbol{z}\|_2\le 1\}$.

Suppose we made a sequence of decisions $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_T$, which can be written as $\boldsymbol{x}_t = \boldsymbol{x}_0+\epsilon\boldsymbol{z}_t$ for $\boldsymbol{z}_t\in\tilde{B}$. Then, it is clear that for any sequence of losses $\boldsymbol{f}_1,\ldots,\boldsymbol{f}_T$, we have

$$\sum_{t=1}^T \omega_t\langle\boldsymbol{f}_t,\boldsymbol{x}_t\rangle - \min_{\boldsymbol{x}\in\mathcal{X}}\sum_{t=1}^T \omega_t\langle\boldsymbol{f}_t,\boldsymbol{x}\rangle = \epsilon_x\left(\sum_{t=1}^T \omega_t\langle\boldsymbol{f}_t,\boldsymbol{z}_t\rangle - \min_{\boldsymbol{z}\in\tilde{B}}\sum_{t=1}^T \omega_t\langle\boldsymbol{f}_t,\boldsymbol{z}\rangle\right). \tag{H.5}$$

Therefore, if we run CBA$^+$ on the set $\tilde{B}$ to obtain $O(\sqrt{T})$ growth of the right-hand side of (H.5), we obtain a no-regret algorithm for $\mathcal{X}$. We now show how to run CBA$^+$ for the set $\tilde{B}$. Let $\mathcal{V} = \{\boldsymbol{v}\in\mathbb{R}^n \,|\, \boldsymbol{v}^\top\boldsymbol{e} = 0\}$. We use the following orthonormal basis of $\mathcal{V}$; let $\boldsymbol{v}_1,\ldots,\boldsymbol{v}_{n-1}\in\mathbb{R}^n$ be the vectors $\boldsymbol{v}_i = \sqrt{i/(i+1)}(1/i,\ldots,1/i,-1,0,\ldots,0)$, $\forall i=1,\ldots,n-1$, where the component $1/i$ is repeated i times. The vectors $\boldsymbol{v}_1,\ldots,\boldsymbol{v}_{n-1}$ are orthonormal and constitute a basis of $\mathcal{V}$ (Egozcue et al. [25]). Writing $\boldsymbol{V} = (\boldsymbol{v}_1,\ldots,\boldsymbol{v}_{n-1})\in\mathbb{R}^{n\times(n-1)}$ and noting that $\boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{I}$, we can write $\tilde{B} = \{\boldsymbol{V}\boldsymbol{s} \,|\, \boldsymbol{s}\in\mathbb{R}^{n-1}, \|\boldsymbol{s}\|_2\le 1\}$. Now, if $\boldsymbol{x} = \boldsymbol{x}_0+\epsilon_x\boldsymbol{z}_t$ with $\boldsymbol{z}_t\in\mathcal{V}$, we have $\boldsymbol{z}_t = \boldsymbol{V}\boldsymbol{s}_t$

Figure I.1. (Color online) Impact of using alternation and linear averaging on the empirical performance of SP-CBA$^+$. (a) Uniform distribution. (b) Normal distribution.



for $s_t \in \mathbb{R}^{n-1}$ and $\|s\|_2 \leq 1$. Finally, $\sum_{t=1}^{T} \omega_t \langle f_t, x_t \rangle - \min_{x \in \mathcal{X}} \sum_{t=1}^{T} \omega_t \langle f_t, x \rangle$ is equal to

$$\epsilon_x \left(\sum_{t=1}^{T} \omega_t \langle V^\top f_t, s_t \rangle - \min_{s \in \mathbb{R}^{n-1}, \|s\|_2 \leq 1} \sum_{t=1}^{T} \omega_t \langle V^\top f_t, s \rangle \right). \tag{H.6}$$

Therefore, to obtain a regret minimizer for (H.6) with observed losses $(f)_{t \geq 1}$, we can run CBA$^+$ on the right-hand side, where the decision set is an ℓ_2 ball and the sequence of observed losses is $(V^\top f_t)_{t \geq 1}$. In the previous section, we showed how to efficiently instantiate CBA$^+$ in this setting (see Proposition 3). □

Remark H.1. In this section, we have highlighted a sequence of reformulations of the regret from (H.5) to (H.6). We essentially showed how to instantiate CBA$^+$ for settings where the decision set $\mathcal{X}$ is the intersection of an ℓ_2 ball with a hyperplane for which we have an orthonormal basis.

Appendix I. Strongest Empirical Setup for SP-CBA$^+$

In this appendix, we empirically highlight the benefits of using alternation and linear averaging (i.e., linear weights on the decisions and uniform weights on the instantaneous payoff vectors) when implementing SP-CBA$^+$ for solving saddle-point problems. To keep things simple, we focus on the simplest instances that we use in our numerical experiments, solving matrix games (Section 5.1) with random distributions for the coefficients of the matrix $A \in \mathbb{R}^{n \times m}$ (uniform or normal), with $(n, m) = (30, 30)$. Figure I.1 shows the performance of three variations of SP-CBA$^+$. The first algorithm is a vanilla implementation (*vanilla* SP-CBA$^+$), where we use CBA$^+$ as a regret minimizer in the repeated game framework, without alternation and with uniform averaging on both the instantaneous payoffs and the decisions. The second algorithm (*vanilla* SP-CBA$^+$ *(alt.)*) improves upon vanilla SP-CBA$^+$ by using alternation and uniform averaging on both the instantaneous payoffs and the decisions. The third algorithm, which we call SP-CBA$^+$ in this figure and in Section 5, corresponds to using CBA$^+$ in the repeated game framework, as well as alternation and linear averaging on the decisions and uniform weights on the instantaneous payoffs. As seen in the figure, SP-CBA$^+$ has by far the strongest empirical performance, and thus, we use alternation and linear averaging in all our simulations in Section 5. The same observations have been made for solving extensive-form games with CFR$^+$, which combines RM$^+$ as a regret minimizer with alternation and linear averaging.

Appendix J. Details on OMD, FTRL, and Optimistic Variants

J.1. Algorithms

For solving our instances of distributionally robust optimization, we compare SP-CBA$^+$ with the following four state-of-the-art algorithms; at iteration $t \geq 1$, for a step size $\eta_t > 0$, the updates are as follows.

1. FTRL (Abernethy et al. [2], McMahan [43]).

$$x_{t+1} \in \arg\min_{x \in \mathcal{X}} \left\langle \sum_{\tau=1}^{t} f_\tau, x \right\rangle + \frac{1}{\eta_t} \|x\|_2^2. \tag{FTRL}$$

OFTRL (Rakhlin and Sridharan [57]). Given estimation m^{t+1} of loss at iteration $t+1$, choose

$$x_{t+1} \in \arg\min_{x \in \mathcal{X}} \left\langle \sum_{\tau=1}^{t} f_\tau + m^{t+1}, x \right\rangle + \frac{1}{\eta_t} \|x\|_2^2. \tag{OFTRL}$$

2. OMD (Beck and Teboulle [6], Nemirovski and Yudin [49]).

$$\boldsymbol{x}_{t+1} \in \min_{\boldsymbol{x}\in\mathcal{X}} \langle \boldsymbol{f}_t, \boldsymbol{x}\rangle + \frac{1}{\eta_t}\|\boldsymbol{x} - \boldsymbol{x}_t\|_2^2. \tag{OMD}$$

OOMD (Chiang et al. [20]). Given estimation $\boldsymbol{m}^{t+1}$ of loss at iteration $t+1$,

$$\begin{aligned} &\boldsymbol{z}_{t+1} \in \min_{\boldsymbol{z}\in\mathcal{X}} \langle \boldsymbol{m}_{t+1}, \boldsymbol{z}\rangle + \frac{1}{\eta_t}\|\boldsymbol{z} - \boldsymbol{x}_t\|_2^2,\\ &\text{Observe the loss } \boldsymbol{f}_{t+1} \text{ related to } \boldsymbol{z}_{t+1},\\ &\boldsymbol{x}_{t+1} \in \min_{\boldsymbol{x}\in\mathcal{X}} \langle \boldsymbol{f}_{t+1}, \boldsymbol{x}\rangle + \frac{1}{\eta_t}\|\boldsymbol{x} - \boldsymbol{x}_t\|_2^2. \end{aligned} \tag{OOMD}$$

Note that these algorithms can be written more generally using Bregman divergence (e.g., Ben-Tal and Nemirovski [7]). We choose to work with $\|\cdot\|_2$ instead of Kullback–Leibler divergence as this ℓ_2 setup is usually associated with faster empirical convergence rates (Chambolle and Pock [19], Gao et al. [29]). Additionally, following Chiang et al. [20] and Rakhlin and Sridharan [57], we use the last observed loss as the predictor for the next loss (i.e., we set $\boldsymbol{m}^{t+1} = \boldsymbol{f}_t$).

J.2. Implementations

The proximal updates defined in the previous section need to be resolved for the decision sets of both players of the distributionally robust optimization Problem (19). We present the details of our implementation here. The results in the rest of this section are reminiscent to the novel tractable proximal setups presented in Grand-Clément and Kroer [31] and Grand-Clément and Kroer [33].

J.2.1. Computing the Projection Steps for the First Player. For $\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^n \,|\, \|\boldsymbol{x} - \boldsymbol{x}_0\|_2 \le \epsilon_x\}$, $\boldsymbol{c}, \boldsymbol{x}' \in \mathbb{R}^n$ and a step size $\eta > 0$, the proximal update becomes

$$\min_{\|\boldsymbol{x}-\boldsymbol{x}_0\|_2 \le \epsilon_x} \langle \boldsymbol{c}, \boldsymbol{x}\rangle + \frac{1}{2\eta}\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2.$$

Using a change of variable, we find that the optimal solution $\boldsymbol{x}^*$ to the problem is

$$\boldsymbol{x}^* = \boldsymbol{x}_0 + \epsilon_x \frac{\boldsymbol{x}' - \eta\boldsymbol{c} - \boldsymbol{x}_0}{\max\{\epsilon_x, \|\boldsymbol{x}' - \eta\boldsymbol{c} - \boldsymbol{x}_0\|_2\}}.$$

J.2.2. Computing the Projection Steps for the Second Player. For $\mathcal{Y} = \{\boldsymbol{y} \in \Delta(m) \,|\, \|\boldsymbol{y} - \boldsymbol{y}_0\|_2 \le \epsilon_y\}$, the proximal update of the second player from a previous point $\boldsymbol{y}'$ and a step size of $\eta > 0$ becomes

$$\min_{\|\boldsymbol{y}-\boldsymbol{y}_0\|_2 \le \epsilon_y,\, \boldsymbol{y}\in\Delta(m)} \langle \boldsymbol{c}, \boldsymbol{y}\rangle + \frac{1}{2\eta}\|\boldsymbol{y} - \boldsymbol{y}'\|_2^2. \tag{J.1}$$

If we dualize the ℓ_2 constraint with a Lagrangian multiplier $\mu \ge 0$, we obtain the relaxed problem $q(\mu)$, where

$$q(\mu) = -(1/2)\epsilon_y^2\mu + \min_{\boldsymbol{y}\in\Delta(m)} \langle \boldsymbol{c}, \boldsymbol{y}\rangle + \frac{1}{2\eta}\|\boldsymbol{y} - \boldsymbol{y}'\|_2^2 + \frac{\mu}{2}\|\boldsymbol{y} - \boldsymbol{y}_0\|_2^2. \tag{J.2}$$

Note that the arg min in

$$\min_{\boldsymbol{y}\in\Delta(m)} \langle \boldsymbol{c}, \boldsymbol{y}\rangle + \frac{1}{2\eta}\|\boldsymbol{y} - \boldsymbol{y}'\|_2^2 + \frac{\mu}{2}\|\boldsymbol{y} - \boldsymbol{y}_0\|_2^2$$

is the same arg min as in

$$\min_{\boldsymbol{y}\in\Delta(m)} \left\| \boldsymbol{y} - \frac{\eta}{\eta\mu + 1}\left(\frac{1}{\eta}\boldsymbol{y}' + \mu\boldsymbol{y}_0 - \boldsymbol{c}\right)\right\|_2^2. \tag{J.3}$$

Note that (J.3) is an orthogonal projection onto the simplex. Therefore, it can be solved efficiently (Duchi et al. [24]). We call $\boldsymbol{y}(\mu)$ an optimal solution of (J.3). Then, $q(\mu)$ can be rewritten

$$q(\mu) = -(1/2)\epsilon_y^2\mu + \langle \boldsymbol{c}, \boldsymbol{y}(\mu)\rangle + \frac{1}{2\eta}\|\boldsymbol{y}(\mu) - \boldsymbol{y}'\|_2^2 + \frac{\mu}{2}\|\boldsymbol{y}(\mu) - \boldsymbol{y}_0\|_2^2.$$

We can therefore binary search $q(\mu)$ as in the previous expression. An upper bound $\overline{\mu}$ for μ^* can be computed as follows. Note that

$$q(\mu) \le -(1/2)\epsilon_y^2\mu + \langle \boldsymbol{c}, \boldsymbol{y}_0\rangle + \frac{1}{2\eta}\|\boldsymbol{y}_0 - \boldsymbol{y}'\|_2^2.$$

Because $\mu \longmapsto q(\mu)$ is concave, we can choose $\overline{\mu}$ such that $q(\overline{\mu}) \leq q(0)$. Using the previous inequality, this yields

$$\overline{\mu} = \frac{2}{\epsilon_y^2}\left(\langle \boldsymbol{c}, \boldsymbol{y}_0\rangle + \frac{1}{2\eta}\|\boldsymbol{y}_0 - \boldsymbol{y}'\|_2^2 - q(0)\right).$$

In our simulations, we search for an optimal μ using the minimize_scalar function from the sklearn Python package, with an accuracy of $\epsilon = 0.001$.

J.3. Computing the Theoretical Fixed Step Sizes for Section 5.3

For OMD and FTRL, in theory (e.g., Ben-Tal and Nemirovski [7]), for a player with decision set $\mathcal{X}$, we can choose $\eta_{\text{th}} = \sqrt{2}\Omega/L\sqrt{T}$ with $\Omega = \max_{x,x'\in\mathcal{X}}\|\boldsymbol{x}-\boldsymbol{x}'\|_2$ and L an upper bound on the norm of any observed loss $\boldsymbol{f}_t$: $\|\boldsymbol{f}_t\|_2 \leq L,\ \forall t \geq 1$. Note that this requires us to know (1) the number of iterations T and (2) the upper bound L on the norm of any observed loss $\boldsymbol{f}_t$ before the losses are generated. For OOMD, we can choose $\eta_{\text{th}} = 1/\sqrt{8}L$ (Syrgkanis et al. [60, corollary 6]), and for OFTRL, we can choose $\eta_{\text{th}} = 1/2L$ (Syrgkanis et al. [60, corollary 8]). We now show how to compute L_x and L_y (for the first player and the second player) for an instance of the distributionally robust logistic regression Problem (19).

1. For the first player, $\boldsymbol{f}_t = \boldsymbol{A}^t\boldsymbol{y}_t$, with $\boldsymbol{A}^t$ the matrix of subgradients of $\boldsymbol{x} \longmapsto F(\boldsymbol{x},\boldsymbol{y}_t)$ at $\boldsymbol{x}_t$:

$$A_{ij}^t = \frac{-b_i a_{i,j}\exp(-b_i\boldsymbol{a}_i^\top\boldsymbol{x}_t)}{1+\exp(-b_i\boldsymbol{a}_i^\top\boldsymbol{x}_t)} + \mu x_j,\ \forall (i,j) \in \{1,\ldots,m\}\times\{1,\ldots,n\}.$$

Therefore, $\|\boldsymbol{f}_t\|_2 \leq \|\boldsymbol{A}^t\|_2\|\boldsymbol{y}_t\|_2 \leq \|\boldsymbol{A}^t\|_2$ because $\boldsymbol{y} \in \Delta(m)$. Now, we have $\|\boldsymbol{A}^t\|_2 \leq \|\boldsymbol{A}^t\|_F = \sqrt{\sum_{i,j}|A_{ij}^t|^2}$. Note that

$$\sqrt{\sum_{i,j}|A_{ij}^t|^2} \leq \sum_{i,j}|A_{ij}^t|.$$

We also have $|A_{ij}^t| \leq |b_i a_{i,j}| + \mu|x_j|$. Recall that we have $\boldsymbol{x} \in \mathbb{R}^n$ such that $\|\boldsymbol{x}-\boldsymbol{x}_0\|_2 \leq \epsilon_x$. We obtain the following upper bound:

$$L_x = \sum_{i,j}|b_i a_{i,j}| + \mu\cdot m\cdot(\|\boldsymbol{x}_0\|_1 + \sqrt{n}\epsilon_x).$$

2. For the second player, the loss $\boldsymbol{f}_t$ is $\boldsymbol{f}_t = (\ell_i(\boldsymbol{x}_t))_{i\in[1,m]}$, with $\ell_i(\boldsymbol{x}) = \log(1+\exp(-b_i\boldsymbol{a}_i^\top\boldsymbol{x}))$. For each $i \in [1,m]$, we have $|\ell_i(\boldsymbol{x})| \leq \log(1+\exp(|b_i|\epsilon_x\|\boldsymbol{a}_i\|_2))$, and we can conclude that

$$L_y = \sqrt{\sum_{i=1}^m \log(1+\exp(|b_i|\epsilon_x\|\boldsymbol{a}_i\|_2))^2}.$$

Appendix K. Computing the Theoretical Step Sizes for Section 5.4

In the saddle-point formulation of MDP, the objective function is $F(\boldsymbol{v},\boldsymbol{\mu}) = (1-\lambda)\boldsymbol{p}_0^\top\boldsymbol{v} + \sum_{s=1}^n\sum_{a=1}^A \mu_{sa}(r_{sa} + \lambda\boldsymbol{P}_{sa}^\top\boldsymbol{v} - v_s)$, for $\boldsymbol{v} \in \mathbb{R}^n, \|\boldsymbol{v}\|_2 \leq \sqrt{n}r_\infty/(1-\lambda)$ and $\mu \in \Delta(n\times A)$. The function F is differentiable, and we have $\nabla_v F(\boldsymbol{v},\boldsymbol{\mu}) \in \mathbb{R}^n, \nabla_\mu F(\boldsymbol{v},\boldsymbol{\mu}) \in \mathbb{R}^{n\times A}$ with

$$\left(\nabla_v F(\boldsymbol{v},\boldsymbol{\mu})\right)_{s'} = (1-\lambda)p_{0s'} + \lambda\sum_{s,a}\mu_{sa}P_{sas'} - \sum_a \mu_{s'a},\ \forall s' \in [n],$$
$$\left(\nabla_\mu F(\boldsymbol{v},\boldsymbol{\mu})\right)_{sa} = r_{sa} + \lambda\boldsymbol{P}_{sa}^\top\boldsymbol{v} - v_s,\ \forall (s,a) \in [n]\times[A].$$

We now provide upper bounds L_v and L_μ on $\|\nabla_v F(\boldsymbol{v},\boldsymbol{\mu})\|_2$ and $\|\nabla_\mu F(\boldsymbol{v},\boldsymbol{\mu})\|_2$. Using the equivalence between $\|\cdot\|_2$ and $\|\cdot\|_1$, we have, for $\boldsymbol{\mu} \in \Delta(n\times A)$,

$$\|\nabla_v F(\boldsymbol{v},\boldsymbol{\mu})\|_2 \leq \|\nabla_v F(\boldsymbol{v},\boldsymbol{\mu})\|_1 \leq (1-\lambda) + \lambda\sum_{s',a,s}\mu_{sa}P_{sas'} + \sum_{s',a}\mu_{s'a} \leq (1-\lambda) + \lambda + 1 \leq 2.$$

For bounding $\|\nabla_\mu F(\boldsymbol{v},\boldsymbol{\mu})\|_2$, using Cauchy–Schwarz's inequality and $\|\boldsymbol{v}\|_2 \leq \sqrt{n}r_\infty/(1-\lambda)$, we have

$$\|\nabla_\mu F(\boldsymbol{v},\boldsymbol{\mu})\|_2 \leq \|\boldsymbol{r}\|_2 + \frac{\sqrt{n}r_\infty}{1-\lambda}(A(\lambda n+1)).$$

Overall, we can choose

$$L_v = 2, L_\mu = \|\boldsymbol{r}\|_2 + \frac{\sqrt{n}r_\infty}{1-\lambda}(A(\lambda n+1)).$$

References

[1] Abernethy J, Bartlett PL, Hazan E (2011) Blackwell approachability and no-regret learning are equivalent. *Proc. 24th Annual Conf. Learn. Theory*, 27–46.

[2] Abernethy JD, Hazan E, Rakhlin A (2008) Competing in the dark: An efficient algorithm for bandit linear optimization. Servedio RA, Zhang T, eds. *Proc. Conf. Learn. Theory (COLT)* (Omnipress), 263–274.

[3] Alagoz O, Hsu H, Schaefer AJ, Roberts MS (2010) Markov decision processes: A tool for sequential decision making under uncertainty. *Medical Decision Making* 30(4):474–483.

[4] Archibald T, McKinnon K, Thomas L (1995) On the generation of Markov decision processes. *J. Oper. Res. Soc.* 46(3):354–361.

[5] Aumann RJ, Maschler M, Stearns RE (1995) *Repeated Games with Incomplete Information* (MIT Press, Cambridge, MA).

[6] Beck A, Teboulle M (2003) Mirror descent and nonlinear projected subgradient methods for convex optimization. *Oper. Res. Lett.* 31(3):167–175.

[7] Ben-Tal A, Nemirovski A (2001) *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, vol. 2 (SIAM, Philadelphia).

[8] Ben-Tal A, Hazan E, Koren T, Mannor S (2015) Oracle-based robust optimization via online learning. *Oper. Res.* 63(3):628–638.

[9] Bertsimas D, Den Hertog D, Pauphilet J (2021) Probabilistic guarantees in robust optimization. *SIAM J. Optim.* 31(4):2893–2920.

[10] Bhatnagar S, Sutton R, Bowling M, Ghavamzadeh M, Lee M (2007) Natural-gradient Actor-Critic algorithms. *Proc. Adv. Neural Inform. Processing Systems* (NeurIPS, San Diego), 19–26.

[11] Blackwell D (1956) An analog of the minimax theorem for vector payoffs. *Pacific J. Math.* 6(1):1–8.

[12] Blackwell D (1954) Controlled random walks. *Proc. Internat. Congress Math.*, vol. 3, 336–338.

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

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

[15] Brown N, Sandholm T (2019) Solving imperfect-information games via discounted regret minimization. *Proc. Conf. AAAI Artificial Intelligence* (AAAI Press, Palo Alto, CA), vol. 33, 1829–1836.

[16] Brown N, Sandholm T (2019) Superhuman AI for multiplayer poker. *Science* 365(6456):885–890.

[17] Burch N, Moravčík M, Schmid M (2019) Revisiting CFR+ and alternating updates. *J. Artificial Intelligence Res.* 64:429–443.

[18] Chambolle A, Pock T (2011) A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vision* 40(1):120–145.

[19] Chambolle A, Pock T (2016) On the ergodic convergence rates of a first-order primal–dual algorithm. *Math. Programming* 159(1–2):253–287.

[20] Chiang C-K, Yang T, Lee C-J, Mahdavi M, Lu C-J, Jin R, Zhu S (2012) Online optimization with gradual variations. *Proc. Conf. Learn. Theory* (COLT), vol. 23, 6.1–6.20.

[21] Chzhen E, Giraud C, Stoltz G (2021) A unified approach to fair online learning via blackwell approachability. *Adv. Neural Inform. Processing Systems*, vol. 34 (NeurIPS, SanDiego), 18280–18292.

[22] Combettes PL, Reyes NN (2013) Moreau's decomposition in Banach spaces. *Math. Programming* 139(1):103–114.

[23] De Rooij S, Van Erven T, Grünwald PD, Koolen WM (2014) Follow the leader if you can, hedge if you must. *J. Machine Learn. Res.* 15(1):1281–1316.

[24] Duchi J, Shalev-Shwartz S, Singer Y, Chandra T (2008) Efficient projections onto the ℓ_1 ball for learning in high dimensions. *Proc. 25th Internat. Conf. Machine Learn.* (PMLR, New York), 272–279.

[25] Egozcue JJ, Pawlowsky-Glahn V, Mateu-Figueras G, Barcelo-Vidal C (2003) Isometric logratio transformations for compositional data analysis. *Math. Geology* 35(3):279–300.

[26] Farina G, Kroer C, Sandholm T (2019) Online convex optimization for sequential decision processes and extensive-form games. *Proc. Conf. AAAI Artificial Intelligence* (AAAI Press, Palo Alto, CA), vol. 33, 1917–1925.

[27] Farina G, Kroer C, Sandholm T (2019) Optimistic regret minimization for extensive-form games via dilated distance-generating functions. *Adv. Neural Inform. Processing Systems* (NeurIPS, San Diego), 5222–5232.

[28] Farina G, Kroer C, Sandholm T (2021) Faster game solving via predictive Blackwell approachability: Connecting regret matching and mirror descent. *Proc. AAAI Conf. on Artificial Intelligence* 35(6):5363–5371.

[29] Gao Y, Kroer C, Goldfarb D (2021) Increasing iterate averaging for solving saddle-point problems. *Proc. Conf. AAAI Artificial Intelligence* (AAAI Press, Palo Alto, CA), vol. 35, 7537–7544.

[30] Goyal V, Grand-Clement J (2023) Robust Markov decision processes: Beyond rectangularity. *Math. Oper. Res.* 48(1):203–226.

[31] Grand-Clément J, Kroer C (2020) First-order methods for Wasserstein distributionally robust MDP. *Internat. Conf. Machine Learn.* (PMLR, New York), 2010–2019.

[32] Grand-Clément J, Kroer C (2021) Conic Blackwell algorithm: Parameter-free convex-concave saddle-point solving. *Adv. Neural Inform. Processing Systems*, vol. 34 (NeurIPS, San Diego), 9587–9599.

[33] Grand-Clément J, Kroer C (2021) Scalable first-order methods for robust MDPs. *Proc. Conf. AAAI Artificial Intelligence* (AAAI Press, Palo Alto, CA), vol. 35, 12086–12094.

[34] Grand-Clément J, Chan CW, Goyal V, Escobar G (2022) Robustness of Proactive Intensive Care Unit Transfer Policies. *Oper. Res.*, ePub ahead of print November 22, https://doi.org/10.1287/opre.2022.2403.

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

[36] Herbster M, Warmuth MK (1998) Tracking the best expert. *Machine Learn.* 32(2):151–178.

[37] Iyengar G (2005) Robust dynamic programming. *Math. Oper. Res.* 30(2):257–280.

[38] Jin Y, Sidford A (2020) Efficiently solving MDPs with stochastic mirror descent. *Internat. Conf. Machine Learn.* (PMLR), 4890–4900.

[39] Kroer C (2020) IEOR8100: Economics, AI, and optimization lecture note 5: Computing Nash equilibrium via regret minimization. Working paper.

[40] Kroer C, Farina G, Sandholm T (2018) Solving large sequential games with the excessive gap technique. *Adv. Neural Inform. Processing Systems* (NeurIPS, San Diego), 864–874.

[41] Kroer C, Peysakhovich A, Sodomka E, Stier-Moses NE (2022) Computing large market equilibria using abstractions. *Oper. Res.* 70(1):329–351.

[42] Kroer C, Waugh K, Kılınç-Karzan F, Sandholm T (2020) Faster algorithms for extensive-form game solving via improved smoothing functions. *Math. Programming* 179:385–417.

[43] McMahan B (2011) Follow-the-regularized-leader and mirror descent: Equivalence theorems and l_1 regularization. *Proc. Fourteenth Internat. Conf. Artificial Intelligence Statist.*, 525–533.

[44] Milman E (2006) Approachable sets of vector payoffs in stochastic games. *Games Econom. Behav.* 56(1):135–147.

[45] Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, et al. (2015) Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.

[46] Moravčík M, Schmid M, Burch N, Lisỳ V, Morrill D, Bard N, Davis T, Waugh K, Johanson M, Bowling M (2017) Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science* 356(6337):508–513.

[47] Namkoong H, Duchi JC (2016) Stochastic gradient methods for distributionally robust optimization with f-divergences. *NIPS*, vol. 29, 2208–2216.

[48] Nemirovski A (2004) Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM J. Optim.* 15(1):229–251.

[49] Nemirovski A, Yudin D (1983) *Problem Complexity and Method Efficiency in Optimization* (Wiley, New York).

[50] Nesterov Y (2005) Smooth minimization of non-smooth functions. *Math. Programming* 103(1):127–152.

[51] Niazadeh R, Golrezaei N, Wang J, Susan F, Badanidiyuru A (2021) Online learning via offline greedy algorithms: Applications in market design and optimization. *Proc. 22nd ACM Conf. Econom. Comput.* (ACM, New York), 737–738.

[52] Orabona F, Pál D (2015) Scale-free algorithms for online linear optimization. *Internat. Conf. Algorithmic Learn. Theory* (Springer), 287–301.

[53] Perchet V (2010) Approachability, calibration and regret in games with partial observations. PhD thesis, Université Pierre et Marie Curie, Paris.

[54] Perchet V (2014) Approachability, regret and calibration: Implications and equivalences. *J. Dynamics Games* 1(2):181–254.

[55] Puterman M (1994) *Markov Decision Processes: Discrete Stochastic Dynamic Programming* (John Wiley & Sons, New York).

[56] Rahimian H, Mehrotra S (2019) Distributionally robust optimization: A review. *Open J. Math. Optim.* 3(2022).

[57] Rakhlin A, Sridharan K (2013) Online learning with predictable sequences. *Conf. Learn. Theory* (PMLR, New York), 993–1019.

[58] Sidford A, Tian K (2018) Coordinate methods for accelerating l_∞ regression and faster approximate maximum flow. *Proc. 2018 IEEE 59th Annual Sympos. Foundations Comput. Sci. (FOCS)* (IEEE, Piscataway, NJ), 922–933.

[59] Steimle LN, Denton BT (2017) Markov decision processes for screening and treatment of chronic diseases. Boucherie R, van Dijk N, eds. *Markov Decision Processes in Practice*, International Series in Operations Research & Management Science, vol. 248 (Springer, Cham, Switzerland), 189–222.

[60] Syrgkanis V, Agarwal A, Luo H, Schapire RE (2015) Fast convergence of regularized learning in games. *Adv. Neural Inform. Processing Systems 28 (NIPS 2015)* (NeurIPS, San Diego).

[61] Tammelin O, Burch N, Johanson M, Bowling M (2015) Solving heads-up limit Texas hold'em. *Proc. Twenty-Fourth Internat. Joint Conf. Artificial Intelligence.*

[62] Tseng P (1995) On linear convergence of iterative methods for the variational inequality problem. *J. Comput. Appl. Math.* 60(1–2):237–252.

[63] von Stengel B (1996) Efficient computation of behavior strategies. *Games Econom. Behav.* 14(2):220–246.

[64] Wei C-Y, Lee C-W, Zhang M, Luo H (2020) Linear last-iterate convergence in constrained saddle-point optimization. *Internat. Conf. Learn. Representations* (ICLR, Appleton, WI).

[65] Wiesemann W, Kuhn D, Rustem B (2013) Robust Markov decision processes. *Math. Oper. Res.* 38(1):153–183.

[66] Zinkevich M, Johanson M, Bowling M, Piccione C (2007) Regret minimization in games with incomplete information. *Adv. Neural Inform. Processing Systems* (NeurIPS, San Diego), 1729–1736.