BREGMAN AUGMENTED LAGRANGIAN METHOD: CONVERGENCE, ACCELERATION, AND APPLICATIONS IN REINFORCEMENT LEARNING

BY

SHEN YAN

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Industrial Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2020

Urbana, Illinois

Adviser:

Prof. Niao He

ABSTRACT

In this thesis, the algorithm Bergman proximal point method (BPP), and its application to Bregman augmented Lagrangian method(BALM) is considered. Unlike classical augmented Lagrangian method (ALM), whose convergence rate and its relation with the proximal point method is well-understood, the convergence rate for BALM has not yet been thoroughly studied in the literature. We analyze, in this thesis, the convergence rates of BALM in terms of the primal objective as well as the feasibility violation. We show that the algorithm can also be applied to variational inequality problems with convex constraints, and fully characterize the iteration complexity of the algorithm derived from the inexact version of BALM. Furthermore, we develop, for the first time, an accelerated Bregman proximal point method, that improves the convergence rate from $\mathcal{O}(1/\sum_{k=0}^{T-1}\eta_k)$ to $\mathcal{O}(1/(\sum_{k=0}^{T-1}\sqrt{\eta_k})^2)$, where $\{\eta_k\}_{k=0}^{T-1}$ is the sequence of proximal parameters. When applied to the dual of convex constrained convex programs, this leads to the construction of an accelerated BALM, that achieves the improved rates for both primal and dual convergences. Finally, numerical experiments comparing the performance of different Bregman divergences as well as the acceleration versions, with applications to Markov decision problems/reinforcement learning are presented at the end.

ACKNOWLEDGMENTS

I would like thank my advisor, Prof. Niao He, for her support and help during my graduate study at UIUC.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION AND BACKGROUND

In this chapter we introduce some notations used through the thesis. After that, we discuss some known results, as the background material, and we use them later to prove our results.

1.1 Convex Minimization Problem

In this thesis we will consider the following convex minimization problem:

$$\min_{x\in\mathcal{X}} \ f(x) \quad \text{s.t. } G(x) \leq 0, \tag{1.1}$$

where $f(x)$ is a closed and convex function, $\mathcal{X} \subset \mathbb{R}^n$ is a closed and convex set, $G(x) := (g^1(x), g^2(x), \cdots, g^m(x))$ and $\{g^i(x)\}_{i=1,2,\cdots,m}$ are closed and convex. The inequality should be understood as element-wise. Such problems occur pervasively in machine learning, signal processing, and many other engineering fields, including basis pursuit, support vector machine, portfolio optimization, and finding the optimal policy for Markov decision problems.

The Lagrangian dual of the previous convex programs can be written in the form of

$$\max_{\lambda\in\Lambda} \ d(\lambda), \ \text{where } d(\lambda) = \min_{x\in\mathcal{X}} \{L(x,\lambda) := f(x) + \lambda^\top G(x)\}, \tag{1.2}$$

where $\Lambda = \mathbb{R}^m_+$. Assuming that the strong duality holds,

Assumption 1.1.1 *There exists optimal (x^*, λ^*) such that the KKT condi-*

tions are satisfied

$$\begin{cases} \langle \nabla f(x^*) + \nabla G(x^*)^\top \lambda^*, x - x^* \rangle \geq 0 \quad \forall x \in \mathcal{X} \\ G(x^*) \leq 0 \\ \lambda_i^* g^i(x^*) = 0, \quad \forall i = 1, \ldots, m \\ \lambda^* \geq 0 \end{cases} \tag{1.3}$$

where (x^*, λ^*) *is a pair of optimal primal dual solution.* $\nabla G(x^*)$ *is the Jacobian matrix of* $G(x^*)$*.*

These conditions are equivalent to the following

$$f(x) - f(x^*) + \langle \lambda^*, \nabla G(x^*)(x - x^*) \rangle - \langle \lambda - \lambda^*, G(x^*) \rangle \geq 0 \quad \forall x \in \mathcal{X}, \lambda \in \Lambda \tag{1.4}$$

By taking $\lambda = 0$, we obtain

$$f(x) - f(x^*) + \langle \lambda^*, G(x) \rangle \geq 0, \quad \forall x \in \mathcal{X}. \tag{1.5}$$

where we have already used the convexity of $G(x)$:

$$G(x) - G(x^*) \geq \nabla G(x^*)(x - x^*)$$

The above inequality can also be derived from the saddle point interpretation of Lagrangian. Since $L(x, \lambda)$ is convex in x and concave(linear) in λ, an optimal KKT point (x^*, λ^*) is also the saddle point of the Lagrangian, i.e. $L(x, \lambda^*) - L(x^*, \lambda) \geq 0, \forall x \in \mathcal{X}, \lambda \in \Lambda$, which by setting $\lambda = 0$ also gives Eqn (1.5).

1.2 Convex Variational Inequality Problem

We will also consider in this thesis a specific form of convex variational inequality problem, which is also often called convex generalized Nash equilibrium problem (GNE). Let $Z_i \subseteq \mathbb{R}^{n_i}, i = 1, \ldots, m$, and $F_i(z) = F_i(z^i, z^{-i}) : \mathbb{R}^{n_1 + \cdots + n_m} \to \mathbb{R}$, where z^{-i} represents all $(z^1, \ldots, z^{i-1}, z^{i+1}, \ldots, z^m)$, $z^i \in \mathbb{R}^{n_i}$. Each $F_i(z)$ can be thought of as the loss function of player i taking action z^i given others take actions z^{-i}. The GNE problem we consider is to find

$z_* \in Z := Z^1 \times \cdots \times Z^m$ such that

$$\max_{x \in Z \cap \mathcal{G}} \epsilon(z_*; x) := \sum_{i=1}^{m} \left[F_i(z_*^i, z_*^{-i}) - F_i(x^i, z_*^{-i}) \right] = 0, \tag{1.6}$$

where $\mathcal{G} := \{z | G(z) := (g^1(z), \ldots, g^l(z)) \leq 0\}$. The intuition behind this is clear: for an optimal solution z_*, player i's action z_*^i should be optimal given all other players choosing z_*^{-i}.

We make the following assumptions.

Assumption 1.2.1 *$Z := Z^1 \times \cdots \times Z^m$ is convex and compact, which guarantees the existence of a Nash equilibrium.*

Assumption 1.2.2 *Each $g^j(z)$ is convex in z.*

Assumption 1.2.3 *Each $F_i(z^i, z^{-i})$ is convex in z^i. Additionally, $F_i(z^i, z^{-i})$ is concave in z^{-i}, and $\sum_{i=1}^m F_i(z)$ is convex in z. Compared with normal convex NE we have two extra assumptions to make sure that $\max_x \epsilon(z; x)$ is convex in z. These assumptions seem necessary when deriving non-asymptotic convergence rate as discussed in [1].*

If we define a point-to-set operator by

$$H(z) = [H^1(z), H^2(x), \ldots, H^m(z)], \quad H^i(z) \in \partial_{z^i} F_i(z^i, z^{-i}), \quad i = 1 \ldots, m$$

By the convexity F, $H(z)$ is a monotone operator. Then the above problem is equivalent to solving the strong variational inequality(strong VI) problem: find $z_* \in Z \cap \mathcal{G}$ such that

$$\max_{x \in Z \cap \mathcal{G}} \langle H(z_*), z_* - x \rangle = 0, \tag{1.7}$$

Under the certain conditions such as continuity of H, the above strong VI is equivalent to the following weak VI [2]: find $z_* \in Z \cap \mathcal{G}$ such that

$$\max_{x \in Z \cap \mathcal{G}} \langle H(x), z_* - x \rangle = 0,$$

Assumption 1.2.4 *There exists Lagrangian multiplier $\lambda^* \geq 0$ such that*

$$\begin{cases} \langle H(z_*) + \nabla G(z_*)^\top \lambda^*, x - z_* \rangle \geq 0, \quad \forall x \in Z \\ G(z_*) \leq 0 \\ \lambda_i^* g^i(z_*) = 0, \quad \forall i = 1, \ldots, m \\ \lambda^* \geq 0 \end{cases} \tag{1.8}$$

Similar to the minimization case, there is also a equivalent form, namely

$$\langle H(z_*), x - z_* \rangle + \langle \lambda^*, \nabla G(z_*)(x - z_*) \rangle - \langle G(z_*), \lambda - \lambda^* \rangle \geq 0, \quad \forall x \in Z, \lambda \geq 0 \tag{1.9}$$

which again implies

$$\langle H(z_*), x - z_* \rangle + \langle \lambda^*, G(x) \rangle \geq 0, \quad \forall x \in Z \tag{1.10}$$

which actually also implies

$$-\epsilon(z_*; x) + \langle \lambda^*, G(x) \rangle \geq 0, \quad \forall x \in Z \tag{1.11}$$

A special case of such problem is convex-concave saddle point problem with joint constraints:

$$\begin{aligned} &\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} f(x, y) \\ &\quad \text{s.t.} \quad G(x, y) \leq 0 \end{aligned} \tag{1.12}$$

which have application to distributionally robust optimization [3, 4, 5, 6, 7].

1.3 Bregman Functions and Divergences

Let h be a proper, continuously differentiable, and strictly convex function on $\Lambda \subseteq \mathbb{R}^m$. The Bregman divergence induced by function h is given as follows:

$$D_h(\lambda, \tilde{\lambda}) := h(\lambda) - h(\tilde{\lambda}) - \nabla h(\tilde{\lambda})^\top (\lambda - \tilde{\lambda}) \quad \forall \lambda \in \Lambda, \tilde{\lambda} \in \Lambda.$$

By strict convexity, $D_h(\lambda, \tilde{\lambda}) \geq 0$, and $D_h(\lambda, \tilde{\lambda}) = 0$ only when $\lambda = \tilde{\lambda}$. For example, when $\Lambda = \mathbb{R}^m$, the simplest choice of Bregman divergence is $D_h(\lambda, \tilde{\lambda}) = \frac{1}{2}\|\lambda - \tilde{\lambda}\|_2^2$, where $h(\lambda) = \frac{1}{2}\|\lambda\|_2^2$; when $\Lambda = \mathbb{R}_+^m$, a common choice of Bregman divergence is $D_h(\lambda, \tilde{\lambda}) = \sum_i \left(\lambda^{(i)} \log \lambda^{(i)} - \lambda^{(i)} \log \tilde{\lambda}^{(i)} - \lambda^{(i)} + \tilde{\lambda}^{(i)}\right)$, where $h(\lambda) = \sum_i \lambda^{(i)} \log \lambda^{(i)}$. We also list the well-known three-point identity property [8] of Bregman divergence, which will be heavily used in the analysis: $\forall \lambda_1, \lambda_2, \lambda_3 \in \Lambda$,

$$D_h(\lambda_1, \lambda_2) + D_h(\lambda_2, \lambda_3) - D_h(\lambda_1, \lambda_3) = \langle \nabla h(\lambda_2) - \nabla h(\lambda_3), \lambda_2 - \lambda_1 \rangle. \quad (1.13)$$

See [9, 8, 10] for a more detailed discussion on Bregman divergences.

CHAPTER 2

ANALYSIS OF BALM

2.1 Minimization Problems

2.1.1 Existing Works on Classical ALM

The classical augmented Lagrangian method (ALM), originally introduced in [11, 12], has been one of the most fundamental and popular algorithms for solving problems with linearly constrained convex programs; see e.g, [13] for a comprehensive overview. Particularly for (1.1), the key steps for ALM are as simple as follows:

$$\begin{cases} x_{k+1} \in \arg\min_{x\in\mathcal{X}}\{f(x) + \frac{1}{2\eta_k}\|[\lambda_k + \eta_k G(x)]_+\|_2^2\} \\ \lambda_{k+1} = [\lambda_k + \eta_k G(x_{k+1})]_+ \end{cases} \tag{2.1}$$

where η_k is the proximal parameter at step k and $[x]_+$ stands for taking $\max(x, 0)$ for each element. It is well-known that ALM can be interpreted as applying the proximal point method on the Lagrangian dual [14]. Defining the Lagrange function as $L(x, \lambda) = f(x) + \langle\lambda, G(x)\rangle$, the proximal minimization perspective allows us to write the ALM in a simple way:

$$(x_{k+1}, \lambda_{k+1}) \in \operatorname*{argmax}_{\lambda\in\Lambda} \min_{x\in\mathcal{X}} \left\{ L(x, \lambda) - \frac{1}{2\eta_k}\|\lambda - \lambda_k\|_2^2 \right\}, \tag{2.2}$$

The convergence of ALM has been extensively studied in the literature; due to the large volume of literature on this topic, we only list a few results here. The asymptotic convergence in the convex case was provided in [14] from the proximal minimization viewpoint. Understanding the non-asymptotic convergence of ALM and the iteration complexity of its inexact variations has been the main focus in several recent works. For example, for the linear equality constrained problem, [15] analyzed the convergence rate

for the primal problem, assuming the subproblems are only approximately solved through some first-order subroutines. [16] generalized the results to include both equality and inequality constrained problems. There also exist recent works applying ALM to non-convex problems [17, 18]. Moreover, analysis for other variants of ALM also exist, e.g., the linearized augmented Lagrangian method [19, 20].

As for the acceleration of the ALM, there currently exist two such schemes, which can be derived from applying Güler's 1st and 2nd accelerated proximal point methods [21, 22] to the dual problem, respectively. See [23, 24, 25, 26] and [27, 28, 29] for more details on each scheme. Most of these works only proved an accelerated convergence rate of the dual problem, instead of the primal convergence. For example, [27, 28, 29] applied the Güler's 2nd accelerated scheme to the dual problem, and showed that the Lagrangian residual satisfies $L(x^*, \lambda^*) - L(x_T, \lambda_T) \leq \mathcal{O}(1/T^2)$, where (x^*, λ^*) corresponds to the optimal solution and Lagrange multiplier. Notice that this only implies an accelerated rate for the dual convergence. In fact, generally speaking, this algorithm could fail to attain the same accelerated $\mathcal{O}(1/T^2)$ rate in terms primal convergence.

To make our point clear, we summarize different acceleration schemes in the following. In particular, we consider applying accelerated ALMs to a linear equality constrained problem

$$\min_{x \in \mathcal{X}} \ f(x) \quad \text{s.t. } Ax = b,$$

Existing accelerated ALM methods

1. Applying Güler's 1st accelerated proximal point method to the Lagrangian dual [21, 22, 23, 30, 25]:

$$\begin{cases} x_{k+1} = \arg\min_{x \in \mathcal{X}} \{ f(x) + y_k^\top (Ax - b) + \frac{\eta}{2} \|Ax - b\|^2 \} \\ \lambda_{k+1} = y_k + \eta(Ax_{k+1} - b) \\ t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2} \\ y_{k+1} = \lambda_{k+1} + \frac{t_k - 1}{t_{k+1}} (\lambda_{k+1} - \lambda_k) \end{cases} \tag{2.3}$$

2. Applying Güler's 2nd accelerated proximal point method to the La-

grangian dual [21, 22, 27, 28, 29]:

$$\begin{cases} x_{k+1} = \arg\min_{x\in\mathcal{X}}\{f(x) + y_k^\top(Ax-b) + \frac{\eta}{2}\|Ax-b\|^2\} \\ \lambda_{k+1} = y_k + \eta(Ax_{k+1}-b) \\ t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2} \\ y_{k+1} = \lambda_{k+1} + \frac{t_k-1}{t_{k+1}}(\lambda_{k+1}-\lambda_k) + \frac{t_k}{t_{k+1}}(\lambda_{k+1}-y_k) \end{cases} \tag{2.4}$$

3. Applying Nesterov's accelerated dual average method to the augmented Lagrangian dual problem [31, 32, 26]:

$$\begin{cases} x_{k+1} = \arg\min_{x\in\mathcal{X}}\{f(x) + y_k^\top(Ax-b) + \frac{\eta}{2}\|Ax-b\|^2\} \\ \lambda_{k+1} = y_k + \eta(Ax_{k+1}-b) \\ t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2} \\ y_{k+1} = \left(1-\frac{1}{t_{k+1}}\right)\lambda_{k+1} + \frac{1}{t_{k+1}}\left(\lambda_0 + \eta\sum_{j=0}^k t_j(Ax_{j+1}-b)\right) \end{cases} \tag{2.5}$$

Equivalence of (2.3) and (2.5) The last step of (2.5) can be rewritten as the following (with $v_0 = y_0$)

$$\begin{cases} v_{k+1} = v_k + t_k(\lambda_{k+1}-y_k) \\ y_{k+1} = \left(1-\frac{1}{t_{k+1}}\right)\lambda_{k+1} + \frac{1}{t_{k+1}}v_{k+1} \end{cases}$$

Eliminate v_k and using the fact that $v_k = t_k y_k - (t_k-1)\lambda_k$, we have $v_{k+1} = v_k + t_k(\lambda_{k+1}-y_k)$. This implies that

$$t_{k+1}y_{k+1} - (t_{k+1}-1)\lambda_{k+1} = t_k y_k - (t_k-1)\lambda_k + t_k(\lambda_{k+1}-y_k).$$

Equivalent, $y_{k+1} = \lambda_{k+1} + \frac{t_k-1}{t_{k+1}}(\lambda_{k+1}-\lambda_k)$, which recovers last step of (2.3).

Primal convergence rate of (2.4) Existing proofs only indicate an accelerated rate of dual convergence, yet it remains unclear whether this method could guarantee an improved rate of primal acceleration at the same time. Here we provide a simple counter-example showing that the algorithm described in (2.4) could fail to improve the primal convergence. The example we consider is a simple linear program:

$$\min_{x\in\mathbb{R}^n}\{c^\top x : Ax = b\},$$

where we assume $A \in \mathbb{R}^{m\times n}$, $m > n$, and $\text{rank}(A) = n$. The corresponding dual problem is

$$\max_{\lambda\in\mathbb{R}^m}\{-\lambda^\top b : A^\top\lambda + c = 0\}.$$

Assume that the problem is feasible, i.e., $b \in \text{range}(A)$. This implies that there is only one feasible solution, which we shall call x^*. Clearly, $x^* = (A^\top A)^{-1}A^\top b$, and $c^\top x^* = c^\top(A^\top A)^{-1}A^\top b$. Now (2.4) for this specific problem can be written as (with $t_0 = 1, y_0 = \mathbf{0}$)

$$\begin{cases} x_{k+1} = (A^\top A)^{-1}(A^\top b - \frac{1}{\eta}(A^\top y_k + c)) \\ \lambda_{k+1} = y_k + \eta(Ax_{k+1} - b) \\ t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2} \\ y_{k+1} = \lambda_{k+1} + \frac{t_k-1}{t_{k+1}}(\lambda_{k+1} - \lambda_k) + \frac{t_k}{t_{k+1}}(\lambda_{k+1} - y_k) \end{cases}.$$

By substituting x_{k+1}, we get

$$\lambda_{k+1} = (I - A(A^\top A)^{-1}A^\top)y_k - A(A^\top A)^{-1}c,$$

from this expression we can see that the dual variable λ_1 already recovers the optimal solution, since $A^\top\lambda_1 + c = 0$, and $-b^\top\lambda_1 = c^\top(A^\top A)^{-1}A^\top b$. We further obtain the update for y_{k+1}:

$$\begin{aligned} y_{k+1} = (I - A(A^\top A)^{-1}A^\top)\left(y_k + \frac{t_k - 1}{t_{k+1}}(y_k - y_{k-1})\right) - A(A^\top A)^{-1}c \\ - \frac{t_k}{t_{k+1}}\left(A(A^\top A)^{-1}A^\top y_k + A(A^\top A)^{-1}c\right). \end{aligned}$$

To simplify notations, let $z_k = A^\top y_k + c$, we therefore have

$$z_{k+1} = -\frac{t_k}{t_{k+1}}z_k = (-1)^{k+1}\frac{t_0}{t_{k+1}}c.$$

We can then obtain the following update for x_{k+1},

$$\begin{aligned} x_{k+1} &= (A^\top A)^{-1}(A^\top b - \frac{1}{\eta}(A^\top y_k + c)) \\ &= x^* - \frac{1}{\eta}(A^\top A)^{-1} z_k \\ &= x^* - \frac{(-1)^k (A^\top A)^{-1} c}{\eta} \frac{t_0}{t_k}. \end{aligned}$$

This leads to the primal optimality and feasibility gap,

$$\begin{aligned} |c^\top (x^* - x_T)| &= \frac{c^\top (A^\top A)^{-1} c}{\eta} \frac{t_0}{t_{T-1}}, \\ \|Ax_T - b\| &= \frac{\|A(A^\top A)^{-1} c\|}{\eta} \frac{t_0}{t_{T-1}}. \end{aligned}$$

Since $\frac{T+1}{2} \leq t_T \leq T+1$, for this problem, (2.4) only achieves $\mathcal{O}(1/T)$ primal convergence rate.

The formulation Eqn (2.2) naturally leads to the generalization of *Bregman Augmented Lagrangian Method* (we refer to BALM for short), where the Euclidean distance is replaced with a general Bregman divergence. This can also be seen as a direct application of the Bregman proximal point algorithm (BPP) to the dual problem, which was originally introduced in [9, 33]. Let $h(\lambda) : \text{int}(\Lambda) \to \mathbb{R}$ be a strictly convex, continuously differential function on the interior set $\text{int}(\Lambda)$. The Bregman divergence induced by h is given by $D_h(\lambda, \lambda') = h(\lambda) - h(\lambda') - \langle \nabla h(\lambda'), \lambda - \lambda' \rangle$, which is nonnegative and strictly convex. Thus, the key steps of BALM can be viewed as follows:

$$(x_{k+1}, \lambda_{k+1}) \in \underset{\lambda \in \Lambda}{\text{argmax}} \min_{x \in \mathcal{X}} \left\{ L(x, \lambda) - \frac{1}{\eta_k} D_h(\lambda, \lambda_k) \right\}. \tag{2.6}$$

One of the most important advantages of using a Bregman divergence as opposed to the Euclidean distance is that the objective of the subproblems becomes twice-differentiable [34]. The use of Bregman divergence also allows more flexibility to exploit the geometry of dual domain Λ, especially for the linear inequality constrained case. The advantages of BALM have also been observed empirically in practice; see e.g. [35] for image segmentation applications.

However, on the theoretical side, the convergence of BALM has only been

studied in few works. The asymptotic convergence is proven in [36, 8] and [10] when considering generalized Bregman functions. To the authors' knowledge, the non-asymptotic convergence rate of BALM is still absent in the literature, especially in terms of the original objective and constraint violation of the primal sequences. Moreover, while accelerated BALM and accelerated proximal point algorithm [21] has been established in the Euclidean setting, it remains unclear whether such acceleration schemes can be extended to BALM with general Bregman divergences and whether faster convergence rates can be achieved, especially in terms of the primal convergence.

Next, we explicitly define BPP and BALM, then summarize some existing results on BPP and BALM.

Algorithm 1: Bregman Augmented Lagrangian Method (BALM)

Input: $\lambda_0 \in \Lambda, \{\eta_k\}_{k\geq 0}$

for $k \geq 0$ **do**

$x_{k+1} \in \arg\min_{x\in\mathcal{X}}\{f(x) + \max_{\lambda\in\Lambda}\{\lambda^\top G(x) - \frac{1}{\eta_k}D_h(\lambda,\lambda_k)\}\}$

$\lambda_{k+1} = \arg\max_{\lambda\in\Lambda}\{\lambda^\top G(x_{k+1}) - \frac{1}{\eta_k}D_h(\lambda,\lambda_k)\}$

Specifically, the Bregman proximal point (BPP) method [9, 8] for solving the Lagrange dual problem follows the recursion:

$$\lambda_{k+1} := \arg\max_{\lambda\in\Lambda}\left\{d(\lambda) - \frac{1}{\eta_k}D_h(\lambda,\lambda_k)\right\}. \tag{2.7}$$

The operation defined in (2.7) is also known as the *Bregman proximal operator*. Recall that $d(\lambda) = \min_{x\in\mathcal{X}}\{f(x) + \lambda^\top G(x)\}$ is the Lagrange dual function. Solving (2.7) reduces to solving the convex-concave saddle point problem

$$\max_{\lambda\in\Lambda}\min_{x\in\mathcal{X}}\left\{\Phi_{\eta_k}(x,\lambda;\lambda_k) := f(x) + \lambda^\top G(x) - \frac{1}{\eta_k}D_h(\lambda,\lambda_k)\right\}. \tag{2.8}$$

Assuming that both f and h are coercive, based on convex analysis theory [37, 38], problem (2.8) possesses a saddle point, denoted as (x_{k+1},λ_{k+1}), such that

$$\Phi_{\eta_k}(x_{k+1},\lambda;\lambda_k) \leq \Phi_{\eta_k}(x_{k+1},\lambda_{k+1};\lambda_k) \leq \Phi_{\eta_k}(x,\lambda_{k+1};\lambda_k),$$

for any $x\in\mathcal{X}, \lambda\in\Lambda$. Thus, $\lambda_{k+1} = \operatorname{argmax}_{\lambda\in\Lambda}\Phi_{\eta_k}(x_{k+1},\lambda;\lambda_k)$. The Breg-

man ALM (BALM), described in Algorithm 1, can be interpreted as iteratively computing the saddle point of the sequence of subproblems (2.8), or consequently, computing the Bregman proximal operator (2.7). Particularly, when setting $h(\lambda) = \frac{1}{2}\|\lambda\|_2^2$, this leads to the classical ALM; when $h(\lambda) = \sum_{i=i}^{m} \lambda_i \log(\lambda_i)$ leads to the exponential multiplier method [39]. For various other examples of BALM, please see [33, 40] and references therein. Finally, we point out that the convergence analysis of BPP in (2.7) has been well-studied [9, 8]. We list some results below for later references.

Lemma 2.1.1 ([8]) *Let $\{\lambda_k\}_{k\geq 0}$ be a sequence generated from (2.7) with positive parameters $\{\eta_k\}_{k\geq 0}$. Let $\lambda^* \in \Lambda$ be an optimal solution to (1.2). The following holds:*

(a) $\eta_k(d(\lambda) - d(\lambda_{k+1})) \leq D_h(\lambda, \lambda_k) - D_h(\lambda, \lambda_{k+1}) - D_h(\lambda_{k+1}, \lambda_k)$, $\forall \lambda \in \Lambda$;

(b) $d(\lambda_k)$ *is non-decreasing;*

(c) $D_h(\lambda^*, \lambda_k)$ *is non-increasing;*

(d) $d(\lambda^*) - d(\lambda_T) \leq \frac{D_h(\lambda^*, \lambda_0)}{\sum_{k=0}^{T-1} \eta_k}$; *if* $\sum_{k=0}^{\infty} \eta_k = \infty$, *then* $d(\lambda_k) \to d(\lambda^*)$ *as* $k \to \infty$.

The result (a) can be obtained directly from the optimality condition of (2.7) and the three-point identity of the Bregman divergence; (b) follows from (a) by setting $\lambda = \lambda_k$; (c) follows from (a) by setting $\lambda = \lambda^*$; and (d) can be obtained by taking the telescoping sum over (a). Moreover, it can be shown that the sequence $\{\lambda_k\}_{k\geq 0}$ converges to some optimal solution λ^*. Note that the above results hold true for general convex problems in the form of $\max_{\lambda\in\Lambda} d(\lambda)$.

Lemma (2.1.1) immediately implies the convergence of the dual sequence of BALM for solving the constrained convex programs. However, establishing the convergence of the primal sequence, both in terms of the optimality and feasibility, still remains elusive.

2.1.2 Ergodic convergence of BALM

The next lemma characterizes the one-step behavior of the algorithm,

Lemma 2.1.2 *We have for any $x \in \mathcal{X}, \lambda \in \Lambda$,*

$$L(x_{k+1}, \lambda) - L(x, \lambda_{k+1}) \leq \frac{1}{\eta_k}(\nabla h(\lambda_{k+1}) - \nabla h(y_k))^\top (\lambda - \lambda_{k+1}). \tag{2.9}$$

Moreover,

$$L(x_{k+1}, \lambda) - L(x, \lambda_{k+1}) \leq \frac{1}{\eta_k}(D_h(\lambda, \lambda_k) - D_h(\lambda, \lambda_{k+1})). \tag{2.10}$$

Proof. Since BALM is equivalent to solving a convex-concave saddle point problem (2.8) each step, we have the following optimality conditions

$$(x - x_{k+1})^\top (g_{k+1} + \nabla G(x_{k+1})^\top \lambda_{k+1}) \geq 0, \quad \forall x \in \mathcal{X},\ g_{k+1} \in \partial f(x_{k+1}) \tag{2.11}$$

$$(\lambda_{k+1} - \lambda)^\top \left(G(x_{k+1}) - \frac{1}{\eta_k}(\nabla h(\lambda_{k+1}) - \nabla h(\lambda_k)) \right) \geq 0, \quad \forall \lambda \in \Lambda. \tag{2.12}$$

Therefore, we have

$$\begin{aligned}
&L(x_{k+1}, \lambda) - L(x, \lambda_{k+1}) \\
=&f(x_{k+1}) - f(x) + \lambda^\top G(x_{k+1}) - {\lambda_{k+1}}^\top G(x) \\
=&f(x_{k+1}) - f(x) + \lambda_{k+1}^\top G(x_{k+1}) - {\lambda_{k+1}}^\top G(x) + \lambda^\top G(x_{k+1}) - \lambda_{k+1}^\top G(x_{k+1}) \\
\leq&f(x_{k+1}) - f(x) + (x_{k+1} - x)^\top \nabla G(x_{k+1})^\top \lambda_{k+1} - (\lambda_{k+1} - \lambda)^\top G(x_{k+1}).
\end{aligned}$$

Invoking the convexity of $f(x)$ and (2.11), it follows that

$$\begin{aligned}
&f(x_{k+1}) - f(x) + (x_{k+1} - x)^\top \nabla G(x_{k+1})^\top \lambda_{k+1} \\
\leq&(x_{k+1} - x)^\top g_{k+1} + (x_{k+1} - x)^\top \nabla G(x_{k+1})^\top \lambda_{k+1} \\
=&(x_{k+1} - x)^\top (g_{k+1} + \nabla G(x_{k+1})^\top \lambda_{k+1}) \leq 0.
\end{aligned}$$

For the second part, we have

$$\begin{aligned}
-(\lambda_{k+1} - \lambda)^\top G(x_{k+1}) &\leq \frac{1}{\eta_k}(\lambda - \lambda_{k+1})^\top (\nabla h(\lambda_{k+1}) - \nabla h(\lambda_k)) \\
&= \frac{1}{\eta_k}(D_h(\lambda, \lambda_k) - D_h(\lambda, \lambda_{k+1}) - D_h(\lambda_{k+1}, \lambda_k)) \\
&\leq \frac{1}{\eta_k}(D_h(\lambda, \lambda_k) - D_h(\lambda, \lambda_{k+1})).
\end{aligned}$$

where the first inequality uses (2.12), and the second equality uses the three-point identity of Bregman divergence. Summing up the above two inequalities leads to the desired result. ∎

Denote the candidate solution

$$\tilde{x}_T = \frac{\sum_{k=0}^{T-1}\eta_k x_{k+1}}{\sum_{k=0}^{T-1}\eta_k},\ \tilde{\lambda}_T = \frac{\sum_{k=0}^{T-1}\eta_k \lambda_{k+1}}{\sum_{k=0}^{T-1}\eta_k}.$$

From Lemma 2.1.2, we can immediately obtain the following result.

Lemma 2.1.3 *We have*

$$L(\tilde{x}_T, \lambda) - L(x, \tilde{\lambda}_T) \leq \frac{D_h(\lambda, \lambda_0)}{\sum_{k=0}^{T-1}\eta_k}, \quad \forall x \in \mathcal{X}, \lambda \in \Lambda. \tag{2.13}$$

Moreover, by setting $x = x^*$, we further have

$$f(\tilde{x}_T) - f(x^*) + \lambda^\top G(\tilde{x}_T) \leq \frac{D_h(\lambda, \lambda_0)}{\sum_{k=0}^{T-1}\eta_k}, \quad \forall \lambda \in \Lambda. \tag{2.14}$$

Proof. To obtain (2.13), we see that

$$\begin{aligned} L(\tilde{x}_T, \lambda) - L(x, \tilde{\lambda}_T) \leq & \frac{1}{\sum_{k=0}^{T-1}\eta_k} \sum_{k=0}^{T-1} \eta_k \left(L(x_{k+1}, \lambda) - L(x, \lambda_{k+1})\right) \\ \leq & \frac{1}{\sum_{k=0}^{T-1}\eta_k} \sum_{k=0}^{T-1} (D_h(\lambda, \lambda_k) - D_h(\lambda, \lambda_{k+1})) \\ = & \frac{1}{\sum_{k=0}^{T-1}\eta_k} \left(D_h(\lambda, \lambda_0) - D_h(\lambda, \lambda_T)\right) \end{aligned}$$

The first step uses the fact that $L(x, \lambda)$ is convex in x and linear in λ, and the second step uses Lemma 2.1.2. Combining the fact that $D_h(\lambda, \lambda_T) \geq 0$, we end up with (2.13). Setting $x = x^*$, the result in (2.14) follows based on the fact that $\tilde{\lambda}_T^\top G(x^*) \leq 0$. ∎

The following theorem describes the primal convergence rate of BALM applied to the convex constrained problems Eqn (1.1), both in terms of the optimality and the constraint violation.

Theorem 2.1.1 *Define $\rho_* = 2\|\lambda^*\| + 1$. Algorithm 1 satisfies that for prob-*

lem (1.1)

$$\max\{|f(\tilde{x}_T) - f(x^*)|, \|[G(\tilde{x}_T)]_+\|_\star\} \leq \frac{\max_{\lambda\in\mathcal{B}_{\rho_*}^+} D_h(\lambda, \lambda_0)}{\sum_{k=0}^{T-1}\eta_k}, \tag{2.15}$$

where $\mathcal{B}_\rho^+ = \{\lambda \in \mathbb{R}^m : \lambda \geq 0, \|\lambda\| \leq \rho\}$ *and* $[x]_+ = \max(x, 0)$.

Proof. We only focus on the proof for the inequality constrained case. The equality constrained case follows similarly. Setting $\lambda = 0$ in (2.14) implies

$$f(\tilde{x}_T) - f(x^*) \leq \frac{D_h(0, \lambda_0)}{\sum_{k=0}^{T-1}\eta_k}.$$

Taking maximum over $\lambda \in \mathcal{B}_\rho^+$ leads to

$$f(\tilde{x}_T) - f(x^*) + \rho\|[G(\tilde{x}_T)]_+\|_\star \leq \frac{\max_{\lambda\in\mathcal{B}_\rho^+} D_h(\lambda, \lambda_0)}{\sum_{k=0}^{T-1}\eta_k}, \forall\rho > 0. \tag{2.16}$$

Plugging in $x = \tilde{x}_T$ into Eqn (1.5), we have

$$f(x^*) - f(\tilde{x}_T) - \|\lambda^*\|\|[G(\tilde{x}_T)]_+\|_\star \leq f(x^*) - f(\tilde{x}_T) - {\lambda^*}^\top G(\tilde{x}_T) \leq 0. \tag{2.17}$$

Now summing together (2.16) and (2.17), we obtain

$$(\rho - \|\lambda^*\|)\|[G(\tilde{x}_T)]_+\|_\star \leq \frac{\max_{\lambda\in\mathcal{B}_\rho^+} D_h(\lambda, \lambda_0)}{\sum_{k=0}^{T-1}\eta_k}. \tag{2.18}$$

Setting $\rho = 2\|\lambda^*\| + 1$ in (2.18) and combining with the fact that $f(\tilde{x}_T) - f(x^*) \geq -\|\lambda^*\| \cdot \|[G(\tilde{x}_T)]_+\|_\star$ leads to the desired result in (2.15). ∎

The above result generalizes the existing ergodic convergence result for classical ALM, e.g., in [16], which can be viewed as a special case when the Bregman divergence is set to the Euclidean distance. From the analysis, we see that the convergence of the primal objective and constraint violation heavily depends on the chosen norm used to measure the constraint violation. Note that the rate of primal convergence is essentially in the same order as that of the dual convergence discussed in previous section. When the proximal parameters $\{\eta_k\}_{k\geq 0}$ are fixed to a constant, this implies the $\mathcal{O}(1/T)$ convergence rate of both primal and dual sequences.

2.2 Variational Inequality Problems

2.2.1 Introduction

Recall the problem Eqn (1.6), also referred to as the constrained VI or generalized Nash Equilibrium (GNE): find $z_* \in Z := Z^1 \times \cdots \times Z^m$ such that

$$\max_{x \in Z \cap \mathcal{G}} \epsilon(z_*; x) := \sum_{i=1}^{m} \left[F_i(z_*^i, z_*^{-i}) - F_i(x^i, z_*^{-i}) \right] = 0 \tag{2.19}$$

These type of problems encompass the previous constrained minimization problem as a special case, by setting $m = 1$.

Such problems have been considered in existing literature [41, 42, 2, 43]. In particular, such problems with linear constraint can be solved efficiently using first-order method (using the monotone operator only) [43]. However, most existing analysis focus on asymptotic analysis. Here we consider problems with convex functional inequality constrained problems, i.e. the constraints are convex in z. We analyze the iteration complexity by considering the inexact version of BALM and derive the total iteration complexity using first-order method. In particular, we consider using accelerated mirror-prox [44] to solve the subproblem. Note that we do have more assumptions, introduced in section 1.2, in order to make non-asymptotic analysis possible.

2.2.2 Analysis of Inexact BALM for Variational Inequality Problem

To describe the algorithm, first define

$$\phi_k(z) := \max_{\lambda \geq 0} \left\{ \langle \lambda, G(z) \rangle - \frac{1}{\eta_k} D(\lambda, \lambda_k) \right\} \tag{2.20}$$

where k is the iteration number. Then the k^{th} subproblem can be defined to find z_{k+1}^* such that

$$\max_{x \in Z} \ \epsilon(z_{k+1}^*; x) + \phi_k(z_{k+1}^*) - \phi_k(x) = 0 \tag{2.21}$$

Under mild conditions [45], it is equivalent to the weak V.I. solution (since the above is implied by the strong V.I.)

$$\max_{x \in Z} \langle H(x), z_{k+1}^* - x \rangle + \phi_k(z_{k+1}^*) - \phi_k(x) = 0 \tag{2.22}$$

We consider an inexact/stochastic version of the algorithm in algorithm 2. The analysis is very similar to BALM being applied to minimization problem.

Algorithm 2: BALM-VI

Input: $\lambda_0 \geq 0$

for $k \geq 0$ **do**

Obtain z_{k+1} such that

$$\mathbb{E}[\max_{x \in Z} \epsilon(z_{k+1}; x) + \phi_k(z_{k+1}) - \phi_k(x)] \leq \delta_k$$

and

$$\mathbb{E}[\max_{x \in Z} \langle H(x), z_{k+1} - x \rangle + \phi_k(z_{k+1}) - \phi_k(x)] \leq \delta_k$$

$\lambda_{k+1} = \text{argmax}_{\lambda \geq 0} \{ \langle \lambda, G(z_{k+1}) \rangle - \frac{D(\lambda, \lambda_k)}{\eta_k} \}$

Lemma 2.2.1 *The algorithm 2 satisfies*

$$\mathbb{E}\left[\max_{x \in Z \cap \mathcal{G}, \lambda \geq 0} \epsilon(z_{k+1}; x) + \langle \lambda, G(z_{k+1}) \rangle + \frac{1}{\eta_k} \left(D(\lambda, \lambda_{k+1}) - D(\lambda, \lambda_k) \right) \right] \leq \delta_k \tag{2.23}$$

and

$$\mathbb{E}\left[\max_{x \in Z \cap \mathcal{G}, \lambda \geq 0} \langle H(x), z_{k+1} - x \rangle + \langle \lambda, G(z_{k+1}) \rangle + \frac{1}{\eta_k} \left(D(\lambda, \lambda_{k+1}) - D(\lambda, \lambda_k) \right) \right] \leq \delta_k \tag{2.24}$$

Proof. For Eqn (2.23), from the algorithm update we get

$$\mathbb{E}[\max_{x \in Z} \epsilon(z_{k+1}; x) + \langle \lambda_{k+1}, G(z_{k+1}) \rangle - \frac{1}{\eta_k} D(\lambda_{k+1}, \lambda_k) - \max_{\lambda \geq 0} \left\{ \langle \lambda, G(z) \rangle - \frac{1}{\eta_k} D(\lambda, \lambda_k) \right\}] \leq \delta_k$$

For $z \in \mathcal{G}$, $G(z) \leq 0$, thus

$$\mathbb{E}\left[\max_{x \in Z} \epsilon(z_{k+1}; x) + \langle \lambda_{k+1}, G(z_{k+1}) \rangle - \frac{1}{\eta_k} D(\lambda_{k+1}, \lambda_k)\right] \leq \delta_k$$

From the update of λ_k,

$$\langle \lambda_{k+1}, G(z_{k+1}) \rangle - \frac{D(\lambda_{k+1}, \lambda_k)}{\eta_k} \geq \max_{\lambda \geq 0} \{\langle \lambda, G(z_{k+1}) \rangle + \frac{1}{\eta_k}(D(\lambda, \lambda_{k+1}) - D(\lambda, \lambda_k))\}$$

The proof for Eqn (2.24) is similar. ∎

Lemma 2.2.2 *Let*

$$\tilde{z}_T = \frac{\sum_{k=0}^{T-1} \eta_k z_{k+1}}{\sum_{k=0}^{T-1} \eta_k}$$

then

$$\mathbb{E}\left[\max_{x \in Z \cap \mathcal{G}, \lambda \geq 0} \epsilon(\tilde{z}_T; x) + \langle \lambda, G(\tilde{z}_T) \rangle - \frac{D(\lambda, \lambda_0)}{\sum_{k=0}^{T-1} \eta_k}\right] \leq \frac{\sum_{k=0}^{T-1} \eta_k \delta_k}{\sum_{k=0}^{T-1} \eta_k} \tag{2.25}$$

and

$$\mathbb{E}\left[\max_{x \in Z \cap \mathcal{G}, \lambda \geq 0} \langle H(x), \tilde{z}_T - x \rangle + \langle \lambda, G(\tilde{z}_T) \rangle - \frac{D(\lambda, \lambda_0)}{\sum_{k=0}^{T-1} \eta_k}\right] \leq \frac{\sum_{k=0}^{T-1} \eta_k \delta_k}{\sum_{k=0}^{T-1} \eta_k} \tag{2.26}$$

Proof. By assumption, both $\max_{x \in Z \cap \mathcal{G}} \epsilon(z; x)$ and $\max_{x \in Z \cap \mathcal{G}} \langle H(x), z - x \rangle$ are convex in z. Then using telescoping sum gives the results. ∎

Theorem 2.2.1 *Let* $r = \|\lambda^*\| + 1$,

$$\mathbb{E}\left[\max_{x \in Z \cap \mathcal{G}} \langle H(x), \tilde{z}_T - x \rangle\right], \mathbb{E}\left[\max_{x \in Z \cap \mathcal{G}} \epsilon(\tilde{z}_T; x)\right] \leq \frac{D(0, \lambda_0)}{\sum_{k=0}^{T-1} \eta_k} + \frac{\sum_{k=0}^{T-1} \eta_k \delta_k}{\sum_{k=0}^{T-1} \eta_k} \tag{2.27}$$

and

$$\mathbb{E}[\|[G(\tilde{z}_T)]_+\|_\star] \leq \frac{\max_{\lambda \in \mathcal{B}_r^+} \{D(\lambda, \lambda_0)\}}{\sum_{k=0}^{T-1} \eta_k} + \frac{\sum_{k=0}^{T-1} \eta_k \delta_k}{\sum_{k=0}^{T-1} \eta_k} \tag{2.28}$$

where $\mathcal{B}_r^+ = \{\lambda \geq 0 : \|\lambda\| \leq r\}$, $\|\cdot\|_\star$ *is the dual norm of* $\|\cdot\|$, $[\cdot]_+$ *takes elementwise maximum with* 0.

Proof. Take $\lambda = 0$ in Eqn (2.25) and Eqn (2.26), we obtain

$$\mathbb{E}\left[\max_{x\in Z\cap\mathcal{G}}\langle H(x), \tilde{z}_T - x\rangle\right], \mathbb{E}\left[\max_{x\in Z\cap\mathcal{G}}\epsilon(\tilde{z}_T; x)\right] \leq \frac{D(0,\lambda_0)}{\sum_{k=0}^{T-1}\eta_k} + \frac{\sum_{k=0}^{T-1}\eta_k\delta_k}{\sum_{k=0}^{T-1}\eta_k}$$

From Eqn (2.26), let $x = z_*$ and let λ take maximum from $\lambda \in \mathcal{B}_r^+$

$$\mathbb{E}\left[\langle H(z_*), \tilde{z}_T - z_*\rangle + r\|[G(\tilde{z}_T)]_+\|_\star\right] \leq \frac{\max_{\lambda\in\mathcal{B}_r^+}\{D(\lambda,\lambda_0)\}}{\sum_{k=0}^{T-1}\eta_k} + \frac{\sum_{k=0}^{T-1}\eta_k\delta_k}{\sum_{k=0}^{T-1}\eta_k}$$

Also by Eqn (1.10)

$$\langle H(z_*), z_* - \tilde{z}_T\rangle - \|\lambda^*\|\|[G(\tilde{z}_T)]_+\|_\star \leq \langle H(z_*), z_* - \tilde{z}_T\rangle - \langle\lambda^*, G(\tilde{z}_T)\rangle \leq 0$$

Sum up the above two inequalities and we obtain the constraint violation bound. ∎

Note that it is unclear if there exists any acceleration technique for BALM being applied to VI. When the subproblem can be solved exactly, the dual problem is still a maximization problem, so the dual problem can still be accelerated. However, the primal problem may not enjoy the same accelerated rate, and the problem becomes even more elusive when the subproblem is only solved to a certain accuracy. We will not focus on acceleration in this chapter.

2.2.3 Overall Iteration Complexity For Classical ALM

In order to discuss the overall iteration complexity, we need to discuss how to solve the subproblem such that the inexactness appeared in the algorithm is satisfied. In order to do so, we focus on the case when $D(\lambda,\lambda') = \frac{\|\lambda-\lambda'\|_2^2}{2}$, i.e. the classical ALM, which means that

$$\phi_k(z) = \frac{1}{2\eta_k}\sum_{j=1}^{l}\left([\lambda_k(j) + \eta_k g^j(z)]_+^2 - \lambda_k(j)^2\right) \tag{2.29}$$

Recall that we need to solve the sub-problem to obtain z_{k+1} such that

$$\mathbb{E}[\max_{x\in Z}\ \epsilon(z_{k+1}; x) + \phi_k(z_{k+1}) - \phi_k(x)] \leq \delta_k \tag{2.30}$$

and

$$\mathbb{E}[\max_{x\in Z}\langle H(x), z_{k+1}-x\rangle + \phi_k(z_{k+1}) - \phi_k(x)] \leq \delta_k \tag{2.31}$$

Before talking about the subroutine, we first make a few standard assumptions, similar to that of [16].

Assumption 2.2.1 *Each $g_i(z)$ is continuously differentiable, and*

$$\|H(z) - H(z')\| \leq L_0\|z - z'\| \tag{2.32}$$
$$\|\nabla g^j(z) - \nabla g^j(z')\| \leq L_j\|z - z'\|,\ \forall j \tag{2.33}$$
$$\max\{|g^j(z)|, \|\nabla g^j(z)\|\} \leq B_j,\ \forall j \tag{2.34}$$
$$|g^j(z) - g^j(z')| \leq B_j\|z - z'\|,\ \forall j \tag{2.35}$$

Then the gradient of $\phi_k(z)$, $\nabla\phi_k(z) = \sum_{j=1}^{l}[\lambda_k(j) + \eta_k g^j(z)]_+\nabla g^j(z)$, is Lipschitz continuous with constant:

$$L_{\phi_k}(\lambda_k) = \sum_{j=1}^{l}(L_j|\lambda_k(j)| + \eta_k B_j(B_j + L_j)) \tag{2.36}$$

The proof can be found in [16]. Furthermore the dependence on λ_k can also be dealt with by bounding $\|\lambda_k\|$, the details can also be found in [16], here we simply assume that

$$L_{\phi_k}(\lambda_k) \leq \sum_{j=1}^{l}(CL_j + \eta_k B_j(B_j + L_j)) \tag{2.37}$$

With all the assumptions listed above, we can solve the subproblem with the accelerated mirror-prox discussed in [44] (we omit the outer index k for brevity):

where

$$P_r^V(x) := \arg\min_{u\in Z}\{\langle x, u - r\rangle + V(u, r)\} \tag{2.42}$$

and $V(u, r)$ is some Bregman divergence such that $V(u, r) \geq \frac{\mu}{2}\|u - r\|^2$. Let $\max_{z_1,z_2\in Z} V(z_1, z_2) \leq \Omega_Z^2$,then we have the following theorems:

Theorem 2.2.2 ([44]) *Suppose $\tilde{H}(r_t) = H(r_t)$, $\tilde{H}(w_{t+1}) = H(w_{t+1})$, $\tilde{\nabla}\phi\left(w_t^{md}\right) =$*

Algorithm 3: acc-MirrorProx

Input: Choose $r_1 \in Z$, set $w_1 = r_1, w_1^{ag} = r_1$

for $t = 1, \ldots, N-1$ **do**

$$w_t^{md} = (1-\alpha_t)\, w_t^{ag} + \alpha_t r_t \tag{2.38}$$

$$w_{t+1} = P_{r_t}^V\left(\gamma_t \tilde{H}\left(r_t\right) + \gamma_t \tilde{\nabla}\phi\left(w_t^{md}\right)\right) \tag{2.39}$$

$$r_{t+1} = P_{r_t}^V\left(\gamma_t \tilde{H}\left(w_{t+1}\right) + \gamma_t \tilde{\nabla}\phi\left(w_t^{md}\right)\right) \tag{2.40}$$

$$w_{t+1}^{ag} = (1-\alpha_t)\, w_t^{ag} + \alpha_t w_{t+1} \tag{2.41}$$

Output w_{N+1}^{ag}.

$\nabla\phi\left(w_t^{md}\right)$, *i.e. there is no randomness, then by choosing*

$$\alpha_t = \frac{2}{t+1}, \gamma_t = \frac{\mu t}{2(L_\phi + L_0 t)}$$

we have

$$\max_{x\in Z}\langle H(x), w_{t+1}^{ag} - x\rangle + \phi(w_{t+1}^{ag}) - \phi(x) \le \left(\frac{4L_\phi}{\mu t(t+1)} + \frac{4L_0}{\mu t}\right)\Omega_Z^2 \tag{2.43}$$

$$\max_{x\in Z}\epsilon(w_{t+1}^{ag}; x) + \phi(w_{t+1}^{ag}) - \phi(x) \le \left(\frac{4L_\phi}{\mu t(t+1)} + \frac{4L_0}{\mu t}\right)\Omega_Z^2 \tag{2.44}$$

Theorem 2.2.3 ([44]) *Suppose* $\mathbb{E}[\tilde{H}\left(r_t\right)] = H\left(r_t\right)$, $\mathbb{E}[\tilde{H}\left(w_{t+1}\right)] = H\left(w_{t+1}\right)$, $\mathbb{E}[\tilde{\nabla}\phi\left(w_t^{md}\right)] = \nabla\phi\left(w_t^{md}\right)$, *and*

$$\mathbb{E}[\|\tilde{H}(z) - H(z)\|^2] \le \sigma_H^2, \mathbb{E}[\|\tilde{\nabla}\phi(z) - \nabla\phi(z)\|^2\|] \le \sigma_\phi^2$$

by choosing

$$\alpha_t = \frac{2}{t+1}, \gamma_t = \frac{\mu t}{4L_\phi + 3L_0 t + \sigma(t+1)\sqrt{\mu t}/(\sqrt{2}\Omega_Z)}$$

where $\sigma := \sqrt{\sigma_H^2 + \sigma_\phi^2}$. *Then we have*

$$\mathbb{E}[\max_{x\in Z}\langle H(x), w_{t+1}^{ag} - x\rangle + \phi(w_{t+1}^{ag}) - \phi(x)] \leq \frac{16L_\phi\Omega_Z^2}{\mu t(t+1)} + \frac{12L_0\Omega_Z^2}{\mu(t+1)} + \frac{7(\sigma_H+\sigma_\phi)\Omega_Z}{\sqrt{\mu(t-1)}} \tag{2.45}$$

$$\mathbb{E}[\max_{x\in Z}\epsilon(w_{t+1}^{ag}; x) + \phi(w_{t+1}^{ag}) - \phi(x)] \leq \frac{16L_\phi\Omega_Z^2}{\mu t(t+1)} + \frac{12L_0\Omega_Z^2}{\mu(t+1)} + \frac{7(\sigma_H+\sigma_\phi)\Omega_Z}{\sqrt{\mu(t-1)}} \tag{2.46}$$

Discussion In [44], only Eqn (2.43) and Eqn (2.45) were proved, but the proof can be easily extended to Eqn (2.44) and Eqn (2.46) by noting that $\epsilon(z; x)$ is convex in z. The above theorem implies that is we want to obtain z_{k+1} which satisfies the δ_k error, it takes the following number of inner iterations for deterministic and stochastic case:

$$\begin{cases} \mathcal{O}\left(\frac{L_0}{\delta_k} + \sqrt{\frac{L_\phi}{\delta_k}}\right) & \text{(deterministic)} \\ \mathcal{O}\left(\frac{L_0}{\delta_k} + \sqrt{\frac{L_\phi}{\delta_k}} + \frac{\sigma_H+\sigma_\phi}{\delta_k^2}\right) & \text{(stochastic)} \end{cases} \tag{2.47}$$

If we let $\eta_k = \eta, \delta_k = \delta$, then the error bounds Eqn (2.27) and Eqn (2.28) have the form

$$\frac{C(\lambda^*)}{T\eta} + \delta$$

where $C(\lambda^*)$ is a constant depending only on λ^*. Also note that $L_{\phi_k} \leq \sum_{j=1}^{l}(CL_j + \eta B_j(B_j + L_j))$. So we can let T be a fixed constant, and let let $\eta = \frac{1}{\delta}$. Then to obtain a $(C(\lambda^*)/T + 1)\,\delta$-accuracy solution, the overall iteration complexity is

$$\begin{cases} \mathcal{O}\left(\frac{L_0}{\delta} + \sqrt{\frac{\sum_{j=1}^{l} CL_j}{\delta}} + \frac{\sqrt{\sum_{j=1}^{l} B_j(B_j+L_j)}}{\delta}\right) & \text{(deterministic)} \\ \mathcal{O}\left(\frac{L_0}{\delta} + \sqrt{\frac{\sum_{j=1}^{l} CL_j}{\delta}} + \frac{\sqrt{\sum_{j=1}^{l} B_j(B_j+L_j)}}{\delta} + \frac{\sigma_H+\sigma_\phi}{\delta^2}\right) & \text{(stochastic)} \end{cases} \tag{2.48}$$

2.3 Conclusions

In this chapter, we analyzed the covergence rate of BALM, and its inexact version, being applied to minimization problems, and more generally variational inequality problems. In particular, the subproblem can even be solved with a stochastic method. Then we focus on the classical ALM, and we use the accelerated mirror-prox to solve the subproblem in order to derive the total iteration complexity to solve a specific form of generalized Nash Equilibrium problem.

CHAPTER 3

ACCELERATION OF BPP AND BALM

3.1 A Generic Acceleration Scheme of BPP

In the seminal work [21], Güler proposed the first accelerations of the proximal point algorithm based on Nesterov's acceleration scheme [46]. Inexact versions of the accelerated PPA have been later studied in [47, 48] and recent work [25]. While it seems rather natural to extend the accelerated PPA to the non-Euclidean setting, there exists only few attempts in this direction [49, 50]. It came to our attention that these existing works contain fatal flaws, both algorithmically and theoretically.

Motivated by [10, 21], we propose the first theoretically-sound acceleration scheme for Bregman proximal point method, which will later applied to accelerate BALM. Without loss of generality, we consider solving the convex problem

$$\max_{\lambda \in \Lambda} \; d(\lambda), \tag{3.1}$$

where $d(\lambda)$ and Λ are closed and convex. The objective $d(\lambda)$ does not have to be the Lagrange dual of the linearly constrained convex program. Let $D_h(\lambda, \lambda') : \Lambda \times \Lambda \to \infty$ be a Bregman divergence induced by some function h that is continuously differentiable and strictly convex on Λ. In addition, we assume that the Bregman divergence satisfies the so-called *triangle scaling property*, which turns out to be a crucial assumption to achieve faster rates. The triangle scaling property was introduced recently in [51, 52] for analyzing the convergence of Bregman proximal gradient methods for relatively smooth objective functions. To be specific,

Assumption 3.1.1 *There exists some constant $G > 0$ such that the Breg-*

man divergence D_h has the triangle scaling property: for all $\lambda, \lambda_1, \lambda_2 \in int(\Lambda)$,

$$D_h((1-\theta)\lambda + \theta\lambda_1, (1-\theta)\lambda + \theta\lambda_2) \leq G\theta^2 D_h(\lambda_1, \lambda_2), \forall \theta \in [0,1]. \tag{3.2}$$

For detailed discussions about this property, see [52]. For ease of exposition, here we simply adopt G as a uniform constant, which is closely related to the Hessian of the Bregman function. In particular, if the Bregman divergence is both L_h-Lipschitz smooth and σ_h-strongly convex, then $\frac{\sigma_h}{2}\|x-y\|^2 \leq D_h(x,y) \leq \frac{L_h}{2}\|x-y\|^2$, thus Assumption 3.1.1 is satisfied with $G = L_h/\sigma_h$.

The general idea for constructing the acceleration scheme is to first define the following sequence of functions recursively:

$$\begin{cases} \phi_0(\lambda) = d(\lambda_0) - AD_h(\lambda, \lambda_0) \\ \phi_{k+1}(\lambda) = (1-\theta_k)\phi_k(\lambda) + \theta_k(d(Jy_k) + \frac{1}{\eta_k}(\nabla h(Jy_k) - \nabla h(y_k))^\top(\lambda - Jy_k)), \end{cases} \tag{3.3}$$

where y_k is any point (to be specified later) and $Jy_k := \arg\max_{\lambda\in\Lambda}\left\{d(\lambda) - \frac{D_h(\lambda, y_k)}{\eta_k}\right\}$. These functions satisfy the following relation,

Lemma 3.1.1 *For any k and $\lambda \in \Lambda$, it holds that*

$$d(\lambda) - \phi_{k+1}(\lambda) \leq (1-\theta_k)(d(\lambda) - \phi_k(\lambda)). \tag{3.4}$$

Proof. By concavity and optimality condition from the definition of Jy_k, we have

$$d(\lambda) \leq d(Jy_k) + \frac{1}{\eta_k}(\nabla h(Jy_k) - \nabla h(y_k))^\top(\lambda - Jy_k). \tag{3.5}$$

Hence, it immediately implies that

$$\begin{aligned} d(\lambda) - \phi_{k+1}(\lambda) &= (1-\theta_k)(d(\lambda) - \phi_k(\lambda)) + \theta_k(d(\lambda) - d(Jy_k) \qquad (3.6) \\ &\quad - \frac{1}{\eta_k}(\nabla h(Jy_k) - \nabla h(y_k))^\top(\lambda - Jy_k)) \\ &\leq (1-\theta_k)(d(\lambda) - \phi_k(\lambda)). \qquad (3.7) \end{aligned}$$

■

Our goal is to obtain λ_k such that $d(\lambda_k) \geq \max_{\lambda\in\Lambda}\phi_k(\lambda)$. From the con-

struction of $\phi_k(\lambda)$, we can see that

$$\phi_k(\lambda) = l_k(\lambda) - A_k D_h(\lambda, \lambda_0), \tag{3.8}$$

where $l_k(\lambda)$ is an affine function, and $A_k = \prod_{i=0}^{k-1}(1-\theta_k)A$. Using the three-point identity of the Bregman divergence, it can be easily shown that

$$\phi_k(\lambda) = \phi_k(\lambda') + \nabla\phi_k(\lambda')^\top(\lambda - \lambda') - A_k D_h(\lambda, \lambda'), \forall \lambda, \lambda' \in \Lambda. \tag{3.9}$$

This means that if we let $v_k := \arg\max_{\lambda\in\Lambda} \phi_k(\lambda)$, we have

$$\phi_k(\lambda) \le \phi_k(v_k) - A_k D_h(\lambda, v_k), \forall \lambda \in \Lambda. \tag{3.10}$$

The following lemma shows how to construct the desired λ_{k+1}, given that we already have $d(\lambda_k) \ge \phi_k(v_k)$.

Lemma 3.1.2 *Suppose we already have λ_k such that $\phi_k(v_k) \le d(\lambda_k)$, then choosing*

$$\frac{G\theta_k^2}{\eta_k} = (1-\theta_k)A_k, y_k = \theta_k v_k + (1-\theta_k)\lambda_k, \lambda_{k+1} = Jy_k$$

would ensure $\phi_k(v_{k+1}) \le d(\lambda_{k+1})$.

Proof. Denote $\Delta_k = \frac{1}{\eta_k}[\nabla h(Jy_k) - \nabla h(y_k)]$. We can show that

$$\begin{aligned}
\phi_{k+1}(v_{k+1}) &= \max_{\lambda\in\Lambda}\{(1-\theta_k)\phi_k(\lambda) + \theta_k(d(Jy_k) + \Delta_k^\top(\lambda - Jy_k))\} \\
&\le \max_{\lambda\in\Lambda}\{(1-\theta_k)(\phi_k(v_k) - A_k D_h(\lambda, v_k)) + \theta_k(d(Jy_k) + \Delta_k^\top(\lambda - Jy_k))\} \\
&\le \max_{\lambda\in\Lambda}\{(1-\theta_k)(d(\lambda_k) - A_k D_h(\lambda, v_k)) + \theta_k(d(Jy_k) + \Delta_k^\top(\lambda - Jy_k))\} \\
&\le d(Jy_k) + \max_{\lambda\in\Lambda}\{-(1-\theta_k)A_k D_h(\lambda, v_k) + \Delta_k^\top(\theta_k\lambda + (1-\theta_k)\lambda_k - Jy_k)\} \\
&\le d(Jy_k) + \max_{\lambda\in\Lambda}\{-(1-\theta_k)A_k D_h(\lambda, v_k) + \frac{1}{\eta_k} D_h(\theta_k\lambda + (1-\theta_k)\lambda_k, y_k)\}.
\end{aligned}$$

Here the first inequality uses (3.10); the second inequality uses the induction hypothesis; the third inequality applies (3.5) with $\lambda = \lambda_k$; and the last inequality uses the three-point identity (1.13). Next, based on assumption

(3.1.1) and the fact that $y_k = \theta_k v_k + (1-\theta_k)\lambda_k$, we can further obtain:

$$\begin{aligned}\phi_{k+1}(v_{k+1}) &\le d(Jy_k) + \max_{\lambda\in\Lambda}\Big\{ -(1-\theta_k)A_k D_h(\lambda, v_k) \\ &\qquad + \frac{1}{\eta_k} D_h(\theta_k\lambda + (1-\theta_k)\lambda_k, \theta_k v_k + (1-\theta_k)\lambda_k)\Big\} \\ &\le d(Jy_k) + \max_{\lambda\in\Lambda}\{-(1-\theta_k)A_k D_h(\lambda, v_k) + \frac{G\theta_k^2}{\eta_k} D_h(\lambda, v_k)\}\end{aligned} \tag{3.11}$$

If we choose $\lambda_{k+1} = Jy_k$, and $\frac{G\theta_k^2}{\eta_k} = (1-\theta_k)A_k$, by induction, we have $d(\lambda_{k+1}) \ge \phi_{k+1}(v_{k+1})$. ∎

Now we are in the position to present the generic accelerated scheme of Bregman proximal point algorithm (acc-BPP) in Algorithm 4. The computation of v_k can be carried out in a closed form in most cases, since $\phi_k(\lambda)$ composes an affine term and a Bregman divergence term as expressed in (3.8).

Algorithm 4: Accelerated Bregman Proximal Point Algorithm (acc-BPP)

Input: $\lambda_0 \in \Lambda, v_0 = \lambda_0, A_0 = A \in (0, +\infty), \phi_0(\lambda) = d(\lambda_0) - AD_h(\lambda, \lambda_0), \{\eta_k\}_{k\ge 0}, G$

for $k \ge 0$ **do**

- Choose θ_k such that $\frac{\eta_k A_k(1-\theta_k)}{G} = \theta_k^2$, i.e. $\theta_k = \frac{\sqrt{(A_k\eta_k/G)^2 + 4A_k\eta_k/G} - A_k\eta_k/G}{2}$
- $y_k = \theta_k v_k + (1-\theta_k)\lambda_k$
- $\lambda_{k+1} \in \arg\max_{\lambda\in\Lambda}\{d(\lambda) - \frac{1}{\eta_k} D_h(\lambda, y_k)\}$
- $A_{k+1} = (1-\theta_k)A_k$
- $\phi_{k+1}(\lambda) = (1-\theta_k)\phi_k(\lambda) + \theta_k(d(\lambda_{k+1}) + \frac{1}{\eta_k}(\nabla h(\lambda_{k+1}) - \nabla h(y_k))^\top(\lambda - \lambda_{k+1}))$
- $v_{k+1} = \arg\max_{\lambda\in\Lambda}\phi_{k+1}(\lambda)$

The following theorem characterizes the convergence rate of Algorithm 4.

Theorem 3.1.1 *Under Assumption 3.1.1, Algorithm 4 satisfies that*

$$d(\lambda^*) - d(\lambda_T) \le \left[\prod_{k=0}^{T-1}(1-\theta_i)\right](d(\lambda^*) - d(\lambda_0) + AD_h(\lambda^*, \lambda_0)), \tag{3.12}$$

and

$$\frac{1}{\left(1+\sqrt{A/G}\sum_{k=0}^{T-1}\sqrt{\eta_k}\right)^2} \leq \prod_{k=0}^{T-1}(1-\theta_i) \leq \frac{1}{\left(1+(\sqrt{A/G}/2)\sum_{k=0}^{T-1}\sqrt{\eta_k}\right)^2}. \tag{3.13}$$

Proof. By the construction of $\{\phi_k(\lambda)\}_k$, we have

$$d(\lambda)-\phi_k(\lambda) \leq \prod_{i=0}^{k-1}(1-\theta_i)(d(\lambda)-\phi_0(\lambda)), \quad \forall \lambda \in \Lambda.$$

And the inductive construction of λ_k guarantees that $d(\lambda_k) \geq \max_{\lambda\in\Lambda}\phi_k(\lambda)$, which proves (3.12). The inequality (3.13) is proved in [21]. ∎

The above theorem indicates that acc-BPP improves the convergence rate of BPP from $\mathcal{O}\left(1/\sum_{j=0}^{T-1}\eta_j\right)$ to $\mathcal{O}\big(1/(\sum_{j=0}^{T-1}\sqrt{\eta_j})^2\big)$. This recovers the result in [21] as a special case. In particular, if we choose $\eta_k = \eta, k = 0, 1, \cdots, T-1$ to be a constant, this automatically leads to the $\mathcal{O}(1/T^2)$ convergence rate. Note however, η_k can be arbitrarily chosen in practice.

Note that the triangle scaling property of Bregman divergences naturally arises when designing the generic acceleration scheme. While this condition may not be satisfied by some Bregman divergences, it should be noticed from the proof that the condition only needs to hold true at $\theta_k, k = 0, 1, \cdots$. When θ_k's are bounded away from 0, which is almost always the case in practice, there exists a constant G such that the property is satisfied, albeit possibly being large and difficult to estimate. In the numerical experiments, we find that setting G to be any positive constant provides accelerated performance.

Remark Recall that existing accelerated Bregman proximal gradient methods (ABPG) [52, 51] attain the $\mathcal{O}(1/T^2)$ rate of convergence when solving the composite optimization, i.e., $\max_{\lambda\in\Lambda} g(\lambda) + d(\lambda)$, where $g(\lambda)$ is (relatively) Lipschitz smooth and $d(\lambda)$ is a simple concave function admitting easy-to-compute Bregman operators. One may be tempted to think that ABPG reveals an "accelerated" version of Bregman proximal point method when setting $g(\lambda) = 0$. Take the Algorithm 1 in [52] for an example. Setting

$g(\lambda) = 0$ leads to the following "accelerated" algorithm

$$\begin{cases} y_k = (1-\theta_k)\mu_k + \theta_k \lambda_k \\ \lambda_{k+1} = \arg\max_{\lambda\in\Lambda}\{d(\lambda) - \theta_k L D_h(\lambda, \lambda_k)\} \\ \mu_{k+1} = (1-\theta_k)\mu_k + \theta_k \lambda_{k+1} \\ \frac{1-\theta_{k+1}}{\theta_{k+1}^2} = \frac{1}{\theta_k^2} \end{cases} \tag{3.14}$$

The choice of L is arbitrary, but it is fixed once chosen. Based on the convergence analysis in [51], the above algorithm inherits the convergence rate of $d(\lambda^*) - d(\mu_T) \leq \frac{4LD_h(\lambda^*, \mu_0)}{T^2}$. However, it is also shown in [51] that $\theta_k \leq \frac{2}{k+1}$, or equivalently, the proximal parameters $\eta_k \geq \frac{k+1}{2L}$ in the proximal point scheme. Notably, if we choose $\eta_k \geq \frac{k+1}{2L}$ in the vanilla BPP, the achievable convergence rate would be $\mathcal{O}\left(\frac{1}{\sum_{k=0}^{T-1}\eta_k}\right) = \mathcal{O}(1/T^2)$, which already attains the same rate as the above "accelerated" algorithm. In contrast, the proposed acc-BPP would achieve the rate $\mathcal{O}\left(\frac{1}{(\sum_{k=0}^{T-1}\sqrt{\eta_k})^2}\right) = \mathcal{O}(1/T^3)$ with such proximal parameters, which is much faster than $\mathcal{O}(1/T^2)$ rate. Therefore, we emphasize that the freedom in choosing arbitrary $\{\eta_k\}_{k\geq 0}$ is crucial here, and also distinct our acceleration with the one above.

3.2 Two variations of the accelerated Bregman Proximal Point method

In this section, we introduce two variations (or special cases) of acc-BPP that enjoy much more compact forms as well as simpler convergence analysis.

3.2.1 Memoryless form

In the previous generic acceleration scheme, v_k is defined as $v_k := \arg\max_{\lambda\in\Lambda}\phi_k(\lambda)$. This requires keeping track of the explicit form of functions $\{\phi_k\}_{k\geq 0}$ and computing its maximizer, which may not necessarily admit a closed form. This issue can be alleviated by setting $v_k := \arg\max_{\lambda}\phi_k(\lambda)$, instead. In fact, it can be easily seen that the proof still remains valid by doing so, if additional relaxing Assumption 3.1.1 to hold on the entire domain of h instead of Λ. In this case, we can obtain a closed-form for v_k.

Lemma 3.2.1 ([10]) *Let $\phi_k(\lambda)$ be recursively defined as (3.3) with $Jy_k =$*

λ_{k+1}, *then* $v_k = \arg\max_\lambda \phi_k(\lambda)$ *satisfies the following recursion relation*

$$\begin{aligned} v_{k+1} &= \arg\max_{\lambda} \left\{ \frac{\theta_k}{\eta_k}(\nabla h(\lambda_{k+1}) - \nabla h(y_k))^\top \lambda - A_{k+1} D_h(\lambda, v_k) \right\}, \\ &= \arg\max_{\lambda} \left\{ \frac{1}{G\theta_k}(\nabla h(\lambda_{k+1}) - \nabla h(y_k))^\top \lambda - D_h(\lambda, v_k) \right\}. \end{aligned} \tag{3.15}$$

This implies an explicit v-update: $v_{k+1} = \nabla h^*\big(\nabla h(v_k) + \frac{1}{G\theta_k}(\nabla h(\lambda_{k+1}) - \nabla h(y_k))\big)$.

Algorithm 5: acc-BPP2

Input: $\lambda_0 \in \Lambda, v_0 = \lambda_0, \theta_0 = 1, \{\eta_k\}_{k\geq 0}, G$

for $k \geq 0$ **do**

- $y_k = \theta_k v_k + (1-\theta_k)\lambda_k$
- $\lambda_{k+1} \in \arg\max_{\lambda\in\Lambda}\{d(\lambda) - \frac{1}{\eta_k} D_h(\lambda, y_k)\}$
- $v_{k+1} = \nabla h^*\left(\nabla h(v_k) + \frac{1}{G\theta_k}(\nabla h(\lambda_{k+1}) - \nabla h(y_k))\right)$
- Update θ_{k+1} such that $\frac{\eta_k}{\theta_k^2} = \frac{\eta_{k+1}}{\theta_{k+1}^2} - \frac{\eta_{k+1}}{\theta_{k+1}}$.

As a result, acc-BPP simply reduces to Algorithm 5, which no longer requires to store the representation of $\{\phi_k\}$ an has much cheaper memory cost. The proof for the convergence rate of acc-BPP2 follows exactly that of acc-BPP. Note that choosing $\theta_0 = 1$[1] amounts to setting $A = +\infty$ in acc-BPP, thus acc-BPP2 satisfies

$$d(\lambda^*) - d(\lambda_T) \leq \lim_{A\to+\infty} \frac{d(\lambda^*) - d(\lambda_0) + AD_h(\lambda^*, \lambda_0)}{(1 + (\sqrt{A/G}/2)\sum_{j=0}^{T-1}\sqrt{\eta_j})^2} = \frac{4GD_h(\lambda^*, \lambda_0)}{\left(\sum_{k=0}^{T-1}\sqrt{\eta_k}\right)^2} \tag{3.16}$$

3.2.2 Dual averaging form

Below, we show that the above special case of acc-BPP admits another form that resembles the Nesterov's accelerated dual average method [31, 32, 30]. Recall the definition of $\{\phi_k\}_{k\geq 0}$ and $\{A_k\}_{k\geq 0}$, the computation of v_{k+1} is also

[1]Notice that θ_0 can also be any real number in $(0, 1]$, accordingly $A = \frac{G\theta_0^2}{\eta_0(1-\theta_0)}$.

equivalent to

$$v_{k+1} = \arg\max_{\lambda\in\Lambda} \Big\{ \sum_{j=0}^{k} \frac{\theta_j}{A_{j+1}\eta_j} \left(\nabla h(\lambda_{j+1}) - \nabla h(y_j)\right)^\top \lambda - D_h(\lambda, \lambda_0) \Big\},$$
$$= \arg\max_{\lambda\in\Lambda} \Big\{ \sum_{j=0}^{k} \frac{1}{G\theta_j} \left(\nabla h(\lambda_{j+1}) - \nabla h(y_j)\right)^\top \lambda - D_h(\lambda, \lambda_0) \Big\}. \quad (3.17)$$

Hence, acc-BPP can also be rewritten as Algorithm 6. Based on the dual averaging interpretation, we show that Algorithm 6 admits a simpler convergence proof, which will be further used to prove the primal convergence of accelerated BALM in the next section.

Algorithm 6: acc-BPP3

Input: $\lambda_0 \in \Lambda, v_0 = \lambda_0, G, \theta_0 = 1, \{\eta_k\}_{k\geq 0}$

for $k \geq 0$ **do**

- $y_k = \theta_k v_k + (1-\theta_k)\lambda_k$
- $\lambda_{k+1} \in \arg\max_{\lambda\in\Lambda}\{d(\lambda) - \frac{1}{\eta_k} D_h(\lambda, y_k)\}$
- $v_{k+1} = \arg\max_{\lambda\in\Lambda}\{-GD_h(\lambda,\lambda_0) + \sum_{j=0}^{k} \frac{\eta_j}{\theta_j}\Big(d(\lambda_{j+1}) + \frac{1}{\eta_j}(\nabla h(\lambda_{j+1}) - \nabla h(y_j))^\top(\lambda - \lambda_{j+1})\Big)\}$
- Update θ_{k+1} such that $\sum_{j=0}^{k+1} \frac{\eta_j}{\theta_j} = \frac{\eta_{k+1}}{\theta_{k+1}^2}$, satisfied by $\frac{\eta_k}{\theta_k^2} = \frac{\eta_{k+1}}{\theta_{k+1}^2} - \frac{\eta_{k+1}}{\theta_{k+1}}$.

Theorem 3.2.1 *Let $S_k = \sum_{j=0}^{k} \frac{\eta_j}{\theta_j}$. Under Assumption 3.1.1, Algorithm 6 satisfies the relation:*

$$S_k d(\lambda_{k+1}) \geq \max_{\lambda\in\Lambda} \tilde{\phi}_k(\lambda), \quad (3.18)$$

where $\tilde{\phi}_k(\lambda) := -GD_h(\lambda,\lambda_0) + \sum_{j=0}^{k} \frac{\eta_j}{\theta_j}\big(d(\lambda_{j+1}) + \frac{1}{\eta_j}(\nabla h(\lambda_{j+1}) - \nabla h(y_j))^\top(\lambda - \lambda_{j+1})\big)$.

Proof. We prove the claim by induction. When $k = 0, \forall \lambda \in \Lambda$,

$$\begin{aligned}
\tilde{\phi}_0(\lambda) =& -GD_h(\lambda,\lambda_0) + \frac{\eta_0}{\theta_0} d(\lambda_1) + \frac{1}{\theta_0}(\nabla h(\lambda_1) - \nabla h(y_0))^\top(\lambda - \lambda_1) \\
=& -GD_h(\lambda,\lambda_0) + \frac{\eta_0}{\theta_0} d(\lambda_1) + \frac{1}{\theta_0}(D_h(\lambda, y_0) - D_h(\lambda,\lambda_1) - D_h(\lambda_1, y_0)) \\
\leq& -GD_h(\lambda,\lambda_0) + \frac{\eta_0}{\theta_0} d(\lambda_1) + \frac{1}{\theta_0} D_h(\lambda,\lambda_0) \\
\leq& \frac{\eta_0}{\theta_0} d(\lambda_1).
\end{aligned}$$

Denote $\Delta_k = \frac{1}{\eta_k}[\nabla h(\lambda_{k+1}) - \nabla h(y_k)]$. Suppose now the relation is satisfied for k, namely, $\tilde{\phi}_k(\lambda) \le S_k d(\lambda_{k+1}), \forall \lambda \in \Lambda$. Then we have,

$$\begin{aligned}
\tilde{\phi}_{k+1}(\lambda) &\overset{\langle 1\rangle}{=} \tilde{\phi}_k(\lambda) + \frac{\eta_{k+1}}{\theta_{k+1}}\left(d(\lambda_{k+2}) + \Delta_{k+1}^\top(\lambda - \lambda_{k+2})\right) \\
&\overset{\langle 2\rangle}{\le} \tilde{\phi}_k(v_{k+1}) - GD_h(\lambda, v_{k+1}) + \frac{\eta_{k+1}}{\theta_{k+1}}\left(d(\lambda_{k+2}) + \Delta_{k+1}^\top(\lambda - \lambda_{k+2})\right) \\
&\overset{\langle 3\rangle}{\le} S_k d(\lambda_{k+1}) + \frac{\eta_{k+1}}{\theta_{k+1}}\left(d(\lambda_{k+2}) + \Delta_{k+1}^\top(\lambda - \lambda_{k+2})\right) - GD_h(\lambda, v_{k+1}) \\
&\overset{\langle 4\rangle}{\le} S_k\left(d(\lambda_{k+2}) + \Delta_{k+1}^\top(\lambda_{k+1} - \lambda_{k+2})\right) \\
&\qquad + \frac{\eta_{k+1}}{\theta_{k+1}}\left(d(\lambda_{k+2}) + \Delta_{k+1}^\top(\lambda - \lambda_{k+2})\right) - GD_h(\lambda, v_{k+1})
\end{aligned}$$

where step $\langle 1\rangle$ is a trivial identity based on the definition of $\tilde{\phi}_{k+1}$, step $\langle 2\rangle$ uses the strong convexity of $\tilde{\phi}_k$, step $\langle 3\rangle$ uses the induction hypothesis, step $\langle 4\rangle$ uses (3.5). Now since

$$\frac{S_k}{\eta_{k+1}(1-\theta_{k+1})} = \frac{\eta_{k+1}(1-\theta_{k+1})/\theta_{k+1}^2}{\eta_{k+1}(1-\theta_{k+1})} = \frac{1}{\theta_{k+1}^2},$$

we have

$$\begin{aligned}
&S_k\Delta_{k+1}^\top(\lambda_{k+1} - \lambda_{k+2}) + \frac{\eta_{k+1}}{\theta_{k+1}}\Delta_{k+1}^\top(\lambda - \lambda_{k+2}) \\
=&\frac{1}{\theta_{k+1}^2}(\nabla h(\lambda_{k+2}) - \nabla h(y_{k+1}))^\top((1-\theta_{k+1})\lambda_{k+1} + \theta_{k+1}\lambda - \lambda_{k+2}) \\
\overset{\langle 5\rangle}{\le}&\frac{1}{\theta_{k+1}^2}D_h((1-\theta_{k+1})\lambda_{k+1} + \theta_{k+1}\lambda, y_{k+1}) \\
\overset{\langle 6\rangle}{\le}&GD_h(\lambda, v_{k+1}).
\end{aligned}$$

Here step $\langle 5\rangle$ applies the three-point identity (1.13), and step $\langle 6\rangle$ uses the triangle-scaling property (3.2). As a result, it follows that $\tilde{\phi}_{k+1}(\lambda) \le S_{k+1}d(\lambda_{k+2}), \forall \lambda \in \Lambda$. ∎

Recall the optimality condition (3.5) for the λ-update, we have $\forall \lambda \in \Lambda$,

$$\sum_{j=0}^{k}\frac{\eta_j}{\theta_j}\left(d(\lambda_{j+1}) + \frac{1}{\eta_j}(\nabla h(\lambda_{j+1}) - \nabla h(y_j))^\top(\lambda - \lambda_{j+1})\right) \ge S_k d(\lambda).$$

Combining with the above theorem, this implies that $S_k d(\lambda_{k+1}) \ge -GD_h(\lambda, \lambda_0)+$

$S_k d(\lambda)$. Therefore, we immediately obtain the convergence result:

Corollary 3.2.1 *For any $\lambda \in \Lambda$,*

$$d(\lambda) - d(\lambda_{k+1}) \leq \frac{GD_h(\lambda, \lambda_0)}{S_k} = \frac{\theta_k^2 GD_h(\lambda, \lambda_0)}{\eta_k}. \tag{3.19}$$

Next, we establish a bound on θ_k.

Proposition 3.2.1

$$\frac{\sqrt{\eta_k}}{\sum_{i=0}^{k} \sqrt{\eta_i}} \leq \theta_k \leq \frac{2\sqrt{\eta_k}}{\sum_{i=0}^{k} \sqrt{\eta_i}} \tag{3.20}$$

Proof. Let $t_i = \frac{1}{\theta_i}$, then the update $\frac{\eta_i}{\theta_i^2} = \frac{\eta_{i+1}}{\theta_{i+1}^2} - \frac{\eta_{i+1}}{\theta_{i+1}}$ is equivalent to

$$t_{i+1}^2 - t_{i+1} - \frac{\eta_i}{\eta_{i+1}} t_i^2 = 0, \text{ i.e., } t_{i+1} = \frac{1 + \sqrt{1 + 4\frac{\eta_i}{\eta_{i+1}} t_i^2}}{2}.$$

Therefore,

$$\frac{1}{2} + \frac{1}{2}\left(1 + 2\sqrt{\frac{\eta_i}{\eta_{i+1}}} t_i\right) \geq t_{i+1} \geq \frac{1}{2} + \sqrt{\frac{\eta_i}{\eta_{i+1}}} t_i.$$

which further implies $\sqrt{\eta_{i+1}} \geq \sqrt{\eta_{i+1}} t_{i+1} - \sqrt{\eta_i} t_i \geq \frac{1}{2}\sqrt{\eta_{i+1}}$. Taking summation over $i = 0, \ldots, k-1$, this leads to

$$\sum_{i=0}^{k-1} \sqrt{\eta_{i+1}} \geq \sqrt{\eta_k} t_k - \sqrt{\eta_0} t_0 \geq \frac{1}{2} \sum_{i=0}^{k-1} \sqrt{\eta_{i+1}}.$$

Therefore,

$$\frac{\sum_{i=0}^{k} \sqrt{\eta_i}}{\sqrt{\eta_k}} \geq t_k \geq \frac{\sum_{i=0}^{k} \sqrt{\eta_i}}{2\sqrt{\eta_k}}$$

and we obtain the desired result. ∎

From this result, we can conclude that,

$$d(\lambda) - d(\lambda_T) \leq \frac{4GD_h(\lambda, \lambda_0)}{\left(\sum_{k=0}^{T-1} \sqrt{\eta_k}\right)^2}, \quad \forall \lambda \in \Lambda,$$

which is the same as the previous case when $A \to +\infty$ (namely $\theta_0 = 1$).

3.3 Accelerated Bregman Augmented Lagrangian Method

Applying the acc-BPP algorithms to the dual problem associated with the linearly constrained convex programs would then lead to accelerated versions of BALM. We present in Algorithm 7, an accelerated BALM algorithm based on Algorithm 6, denoted as acc-BALM. As an immediate result, the dual sequence from acc-BALM converges in the rate of $\mathcal{O}\big(1/(\sum_{k=0}^{T-1}\sqrt{\eta_k})^2\big)$, which improves over the $\mathcal{O}(1/\sum_{k=0}^{T-1}\eta_k)$ of BALM. However, as discussed in the introduction, algorithms with an accelerated rate of dual convergence does not necessarily exhibit accelerated primal convergence. For example, the accelerated algorithm of ALM established in [27, 28, 29] with constant proximal parameters only ensures a $\mathcal{O}(1/T^2)$ rate for the dual convergence, namely, $L(x^*,\lambda^*) - L(x_T,\lambda_T) \leq \mathcal{O}(1/T^2)$, whereas the primal sequence converges only in the rate of $\mathcal{O}(1/T)$.

Below we show that the proposed acc-BALM algorithm based on the previous acceleration scheme also ensures acceleration on the primal convergence. From Section 2.1.2, we know that the key to prove primal convergence is to bound $L(\tilde{x}_T,\lambda) - L(x,\tilde{\lambda}_T)$, which further implies bounds for the primal objective $|f(\tilde{x}_T) - f(x^*)|$ and feasibility violation $\|A\tilde{x}_T - b\|$. The next theorem establishes the primal convergence rate for Algorithm 7.

Algorithm 7: acc-BALM

Input: $\lambda_0 \in \Lambda, v_0 = \lambda_0, G, \theta_0 = 1, \{\eta_k\}_{k\geq 0}$

for $k \geq 0$ **do**

$y_k = \theta_k v_k + (1-\theta_k)\lambda_k$

$x_{k+1} \in \arg\min\Big\{f(x) + \max_{\lambda\in\Lambda}\{\lambda^\top G(x) - \frac{1}{\eta_k}D_h(\lambda, y_k)\}\Big\}$

$\lambda_{k+1} = \arg\max_{\lambda\in\Lambda}\{\lambda^\top G(x_{k+1}) - \frac{1}{\eta_k}D_h(\lambda, y_k)\}$

$v_{k+1} = \arg\max_{\lambda\in\Lambda}\{-GD_h(\lambda,\lambda_0) + \sum_{j=0}^{k}\frac{\eta_j}{\theta_j}\Big(d(\lambda_{j+1}) + \frac{1}{\eta_j}(\nabla h(\lambda_{j+1}) - \nabla h(y_j))^\top(\lambda - \lambda_{j+1})\Big)\}$

Update θ_{k+1} such that $\sum_{j=0}^{k+1}\frac{\eta_j}{\theta_j} = \frac{\eta_{k+1}}{\theta_{k+1}^2}$, satisfied by $\frac{\eta_k}{\theta_k^2} = \frac{\eta_{k+1}}{\theta_{k+1}^2} - \frac{\eta_{k+1}}{\theta_{k+1}}$.

Theorem 3.3.1 *Denote*

$$\tilde{x}_T = \frac{\sum_{k=0}^{T-1}(\eta_k/\theta_k)x_{k+1}}{\sum_{k=0}^{T-1}\eta_k/\theta_k},\ \tilde{\lambda}_T = \frac{\sum_{k=0}^{T-1}(\eta_k/\theta_k)\lambda_{k+1}}{\sum_{k=0}^{T-1}\eta_k/\theta_k}.$$

Then for any $x \in \mathcal{X}, \lambda \in \Lambda$, we have

$$L(\tilde{x}_T, \lambda) - L(x, \tilde{\lambda}_T) \leq \frac{GD_h(\lambda, \lambda_0) + GD_h(\lambda^*, \lambda_0)\sum_{k=0}^{T-1}\theta_k}{S_{T-1}}, \tag{3.21}$$

where $S_{T-1} = \sum_{k=0}^{T-1}\eta_k/\theta_k$.

Proof. Using Theorem 3.2.1 and Lemma 2.1.2, we have $\forall x \in \mathcal{X}, \lambda \in \Lambda$,

$$\begin{aligned}
& L(\tilde{x}_T, \lambda) - L(x, \tilde{\lambda}_T) \\
\leq\ & \frac{1}{S_{T-1}}\sum_{k=0}^{T-1}(\eta_k/\theta_k)\left(L(x_{k+1}, \lambda) - L(x, \lambda_{k+1})\right) \\
\leq\ & \frac{1}{S_{T-1}}\Big(\sum_{k=0}^{T-1}(\eta_k/\theta_k)\big(\frac{1}{\eta_k}(\nabla h(\lambda_{k+1}) - \nabla h(y_k))^\top(\lambda - \lambda_{k+1})\big)\Big) \\
\leq\ & \frac{1}{S_{T-1}}\Big(GD_h(\lambda, \lambda_0) + \sum_{k=0}^{T-1}(\eta_k/\theta_k)(d(\lambda_T) - d(\lambda_{k+1}))\Big)
\end{aligned}$$

where the first inequality uses the fact that $L(x, \lambda)$ is convex in x and linear in λ, the second inequality applies Lemma 2.1.2, and the third inequality comes from Theorem 3.2.1. From Corollary 3.2.1, we have $d(\lambda_T) - d(\lambda_{k+1}) \leq d(\lambda^*) - d(\lambda_{k+1}) \leq \frac{\theta_k^2 GD_h(\lambda^*, \lambda_0)}{\eta_k}, \forall k$. It then leads to the desired result. ∎

The following result can be obtained following the same proof as Theorem 2.1.1.

Corollary 3.3.1 *Define $\rho_* = 2\|\lambda^*\| + 1$. Algorithm 7 satisfies that for problem* (1.1)

$$\max\left\{|f(\tilde{x}_T) - f(x^*)|, \|[G(\tilde{x}_T)]_+\|_\star\right\} \leq \frac{\max_{\lambda \in \mathcal{B}_{\rho_*}^+} GD_h(\lambda, \lambda_0)(1 + \sum_{k=0}^{T-1}\theta_k)}{S_{T-1}},$$

where $\mathcal{B}_\rho^+ = \{\lambda \in \mathbb{R}^m : \lambda \geq 0, \|\lambda\| \leq \rho\}$.

Discussions From Proposition 3.2.1, it is clear that $S_{T-1} \geq \left(\frac{\sum_{k=0}^{T-1}\sqrt{\eta_k}}{2}\right)^2$, and $\sum_{k=0}^{T-1}\theta_k \leq \sum_{k=0}^{T-1}\frac{2\sqrt{\eta_k}}{\sum_{i=0}^{k}\sqrt{\eta_i}}$. We now discuss the consequences of special

choices of $\{\eta_k\}_{k\geq 0}$. In particular, we consider choosing $\eta_k = \eta(k+1)^p, k = 0, 1, \cdots$, where $p \geq 0$. First observe that

$$\sum_{k=0}^{T-1} \sqrt{\eta_k} \geq \sum_{k=0}^{T-1} \sqrt{\eta} k^{p/2} \geq \sqrt{\eta} \int_0^T x^{p/2} dx \geq \frac{\sqrt{\eta} T^{p/2+1}}{p/2+1},$$

thus $S_{T-1} \geq \frac{\eta T^{p+2}}{(p+2)^2}$. In addition, we have

$$\sum_{k=0}^{T-1} \theta_k \leq \sum_{k=0}^{T-1} \frac{(p+2)(k+1)^{p/2}}{(k+1)^{p/2+1}} \leq \sum_{k=0}^{T-1} \frac{p+2}{k+1} \leq (p+2)\ln(T).$$

Therefore, when the proximal parameters are set to $\eta_k = \eta(k+1)^p$, the primal convergence rate of acc-BALM becomes $\mathcal{O}\left(\frac{\ln T}{T^{p+2}}\right)$, whereas the primal convergence rate of BALM is $\mathcal{O}\left(\frac{1}{T^{p+1}}\right)$. In particular, when the proximal parameters are fixed to a constant, namely $p = 0$, acc-BALM improves the primal convergence from $\mathcal{O}(\frac{1}{T})$ to $\mathcal{O}\left(\frac{\ln(T)}{T^2}\right)$. When the Bregman divergence is set to the Euclidean distance, and the constraints are linear equality constraints, our result recovers the existing result in [26] as a special case. However, it is worth mentioning that our acceleration scheme and proof techniques are fairly general, whereas the accelerated algorithm and the convergence proof in [26] heavily rely on the structure of linear equality constraints and only apply to classical ALM. We believe our result gives the first primal convergence analysis of accelerated ALM for problems with inequality constraints.

3.4 Numerical Experiments

In this section, we test the numerical performance of the proposed algorithms, particularly, acc-BPP in Algorithm 5 and acc-BALM in Algorithm 7, and compare to their non-accelerated counterparts.

We first consider two different convex problems:

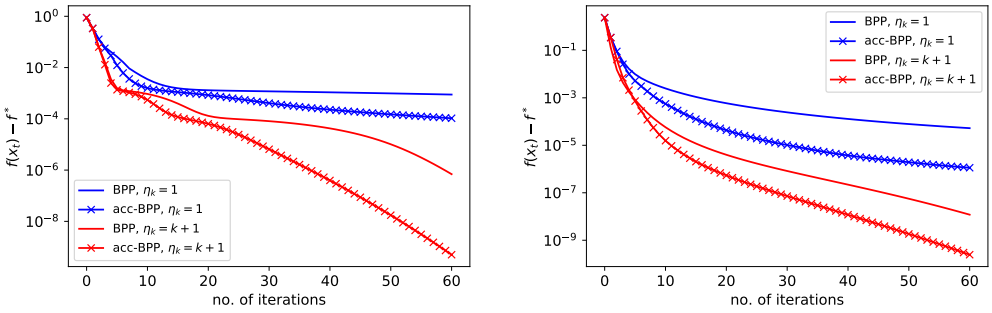
$$\min_{x\in\Delta_n} f_1(x) := \max\{c_1^\top x, \cdots, c_m^\top x\}, \tag{3.22a}$$

$$\min_{x\in\Delta_n} f_2(x) := \sum\nolimits_{j=1}^{m} \exp(a_j^\top x), \tag{3.22b}$$

where $\Delta_n := \{x \in \mathbb{R}^n : \sum_{i=1}^n x_i = 1, x \geq 0\}$ is the simplex set. These two examples represent nonsmooth and smooth convex objectives, respectively.

For both problems, we consider $m = 15, n = 20$. For problem (3.22a), each c_j is randomly generated from $U[-1, 1]^n$. For problem (3.22b), each a_j is also uniformly generated from $U[-1, 1]^n$.

We run BPP and acc-BPP to solve the above two problems. We choose the Bregman function $h(x) = \sum_{i=1}^n x_i \ln(x_i) : \Delta_n \to \mathbb{R}$, and the Bregman divergence is $D_h(x, y) = \sum_{i=1}^n (x_i \ln(x_i/y_i) - x_i + y_i)$, which is also known as the *generalized KL-divergence*. The Bregman operators (i.e., the optimal solutions to the proximal minimization steps) are obtained using ECOS conic programming solver version 2.0.7 [53]. Even though $D_h(x, y)$ here does not strictly satisfy the triangle-scaling property (3.2), we simply setting $G = 1$ in the experiment and consider two different choices of proximal parameters: $\eta_k = 1$ and $\eta_k = k+1$. Results are summarized in Figure 3.1, which indicates that acc-BPP achieves faster convergences than BPP under both settings.



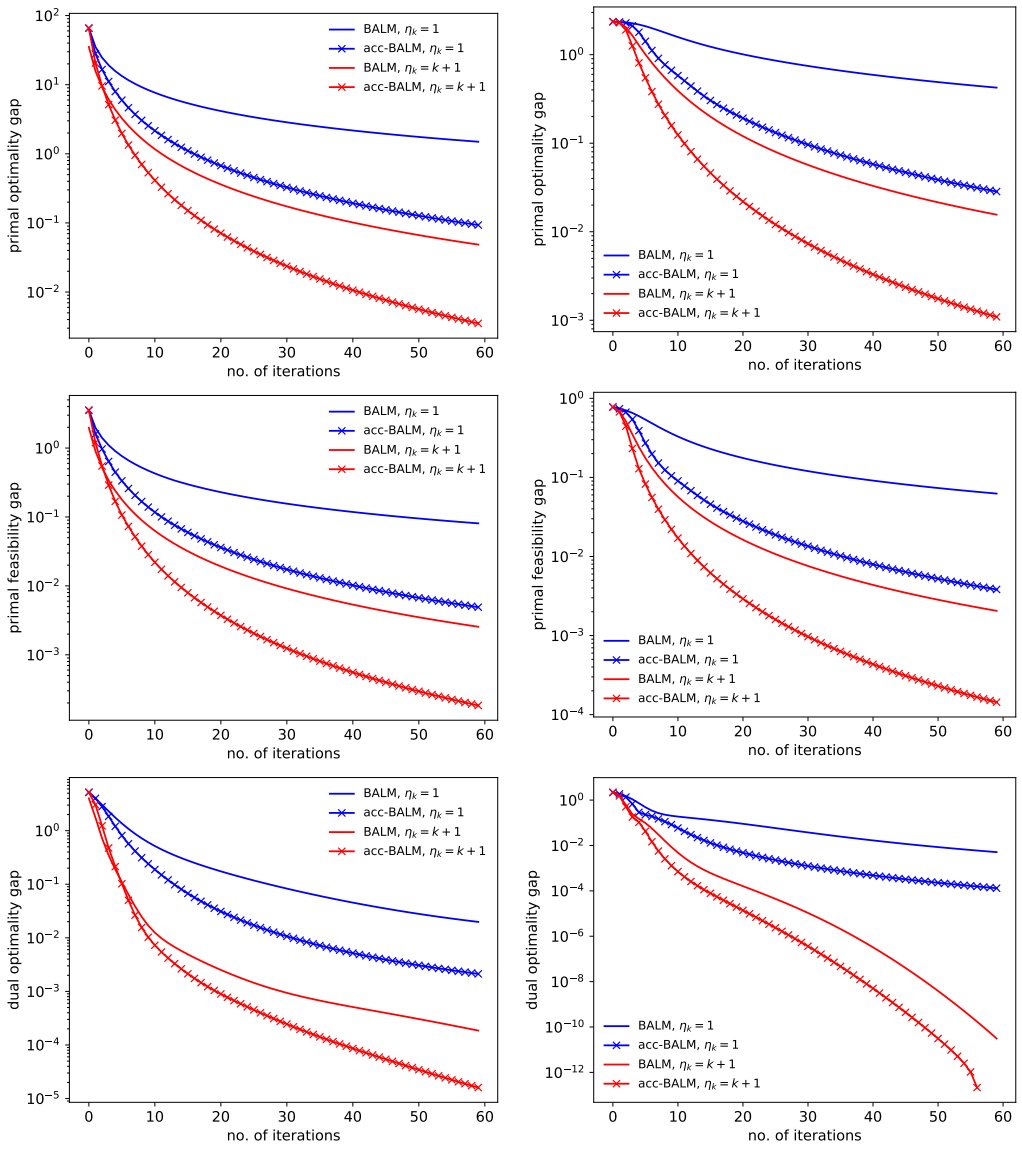
(a) problem (3.22a) (b) problem (3.22b)

Figure 3.1: Comparison of BPP and acc-BPP on convex problems (3.22a) and (3.22b)

Next, we consider another two convex minimization problems with linear inequality constraints for evaluating the performance of BALM and acc-BALM:

$$\min_x \{c^\top x : Ax \le b\}, \quad (3.23a) \qquad \min_x \left\{ \frac{1}{2} x^\top W x : Ax \le b \right\}, \quad (3.23b)$$

where $A \in \mathbb{R}^{m \times n}$. For both problems, we set $m = 150, n = 30$. For problem (3.23a), we first generate an instance of *Markov decision problem* [54], where each entry of transition probability is randomly generated from $U[0, 1]$ with normalization, and rewards are also uniformly generated from $U[0, 1]$. We then consider the linear program formulation associated with this finite



(a) problem (3.23a) (b) problem (3.23b)

Figure 3.2: Comparison of BALM and acc-BALM on convex problems (3.23a) and (3.23b)

MDP. For problem (3.23b), we set $W = \omega^\top \omega$, where $\omega \sim U[0,2]^n$, and $A \sim U[0,1]^{m\times n}, b \sim U[-1,1]^m$. We set the Bregman divergence to be the generalized KL-divergence for both BALM and acc-BALM. The optimal solutions to the subproblems are obtained using ECOS solver version 2.0.7 [53]. Again, we simply setting $G = 1$ in the experiment and consider two different choices of proximal parameters: $\eta_k = 1$ and $\eta_k = k+1$. Results are summarized in Figure 3.2, which clearly indicates that acc-BALM achieves faster

convergences than BALM under both settings.

3.5 Conclusions

We have established the first non-asymptotic primal convergence rate for BALM, which generalizes the classical ALM method. A generic accelerated scheme of the Bregman proximal point method is proposed, which is further used to construct the first accelerated BALM with both improved dual and primal convergence rates. Numerical experiments demonstrate that these accelerated algorithms achieve superior performance in practice. In the next section, we will explore the total iteration complexity of BALM and when subproblems are solved inexactly through some (stochastic) first-order subroutines, and the problem being considered will also be more general.

CHAPTER 4

APPLICATIONS TO REINFORCEMENT LEARNING

In this chapter, we apply BPP/BALM to reinforcement learning problems, and present numerical experiments demonstrating further the effectiveness of the acceleration scheme even when applied to problems with randomness.

4.1 Application of BPP to Reinforcement Learning: REPS

In [55, 56], BALM, or Bregman proximal point algorithm, is used to construct algorithms solving Markov decision problems with unknown transitions, also referred to as the reinforcement learning problem. The resulting algorithm is known as the Relative Entropy Policy Search, or REPS for short.

We consider the discounted MDP: $M = (\mathcal{X}, \mathcal{A}, P, r, \gamma)$. This model can be described as follows: a player observes a state $x_t \in \mathcal{X}$, selects and action $a_t \in \mathcal{A}$, the environment will bring the player to the next state according to the transition probability: $x_{t+1} \sim P(\cdot|x_t, a_t)$, and the player gets a reward $r(x_t, a_t)$. The goal of the player is to maximize long term discounted reward: $R = (1-\gamma)\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r(x_t, a_t)] = \sum_{x,a} \lambda(x,a) r(x,a)$, where $\lambda(x,a) = (1-\gamma)\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \mathbb{I}\{(x_t, a_t) = (x,a)\}]$ is the occupancy measure. For every occupancy measure λ, there is a corresponding policy $\pi_\lambda(a|x) = \frac{\lambda(x,a)}{\sum_{a'} \lambda(x,a')}$.

It is well-known that the MDP problem can be formulated as a linear programming problem [54].

$$\begin{aligned} \max_{\lambda \in \mathbb{R}^{\mathcal{X}\times\mathcal{A}}} \ & \langle \lambda, r \rangle \\ \text{s.t. } & E^\top \lambda = \gamma P^\top \lambda + (1-\gamma)\nu_0 \\ & \lambda \geq 0 \end{aligned} \tag{4.1a}$$

$$\begin{aligned} \min_{V \in \mathbb{R}^{\mathcal{X}}} \ & (1-\gamma)\langle \nu_0, V \rangle \\ \text{s.t. } & EV \geq r + \gamma PV \end{aligned} \tag{4.1b}$$

where ν_0 can be interpreted as the initial state distribution. $(E^\top\lambda)(x) = \sum_a \lambda(x,a)$, $(P^\top\lambda)(x) = \sum_{x',a'} P(x|x',a')\lambda(x',a')$. Therefore the two constraints above can be written explicitly as

$$\sum_a \lambda(x,a) = \gamma \sum_{x',a'} P(x|x',a')\lambda(x',a') + (1-\gamma)\nu_0(x), \quad \forall x$$

$$V(x) \geq r(x,a) + \gamma \sum_{x'} P(x'|x,a)V(x'), \quad \forall x,a$$

REPS can be obtained by by applying BPP to (4.1a). In Algorithm 8 and Algorithm 9, we present REPS with KL divergence and Euclidean distance.

Algorithm 8: REPS-KL

initialize λ_0

for $k = 0, 1, 2, \ldots, K-1$ **do**

$V_{k+1} = \min_V\ (1-\gamma)\langle \nu_0, V\rangle + \frac{1}{\eta}\sum_{x,a} \lambda_k(x,a) e^{\eta\delta(x,a)}$

$\lambda_{k+1}(x,a) = \lambda_k(x,a) e^{\eta\delta_{k+1}(x,a)}$

end

Algorithm 9: REPS-SQ

initialize λ_0

for $k = 0, 1, 2, \ldots, K-1$ **do**

$V_{k+1} = \min_V\ (1-\gamma)\langle \nu_0, V\rangle + \frac{1}{2\eta}\sum_{x,a} \max\{0, \lambda_k(x,a) + \eta\delta(x,a)\}^2$

$\lambda_{k+1}(x,a) = \max\{0, \lambda_k(x,a) + \eta\delta_{k+1}(x,a)\}$

end

where $\delta(x,a) = r(x,a) + \gamma\mathbb{E}_{x'\sim P(\cdot|x,a)}[V(x')|x,a] - V(x)$, and $\delta_k(x,a) = r(x,a) + \gamma\mathbb{E}_{x'\sim P(\cdot|x,a)}[V_k(x')|x,a] - V_k(x)$.

When the transition probability P is known, we already have experiments showing that the REPS-SQ is better than REPS-KL. However, when P is not known, the situation could be different. In this case, we assume that we have access to a sampling oracle such that we can sample transition pairs: (X, A, X', R). With these transition samples, one can define empirical objective function and perform λ update. More specifically, suppose we have

N samples $\{(X_i, A_i, X_i', R_i)\}_1^N$, then the update for REPS-KL becomes:

$$\begin{cases} V_{k+1} = \min\limits_{V} (1-\gamma)\langle \nu_0, V\rangle + \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} \dfrac{1}{\eta} e^{\eta(R_i+\gamma V(X_i')-V(X_i))}, \\ \lambda_{k+1}(x,a) \propto \lambda_k(x,a) e^{\eta(R_i+\gamma V_{k+1}(X_i')-V_{k+1}(X_i))}, \text{ (with normalization)} \end{cases} \tag{4.2}$$

the update for REPS-SQ becomes:

$$\begin{cases} V_{k+1} = \min\limits_{V} (1-\gamma)\langle \nu_0, V\rangle + \\ \qquad\qquad \dfrac{1}{2\eta N}\displaystyle\sum_{i=1}^{N} \max\{0, \lambda_k(x,a) + \eta(R_i + \gamma V(X_i') - V(X_i))\}^2, \\ \lambda_{k+1}(x,a) = \max\{0, \lambda_k(x,a) + \eta(R_i + \gamma V(X_i') - V(X_i))\}, \\ \text{(with projection onto the simplex)} \end{cases} \tag{4.3}$$

It should be noted that for REPS-KL, the state-action pairs are sampled from λ_k, while for REPS-SQ they are sampled from uniform distribution. Of course one can use samples from other distribution (importance sampling), provided that the distribution is known. Once the empirical objective is constructed, one can use methods such as gradient descent to solve the sub-problems.

Based on the acc-BPP described in Algorithm 5, one can design accelerated versions of REPS-SQ and REPS-KL, the sampled versions are similarly defined. In particular,

Algorithm 10: acc-REPS-KL

initialize $s_0 = \lambda_0, \theta_0 = 1, G$

for $k = 0, 1, 2, \ldots, K-1$ **do**

$y_k = \theta_k s_k + (1-\theta_k)\lambda_k$

$V_{k+1} = \min_V \ (1-\gamma)\langle \nu_0, V\rangle + \frac{1}{\eta}\sum_{x,a} y_k(x,a) e^{\eta\delta(x,a)}$

$\lambda_{k+1}(x,a) = y_k(x,a) e^{\eta\delta_{k+1}(x,a)}$

$s_{k+1} = s_k \left(\frac{\lambda_{k+1}}{y_k}\right)^{1/G\theta_k}$

Update θ_{k+1} such that $\frac{1}{\theta_k^2} = \frac{1}{\theta_{k+1}^2} - \frac{1}{\theta_{k+1}}$

end

Algorithm 11: acc-REPS-SQ

initialize $s_0 = \lambda_0, \theta_0 = 1, G$

for $k = 0, 1, 2, \ldots, K-1$ **do**

$y_k = \theta_k s_k + (1-\theta_k)\lambda_k$

$V_{k+1} = \min_V \ (1-\gamma)\langle \nu_0, V\rangle + \frac{1}{2\eta}\sum_{x,a} \max\{0, y_k(x,a) + \eta\delta(x,a)\}^2$

$\lambda_{k+1}(x,a) = \max\{0, y_k(x,a) + \eta\delta_{k+1}(x,a)\}$

$s_{k+1} = s_k + \frac{1}{G\theta_k}(\lambda_k - y_k)$

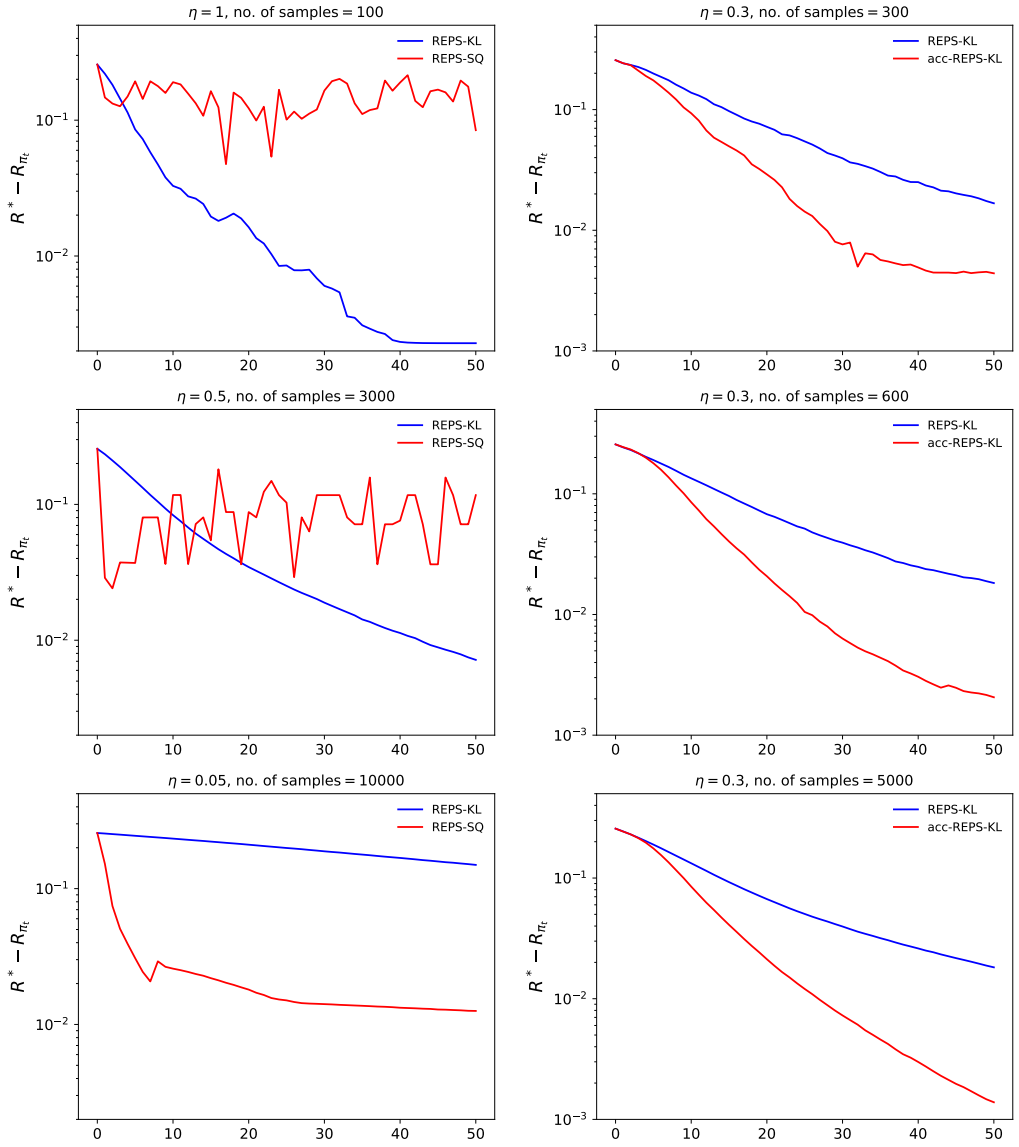
Update θ_{k+1} such that $\frac{1}{\theta_k^2} = \frac{1}{\theta_{k+1}^2} - \frac{1}{\theta_{k+1}}$

end

The plots in Figure 4.1 summarize comparison of REPS-KL with REPS-SQ, as well as comparison of REPS-KL with acc-REPS-KL for solving a randomly generated MDP problem with 10 states and 4 actions. The x-axis is the number of iterations, and the y-axis is the gap between the reward of the optimal policy and that of the output policy of the algorithm. All subproblems are solved by applying gradient descent with a fixed number of iterations.

On the left column, we compare REPS-KL with REPS-SQ for different choices of η and sample sizes. One can see that when η is large and sample size is small, which would lead to larger bias of the empirical objective of the subproblem, REPS-KL performs better and much more stable than REPS-SQ. On the other hand, when the η decreases and the sample size increases, which would cause the bias to go down, the REPS-SQ performs better than REPS-KL, which is consistent with the previous numerical results on exact BALM. In practice when sample size is limited, REPS-KL seems to be a better choice than REPS-SQ.

On the right column, we compare REPS-KL with acc-REPS-KL. It is demonstrated that on our example the acc-REPS-KL outperforms REPS-KL when the parameters are properly tuned, and that the acceleration effect is more significant with larger sample sizes.



(a) REPS-KL vs REPS-SQ (b) REPS-KL vs acc-REPS-KL

Figure 4.1: Comparison of REPS-KL with REPS-SQ, Comparison of REPS-KL with acc-REPS-KL.

REFERENCES

[1] A. Nemirovski, S. Onn, and U. G. Rothblum, "Accuracy certificates for computational problems with convex structure," *Mathematics of Operations Research*, vol. 35, no. 1, pp. 52–78, 2010.

[2] F. Facchinei and J.-S. Pang, *Finite-dimensional variational inequalities and complementarity problems*. Springer Science & Business Media, 2007.

[3] D. Kuhn, P. M. Esfahani, V. A. Nguyen, and S. Shafieezadeh-Abadeh, "Wasserstein distributionally robust optimization: Theory and applications in machine learning," in *Operations Research & Management Science in the Age of Analytics*. INFORMS, 2019, pp. 130–166.

[4] P. M. Esfahani and D. Kuhn, "Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations," *Mathematical Programming*, vol. 171, no. 1-2, pp. 115–166, 2018.

[5] Y. Liu, X. Yuan, S. Zeng, and J. Zhang, "Primal–dual hybrid gradient method for distributionally robust optimization problems," *Operations Research Letters*, vol. 45, no. 6, pp. 625–630, 2017.

[6] Y. Liu, A. Pichler, and H. Xu, "Discrete approximation and quantification in distributionally robust optimization," *Mathematics of Operations Research*, vol. 44, no. 1, pp. 19–37, 2019.

[7] F. Luo and S. Mehrotra, "Distributionally robust optimization with decision dependent ambiguity sets," *arXiv preprint arXiv:1806.09215*, 2018.

[8] G. Chen and M. Teboulle, "Convergence analysis of a proximal-like minimization algorithm using bregman functions," *SIAM Journal on Optimization*, vol. 3, no. 3, pp. 538–543, 1993.

[9] Y. Censor and S. A. Zenios, "Proximal minimization algorithm with D-functions," *Journal of Optimization Theory and Applications*, vol. 73, no. 3, pp. 451–464, 1992.

[10] A. Auslender and M. Teboulle, "Interior gradient and proximal methods for convex and conic optimization," *SIAM Journal on Optimization*, vol. 16, no. 3, pp. 697–725, 2006.

[11] M. R. Hestenes, "Multiplier and gradient methods," *Journal of optimization theory and applications*, vol. 4, no. 5, pp. 303–320, 1969.

[12] M. J. Powell, "A method for nonlinear constraints in minimization problems," *Optimization*, pp. 283–298, 1969.

[13] D. P. Bertsekas, *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.

[14] R. T. Rockafellar, "Augmented lagrangians and applications of the proximal point algorithm in convex programming," *Mathematics of operations research*, vol. 1, no. 2, pp. 97–116, 1976.

[15] G. Lan and R. D. Monteiro, "Iteration-complexity of first-order augmented lagrangian methods for convex programming," *Mathematical Programming*, vol. 155, no. 1-2, pp. 511–547, 2016.

[16] Y. Xu, "Iteration complexity of inexact augmented lagrangian methods for constrained convex programming," *arXiv preprint arXiv:1711.05812*, 2017.

[17] X. Chen, L. Guo, Z. Lu, and J. J. Ye, "An augmented lagrangian method for non-lipschitz nonconvex programming," *SIAM Journal on Numerical Analysis*, vol. 55, no. 1, pp. 168–193, 2017.

[18] M. F. Sahin, A. Alacaoglu, F. Latorre, V. Cevher et al., "An inexact augmented lagrangian framework for nonconvex optimization with nonlinear constraints," in *Advances in Neural Information Processing Systems*, 2019, pp. 13 943–13 955.

[19] H. Ouyang, N. He, L. Tran, and A. Gray, "Stochastic alternating direction method of multipliers," in *International Conference on Machine Learning*, 2013, pp. 80–88.

[20] Y. Xu, "Accelerated first-order primal-dual proximal methods for linearly constrained composite convex programming," *SIAM Journal on Optimization*, vol. 27, no. 3, pp. 1459–1484, 2017.

[21] O. Güler, "New proximal point algorithms for convex minimization," *SIAM Journal on Optimization*, vol. 2, no. 4, pp. 649–664, 1992.

[22] D. Kim, "Accelerated proximal point method for maximally monotone operators," *arXiv preprint arXiv:1905.05149*, 2019.

[23] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[24] P. Tseng, "Approximation accuracy, gradient methods, and error bound for structured convex optimization," *Mathematical Programming*, vol. 125, no. 2, pp. 263–295, 2010.

[25] H. Lin, J. Mairal, and Z. Harchaoui, "Catalyst acceleration for first-order convex optimization: from theory to practice," 2018.

[26] V. Nedelcu, I. Necoara, and Q. Tran-Dinh, "Computational complexity of inexact gradient augmented lagrangian methods: application to constrained mpc," *SIAM Journal on Control and Optimization*, vol. 52, no. 5, pp. 3109–3134, 2014.

[27] B. He and X. Yuan, "On the acceleration of augmented lagrangian method for linearly constrained optimization," 2010.

[28] Y.-f. Ke and C.-f. Ma, "An accelerated augmented lagrangian method for linearly constrained convex programming with the rate of convergence o (1/k 2)," *Applied Mathematics-A Journal of Chinese Universities*, vol. 32, no. 1, pp. 117–126, 2017.

[29] M. Kang, S. Yun, H. Woo, and M. Kang, "Accelerated bregman method for linearly constrained $\ell_1-\ell_2$ minimization," *Journal of Scientific Computing*, vol. 56, no. 3, pp. 515–534, 2013.

[30] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization," *submitted to SIAM Journal on Optimization*, vol. 2, p. 3, 2008.

[31] Y. Nesterov, "Smooth minimization of non-smooth functions," *Mathematical programming*, vol. 103, no. 1, pp. 127–152, 2005.

[32] O. Devolder, F. Glineur, and Y. Nesterov, "First-order methods of smooth convex optimization with inexact oracle," *Mathematical Programming*, vol. 146, no. 1-2, pp. 37–75, 2014.

[33] M. Teboulle, "Entropic proximal mappings with applications to nonlinear programming," *Mathematics of Operations Research*, vol. 17, no. 3, pp. 670–690, 1992.

[34] M. Teboulle, "A simplified view of first order methods for optimization," *Mathematical Programming*, vol. 170, no. 1, pp. 67–96, 2018.

[35] J. Yuan, K. Yin, Y.-G. Bai, X.-C. Feng, and X.-C. Tai, "Bregman-proximal augmented lagrangian approach to multiphase image segmentation," in *International Conference on Scale Space and Variational Methods in Computer Vision*. Springer, 2017, pp. 524–534.

[36] J. Eckstein, "Nonlinear proximal point algorithms using bregman functions, with applications to convex programming," *Mathematics of Operations Research*, vol. 18, no. 1, pp. 202–226, 1993.

[37] R. T. Rockafellar, *Convex analysis*. Princeton university press, 2015.

[38] I. Ekeland and R. Temam, *Convex analysis and variational problems*. SIAM, 1999, vol. 28.

[39] P. Tseng and D. P. Bertsekas, "On the convergence of the exponential multiplier method for convex programming," *Mathematical Programming*, vol. 60, no. 1-3, pp. 1–19, 1993.

[40] A. N. Iusem, "Augmented lagrangian methods and proximal point methods for convex optimization," *Investigación Operativa*, vol. 8, no. 11-49, p. 7, 1999.

[41] C. Kanzow and D. Steck, "Augmented lagrangian methods for the solution of generalized nash equilibrium problems," *SIAM Journal on Optimization*, vol. 26, no. 4, pp. 2034–2058, 2016.

[42] G. Scutari, D. P. Palomar, F. Facchinei, and J.-S. Pang, "Convex optimization, game theory, and variational inequality theory," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 35–49, 2010.

[43] B. Franci and S. Grammatico, "Distributed forward-backward algorithms for stochastic generalized nash equilibrium seeking," *arXiv preprint arXiv:1912.04165*, 2019.

[44] Y. Chen, G. Lan, and Y. Ouyang, "Accelerated schemes for a class of variational inequalities," *Mathematical Programming*, vol. 165, no. 1, pp. 113–149, 2017.

[45] A. Nemirovski, "Prox-method with rate of convergence o (1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems," *SIAM Journal on Optimization*, vol. 15, no. 1, pp. 229–251, 2004.

[46] Y. Nesterov, "On an approach to the construction of optimal methods of minimization of smooth convex functions," *Ekonomika i Mateaticheskie Metody*, vol. 24, no. 3, pp. 509–517, 1988.

[47] B. He and X. Yuan, "An accelerated inexact proximal point algorithm for convex minimization," *Journal of Optimization Theory and Applications*, vol. 154, no. 2, pp. 536–548, 2012.

[48] S. Salzo and S. Villa, "Inexact and accelerated proximal point algorithms," *Journal of Convex analysis*, vol. 19, no. 4, pp. 1167–1192, 2012.

[49] A. Hamdi and A. A. Mukheimer, "Convergence of an augmented lagrangian algorithm for solving minimization problems," *International Journal of Optimization: Theory, Methods and Applications*, vol. 1, no. 4, pp. 381–394, 2011.

[50] A. Hamdi, M. A. Noor, and A. Mukheimer, "Convergence of a proximal point algorithm for solving minimization problems," *Journal of Applied Mathematics*, vol. 2012, 2012.

[51] D. H. Gutman and J. F. Peña, "A unified framework for bregman proximal methods: subgradient, gradient, and accelerated gradient schemes," *arXiv preprint arXiv:1812.10198*, 2018.

[52] F. Hanzely, P. Richtarik, and L. Xiao, "Accelerated bregman proximal gradient methods for relatively smooth convex optimization," *arXiv preprint arXiv:1808.03045*, 2018.

[53] A. Domahidi, E. Chu, and S. Boyd, "Ecos: An socp solver for embedded systems," in *2013 European Control Conference (ECC)*. IEEE, 2013, pp. 3071–3076.

[54] M. L. Puterman, *Markov Decision Processes.: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

[55] J. Peters, K. Mülling, and Y. Altun, "Relative entropy policy search." in *AAAI*, vol. 10. Atlanta, 2010, pp. 1607–1612.

[56] J. Bas-Serrano, S. Curi, A. Krause, and G. Neu, "Logistic q-learning," *arXiv preprint arXiv:2010.11151*, 2020.