Convex Optimization Under Inexact First-order Information

A Thesis
Presented to
The Academic Faculty

by

Guanghui Lan

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology
August 2009

Convex Optimization Under Inexact First-order Information

Approved by:

Dr. Arkadi Nemirovski, Advisor
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Alexander Shapiro, Co-advisor
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Renato D.C. Monteiro, Co-advisor
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Shabbir Ahmed
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Anatoli Jouditski
Joseph Fourier University, Grenoble, France

Date Approved: June 18 2009

To my wife Zhaohui (Julene) Tong, my son Jesse and my forthcoming younger daughter

and

To my parents Yanxi Lan and Xuanhua Zhang

ACKNOWLEDGEMENTS

There have been many people who have helped me through my study at Georgia Tech. In the next few paragraphs, I would like to point out a few of these people to whom I am especially in debt.

First and foremost, I would like to express my deepest appreciation to my advisors, Arkadi Nemirovski, Alexander Shapiro and Ranato D.C. Monteiro. Professor Nemirovski has provided me with innumerous help during my Ph.D. study at Georgia Tech. Without his encouragement, support and guidance this thesis would not happen. Professor Shapiro has given me invaluable support and suggestions with respect to my research and career development during the past few years. Special thanks should go to Professor Monteiro, who has spent countless hours over the past four years guiding me through my academic pursuits. I would also like to thank Professors Shabbir Ahmed and Anatoli Jouditski, for serving as my thesis committee members. Their guidance and advice are invaluable to the completion of this thesis.

Next, I would like to thank Professor Craig Tovey, who found me from the dust during my first-year graduate study at Georgia Tech. His encouragement, help and suggestions are so important to me in the very begining of my academic life.

Then, I would like to thank my friends and fellow graduate students, to name a few, Altner Doug, Ricardo Fukasawa, Yi He, Fatma Kilinc-Karzan, Wenjing Li, Andriy Shapoval, Alejandro Toriello, Juan Pablo Vielma, Yang Zhang and Jieyun Zhou, who have made my life at school and away from school lots of fun. I hope that they know how much I have valued their friendships over the past few years, and I wish them all the best of luck in the future.

Finally, I would like to thank my family. I am deeply indebted to my wife, Zhaohui Tong, for her love, support and understanding and my lovely son, Jesse, for bringing me the happiest time in my life. I would like to thank my parents, my sisters and brother, for

their endless love, understanding and encouragement. Moreover, I would like to thank my mother-in-law, Hexiang Long, for being very supportive to us during my graduate study at Georgia Tech.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ABBREVIATIONS AND SYMBOLS

CP	convex programming or convex optimization
$\mathcal{FO}$	first-order oracle
SDP	semidefinite programming
SA	stochastic approximation
SAA	sample average approximation
AL	augmented Lagrangian
SCO	stochastic composite optimization
VI	variational inequality
$\mathcal{SO}$	stochastic oracle
E-SA	Euclidean stochastic approximation
N-SA	Non-Euclidean stochastic approximation
LHS	Latin Hyperplane Sampling
EU	expected utility
CVaR	Conditional Value-at-Risk
AC-SA	accelerated stochastic approximation
I-AL	inexact augmented Lagrangian

SUMMARY

The research on convex optimization under the first-order oracle started in 1970, and reached the first peak period from 1975 to 1985 that was terminated by the explosion of interior-point methods. Since the iteration cost of Newton-based interior-point methods is highly demanding for large-scale convex programming, first-order methods recently attract a lot of interest for their cheap iteration cost. These methods are advantageous over interior-point methods when the desired solution accuracy is moderate. The past few years has witnessed the success of first-order methods in solving a diverse set of problems arising from combinatorial optimization, machine learning, data mining, compressed sensing, etc.

In many situations the information returned by the first-order oracle is inexact. One prominent example is given by the classic stochastic programming where the objective function is given in the form of expectation. One can only expect to obtain an unbiased estimator of the objective value and its subgradient due to the difficulty of computing the expectation to high accuracy. Moreover, inexact first-order information often appears in certain deterministic optimization techniques which operate on the (sub)gradients of the dual problem. Sometimes, it is difficult to compute the exact (sub)gradients of the dual problem, and as a consequence, only approximate first-order information is available in reality for the circumstances described above. In this study we investigate the design and complexity analysis of the algorithms to solve convex optimization problems under inexact first-order information.

In the first part of this thesis we focus on the general non-smooth convex optimization under a stochastic oracle. We start by introducing the Mirror-descent Stochastic Approximation (SA) algorithm due to Nemirovski et. al. (2009) for solving this class of problems. By incorporating two important elements, namely, averaging the iterates and adapting to the problem's geometry, into the classic SA algorithm, this modified SA algorithm can significantly outperform other approaches, such as, the Sample Average Approximation (SAA)

for a certain class of convex programming problems. However, some issues related to the mirror-descent SA algorithm remain to be addressed. First of all, a long-standing problem for the SA methods is the absence of a validation procedure to estimate the accuracy of the generated solutions. On the other hand, an important methodological property of the SAA approach is that, with some additional effort, it can provide such estimates. To this end we show that while running a mirror descent SA procedure one can compute, with a small additional effort, lower and upper statistical bounds for the optimal objective value. We demonstrate that for a certain class of convex stochastic programing problems these bounds are comparable in quality with similar bounds computed by the SAA method, while their computational cost is considerably smaller. Moreover, the numerical study in Nemirovski et. al. (2009) focuses only on problems where the feasible set is a standard simplex. It is not clear how this algorithm behaves in practice for solving other convex stochastic programming problems. We then conduct extensive numerical experiments to understand the performance of the mirror descent SA algorithm for solving stochastic programing problems with a feasible set more complicated than a standard simplex.

In the second part of this thesis we consider the Stochastic Composite Optimization (SCO), an important class of convex programming problems whose objective function is given by the summation of a smooth and non-smooth component. Moreover, it is assumed that the only information available for the numerical scheme to solve these problems is the subgradients of the composite function contaminated by stochastic noise. Since SCO covers smooth, non-smooth and stochastic convex optimization as certain special cases, a lower bound on the rate of convergence for solving this class of problems immediately follows from the classical complexity theory for convex optimization. Note however that the optimization algorithms that can achieve this lower bound had never been developed. This is partly due to the difficulty that, although either smooth or nonsmooth minimization has been well-studied separately in the literature, a unified treatment for both of them seems highly non-trivial. Our contribution mainly consists of the following aspects. Firstly, with a novel analysis, it is demonstrated that the simple mirror descent SA algorithm applied to the aforementioned problems exhibits the best known so far rate of convergence guaranteed by a

more involved stochastic mirror-prox algorithm. Moreover, by properly modifying a variant of Nesterov's optimal method for smooth convex optimization, we propose an accelerated SA, which can achieve the theoretically optimal rate of convergence for solving this class of problems. Clearly, the accelerated SA algorithm is a universally optimal method for non-smooth, smooth and stochastic convex optimization. It should be stressed that Nesterov's optimal method and/or its variants were designed for solving deterministic smooth convex optimization problems. These algorithms, with very aggressive stepsizes employed, were believed to be too sophisticated to solve non-smooth and stochastic convex optimization problems. We, however, substantially extend the analysis of Nesterov's optimal method to non-smooth and stochastic convex optimization, and devise a novel (actually increasing) stepsize policy for solving these problems. Thirdly, we investigate this accelerated SA in more details, for example, derive the exponential bounds for the large deviations of the resulting solution inaccuracy from the expected one, provided the noise from the stochastic oracle is "light-tailed". Finally, the significant advantages of the accelerated SA over the existing algorithms are illustrated in the context of solving a class of stochastic programming problems whose feasible region is a simple compact convex set intersected with an affine manifold.

In the third part of this work, we investigate certain deterministic optimization technique, namely, the augmented Lagrangian method, applied to solve a special class of convex programming problems. It is well-known that the exact augmented Lagrangian method can be viewed as the gradient ascent method applied to the augmented dual. Moreover, to compute the gradient of the augmented dual, it is necessary to solve the so-called augmented subproblem. Since in most applications, the augmented subproblem can only be solved approximately, we are interested in analyzing the inexact version of the augmented Lagrangian (AL) method where the subproblems are approximately solved by means of Nesterov's optimal method. We establish a bound on the total number of Nesterov's optimal iterations, i.e., the inner iterations, performed throughout the entire inexact AL method to obtain a near primal-dual optimal solution. We also present variants with better iteration-complexity

bounds than the original inexact AL method, which consist of applying the original approach directly to a perturbed problem obtained by adding a strongly convex component to the objective function of the CP problem. We show that the iteration-complexity of the inexact AL methods for obtaining a near primal-dual optimal solution compares favorably with other penalty based approaches, such as the quadratic and exact penaly methods, and another possible approach for solving variational inequalities.

CHAPTER I

INTRODUCTION

In this chapter, we introduce some background and discuss the motivation for our research. In particular, we review the classic complexity theory for convex optimization and discuss some recent advancement in first-order convex programming (CP) methods in Section 1.1. In Section 1.2, we describe convex programming under the stochastic first-order oracle and review its main solution approaches. We then extend our discussion to the situation where the first-order information contains controllable deterministic errors in Section 1.3.

1.1 Complexity theory for convex optimization

In this section, we review a few classic complexity results for convex optimization, which were established by Nemirovski and Yudin through their fundamental work in [44].

1.1.1 General non-smooth convex optimization

Consider the convex programming problem

$$\begin{array}{lll} f^* := \min\limits_x & f(x) & \\ \text{s.t.} & g_i(x) \le 0, i = 1, \ldots, m, & \\ & x \in X, & \end{array} \tag{1.1.1}$$

where X is a compact convex set with a nonempty interior, the objective function f and constraints $g_i, i = 1, \ldots, m$, are convex continuous functions over X. Let $\mathcal{P}_m(X)$ denote the family of all feasible convex programming problems given in the above form. Clearly, due to the feasibility assumption and the compactness of X, the optimal value of (1.1.1) must be attained at certain feasible solution, i.e., problem (1.1.1) is solvable. We identify an instance $\mathcal{I}$ from the family $\mathcal{P}_m(X)$ by $\mathcal{I} = (f, g_1, \ldots, g_m)$.

In what follows we assume that $\mathcal{P}_m(X)$ is represented by the first-order oracle $\mathcal{FO}$ which, given the instance $\mathcal{I}$ and an input vector $x \in \operatorname{int} X$, outputs the values and some

subgradients of the objective function and constraints at the point x. Hence, $\mathcal{FO}$ can be defined as a map from $\operatorname{int} X$ to $\Re^{(m+1)\times(n+1)}$ given by

$$x \mapsto \mathcal{FO}(\mathcal{I}, x) = \left(f(x), f'(x); g_1(x), g_1'(x); \ldots; g_m(x), g_m'(x)\right).$$

A solution method, denoted by $\mathcal{M}$, when applied to instance $\mathcal{I}$, performs sequential calls to $\mathcal{FO}$ by supplying it with certain input x_i, which is called the i-th search point. While at the very first step, search point x_1 is generated by the method without knowing any information about $\mathcal{I}$, the i-th search point at step i is generated by the method based on the accumulated information. In other words, search point x_i can be defined as a certain function of the information obtained from $\mathcal{FO}$ during all the previous steps. The method should also perform the termination test from time to time and compute the output $\bar{x}(\mathcal{M}, \mathcal{I})$ whenever it decides to terminate. Note that both the termination test and the rule of computing the output should depend only on the first-order information accumulated to their corresponding moments. The total number of steps performed by the method $\mathcal{M}$, as applied to instance $\mathcal{I}$, is called the complexity (or iteration-complexity) $\operatorname{Compl}(\mathcal{M}, \mathcal{I})$ of $\mathcal{M}$ at $\mathcal{I}$. This quantity can be $+\infty$ if the method does not terminate for instance $\mathcal{I}$. Accordingly, the complexity for method $\mathcal{M}$ on the whole family $\mathcal{P}_M(X)$ is defined as

$$\operatorname{Compl}(\mathcal{M}) := \sup_{\mathcal{I} \in \mathcal{P}_M(X)} \operatorname{Compl}(\mathcal{M}, \mathcal{I}).$$

Moreover, given an approximate solution $x \in X$ for instance $\mathcal{I}$, we measure its accuracy by

$$\epsilon_r(\mathcal{I}, x) := \max\left\{\frac{f(x) - f^*}{\max_{x \in X} f(x) - f^*}, \frac{[g_1(x)]_+}{\max_{x \in X}[g_1(x)]_+}, \ldots, \frac{[g_m(x)]_+}{\max_{x \in X}[g_m(x)]_+}\right\}, \quad (1.1.2)$$

where $[\cdot]_+ := \max\{\cdot, 0\}$. We define the accuracy of method $\mathcal{M}$ applied to instance $\mathcal{I}$ by the accuracy of its output $\bar{x}(\mathcal{M}, \mathcal{I})$, i.e.,

$$\operatorname{Accur}(\mathcal{M}, \mathcal{I}) := \epsilon_r(p, \bar{x}(\mathcal{M}, \mathcal{I})), \quad (1.1.3)$$

and the accuracy of method $\mathcal{M}$ applied to the whole family $\mathcal{P}_M(X)$ by

$$\operatorname{Accur}(\mathcal{M}) := \sup_{\mathcal{I} \in \mathcal{P}_M(X)} \operatorname{Accur}(\mathcal{M}, \mathcal{I}).$$

Finally, the complexity of the family $\mathcal{P}_M(X)$ is defined as the best complexity of a method for solving problems from this family to a given accuracy, i.e.,

$$\text{Compl}\,(\epsilon) := \min_{\mathcal{M}}\{\text{Compl}\,(\mathcal{M}) : \text{Accur}\,(\mathcal{M}) \leq \epsilon\}. \tag{1.1.4}$$

The optimization methods that can achieve this complexity are called optimal methods for $\mathcal{P}_m(X)$.

In complexity theory, we are interested in establishing the lower and upper bounds for $\text{Compl}\,(\epsilon)$ defined in (1.1.4). A lower bound of $\text{Compl}\,(\epsilon)$ means that for whatever algorithms solving problems in $\mathcal{P}_M(X)$, there always exist a "bad" problem instance such that the number of steps performed by these algorithms can not be smaller than $\text{Compl}\,(\epsilon)$. On the other hand, an upper bound for $\text{Compl}\,(\epsilon)$ is always associated with a particular optimization algorithm and provides a bound on the total number of steps performed by this algorithm applied to the whole family $\mathcal{P}_M(X)$. We first state a major complexity result by Nemirovski and Yudin (1983) that provides the lower and upper bounds for solving general convex programming problems.

Theorem 1.1.1 *The complexity of the family $P_m(X)$ of general convex programming problems with m-constraints over a convex compact set $X \in \Re^n$ can be bounded by*

$$n\left\lfloor \frac{\ln(1/\epsilon)}{6\ln 2}\right\rfloor - 1 \leq \text{Compl}\,(\epsilon) \leq \lceil 2.181 n \ln(1/\epsilon)\rceil\,, \tag{1.1.5}$$

where the upper bound holds for any $\epsilon < 1$ and the lower bound holds for any $\epsilon < \bar{\epsilon}(X)$ where $\bar{\epsilon}(X) \geq 1/n^3$ for all $X \subseteq \Re^n$. In particular, we have that $\bar{\epsilon}(X) = 1$ if X is a paralellope and that $\bar{\epsilon}(X) = 1/n$ if X is an ellipsoid.

We now add a few remarks about the results stated in Theorem 1.1.1. First, the upper bound in (1.1.5) is given by the remarkable Ellipsoid method, the first linearly convergent method invented by Nemirovski and Yudin in 1976 ([45]) for solving general convex programming problems. By using the Ellipsoid method as a tool, Khachian ([28]) established the polynomial solvability of linear programming in 1979. In fact, with the invention of the Ellipsoid method, a generic convex optimization problem, under mild computability and regularity assumptions, became polynomially solvable (and thus "computationally tractable")

(see [6]). The Ellipsoid method, in the worst case, is incomparably better than Dantzig's Simplex method ([14]) for linear programming, but in practice the Ellipsoid method works more or less according to its theoretical efficiency bound while the Simplex method in real-world applications usually outperforms the Ellipsoid method. In 1984, Karmarkar in his seminal paper [27] proposed a completely new polynomial time algorithm for Linear Programming, namely, the interior point method. Karmarkar's algorithm turned out to be very efficient in practice and led to the so-called era of interior point methods for convex optimization (see, for example, [54], [77], [6] and [66]).

Second, while the upper bound of the complexity stated in (1.1.5) is valid for every $\epsilon \in (0,1)$, the lower bound is valid only for small enough ϵ, i.e., $\epsilon \leq \bar{\epsilon}(X)$. For example, if X is a standard Euclidean ball, then the lower bound in (1.1.5) is valid only for $\epsilon \leq 1/n$. Clearly, in view of Theorem 1.1.1, for small enough ϵ, namely, for $0 < \epsilon < \bar{\epsilon}(X)$, the Ellipsoid method is an optimal method, up to an absolute constant factor, for solving $P_m(X)$. Note however that there exists an interval $[\bar{\epsilon}(X), 1)$ for which the lower bound in (1.1.5) is not valid and thus the Ellipsoid method is not optimal. Moreover, as the dimension n increases, the value of $\bar{\epsilon}(X)$ for general X decreases and hence the interval $[\bar{\epsilon}(X), 1)$ tends to cover all possible values of $\epsilon \in (0,1)$. Below we review an important result which provides valid lower and upper bounds on $\mathrm{Compl}\,(\epsilon)$ for every $\epsilon \in (0,1)$.

Before stating this result, we introduce a notion, namely, the *asphericity* κ of X, which essentially tells us how the set X differs from an Euclidean ball. More specifically, the *asphericity* κ is defined as the smallest ratio of radii of two concentric Euclidean balls V_{in} and V_{out} such that $V_{in} \subseteq X \subseteq V_{out}$.

Theorem 1.1.2 *The complexity of the family $P_m(X)$ of general convex programming problems with m-constraints over a convex compact set $X \in \Re^n$ of asphericity κ can be bounded by*

$$\min\left\{n, \left\lfloor \frac{1}{(2\kappa\epsilon)^2} \right\rfloor\right\} \leq \mathrm{Compl}\,(\epsilon) \leq \left\lceil \frac{4\kappa^2}{\epsilon^2} \right\rceil, 0 < \epsilon < 1. \tag{1.1.6}$$

We now make a few comments about the above result. First, the upper bound on $\mathrm{Compl}\,(\epsilon)$ stated in (1.1.6) is achieved by the simple subgradient method. Notice that, for

a given κ, this upper bound is independent on the dimension n of the problem. Second, when the dimension of the domain is large enough for given κ and ϵ, i.e., when

$$n \geq \frac{1}{(2\kappa\epsilon)^2}, \tag{1.1.7}$$

it can be easily seen that the upper bound stated in (1.1.6) is equivalent to the lower complexity bound up to a constant factor. Therefore, the subgradient method is an optimal method for solving general large-scale convex programming problems $P_m(X)$ for which condition (1.1.7) holds. Second, in view of Theorem 1.1.2, theoretically speaking, no algorithms can perform much better than the simple subgradient method for solving general large-scale convex programming problems. The only way to improve the performance of the algorithms would be to develop specialized algorithms for solving certain subclasses of problems in $P_m(X)$. We are about to review in next subsection an important complexity result of this type, namely, the complexity of smooth convex optimization.

1.1.2 Smooth convex optimization and Nesterov's optimal method

Consider the minimization problem

$$f^* := \min_{x \in X} f(x), \tag{1.1.8}$$

where $X \subseteq \Re^n$ is a closed convex set and f is convex and continuously differentiable with Lipschitz continuous gradient over X with respect to a given arbitrary norm $\|\cdot\|$ in $\Re^n$, i.e.,

$$\|\nabla f(x) - \nabla f(y)\|_* \leq L\|x - y\|, \quad \forall x, y \in \Re^n,$$

where $\|\cdot\|_*$ denotes the conjugate norm given by $\|g\|_* := \max_{\|x\| \leq 1} \langle g, x \rangle$. We assume that the optimal value f^* of problem (1.1.8) is finite and that its set of optimal solutions $X^* := \mathrm{Argmin}_{x \in \Re^n} f(x)$ is nonempty. Moreover, the distance from the initial point x_0 to the set of optimal solutions is bounded by R, i.e.,

$$\inf_{x^* \in X^*} \|x_0 - x^*\| \leq R.$$

Letting $\mathcal{S}_n(L, R)$ denote the family of convex programming problems given in this form, we identify an instance $\mathcal{I}$ from $\mathcal{S}_n(L, R)$ by $\mathcal{I} = (f)$.

Note that, while we use the relative accuracy (cf. (1.1.2)) to measure the quality of an approximate solution in the previous subsection, we measure the quality of a given approximation solution $x \in X$ for an instance $\mathcal{I}$ from $\mathcal{S}_{L,R}$ by its absolute accuracy, i.e.,

$$\epsilon_a(x, I) := f(x) - f^*.$$

Accordingly, we replace $\epsilon_r(x, I)$ in the definition of Accur $(\mathcal{M}, \mathcal{I})$ (cf. (1.1.3)) by $\epsilon_a(x, I)$.

Nemirovski and Yudin [44] provides the following lower bound regarding the complexity of $\mathcal{S}_n(L, R)$. It is worth noting that this lower bound is obtained through the construction of a class of unconstrained quadratic programming instances from $\mathcal{S}_n(L, R)$.

Theorem 1.1.3 *The complexity of the family $\mathcal{S}_n(L, R)$ of smooth convex programming problems can be bounded from below by*

$$\text{Compl}\,(\epsilon) \geq \min\left\{\frac{n-1}{2}, \sqrt{\frac{3LR^2}{32\epsilon}} - 1\right\}. \tag{1.1.9}$$

In [44], Nemirovski and Yudin also provide a nearly optimal method, up to a logarithmic factor, for solving $\mathcal{S}_n(L, R)$. In a series of work ([47, 48]), Nesterov presented novel algorithms for solving problem (1.1.8) whose iteration-complexity is bounded by $\mathcal{O}(\sqrt{LR^2/\epsilon})$. Clearly, in view of (1.1.9), Nesterov's methods are optimal, up to a constant factor, for solving $\mathcal{S}_n(L, R)$ when the dimension n is large enough, i.e.,

$$n \geq \sqrt{\frac{3LR^2}{8\epsilon}} + 1.$$

Nesterov's method was further studied in [49], [1] and [50] using Bregman distance (see definition below). Nesterov ([49]) also developed certain extension of his method which exhibits nearly optimal, up to a logarithmic factor, rate of convergence for solving smooth convex optimization problems with smooth functional constraints. In this subsection, we will review a version of Nesterov's method for solving (1.1.8) presented in [50], while other variants of Nesterov's method for solving (1.1.8) can also be found, for example, in [32] and [73]. It is interesting to note that the construction of Nesterov's optimal method is a very nice example which demonstrates the importance of the complexity approach; the method for solving $\mathcal{S}_n(L, R)$ with the optimal rate of convergence was found mainly because the investigation of complexity enforced researchers to believe that such a method should exist.

Let $\omega : X \to \mathbf{R}$ be a differentiable strongly convex function with modulus $\alpha > 0$ with respect to $\|\cdot\|$, i.e.,

$$\omega(x) \geq \omega(\tilde{x}) + \langle \nabla\omega(\tilde{x}), x - \tilde{x}\rangle + \frac{\alpha}{2}\|x - \tilde{x}\|^2, \quad \forall x, \tilde{x} \in X. \tag{1.1.10}$$

The Bregman distance $d_\omega : X \times X \to \mathbf{R}$ associated with ω is defined as

$$d_\omega(x; \tilde{x}) \equiv \omega(x) - l_\omega(x; \tilde{x}), \quad \forall x, \tilde{x} \in X, \tag{1.1.11}$$

where $l_\omega : \Re^n \times X \to \mathbf{R}$ is the "linear approximation" of ω defined as

$$l_\omega(x; \tilde{x}) = \omega(\tilde{x}) + \langle \nabla\omega(\tilde{x}), x - \tilde{x}\rangle, \quad \forall (x, \tilde{x}) \in \Re^n \times X.$$

We are now ready to state Nesterov's smooth first-order method for solving (1.1.8). We use the superscript "*sd*" in the sequence obtained by taking a steepest descent step and the superscript "*ag*" (which stands for "aggregated gradient") in the sequence obtained by using all past gradients.

Nesterov's Algorithm:

0) Let $x_0^{sd} = x_0^{ag} \in X$ be given and set $k = 0$.

1) Set $x_k = \frac{2}{k+2} x_k^{ag} + \frac{k}{k+2} x_k^{sd}$ and compute $f(x_k)$ and $\nabla f(x_k)$.

2) Compute $(x_{k+1}^{sd}, x_{k+1}^{ag}) \in X \times X$ as

$$x_{k+1}^{sd} \in \operatorname{Argmin}\left\{ l_f(x; x_k) + \frac{L}{2}\|x - x_k\|^2 : x \in X \right\}, \tag{1.1.12}$$

$$x_{k+1}^{ag} \equiv \operatorname{argmin}\left\{ \frac{L}{\alpha} d_\omega(x; x_0) + \sum_{i=0}^{k} \frac{i+1}{2} [l_f(x; x_i)] : x \in X \right\}. \tag{1.1.13}$$

3) Set $k \leftarrow k + 1$ and go to step 1.

end

The main convergence result established by Nesterov [50] regarding the above algorithm is summarized in the following theorem.

Theorem 1.1.4 *The sequence $\{x_k^{sd}\}$ generated by Nesterov's optimal method satisfies*

$$f(x_k^{sd}) - f^* \leq \frac{4L\, d_\omega(x^*; x_0^{sd})}{\alpha\, k(k+1)}, \quad \forall k \geq 1,$$

where x^* is an optimal solution of (1.1.8). As a consequence, given any $\epsilon > 0$, an iterate $x_k^{sd} \in X$ satisfying $f(x_k^{sd}) - f^* \leq \epsilon$ can be found in no more than

$$\left\lceil\, 2\sqrt{\frac{d_\omega(x^*; x_0^{sd})L}{\alpha\epsilon}}\, \right\rceil \tag{1.1.14}$$

iterations.

The result stated in Theorem 1.1.4 gives us a bound on the estimated error at each iteration k in terms of the objective value, which is usually refered to as the *rate of convergence* or *convergence rate* of an optimization method. Clearly, we can derive the iteration-complexity of an optimization method from the convergence rate results. The following iteration-complexity result follows as an immediate special case of Theorem 1.1.4.

Corollary 1.1.1 *Suppose that $\|\cdot\|$ is a inner product norm and $h : X \to \Re$ is chosen as $\omega(\cdot) = \|\cdot\|^2/2$ in Nesterov's optimal method. Then, for any $\epsilon > 0$, an iterate $x_k^{sd} \in X$ satisfying $f(x_k^{sd}) - f^* \leq \epsilon$ can be found in no more than*

$$\left\lceil\, \|x_0^{sd} - x^*\| \sqrt{\frac{2L}{\epsilon}}\, \right\rceil \tag{1.1.15}$$

iterations, where x^* is an optimal solution of (1.1.8).

Proof. If $\omega(x) = \|x\|^2/2$, then (1.1.11) implies that $d_\omega(x^*; x_0^{sd}) = \|x_0^{sd} - x^*\|^2/2$. The corollary clearly follows from this fact and Theorem 1.1.4. ∎

Now assume that the objective function f is strongly convex over X, i.e., for some $\mu > 0$,

$$\langle \nabla f(x) - \nabla f(\tilde{x}), x - \tilde{x} \rangle \geq \mu \|x - \tilde{x}\|^2, \quad \forall x, \tilde{x} \in X. \tag{1.1.16}$$

Nesterov shows in Theorem 2.2.2 of [49] that, under the assumptions of Corollary 1.1.1, a variant of his optimal method finds a solution $x_k \in X$ satisfying $f(x_k) - f^* \leq \epsilon$ in no more than

$$\left\lceil\, \sqrt{\frac{L}{\mu}}\, \log \frac{L\, \|x_0^{sd} - x^*\|^2}{\epsilon}\, \right\rceil \tag{1.1.17}$$

iterations. The following result gives a slightly sharper iteration-complexity bound for Nesterov's optimal method that replaces the term $\log(L\|x_0^{sd} - x^*\|^2/\epsilon)$ in (1.1.17) with $\log(\mu\|x_0^{sd} - x^*\|^2/\epsilon)$. The proof of this result is given in Theorem 8 of [33].

Theorem 1.1.5 *Let $\epsilon > 0$ be given and suppose that the assumptions of Corollary 1.1.1 hold and that the function f is strongly convex with modulus μ. Then, the variant where we restart Nesterov's optimal method, with proximal function $\omega(\cdot) = \|\cdot\|^2/2$, every*

$$K := \left\lceil \sqrt{\frac{8L}{\mu}} \right\rceil \tag{1.1.18}$$

iterations finds a solution $\tilde{x} \in X$ satisfying $f(\tilde{x}) - f^* \leq \epsilon$ in no more than $K \max\{1, \lceil \log \mathcal{Q} \rceil\}$ iterations, where

$$\mathcal{Q} := \frac{\mu \|x_0^{sd} - x^*\|^2}{2\epsilon} \tag{1.1.19}$$

and $x^* := \operatorname{argmin}_{x \in X} f(x)$.

1.1.3 Recent advancement on first-order methods for convex optimization

As discussed in Subsection 1.1.2 (see Theorem 1.1.2), the iteration-complexity of any algorithms solving the general large-scale non-smooth convex programming problems can not be smaller than $\mathcal{O}(1/\epsilon^2)$. More recently, Nesterov in a very relevant paper [50] presented a first-order method to solve convex programming problems of the form (1.1.8) for an important and broad class of non-smooth convex objective functions with iteration-complexity bounded by $\mathcal{O}(1/\epsilon)$. Nesterov's approach consists of approximating an arbitrary function f from the class by a sufficiently close smooth one with Lipschitz continuous gradient and applying the optimal smooth method in [47, 50] to the resulting CP problem with f replaced by its approximation function. In a subsequent paper, Nemirovski [42] proposed an extra-gradient type first-order method for solving a slightly more general class of CP problems than the one considered by Nesterov [50] and also established an $\mathcal{O}(1/\epsilon)$ iteration-complexity bound for his method.

The theoretical breakthrough due to Nesterov [50] and Nemirovski [42] certainly attract a lot of attention and first-order methods are starting to regain the status of practically and

provably efficient algorithms for large-scale problems, effectively competing with interior-point methods in cases when running even a single iteration of a higher-order method becomes practically intractable. For example, these first-order methods due to Nesterov [47, 50] and Nemirovski [42] have recently been applied to certain semidefinite programming (SDP) problems with some special structures (see Lu et al. [39], Nesterov [53] and d'Aspremont [15]). Peña [56] used Nesterov's smooth method [47, 50] to successfully solve a special class of large-scale linear programming problems. Lan et. al. [32] proposed primal-dual convex (smooth and/or nonsmooth) minimization reformulations for general cone programming and compared three aforementioned first-order methods, namely, Nesterov's optimal method [47, 50], Nesterov's smooth approximation scheme [50], and Nemirovski's prox-method [42], applied to these reformulations. Partly motivated by this work, Tseng [73] developed some new variants for these methods mentioned above. In the application side, these first-order methods have been successfully used in sparse covariance selection, rank reduction in multivariate linear regression and compressed sensing etc. (see, for example, [16, 37, 38, 52, 4]).

As a final note of this section, in spite of the fact that either nonsmooth or smooth convex optimization has been well-studied separately in the literature and the effort that has been recently taken to apply the smooth convex optimization techniques for solving certain non-smooth convex optimization problems ([50, 42, 52, 73]), a unified treatment for solving general non-smooth and smooth convex optimization seems highly non-trivial. As a result, there does not exist an algorithm which can achieve the optimal rate of convergence for solving both smooth and nonsmooth convex optimization problems. We will demonstrate in Chapter 4 of this thesis that such a unified treatment is possible and then present an universally optimal method for solving both $\mathcal{P}_M(X)$ and $\mathcal{S}(L, R)$ based on properly modifying a variant of Nesterov's method.

1.2 Convex optimization under a stochastic first-order oracle

In the previous section, we reviewed a few important results for convex optimization under exact first-order information. In many situations the information returned by the first-order oracle is inexact. One prominent example is given by the classic stochastic programming:

$$\min_{x \in X} \big\{ f(x) := \mathbb{E}[F(x, \xi)] \big\}, \tag{1.2.1}$$

where $X \subset \mathbb{R}^n$ is a nonempty bounded closed convex set, ξ is a random vector whose probability distribution P is supported on set $\Xi \subset \mathbb{R}^d$ and $F : X \times \Xi \to \mathbb{R}$. We assume that for every $\xi \in \Xi$ the function $F(\cdot, \xi)$ is convex on X, and that the expectation

$$\mathbb{E}[F(x, \xi)] = \textstyle\int_{\Xi} F(x, \xi) dP(\xi) \tag{1.2.2}$$

is well defined and finite valued for every $x \in X$. It follows that function $f(\cdot)$ is convex and finite valued on X. Moreover, we assume that $f(\cdot)$ is continuous on X. Of course, continuity of $f(\cdot)$ follows from convexity if $f(\cdot)$ is finite valued and convex on a neighborhood of X. With these assumptions, (1.2.1) becomes a convex programming problem.

A basic difficulty of solving stochastic optimization problem (1.2.1) is that the multi-dimensional integral (expectation) (1.2.2) cannot be computed with a high accuracy for dimension d, say, greater than 5. There exist two competitive computational approaches for solving (1.2.1) based on Monte Carlo sampling techniques, namely, the *Stochastic Approximation* (SA) and the *Sample Average Approximation* (SAA) methods. To this end we make the following assumptions.

(A1) It is possible to generate an iid sample $\xi_1, \xi_2, ...$, of realizations of random vector ξ.

(A2) There is a mechanism which for every given $x \in X$ and $\xi \in \Xi$ returns value $F(x, \xi)$ and a *stochastic subgradient* – a vector $\mathsf{G}(x, \xi)$ such that $\mathsf{g}(x) := \mathbb{E}[\mathsf{G}(x, \xi)]$ is well defined and is a subgradient of $f(\cdot)$ at x, i.e., $\mathsf{g}(x) \in \partial f(x)$. This mechanism will be referred to as the *Stochastic Oracle* ($\mathcal{SO}$).

Recall that if $F(\cdot, \xi)$, $\xi \in \Xi$, is convex and $f(\cdot)$ is finite valued in a neighborhood of a

point x, then (cf., Strassen [71])

$$\partial f(x) = \mathbb{E}\left[\partial_x F(x,\xi)\right]. \tag{1.2.3}$$

In that case we can employ a measurable selection $\mathsf{G}(x,\xi) \in \partial_x F(x,\xi)$ as a stochastic subgradient.

Both approaches, the SA and SAA methods, have a long history. The SA method is going back to the pioneering paper by Robbins and Monro [61]. Since then stochastic approximation algorithms became widely used in stochastic optimization (see, e.g., [7, 17, 18, 57, 64, 31] and references therein) and, due to especially low demand for computer memory, in signal processing. In the classical analysis of the SA algorithm (it apparently goes back to the works [13] and [65]) it is assumed that f is twice continuously differentiable and strongly convex, and in the case when the minimizer of f belongs to the interior of X, exhibits asymptotically optimal rate of convergence $\mathbb{E}[f(x_t) - f_*] = O(1/t)$ (here x_t is t-th iterate and f_* is the minimal value of $f(x)$ over $x \in X$). This algorithm, however, is very sensitive to a choice of the respective stepsizes. Since "asymptotically optimal" stepsize policy can be very bad in the beginning, the algorithm often performs poorly in practice (e.g., [70], Section 4.5.3.).

An important improvement of the SA method was developed by Polyak [58] and Polyak and Juditsky [59], where longer stepsizes were suggested with consequent averaging of the obtained iterates. Under the outlined "classical" assumptions, the resulting algorithm exhibits the same optimal $O(1/t)$ asymptotical convergence rate, while using an easy to implement and "robust" stepsize policy. It should be mentioned that the main ingredients of Polyak's scheme – long steps and averaging – were, in a different form, proposed already in [46] for the case of problems (1.2.1) with general type Lipschitz continuous convex objectives and for convex-concave saddle point problems. The algorithms from [46] exhibit, in a non-asymptotical fashion, the unimprovable in the general convex case $O(1/\sqrt{t})$-rate of convergence. For a summary of early results in this direction, see [44].

The SAA approach was used by many authors in various contexts under different names.

Its basic idea is rather simple: generate a (random) sample $\xi_1, ..., \xi_N$, of size N, and approximate the "true" problem (1.2.1) by the sample average problem

$$\min_{x \in X} \left\{ \hat{f}_N(x) := N^{-1} \sum_{j=1}^{N} F(x, \xi_j) \right\}. \tag{1.2.4}$$

Note that the SAA method is not an algorithm, the obtained SAA problem (1.2.4) still has to be solved by an appropriate numerical procedure. Recent theoretical studies (cf., [30, 68, 69]) and numerical experiments (see, e.g., [36, 40, 74]) show that the SAA method coupled with a good (deterministic) algorithm could be reasonably efficient for solving certain classes of two stage stochastic programming problems. On the other hand classical SA type numerical procedures typically performed poorly for such problems. Recently, Nemirovski et. al [43] demonstrated that a properly modified SA approach can be competitive and even significantly outperform the SAA method for a certain class of stochastic programming problems. The mirror descent SA method they introduced (cf. Chapter 2) is a direct descendant of the stochastic mirror descent method of Nemirovski and Yudin ([44]). However, the method developed in [43] is more flexible than its "ancestor": the iteration of the method is exactly the prox-step for a chosen prox-function, and the choice of prox-type function is not limited to the norm-type distance-generating functions. Close techniques, based on subgradient averaging, have been proposed in Nesterov [51] and used in [24, 26] to solve certain stochastic optimization problems of the form (1.2.1).

Several issues related to the mirror descent SA algorithm still remain to be addressed. First of all, a long-standing problem for the SA methods is the absence of a validation procedure to estimate the accuracy of the generated solutions. On the other hand, an important methodological property of the SAA approach is that, with some additional effort, it can provide such estimates. Moreover, the numerical study in [43] focuses only on problems where the feasible set is a standard simplex. It is not clear how this algorithm behaves in practice for solving other convex stochastic programming problems. Finally, the mirror descent SA algorithm in [43] does not assume the differentiability of the objective function f. One natural question is whether we gain anything if f is differentiable or contains certain differentiable components. We will address all the above-mentioned issues

in Chapters 3 and 4 of this thesis.

1.3 Convex optimization with approximate first-order information

In the previous section, we considered convex optimization under a stochastic oracle which, upon request, outputs an unbiased estimator for certain subgradient of the objective function. In addition to that, inexact first-order information often appears in certain deterministic optimization techniques which operate on the (sub)gradients of the dual problem. Sometimes, to compute the exact (sub)gradients of the dual can be computationally expensive, for example, requiring to solve a complicated subproblem, and as a result, only approximate first-order information is available in reality for the circumstances described above. In this section, we consider first-order methods for a special class of convex programming problems based on an inexact version of the classical augmented Lagrangian (AL) approach, where the subproblems are approximately solved by means of Nesterov's optimal method.

The basic problem of interest is the CP problem

$$f^* := \min\{f(x) : \mathcal{A}(x) = 0,\, x \in X\}, \tag{1.3.1}$$

where $f : X \to \mathbf{R}$ is a convex function with Lipschitz continuous gradient, $X \subseteq \Re^n$ is a sufficiently simple compact convex set and $\mathcal{A} : \Re^n \to \Re^m$ is an affine function.

For the case where the feasible region consists only of the set X, or equivalently $\mathcal{A} \equiv 0$, Nesterov ([47, 50]) developed methods which can find a point $x \in X$ such that $f(x) - f^* \leq \epsilon$ in at most $\mathcal{O}(\epsilon^{-1/2})$ iterations (see Subsection 1.1.2). Moreover, each iteration of his method requires one gradient evaluation of f and computation of two projections onto X. It is shown that his method achieves, uniformly in the dimension, the lower bound on the number of iterations for minimizing convex functions with Lipschitz continuous gradient over a closed convex set. When $\mathcal{A}$ is not identically 0, Nesterov's optimal method can still be applied directly to problem (1.3.1) but this approach would require the computation of projections onto the feasible region $X \cap \{x : \mathcal{A}(x) = 0\}$, which for most practical problems is as expensive as solving the original problem itself. An alternative approach for solving (1.3.1) when $\mathcal{A}$

is not identically 0 is to use first-order methods whose iterations require only computation of projections onto the simple set X.

Following this line of investigation, Lan and Monteiro [33] studied two first-order methods for solving (1.3.1) based on two well-known penalization approaches, namely: the quadratic and the exact penalization approaches. Iteration-complexity bounds for these methods are then derived to obtain two types of near optimal solutions of (1.3.1), namely: near primal and near primal-dual optimal solutions. Variants with possibly better iteration-complexity bounds than the aforementioned methods are also discussed. In this work, we still consider another first-order approach for solving (1.3.1) based on the classical augmented Lagrangian approach, where the subproblems are approximately solved by means of Nesterov's optimal method. As a by-product, alternative first-order methods for solving (1.3.1) involving only computation of projections onto the simple set X are obtained.

The augmented Lagrangian method, initially proposed by Hestenes [21] and Powell [60] in 1969, is currently regarded as an effective optimization method for solving large-scale nonlinear programming problems (see textbooks or monographs: [8], [9], [20], [55], [63]). More recently, it has been used by the convex programming community to develop specialized first-order methods for solving large-scale semidefinite programming problems (see, for example, Burer and Monteiro [11, 12], Jarre and Rendl [23], Zhao et al. [78]), due to its lower iteration-cost compared to that of Newton-based interior-point methods. The augmented Lagrangian method applied to problem (1.3.1) consists of solving a sequence of subproblems of the form

$$d_\rho(\lambda_k) := \min_{x\in X} \left\{ \mathcal{L}_\rho(x,\lambda_k) := f(x) + \langle \lambda_k, \mathcal{A}(x)\rangle + \frac{\rho}{2}\|\mathcal{A}(x)\|^2 \right\}, \tag{1.3.2}$$

where $\rho > 0$ is a given penalty parameter and $\|\cdot\|$ is the norm associated with a given inner product $\langle\cdot,\cdot\rangle$ in $\Re^m$. The multiplier sequence $\{\lambda_k\}$ is generated according to the iterations

$$\lambda_{k+1} = \lambda_k + \rho\mathcal{A}(x_k^*), \tag{1.3.3}$$

where x_k^* is a solution of problem (1.3.2). Since in most cases (1.3.2) can only be solved approximately, x_k^* in (1.3.3) is replaced by an η_k-approximate solution of (1.3.2), i.e., a point $x_k \in X$ such that $\mathcal{L}_\rho(x,\lambda_k) - d_\rho(\lambda_k) \le \eta_k$. The inexact augmented Lagrangian method

obtained in this manner, where the subproblems (1.3.2) are solved by Nesterov's method, is the main focus of our investigation in this thesis. More specifically, we are interested in establishing a bound on the total number of Nesterov's optimal iterations, i.e., the inner iterations, performed throughout the entire inexact AL method.

Several technical issues arise in the aforementioned iteration-complexity analysis of the inexact AL method. First of all, a termination criterion need to be specified for the inexact AL method. Second, it is well-known that $\mathcal{A}(x_k^*)$ is exactly the gradient of the function d_ρ defined in (1.3.2) at λ_k, and hence that (1.3.3) can be viewed as a steepest ascent iteration with stepsize ρ applied to the function d_ρ. Since, in the inexact AL method, we approximate $d_\rho(\lambda_k) = \mathcal{A}(x_k^*)$ by $\mathcal{A}(x_k)$, where x_k is an approximate solution of (1.3.2), we need to bound the error of the gradient approximation $\mathcal{A}(x_k)$, namely $\|\mathcal{A}(x_k) - \mathcal{A}(x_k^*)\|$, in terms of the accuracy η_k of the approximate solution x_k, and use this result to derive sufficient conditions on the sequence $\{\eta_k\}$ to guarantee the convergence of the corresponding inexact steepest ascent method $\lambda_{k+1} = \lambda_k + \rho\mathcal{A}(x_k)$. Third, as ρ increases, it is well-known that the iteration-complexity of approximately solving each subproblem (1.3.2) increases, while the number of dual iterations (1.3.3), i.e., the outer iterations, decreases. We intend to develop ways of choosing the parameter ρ so as to balance these two opposing criterions. More specifically, ρ is chosen so as to minimize the overall number of inner iterations performed by the inexact AL method. All these issues mentioned above will be addressed in Chapter 5.

1.4 Outline and main results of the thesis

This thesis is organized as follows.

In Chapter 2, we review the classic SA algorithm and introduce the mirror-descent SA method for solving problem (1.2.1). A basic difficulty of solving such stochastic optimization problems is that the involved multidimensional integrals (expectations) cannot be computed with high accuracy. The aim of this chapter is to compare two computational approaches based on Monte Carlo sampling techniques, namely, the SA and the SAA methods. Current opinion is that the SAA method can efficiently use a specific (say linear) structure of the considered problem, while the SA approach is a crude subgradient method which often

performs poorly in practice. We demonstrate that a properly modified SA approach, i.e., the mirror-descent SA method, can be competitive and even significantly outperform the SAA method for a certain class of convex stochastic problems, for example, when the set X is a standard simplex. We also present, in our opinion, highly encouraging results of numerical experiments.

A long-standing problem for the SA methods is the absence of a validation procedure to estimate the accuracy of the generated solutions. The main goal of Chapter 3, is to develop accuracy estimates for stochastic programming problems by employing SA type algorithms. To this end we show that while running a Mirror-descent SA procedure one can compute, with a small additional effort, lower and upper statistical bounds for the optimal objective value. We demonstrate that for a certain class of convex stochastic programs these bounds are comparable in quality with similar bounds computed by the SAA method, while their computational cost is considerably smaller. Moreover, We conduct extensive numerical experiments to understand the performance of the Mirror-descent SA algorithm for solving stochastic programing problems with a feasible set more complicated than a standard simplex.

In Chapter 4 we consider the Stochastic Composite Optimization (SCO), a class of convex programming problems whose objective function is given by the summation of a smooth and non-smooth component. Moreover, the numerical schemes only have access to the subgradients of the composite function itself. Since SCO covers both smooth and non-smooth minimization as certain special cases, a lower bound on the rate of convergence for solving this class of problems immediately follows from the classical complexity theory for convex optimization. Note however that the optimization algorithms that can achieve this lower bound had never been developed. This is partly due to the difficulty that, although either smooth or nonsmooth minimization has been well-studied separately in the literature, a unified treatment for both of them seems highly non-trivial. Our contribution mainly consists of the following aspects. Firstly, with a novel analysis, it is demonstrated that a slightly modified mirror descent SA algorithm applied to the aforementioned problems exhibits the best known so far rate of convergence guaranteed by a more involved stochastic mirror-prox

algorithm. Moreover, by properly modifying a variant of Nesterov's optimal method for smooth convex optimization, we propose an accelerated SA, which can achieve the theoretically optimal rate of convergence for solving this class of problems. Clearly, the accelerated SA algorithm is a universally optimal method for non-smooth, smooth and stochastic convex optimization. It should be stressed that Nesterov's optimal method and/or its variants were designed for solving deterministic smooth convex optimization problems. These algorithms, with very aggressive stepsizes employed, were believed to be too sophisticated to solve non-smooth and stochastic convex optimization problems. We, however, substantially extend the analysis of Nesterov's optimal method to non-smooth and stochastic convex optimization, and devise a novel (actually increasing) stepsize policy for solving these problems. Thirdly, we investigate this accelerated SA in more details, for example, derive the exponential bounds for the large deviations of the resulting solution inaccuracy from the expected one, provided the noise from the stochastic oracle is "light-tailed". Finally, the significant advantages of the accelerated scheme over the existing algorithms are illustrated in the context of solving a class of stochastic programming problems whose feasible region is a simple compact convex set intersected with an affine manifold.

In Chapter 5, we consider a special class of convex programming problems whose feasible regions consist of a simple compact convex set intersected with an affine manifold as described in Section 1.3. We present first-order methods for this class of problems based on an inexact version of the classical augmented Lagrangian approach, where the subproblems are approximately solved by means of Nesterov's optimal method. We then establish a bound on the total number of Nesterov's optimal iterations, i.e., the inner iterations, performed throughout the entire inexact AL method to obtain a near primal-dual optimal solution. We also present variants with better iteration-complexity bounds than the original inexact AL method, which consist of applying the original approach directly to a perturbed problem obtained by adding a strongly convex component to the objective function of the CP problem. We show that the iteration-complexity of the inexact AL methods for obtaining a near primal-dual optimal solution compares favorably with other penalty based approaches,

such as the quadratic and exact penalty method studied in [33], and another possible approach for solving variational inequalities (VI) studied in Nemirovski ([42]), and Monteiro and Svaiter ([41]).

CHAPTER II

MIRROR-DESCENT STOCHASTIC APPROXIMATION

2.1 Overview

In this chapter, we review the classic SA algorithm and the mirror-descent SA method introduced in [43] for solving problem (1.2.1). A basic difficulty of solving such stochastic optimization problems is that the involved multidimensional integrals (expectations) cannot be computed with high accuracy. The aim of this chapter is to compare two computational approaches based on Monte Carlo sampling techniques, namely, the SA and the SAA methods. Current opinion is that the SAA method can efficiently use a specific (say linear) structure of the considered problem, while the SA approach is a crude subgradient method which often performs poorly in practice. We demonstrate that a properly modified SA approach, i.e., the mirror-descent SA method, can be competitive and even significantly outperform the SAA method for a certain class of convex stochastic problems, for example, when the set X is a standard simplex. We also present, in our opinion, highly encouraging results of numerical experiments.

The rest of this chapter is organized as follows. In Section 2.2 we focus on the theory of the SA method applied to problem (1.2.1). We start with outlining the relevant to our goals part of the classical "$O(t^{-1})$" SA theory (Subsection 2.2.1), along with its "$O(t^{-1/2})$" modifications (Subsection 2.2.2). Well-known and simple results presented in these subsections pave the road to our main developments carried out in Subsection 2.2.3. In concluding Section 2.3 we present very promising numerical results for the SA algorithm (Subsection 2.2.3) applied to large-scale stochastic convex minimization problems. Finally, some concluding remarks are made in Section 2.4.

2.2 Stochastic Approximation: Basic Theory

2.2.1 Classical SA Algorithm

The classical SA algorithm solves problem (1.2.1) by mimicking the simplest subgradient descent method. That is, for chosen $x_1 \in X$ and a sequence $\gamma_j > 0$, $j = 1, ...$, of stepsizes, it generates the iterates by the formula

$$x_{j+1} := \Pi_X\big(x_j - \gamma_j \mathsf{G}(x_j, \xi_j)\big), \tag{2.2.1}$$

where Π_X denotes the metric projection operator onto the set X given by

$$\Pi_X(x) = \arg\min_{x' \in X} \|x - x'\|_2.$$

Note that Π_X is a contraction operator, i.e.,

$$\|\Pi_X(x') - \Pi_X(x)\|_2 \leq \|x' - x\|_2, \ \ \forall x', x \in \mathbb{R}^n. \tag{2.2.2}$$

Of course, the crucial question of the classical SA approach is how to choose the stepsizes γ_j. Let x^* be an optimal solution of problem (1.2.1). Note that since the set X is compact and $f(x)$ is continuous, problem (1.2.1) has an optimal solution. Note also that the iterate $x_j = x_j(\xi_{[j-1]})$ is a function of the history $\xi_{[j-1]} := (\xi_1, ..., \xi_{j-1})$ of the generated random process and hence is random.

Denote

$$A_j := \tfrac{1}{2}\|x_j - \bar{x}\|_2^2 \quad \text{and} \quad a_j := \mathbb{E}[A_j] = \tfrac{1}{2}\mathbb{E}\big[\|x_j - \bar{x}\|_2^2\big].$$

By using (2.2.2) and since $x^* \in X$ and hence $\Pi_X(x^*) = x^*$, we can write

$$\begin{aligned} A_{j+1} &= \tfrac{1}{2}\big\|\Pi_X\big(x_j - \gamma_j \mathsf{G}(x_j, \xi_j)\big) - x^*\big\|_2^2 \\ &= \tfrac{1}{2}\big\|\Pi_X\big(x_j - \gamma_j \mathsf{G}(x_j, \xi_j)\big) - \Pi_X(x^*)\big\|_2^2 \\ &\leq \tfrac{1}{2}\big\|x_j - \gamma_j \mathsf{G}(x_j, \xi_j) - x^*\big\|_2^2 \\ &= A_j + \tfrac{1}{2}\gamma_j^2 \|\mathsf{G}(x_j, \xi_j)\|_2^2 - \gamma_j (x_j - x^*)^T \mathsf{G}(x_j, \xi_j). \end{aligned} \tag{2.2.3}$$

Since $x_j = x_j(\xi_{[j-1]})$ is independent of ξ_j, we have

$$\begin{aligned} \mathbb{E}\big[(x_j - x^*)^T \mathsf{G}(x_j, \xi_j)\big] &= \mathbb{E}\big\{\mathbb{E}\big[(x_j - x^*)^T \mathsf{G}(x_j, \xi_j)\big|\xi_{[j-1]}\big]\big\} \\ &= \mathbb{E}\big\{(x_j - x^*)^T \mathbb{E}\big[\mathsf{G}(x_j, \xi_j)\big|\xi_{[j-1]}\big]\big\} \\ &= \mathbb{E}\big[(x_j - x^*)^T \mathsf{g}(x_j)\big]. \end{aligned} \tag{2.2.4}$$

Assume now that there is a positive number M such that

$$\mathbb{E}\left[\|\mathsf{G}(x,\xi)\|_2^2\right] \leq M^2 \quad \forall x \in X. \tag{2.2.5}$$

Then, by taking expectation of both sides of (2.2.3) and using (2.2.5), we obtain

$$a_{j+1} \leq a_j - \gamma_j \mathbb{E}\left[(x_j - x^*)^T \mathsf{g}(x_j)\right] + \tfrac{1}{2}\gamma_j^2 M^2. \tag{2.2.6}$$

Suppose further that the expectation function $f(x)$ is differentiable and strongly convex on X, i.e., there is constant $c > 0$ such that

$$f(x') \geq f(x) + (x' - x)^T \nabla f(x) + \tfrac{1}{2}c\|x' - x\|_2^2, \quad \forall x', x \in X,$$

or equivalently that

$$(x' - x)^T (\nabla f(x') - \nabla f(x)) \geq c\|x' - x\|_2^2, \quad \forall x', x \in X. \tag{2.2.7}$$

Note that strong convexity of $f(x)$ implies that the minimizer x^* is unique. By optimality of x^* we have that

$$(x - x^*)^T \nabla f(x^*) \geq 0, \quad \forall x \in X,$$

which together with (2.2.7) implies that

$$\mathbb{E}\left[(x_j - x^*)^T \nabla f(x_j)\right] \geq \mathbb{E}\left[(x_j - x^*)^T (\nabla f(x_j) - \nabla f(x^*))\right] \geq c\mathbb{E}\left[\|x_j - x^*\|_2^2\right] = 2ca_j.$$

Therefore it follows from (2.2.6) that

$$a_{j+1} \leq (1 - 2c\gamma_j)a_j + \tfrac{1}{2}\gamma_j^2 M^2. \tag{2.2.8}$$

Let us take stepsizes $\gamma_j = \theta/j$ for some constant $\theta > 1/(2c)$. Then, by (2.2.8), we have

$$a_{j+1} \leq (1 - 2c\theta/j)a_j + \tfrac{1}{2}\theta^2 M^2 / j^2.$$

It follows by induction that

$$a_j \leq Q(\theta)/j, \tag{2.2.9}$$

where

$$Q(\theta) := \max\left\{\tfrac{1}{2}\theta^2 M^2 (2c\theta - 1)^{-1}, a_1\right\}.$$

Suppose further that x^* is an *interior* point of X and $\nabla f(x)$ is Lipschitz continuous, i.e., there is constant $L > 0$ such that

$$\|\nabla f(x') - \nabla f(x)\|_2 \leq L\|x' - x\|_2, \ \ \forall x', x \in X. \tag{2.2.10}$$

Then

$$f(x) \leq f(x^*) + \tfrac{1}{2}L\|x - x^*\|_2^2, \ \ \forall x \in X, \tag{2.2.11}$$

and hence

$$\mathbb{E}\big[f(x_j) - f(x^*)\big] \leq La_j \leq LQ(\theta)/j, \tag{2.2.12}$$

where $Q(\theta)$ is defined in (2.2.10).

Under the specified assumptions, it follows from (2.2.11) and (2.2.12), respectively, that after t iterations the expected error of the current solution is of order $O(t^{-1/2})$ and the expected error of the corresponding objective value is of order $O(t^{-1})$, provided that $\theta > 1/(2c)$. We have arrived at the $O(t^{-1})$-rate of convergence mentioned in Section 1.2. Note, however, that the result is highly sensitive to our a priori information on c. What would happen if the parameter c of strong convexity is overestimated? As a simple example consider $f(x) = x^2/10$, $X = [-1, 1] \subset \mathbb{R}$ and assume that there is no noise, i.e., $F(x, \xi) \equiv f(x)$. Suppose, further, that we take $\theta = 1$ (i.e., $\gamma_j = 1/j$), which will be the optimal choice for $c = 1$, while actually here $c = 0.2$. Then the iteration process becomes

$$x_{j+1} = x_j - f'(x_j)/j = \left(1 - \tfrac{1}{5j}\right) x_j,$$

and hence starting with $x_1 = 1$,

$$\begin{aligned} x_j \ &= \textstyle\prod_{s=1}^{j-1}\left(1 - \frac{1}{5s}\right) = \exp\left\{-\sum_{s=1}^{j-1} \ln\left(1 + \frac{1}{5s-1}\right)\right\} > \exp\left\{-\sum_{s=1}^{j-1} \frac{1}{5s-1}\right\} \\ &> \exp\left\{-\left(0.25 + \textstyle\int_1^{j-1} \frac{1}{5t-1}dt\right)\right\} > \exp\left\{-0.25 + 0.2\ln 1.25 - \tfrac{1}{5}\ln j\right\} > 0.8j^{-1/5}. \end{aligned}$$

That is, the convergence is extremely slow. For example for $j = 10^9$ the error of the iterated solution is greater than 0.015. On the other hand for the optimal stepsize factor of $\gamma = 1/c = 5$, the optimal solution $x^* = 0$ is found in one iteration.

It could be added that the stepsizes $\gamma_j = \theta/j$ may become completely unacceptable when f loses strong convexity. For example, when $f(x) = x^4, X = [-1, 1]$, and there is no

noise, these stepsizes result in a disastrously slow convergence: $|x_j| \geq \mathcal{O}([\ln(j+1)]^{-1/2}$. The precise statement here is that with $\gamma_j = \theta/j$ and $0 < x_1 \leq 1/(6\sqrt{\theta})$, we have that

$$x_j \geq \frac{x_1}{\sqrt{1+32\theta x_1^2[1+\ln(j+1)]}}, \quad \forall j = 1, 2, \ldots.$$

We see that in order to make the SA "robust" - applicable to general convex objectives rather than to strongly convex ones - one should replace the classical stepsizes $\gamma_j = \mathcal{O}(j^{-1})$, which can be too small to ensure a reasonable rate of convergence even in the "no noise" case, with "much larger" stepsizes. At the same time, a detailed analysis shows that "large" stepsizes poorly suppress noise. As early as in [46] it was realized that in order to resolve the arising difficulty, it makes sense to separate collecting information on the objective from generating approximate solutions. Specifically, we can use large stepsizes, say, $\gamma_j = \mathcal{O}(j^{-1/2})$ in (2.2.1), thus avoiding too slow motion at the cost of making the trajectory "more noisy". In order to suppress, to some extent, this noisiness, we take, as approximate solutions, appropriate averages of the search points x_j rather than these points themselves.

2.2.2 Robust SA Approach

The results of this subsection go back to [46] and [44]. Let us look again at the basic estimate (2.2.6). By convexity of $f(x)$ we have that for any x, $f(x) \geq f(x_j) + (x - x_j)^T \mathsf{g}(x_j)$, and hence

$$\mathbb{E}\big[(x_j - x^*)^T \mathsf{g}(x_j)\big] \geq \mathbb{E}\big[f(x_j) - f(x^*)\big] = \mathbb{E}\big[f(x_j)\big] - f(x^*).$$

Together with (2.2.6), this implies (recall that $a_j = \mathbb{E}\big[\|x_j - \bar{x}\|_2^2\big]/2$)

$$\gamma_j \mathbb{E}\big[f(x_j) - f(x^*)\big] \leq a_j - a_{j+1} + \tfrac{1}{2}\gamma_j^2 M^2.$$

It follows that

$$\sum_{t=1}^{j} \gamma_t \mathbb{E}\big[f(x_t) - f(x^*)\big] \leq \sum_{t=1}^{j} [a_t - a_{t+1}] + \tfrac{1}{2}M^2 \sum_{t=1}^{j} \gamma_t^2 \leq a_1 + \tfrac{1}{2}M^2 \sum_{t=1}^{j} \gamma_t^2, \tag{2.2.13}$$

and hence, setting $\nu_t := \gamma_t / \sum_{i=1}^{j} \gamma_i$,

$$\mathbb{E}\left[\sum_{t=1}^{j} \nu_t f(x_t) - f(x^*)\right] \leq \frac{a_1 + \frac{1}{2}M^2 \sum_{t=1}^{j} \gamma_t^2}{\sum_{t=1}^{j} \gamma_t}, \tag{2.2.14}$$

Note that $\nu_t \geq 0$ and $\sum_{t=1}^{j} \nu_t = 1$. Consider points

$$\tilde{x}_i^j := \sum_{t=i}^{j} \nu_t x_t. \tag{2.2.15}$$

and let

$$D_X := \max_{x \in X} \|x - x_1\|_2. \tag{2.2.16}$$

By convexity of X, we have $\tilde{x}_i^j \in X$ and by convexity of $f(x)$, we have $f(\tilde{x}_i^j) \leq \sum_{t=i}^{j} \nu_t f(x_t)$. Thus, by (2.2.14) and in view of $a_1 \leq D_X^2$ and $a_i \leq 4D_X^2, i > 1$, we get

$$\begin{array}{ll} (a) & \mathbb{E}\left[f(\tilde{x}_1^j) - f(x^*)\right] \leq \frac{2D_X^2 + M^2 \sum_{t=1}^{j} \gamma_t^2}{2\sum_{t=1}^{j} \gamma_t} \quad \forall 1 \leq j, \\ (b) & \mathbb{E}\left[f(\tilde{x}_i^j) - f(x^*)\right] \leq \frac{8D_X^2 + M^2 \sum_{t=1}^{j} \gamma_t^2}{2\sum_{t=1}^{j} \gamma_t} \quad \forall 1 < i \leq j. \end{array} \tag{2.2.17}$$

Based on the resulting bounds on the expected inaccuracy of approximate solutions $\tilde{x}_i^j$, we can now develop "reasonable" stepsize policies along with the associated efficiency estimates.

Constant stepsizes and basic efficiency estimate Assume that the number N of iterations of the method is fixed in advance and that $\gamma_t = \gamma, t = 1, \ldots, N$. Then it follows by (2.2.17(a)) that

$$\mathbb{E}\left[f(\tilde{x}_1^N) - f(x^*)\right] \leq \frac{2D_X^2 + M^2 N \gamma^2}{2N\gamma}.$$

Minimizing the right-hand side of the above inequality over $\gamma > 0$, we arrive at the constant stepsize policy

$$\gamma_t = \frac{\sqrt{2} D_X}{M\sqrt{N}}, \tag{2.2.18}$$

along with the associated efficiency estimate

$$\mathbb{E}\left[f(\tilde{x}_1^N) - f(x^*)\right] \leq \frac{\sqrt{2} D_X M}{\sqrt{N}}. \tag{2.2.19}$$

With the constant stepsize policy (2.2.18), we also have, for $1 \leq K \leq N$,

$$\mathbb{E}\left[f(\tilde{x}_K^N) - f(x^*)\right] \leq \frac{\sqrt{2} D_X M}{\sqrt{N}} \left(\frac{2N}{N - K + 1} + \frac{1}{2}\right). \tag{2.2.20}$$

When $K/N \leq 1/2$, the right-hand side of (2.2.20) coincides, within an absolute constant factor, with the right-hand side of (2.2.19). Finally, for a constant $\theta > 0$, passing from the stepsizes (2.2.18) to the stepsizes

$$\gamma_t = \frac{\theta\sqrt{2} D_X}{M\sqrt{N}}, \tag{2.2.21}$$

the efficiency estimate becomes

$$\mathbb{E}\left[f(\tilde{x}_K^N) - f(x^*)\right] \leq \max\{\theta, \theta^{-1}\}\frac{\sqrt{2}D_X M}{\sqrt{N}}\left(\frac{2N}{N-K+1} + \frac{1}{2}\right). \tag{2.2.22}$$

Discussion We conclude that the expected error of *Robust SA* algorithm (2.2.1),(2.2.15), with constant stepsize strategy (2.2.18), after N iterations is $O(N^{-1/2})$ in our setting. Of course, this is worse than the convergence rate $O(N^{-1})$ for the classical SA algorithm when the objective function $f(x)$ is strongly convex. However, the error bounds (2.2.19) and (2.2.20) are guaranteed independently of any smoothness and/or strong convexity assumptions on f. All that matters is the convexity of f on the convex compact set X and the validity of (2.2.5). Moreover, scaling the stepsizes by positive constant θ affect the error bound (2.2.22) linearly in $\max\{\theta, \theta^{-1}\}$. This can be compared with a possibly disastrous effect of such scaling in the classical SA algorithm discussed in Subsection 2.2.1. These observations, in particular the fact that there is no necessity in "fine tuning" the stepsizes to the objective function f, explain the adjective "robust" in the name of the method. Finally, it can be shown that without additional, as compared to convexity and (2.2.5), assumptions on f, the accuracy bound (2.2.19) within an absolute constant factor is the best one allowed by statistics (cf. [44]).

Varying stepsizes When the number of steps is not fixed in advance, it makes sense to replace constant stepsizes with the stepsizes

$$\gamma_t = \frac{\theta\sqrt{2}D_X}{M\sqrt{t}}, \tag{2.2.23}$$

from (2.2.17(b)) it follows that

$$\mathbb{E}\left[f(\tilde{x}_K^N) - f(x^*)\right] \leq \frac{\sqrt{2}D_X M}{\sqrt{N}}\left(\frac{2}{\theta}\frac{N}{N-K+1} + \frac{\theta}{2}\sqrt{\frac{N}{K}}\right). \tag{2.2.24}$$

Choosing K as a fixed fraction of N, i.e., setting $K = rN$, with a fixed $r \in (0,1)$, we get the efficiency estimate

$$\mathbb{E}\left[f(\tilde{x}_K^N) - f(x^*)\right] \leq C(r)\max\{\theta, \theta^{-1}\}\frac{D_X M}{\sqrt{N}} \quad N = 1, 2, \ldots, \tag{2.2.25}$$

with an easily computable factor $C(r)$ depending solely on r. This bound, up to a factor depending solely on r and θ, coincides with the bound (2.2.19), with the advantage that our new stepsize policy should not be adjusted to a fixed-in-advance number of steps N.

2.2.3 Mirror Descent SA Method

On a close inspection, the Robust SA algorithm from Subsection 2.2.2 is intrinsically linked to the Euclidean structure of $\mathbb{R}^n$. This structure plays the central role in the very construction of the method (see (2.2.1)), same as in the associated efficiency estimates, like (2.2.19) (since the quantities D_X, M participating in the estimates are defined in terms of the Euclidean norm, see (2.2.16), (2.2.5)). By these reasons, from now on we refer to the algorithm from Section 2.2.2 as to (Robust) *Euclidean SA*. In this section we develop a substantial generalization of the Euclidean SA approach allowing to adjust, to some extent, the method to the geometry, not necessary Euclidean, of the problem in question. We shall see in the mean time that we can gain a lot, both theoretically and numerically, from such an adjustment. A rudimentary form of the generalization to follow can be found in Nemirovski and Yudin [44], from where the name "Mirror Descent" originates.

Let $\|\cdot\|$ be a (general) norm on $\mathbb{R}^n$ and $\|x\|_* = \sup_{\|y\|\le 1} y^T x$ be its dual norm. We say that a function $\omega : X \to \mathbb{R}$ is a *distance generating function* modulus $\alpha > 0$ with respect to $\|\cdot\|$, if ω is convex and continuous on X, the set

$$X^o = \left\{x \in X : \text{there exists } p \in \mathbb{R}^n \text{ such that } x \in \arg\min_{u\in X}[p^T u + \omega(u)]\right\}$$

is convex (note that X^o always contains the relative interior of X), and restricted to X^o, ω is continuously differentiable and strongly convex with parameter α with respect to $\|\cdot\|$, i.e.,

$$(x' - x)^T(\nabla\omega(x') - \nabla\omega(x)) \ge \alpha\|x' - x\|^2, \ \ \forall x', x \in X^o. \tag{2.2.26}$$

The simplest example of a distance generating function is $\omega(x) = \frac{1}{2}\|x\|_2^2$ (modulus 1 with respect to $\|\cdot\|_2$, $X^o = X$).

Let us define function $V : X^o \times X \to \mathbb{R}_+$ as follows

$$V(x, z) = \omega(z) - [\omega(x) + \nabla\omega(x)^T(z - x)]. \tag{2.2.27}$$

In what follows we shall refer to $V(\cdot,\cdot)$ as *prox-function* associated with distance generating function $\omega(x)$. Note that the distance generating function ω here is not necessarily differentiable and strongly convex over the whole domain X and hence that prox-function $V(\cdot,\cdot)$ is slightly more general than the Bregman's distance $d_\omega(\cdot;\cdot)$ given in Subsection 1.1.2, which was studied by Bregman [10] and many others (see [1, 2, 29, 72] and references therein). Note that $V(x,\cdot)$ is nonnegative and is strongly convex modulus α with respect to the norm $\|\cdot\|$. Let us define *prox mapping* $P_x : \mathbb{R}^n \to X^o$, associated with ω and a point $x \in X^o$, viewed as a parameter, as follows:

$$P_x(y) = \arg\min_{z\in X} \big\{y^T(z-x) + V(x,z)\big\}. \tag{2.2.28}$$

Observe that the minimum in the right hand side of (2.2.28) is attained since ω is continuous on X and X is compact, and all the minimizers belong to X^o, whence the minimizer is unique, since $V(x,\cdot)$ is strongly convex on X^o. Thus, the prox-mapping is well defined.

The distance generating function ω also gives rise to the following characteristic entity that will be used frequently in our convergence analysis:

$$D_{\omega,X} := \sqrt{\max_{x\in X}\omega(x) - \min_{x\in X}\omega(x)}, \qquad \forall x \in X. \tag{2.2.29}$$

Let x_1 be the minimizer of ω over X. Observe that $x_1 \in X^o$, whence $\nabla w(x_1)$ is well defined and satisfies $\langle \nabla\omega(x_1), x - x_1\rangle \geq 0$ for all $x \in X$, which combined with the strong convexity of ω implies that

$$\frac{\alpha}{2}\|x - x_1\|^2 \leq V(x_1,x) \leq \omega(x) - \omega(x_1) \leq D^2_{\omega,X}, \ \ \forall x \in X, \tag{2.2.30}$$

and hence

$$\|x - x_1\| \leq \Omega_{\omega,X} := \sqrt{\frac{2}{\alpha}} D_{\omega,X} \text{ and } \|x - x'\| \leq 2\Omega_{\omega,X}, \ \forall x, x' \in X. \tag{2.2.31}$$

For $\omega(x) = \frac{1}{2}\|x\|_2^2$, we have $P_x(y) = \Pi_X(x - y)$, so that (2.2.1) is the recurrence

$$x_{j+1} = P_{x_j}(\gamma_j \mathsf{G}(x_j,\xi_j)), \quad x_1 \in X^o. \tag{2.2.32}$$

Our goal is to demonstrate that the main properties of the recurrence (2.2.1) (which from now on we call the *Euclidean* SA recurrence) are inherited by (2.2.32), *whatever be the underlying distance generating function* $\omega(x)$.

The following statement, whose proof can be found in the appendix of [43], is a simple consequence of the optimality conditions of the right hand side of (2.2.28).

Lemma 2.2.1 *For every $u \in X, x \in X^o$ and $y \in \mathbb{R}^n$ one has*

$$V(P_x(y), u) \leq V(x, u) + y^T(u - x) + \frac{\|y\|_*^2}{2\alpha}. \tag{2.2.33}$$

Using (2.2.33) with $x = x_j$, $y = \gamma_j \mathsf{G}(x_j, \xi_j)$ and $u = x_*$, we get

$$\gamma_j (x_j - x_*)^T \mathsf{G}(x_j, \xi_j) \leq V(x_j, x_*) - V(x_{j+1}, x_*) + \frac{\gamma_j^2}{2\alpha} \|\mathsf{G}(x_j, \xi_j)\|_*^2. \tag{2.2.34}$$

Note that with $\omega(x) = \frac{1}{2}\|x\|_2^2$, one has $V(x, z) = \frac{1}{2}\|x - z\|_2^2$, that is, (2.2.34) becomes nothing but the relation (2.2.3) which played the crucial role in all the developments related to the Euclidean SA. We are about to process, in a completely similar fashion, the relation (2.2.34) in the case of a general distance generating function, thus arriving at the Mirror Descent SA. Specifically, setting

$$\Delta_j = \mathsf{G}(x_j, \xi_j) - \mathsf{g}(x_j), \tag{2.2.35}$$

we can rewrite (2.2.34), with j replaced by t, as

$$\gamma_t (x_t - x_*)^T \mathsf{g}(x_t) \leq V(x_t, x_*) - V(x_{t+1}, x_*) - \gamma_t \Delta_t^T (x_t - x_*) + \frac{\gamma_t^2}{2\alpha} \|\mathsf{G}(x_t, \xi_t)\|_*^2. \tag{2.2.36}$$

Summing up over $t = 1, ..., j$, and taking into account that $V(x_{j+1}, u) \geq 0$, $u \in X$, we get

$$\sum_{t=1}^{j} \gamma_t (x_t - x_*)^T \mathsf{g}(x_t) \leq V(x_1, x_*) + \sum_{t=1}^{j} \frac{\gamma_t^2}{2\alpha} \|\mathsf{G}(x_t, \xi_t)\|_*^2 - \sum_{t=1}^{j} \gamma_t \Delta_t^T (x_t - x_*). \tag{2.2.37}$$

Setting $\nu_t = \frac{\gamma_t}{\sum_{i=1}^{j} \gamma_i}$, $t = 1, ..., j$, and

$$\tilde{x}_1^j = \sum_{t=1}^{j} \nu_t x_t \tag{2.2.38}$$

and invoking convexity of $f(\cdot)$, we have

$$\begin{array}{rcl} \sum_{t=1}^{j} \gamma_t (x_t - x_*)^T \mathsf{g}(x_t) & \geq & \sum_{t=1}^{j} \gamma_t \left[f(x_t) - f(x_*) \right] = \left(\sum_{t=1}^{j} \gamma_t \right) \left[\sum_{t=1}^{j} \nu_t f(x_t) - f(x_*) \right] \\ & \geq & \left(\sum_{t=1}^{j} \gamma_t \right) \left[f(\tilde{x}_j) - f(x_*) \right], \end{array}$$

which combines with (2.2.37) to imply that

$$f(\tilde{x}_1^j) - f(x_*) \leq \frac{V(x_1, x_*) + \sum_{t=1}^{j} \frac{\gamma_t^2}{2\alpha} \|\mathsf{G}(x_t, \xi_t)\|_*^2 - \sum_{t=1}^{j} \gamma_t \Delta_t^T (x_t - x_*)}{\sum_{t=1}^{j} \gamma_t}. \tag{2.2.39}$$

Let us suppose, as in the previous subsection (cf., (2.2.5)), that we are given a positive number M_* such that

$$\mathbb{E}\left[\|\mathsf{G}(x,\xi)\|_*^2\right] \le M_*^2, \ \ \forall x \in X. \tag{2.2.40}$$

Taking expectations of both sides of (2.2.39) and noting that: (i) x_t is a deterministic function of $\xi_{[t-1]} = (\xi_1, ..., \xi_{t-1})$, (ii) conditional on $\xi_{[t-1]}$, the expectation of Δ_t is 0, and (iii) the expectation of $\|\mathsf{G}(x_t, \xi_t)\|_*^2$ does not exceed M_*^2, we obtain

$$\mathbb{E}\left[f(\tilde{x}_1^j) - f(x_*)\right] \le \frac{\max_{u\in X} V(x_1,u) + (2\alpha)^{-1}M_*^2\sum_{t=1}^j \gamma_t^2}{\sum_{t=1}^j \gamma_t}. \tag{2.2.41}$$

Assume from now on that the method starts with the minimizer of ω:

$$x_1 = \mathrm{argmin}_X \omega(x).$$

Then it follows from (2.2.30) and (2.2.41) that

$$\mathbb{E}\left[f(\tilde{x}_1^j) - f(x_*)\right] \le \frac{D_{\omega,X}^2 + \frac{1}{\alpha}M_*^2\sum_{t=1}^j \gamma_t^2}{\sum_{t=1}^j \gamma_j}. \tag{2.2.42}$$

Constant stepsize policy Assuming that the total number of steps N is given in advance and optimizing the right hand side of (2.2.42), evaluated at $j = N$, in $\gamma_t > 0$, $1 \le t \le N$, we arrive at the constant stepsize policy

$$\gamma_t = \frac{\sqrt{2\alpha}D_{\omega,X}}{M_*\sqrt{N}}, \ \ t = 1, ..., N, \tag{2.2.43}$$

and the associated efficiency estimate

$$\mathbb{E}\left[f(\tilde{x}_1^N) - f(x_*)\right] \le D_{\omega,X}M_*\sqrt{\frac{2}{\alpha N}} \tag{2.2.44}$$

(cf., (2.2.18), (2.2.19)). Passing from the stepsizes (2.2.43) to the stepsizes

$$\gamma_t = \frac{\theta\sqrt{2\alpha}D_{\omega,X}}{M_*\sqrt{N}}, \ \ t = 1, ..., N, \tag{2.2.45}$$

the efficiency estimate becomes

$$\mathbb{E}\left[f(\tilde{x}_1^N) - f(x_*)\right] \le \max\left\{\theta, \theta^{-1}\right\} D_{\omega,X}M_*\sqrt{\frac{2}{\alpha N}}. \tag{2.2.46}$$

We refer to the method (2.2.32), (2.2.38), (2.2.45) as (Robust) *Mirror Descent SA* algorithm with constant stepsize policy.

Probabilities of large deviations So far, all our efficiency estimates were upper bounds on the expected non-optimality, in terms of the objective, of approximate solutions generated by the algorithms. Here we complement these results with bounds on probabilities of large deviations. Observe that by Markov inequality, (2.2.46) implies that

$$\mathrm{Prob}\left\{f(\tilde{x}_1^N) - f(x_*) > \varepsilon\right\} \le \frac{\sqrt{2}\max\left\{\theta, \theta^{-1}\right\} D_{\omega,X} M_*}{\varepsilon\sqrt{\alpha N}}, \quad \forall \varepsilon > 0. \tag{2.2.47}$$

It is possible, however, to obtain much finer bounds on deviation probabilities when imposing more restrictive assumptions on the distribution of $\mathsf{G}(x,\xi)$. Specifically, assume that

$$\mathbb{E}\left[\exp\left\{\|\mathsf{G}(x,\xi)\|_*^2 / M_*^2\right\}\right] \le \exp\{1\}, \quad \forall x \in X. \tag{2.2.48}$$

Note that condition (2.2.48) is stronger than (2.2.40). Indeed, if a random variable Y satisfies $\mathbb{E}[\exp\{Y/a\}] \le \exp\{1\}$ for some $a > 0$, then by Jensen inequality $\exp\{\mathbb{E}[Y/a]\} \le \mathbb{E}[\exp\{Y/a\}] \le \exp\{1\}$, and therefore $\mathbb{E}[Y] \le a$. Of course, condition (2.2.48) holds if $\|\mathsf{G}(x,\xi)\|_* \le M_*$ for all $(x,\xi) \in X \times \Xi$. In the case of (2.2.48), for the constant stepsizes (2.2.45), it is shown in Proposition 2.2 of [43] that for any $\Lambda \ge 1$ the following holds

$$\mathrm{Prob}\left\{f(\tilde{x}_1^N) - f(x_*) > \frac{\sqrt{2}\max\left\{\theta,\theta^{-1}\right\} M_* D_{\omega,X}(12+2\Lambda)}{\sqrt{\alpha N}}\right\} \le 2\exp\{-\Lambda\}. \tag{2.2.49}$$

Varying stepsizes Same as in the case of Euclidean SA, we can modify the Mirror Descent SA algorithm to allow for time-varying stepsizes and "sliding averages" of the search points x_t in the role of approximate solutions, thus getting rid of the necessity to fix in advance the number of steps. Specifically, consider

$$\begin{aligned} \overline{D}_{\omega,X} &:= \sqrt{2} \sup_{x \in X^o, z \in X} \left[\omega(z) - \omega(x) - (z-x)^T \nabla\omega(x)\right]^{1/2} \\ &= \sup_{x \in X^o, z \in X} \sqrt{2V(x,z)}, \end{aligned} \tag{2.2.50}$$

and assume that $\overline{D}_{\omega,X}$ is finite. This is definitely so when ω is continuously differentiable on the entire X. Note that for the Euclidean SA, that is, with $\omega(x) = \frac{1}{2}\|x\|_2^2$, $\overline{D}_{\omega,X}$ is the Euclidean diameter of X.

In the case of (2.2.50), setting

$$\tilde{x}_i^j = \frac{\sum_{t=i}^j \gamma_t x_t}{\sum_{t=i}^j \gamma_t}, \tag{2.2.51}$$

summing up inequalities (2.2.34) over $K \le t \le N$ and acting exactly as when deriving (2.2.39), we get for $1 \le K \le N$,

$$f(\tilde{x}_K^N) - f(x_*) \le \frac{V(x_K, x_*) + \sum_{t=K}^N \frac{\gamma_t^2}{2\alpha}\|\mathsf{G}(x_t,\xi_t)\|_*^2 - \sum_{t=K}^N \gamma_t \Delta_t^T (x_t - x_*)}{\sum_{t=K}^N \gamma_t}.$$

Noting that $V(x_K, x_*) \le \frac{1}{2}\overline{D}_{\omega,X}^2$ and taking expectations, we arrive at

$$\mathbb{E}\left[f(\tilde{x}_K^N) - f(x_*)\right] \le \frac{\frac{1}{2}\overline{D}_{\omega,X}^2 + \frac{1}{2\alpha}M_*^2 \sum_{t=K}^N \gamma_t^2}{\sum_{t=K}^N \gamma_t} \tag{2.2.52}$$

(cf., (2.2.42)). It follows that with a decreasing stepsize policy

$$\gamma_t = \frac{\theta \overline{D}_{\omega,X}\sqrt{\alpha}}{M_*\sqrt{t}}, \quad t = 1, 2, ..., \tag{2.2.53}$$

one has for $1 \le K \le N$,

$$\mathbb{E}\left[f(\tilde{x}_K^N) - f(x_*)\right] \le \frac{\overline{D}_{\omega,X} M_*}{\sqrt{\alpha}\sqrt{N}} \left[\frac{2}{\theta}\frac{N}{N-K+1} + \frac{\theta}{2}\sqrt{\frac{N}{K}}\right] \tag{2.2.54}$$

(cf., (2.2.17)). In particular, with $K = \lceil rN \rceil$ for a fixed $r \in (0,1)$, we get an efficiency estimate

$$\mathbb{E}\left[f(\tilde{x}_K^N) - f(x_*)\right] \le C(r) \max\left\{\theta, \theta^{-1}\right\} \frac{\overline{D}_{\omega,X} M_*}{\sqrt{\alpha}\sqrt{N}}, \tag{2.2.55}$$

completely similar to the estimate (2.2.25) for the Euclidean SA.

Discussion Comparing (2.2.19) to (2.2.44) and (2.2.25) to (2.2.55), we see that for both the Euclidean and the Mirror Descent SA, the expected inaccuracy, in terms of the objective, of the approximate solution built in course of N steps is $O(N^{-1/2})$. A benefit of the Mirror Descent over the Euclidean algorithm is in potential possibility to reduce the constant factor hidden in $O(\cdot)$ by adjusting the norm $\|\cdot\|$ and the distance generating function $\omega(\cdot)$ to the geometry of the problem.

Example 2.2.1 Let $X = \{x \in \mathbb{R}^n : \sum_{i=1}^n x_i = 1,\ x \ge 0\}$ be a standard simplex. Consider two setups for the Mirror Descent SA:

— *Euclidean setup*, where $\|\cdot\| = \|\cdot\|_2$ and $\omega(x) = \frac{1}{2}\|x\|_2^2$, and

— *ℓ_1-setup*, where $\|x\| = \|x\|_1 := \sum_{i=1}^n |x_i|$ and ω is the *entropy*

$$\omega(x) = \sum_{i=1}^n x_i \ln x_i. \tag{2.2.56}$$

The Euclidean setup leads to the Euclidean Robust SA which is easily implementable (computing the prox-mapping requires $O(n \ln n)$ operations) and guarantees that

$$\mathbb{E}\left[f(\tilde{x}_1^N) - f(x_*)\right] \leq O(1) \max\left\{\theta, \theta^{-1}\right\} M N^{-1/2}, \tag{2.2.57}$$

with $M^2 = \sup_{x \in X} \mathbb{E}\left[\|\mathsf{G}(x,\xi)\|_2^2\right]$, provided that the constant M is known and the stepsizes (2.2.21) are used (see (2.2.22), (2.2.16) and note that the Euclidean diameter of X is of order of 1). The ℓ_1-setup corresponds to $X^o = \{x \in X : x > 0\}$, $D_{\omega,X} = \sqrt{\ln n}$, $x_1 = \mathrm{argmin}_X \omega = n^{-1}(1, ..., 1)^T$, $\alpha = 1$ and $\|x\|_* = \|x\|_\infty \equiv \max_i |x_i|$ (see Appendix). The associated Mirror Descent SA is easily implementable: the prox-function here is

$$V(x,z) = \sum_{i=1}^n z_i \ln \frac{z_i}{x_i},$$

and the prox mapping $P_x(y) = \mathrm{argmin}_{z \in X}\left[y^T(z-x) + V(x,z)\right]$ can be computed in $O(n)$ operations according to the explicit formula:

$$[P_x(y)]_i = \frac{x_i e^{-y_i}}{\sum_{k=1}^n x_k e^{-y_k}}, \quad i = 1, ..., n.$$

The efficiency estimate guaranteed with the ℓ_1-setup is (2.2.45) is

$$\mathbb{E}\left[f(\tilde{x}_1^N) - f(x_*)\right] \leq O(1) \max\left\{\theta, \theta^{-1}\right\} \sqrt{\ln n} M_* N^{-1/2}, \tag{2.2.58}$$

with

$$M_*^2 = \sup_{x \in X} \mathbb{E}\left[\|\mathsf{G}(x,\xi)\|_\infty^2\right],$$

provided that the constant M_* is known and the constant stepsizes (2.2.45) are used (see (2.2.46), (2.2.40)). To compare (2.2.58) and (2.2.57), observe that $M_* \leq M$, and the ratio M_*/M can be as small as $n^{-1/2}$. Thus, the efficiency estimate for the ℓ_1-setup never is much worse than the estimate for the Euclidean setup, and for large n can be *far better* than the latter estimate:

$$\sqrt{\frac{1}{\ln n}} \leq \frac{M}{\sqrt{\ln n} M_*} \leq \sqrt{\frac{n}{\ln n}}, \; N = 1, 2, ...,$$

both the upper and the lower bounds being achievable. Thus, when X is a standard simplex of large dimension, we have strong reasons to prefer the ℓ_1-setup to the usual Euclidean one.

Note that $\|\cdot\|_1$-norm can be coupled with "good" distance-generating functions different from the entropy one, e.g., with the function

$$\omega(x) = (\ln n) \sum_{i=1}^{n} |x_i|^{1+\frac{1}{\ln n}}, \quad n \geq 3. \tag{2.2.59}$$

Whenever $0 \in X$ and $\mathrm{Diam}_{\|\cdot\|_1}(X) \equiv \max_{x,y\in X} \|x - y\|_1 = 1$ (these conditions can always be ensured by scaling and shifting X), for the just outlined setup one has $\overline{D}_{\omega,X} = O(1)\sqrt{\ln n}$, $\alpha = O(1)$, so that the associated Mirror Descent SA guarantees that with $M_*^2 = \sup_{x\in X} \mathbb{E}\left[\|\mathsf{G}(x,\xi)\|_\infty^2\right]$ and $N \geq 1$,

$$\mathbb{E}\left[f(\tilde{x}^N_{\lceil rN\rceil}) - f(x_*)\right] \leq C(r)\frac{M_*\sqrt{\ln n}}{\sqrt{N}} \tag{2.2.60}$$

(see (2.2.55)), while the efficiency estimate for the Euclidean SA is

$$\mathbb{E}\left[f(\tilde{x}^N_{\lceil rN\rceil}) - f(x_*)\right] \leq C(r)\frac{M\,\mathrm{Diam}_{\|\cdot\|_2}(X)}{\sqrt{N}}, \tag{2.2.61}$$

with

$$M^2 = \sup_{x\in X} \mathbb{E}\left[\|\mathsf{G}(x,\xi)\|_2^2\right] \ \text{ and } \ \mathrm{Diam}_{\|\cdot\|_2}(X) = \max_{x,y\in X} \|x - y\|_2.$$

Ignoring logarithmic in n factors, the second estimate (2.2.61) can be much better than the first estimate (2.2.60) only when $\mathrm{Diam}_{\|\cdot\|_2}(X) \ll 1 = \mathrm{Diam}_{\|\cdot\|_1}(X)$, as it is the case, e.g., when X is an Euclidean ball. On the other hand, when X is an $\|\cdot\|_1$-ball or its nonnegative part (which is the simplex), so that the $\|\cdot\|_1$- and $\|\cdot\|_2$-diameters of X are of the same order, the first estimate (2.2.60) is much more attractive than the estimate (2.2.61) due to potentially much smaller constant M_*.

Comparison with the SAA approach We compare now theoretical complexity estimates for the Mirror Descent SA and the SAA methods. Consider the case when: (i) $X \subset \mathbb{R}^n$ is contained in the $\|\cdot\|_p$-ball of radius R, $p = 1, 2$, and the SA in question is either the Euclidean SA ($p = 2$), or the SA associated with $\|\cdot\|_1$ and the distance-generating function[1](2.2.59), (ii) in SA, the constant stepsize rule (2.2.43) is used, and (iii) the "light tail" assumption (2.2.48) takes place.

[1]In the second case, we apply the SA after the variables are scaled to make X the unit $\|\cdot\|_1$-ball.

Given $\epsilon > 0$, $\delta \in (0, 1/2)$, let us compare the number of steps $N = N_{\rm SA}$ of SA which, with probability $\geq 1-\delta$, results in an approximate solution $\tilde{x}_1^N$ such that $f(\tilde{x}_1^N) - f(x_*) \leq \epsilon$, with the sample size $N = N_{\rm SAA}$ for the SAA resulting in the same accuracy guarantees. According to (2.2.49) we have that

$$\begin{aligned} N_{\rm SA} &= O(1)\ln(n)\ln^2(1/\delta)(RM_*/\epsilon)^2, \quad p = 1, \\ N_{\rm SA} &= O(1)\ln^2(1/\delta)(RM_*/\epsilon)^2, \qquad\quad p = 2, \end{aligned} \tag{2.2.62}$$

where M_* is the constant from (2.2.48). This can be compared with the estimate of the sample size (cf., [68])

$$N_{\rm SAA} = O(1)\big[\ln(1/\delta) + n\ln\left(RM_*/\epsilon\right)\big](RM_*/\epsilon)^2. \tag{2.2.63}$$

We see that both SA and SAA methods have logarithmic in δ and quadratic (or nearly so) in $1/\epsilon$ complexity in terms of the corresponding sample sizes. It should be noted, however, that the SAA method requires solution of the corresponding (deterministic) problem while the SA approach is based on simple calculations as long as stochastic subgradients could be easily computed.

2.3 Numerical results

In this section, we report the results of our computational experiments where we compare the performance of the Mirror Descent SA method and the SAA method applied to three stochastic programming problems, namely: a stochastic utility problem, a stochastic max-flow problem and network planning problem with random demand.

The algorithms we were testing are the two variants of the Mirror Descent SA. The first variant, the *Euclidean* SA (E-SA), is as described in Section 2.2.2; in terms of Section 2.2.3, this is nothing but Mirror Descent SA with Euclidean setup, see Example 2.2.1. The second variant, referred to as the *Non-Euclidean* SA (N-SA), is the Mirror Descent SA with ℓ_1-setup, see Example 2.2.1.

These two variants of SA method are compared with the SAA approach in the following way: fixing an i.i.d. sample (of size N) for the random variable ξ, we apply the three afore-mentioned methods to obtain approximate solutions for the test problem under consideration, and then the quality of the solutions yielded by these algorithms is evaluated

using another i.i.d. sample of size $K >> N$. It should be noted that SAA itself is not an algorithm and in our experiments it was coupled with the Non-Euclidean Restricted Memory Level (NERML) [5] – a powerful deterministic algorithm for solving the sample average problem (1.2.4).

2.3.1 Preliminaries

Algorithmic schemes Both Euclidean and Non-Euclidean SA were implemented according to the description in Section 2.2.3, the number of steps N being the parameter of a particular experiment. In such an experiment, we generated $\approx \log_2 N$ candidate solutions $\tilde{x}_i^N$ with $N-i+1 = \min[2^k, N]$, $k = 0, 1, ..., \lfloor \log_2 N \rfloor$. We then used an additional sample to estimate the objective at these candidate solutions in order to choose the best of these candidates, specifically, as follows: we used a relatively short sample to choose the two "most promising" of the candidate solutions, and then a large sample (of size $K \gg N$) to identify the best of these two candidates, thus getting the "final" solution. The computational effort required by this simple post-processing is *not* reflected in the tables to follow.

The stepsizes At the "pilot stage" of our experimentation, we made a decision on which stepsize policy – (2.2.45) or (2.2.53) to choose, and how to identify the underlying parameters M_* and θ. In all our experiments, M_* was estimated by taking the maxima of $\|\mathsf{G}(\cdot,\cdot)\|_*$ over a small (just 100) calls to the stochastic oracle at randomly generated feasible solutions. As about the value of θ and type of the stepsize policy ((2.2.45) or (2.2.53)), our choice was based on the results of experimentation with a single test problem (instance L1 of the utility problem, see below); some results of this experimentation are presented in Table 1. We have found that the constant stepsize policy (2.2.45) with $\theta = 0.1$ for the Euclidean and $\theta = 5$ for the Non-Euclidean SA slightly outperforms other variants we have considered. This particular policy, combined with the aforementioned scheme for estimating M_*, was used in all subsequent experiments.

Format of test problems All our test problems are of the form $\min_{x\in X} f(x)$, $f(x) = \mathbb{E}[F(x,\xi)]$, where the domain X either is a standard simplex $\{x \in \mathbb{R}^n : x \geq 0, \sum_i x_i = 1\}$,

Table 1: selecting stepsize policy
[method: N-SA, N:2,000, K:10,000, instance: L1]

	θ			
policy	0.1	1	5	10
variable	-7.4733	-7.8865	-7.8789	-7.8547
constant	-6.9371	-7.8637	-7.9037	-7.8971

or can be converted into such a simplex by scaling of the original variables.

Notation in the tables Below,

- n is the design dimension of an instance,
- N is the sample size (i.e., the number of steps in SA, and the size of the sample used to build the stochastic average in SAA),
- `Obj` is the empirical mean of the random variable $F(x, \xi)$, x being the approximate solution generated by the algorithm in question. The empirical mean are taken over a large ($K = 10^4$ elements) dedicated sample,
- `CPU` is the CPU time in seconds,

2.3.2 A stochastic utility problem

Our first experiment was carried out with the utility model

$$\min_{x \in X} \left\{ f(x) = \mathbb{E}\left[\phi\big(\textstyle\sum_{i=1}^{n} (i/n + \xi_i) x_i \big) \right] \right\}, \tag{2.3.1}$$

where $X = \{x \in \mathbb{R}^n : x \geq 0, \sum_{i=1}^{n} x_i = 1\}$, $\xi_i \sim N(0, 1)$ are independent and $\phi(\cdot)$ is a piecewise linear convex function given by $\phi(t) = \max\{v_1 + s_1 t, ..., v_m + s_m t\}$, where v_k and s_k are certain constants. In our experiment, we used $m = 10$ breakpoints, all located on $[0, 1]$. The four instances L1, L2, L3, L4 we dealt with were of dimension varying from 500 to 2000, each instance – with its own randomly generated function ϕ. All the algorithms were coded in ANSI C and the experiments were conducted on a Intel PIV 1.6GHz machine with Microsoft Windows XP professional.

We run each of the three afore-mentioned methods with various sample sizes on every one of the instances. The results are reported in Table 2.

In order to evaluate stability of the algorithms, we run each of them 100 times; the resulting statistics as shown in Table 3. In this relatively time-consuming experiment, we

Table 2: SA vs. SAA on the stochastic utility problem

-		L1: $n = 500$		L2: $n = 1000$		L3: $n = 2000$		L4: $n = 5000$	
ALG.	N	Obj	CPU	Obj	CPU	Obj	CPU	Obj	CPU
N-SA	100	-7.7599	0	-5.8340	0	-7.1419	1	-5.4688	3
	1,000	-7.8781	2	-5.9152	2	-7.2312	6	-5.5716	13
	2,000	-7.8987	2	-5.9243	5	-7.2513	10	-5.5847	25
	4,000	-7.9075	5	-5.9365	12	-7.2595	20	-5.5935	49
E-SA	100	-7.6895	0	-5.7988	1	-7.0165	1	-4.9364	4
	1,000	-7.8559	2	-5.8919	4	-7.2029	7	-5.3895	20
	2,000	-7.8737	3	-5.9067	7	-7.2306	15	-5.4870	39
	4,000	-7.8948	7	-5.9193	13	-7.2441	29	-5.5354	77
SAA	100	-7.6571	7	-5.6346	8	-6.9748	19	-5.3360	44
	1,000	-7.8821	31	-5.9221	68	-7.2393	134	-5.5656	337
	2,000	-7.9100	72	-5.9313	128	-7.2583	261	-5.5878	656
	4,000	-7.9087	113	-5.9384	253	-7.2664	515	-5.5967	1283

Table 3: The variability for the stochastic utility problem

-		N-SA			E-SA			SAA		
		Obj		CPU	Obj		CPU	Obj		CPU
INST	N	MEAN	DEV	(avg.)	MEAN	DEV	(avg.)	MEAN	DEV	(avg.)
L2	1,000	-5.9159	0.0025	2.63	-5.8925	0.0024	4.99	-5.9219	0.0047	67.31
L2	2,000	-5.9258	0.0022	5.03	-5.9063	0.0019	7.09	-5.9328	0.0028	131.25

restrict ourselves with a single instance (L2) and just two sample sizes ($N = 1000$ and 2000). In Table 3, 'MEAN' and 'DEV' are, respectively, the mean and the deviation, over 100 runs, of the objective value Obj at the resulting approximate solution.

The experiments demonstrate that as far as the quality of approximate solutions is concerned, N-SA outperforms E-SA and is almost as good as SAA. At the same time, the solution time for N-SA is significantly smaller than the one for SAA.

2.3.3 Stochastic max-flow problem

In the second experiment, we consider a simple two-stage stochastic linear programming, namely, a stochastic max-flow problem. The problem is to optimize the capacity expansion of a stochastic network. Let $G = (N, A)$ be a diagraph with a source node s and a sink node t. Each arc $(i, j) \in A$ has an existing capacity $p_{ij} \geq 0$, and a random implementing/operating level ξ_{ij}. Moreover, there is a common random degrading factor η for all arcs in A. The goal is to determine how much capacity to add to the arcs, subject to a budget constraint, in order to maximize the expected maximum flow from s to t. Denoting by x_{ij}

Table 4: SA vs. SAA on the stochastic max-flow problem

-		F1		F2		F3		F4	
(m,n)		$(50,500)$		$(100,1000)$		$(100,2000)$		$(250,5000)$	
ALG.	N	Obj	CPU	Obj	CPU	Obj	CPU	Obj	CPU
N-SA	100	0.1140	0	0.0637	0	0.1296	1	0.1278	3
	1000	0.1254	1	0.0686	3	0.1305	6	0.1329	15
	2000	0.1249	3	0.0697	6	0.1318	11	0.1338	29
	4000	0.1246	5	0.0698	11	0.1331	21	0.1334	56
E-SA	100	0.0840	0	0.0618	1	0.1277	2	0.1153	7
	1000	0.1253	3	0.0670	6	0.1281	16	0.1312	39
	2000	0.1246	5	0.0695	13	0.1287	28	0.1312	72
	4000	0.1247	9	0.0696	24	0.1303	53	0.1310	127
SAA	100	0.1212	5	0.0653	12	0.1310	20	0.1253	60
	1000	0.1223	35	0.0694	84	0.1294	157	0.1291	466
	2000	0.1223	70	0.0693	170	0.1304	311	0.1284	986
	4000	0.1221	140	0.0693	323	0.1301	636	0.1293	1885

the capacity to be added to arc (i,j), the problem reads

$$\max_x \left\{ f(x) = \mathbb{E}[F(x;\xi,\eta)] : \sum_{(i,j)\in A} c_{ij}x_{ij} \le b, x_{ij} \ge 0, \quad \forall (i,j) \in A \right\}, \tag{2.3.2}$$

where c_{ij} is the per unit cost for the capacity to be added, b is the total available budget, and $F(x;\xi,\eta)$ denotes the maximum $s-t$ flow in the network when the capacity of an arc (i,j) is $\eta\xi_{ij}(p_{ij}+x_{ij})$. Note that the above is a maximization rather than a minimization problem.

We assume that the random variables ξ_{ij}, θ are independent and uniformly distributed on $[0,1]$ and $[0.5,1]$, respectively, and consider the case of $p_{ij}=0$, $c_{ij}=1$ for all $(i,j)\in E$, and $b=1$. We randomly generated 4 network instances (referred to as F1, F2, F3 and F4) using the network generator GRIDGEN available on DIMACS challenge. The push-relabel algorithm [19] was used to solve the second stage max-flow problem.

In the first test, each algorithm (N-SA, E-SA, SAA) was run once at each test instance; the results are reported in Table 4, where m, n stand for the number of nodes, resp., arcs in G. Similar to the stochastic utility problem, we investigate the stability of the methods by running each of them 100 times. The resulting statistics is presented in Table 5 whose columns have exactly the same meaning as in Table 3.

This experiment fully supports the conclusions on the methods suggested by the experiments with the utility problem.

Table 5: The variability for the stochastic max-flow problem

-		N-SA			E-SA			SAA		
		Obj		AVG.	Obj		AVG.	Obj		AVG.
INST	N	MEAN	DEV	CPU	MEAN	DEV	CPU	MEAN	DEV	CPU
F2	1,000	0.0691	0.0004	3.11	0.0688	0.0006	4.62	0.0694	0.0003	90.15
F2	2,000	0.0694	0.0003	6.07	0.0692	0.0002	6.91	0.0695	0.0003	170.45

2.3.4 A network planning problem with random demand

In the last experiment, we consider the so-called SSN problem of Sen, Doverspike, and Cosares [67]. This problem arises in telecommunications network design where the owner of the network sells private-line services between pairs of nodes in the network, and the demands are treated as random variables based on the historical demand patterns. The optimization problem is to decide where to add capacity to the network to minimize the expected rate of unsatisfied demands. Since this problem has been studied by several authors (see, e.g., [36, 67]), it could be interesting to compare the results. Another purpose of this experiment is to investigate the behavior of the SA method when one variance reduction technique, namely, the Latin Hyperplane Sampling (LHS), is applied.

The problem has been formulated as a two-stage stochastic linear programming as follows:

$$\min_x \left\{ f(x) = \mathbb{E}[F(x,\xi)] : x \geq 0, \sum_i x_i = b \right\}, \tag{2.3.3}$$

where x is the vector of capacities to be added to the arcs of the network, b (the budget) is the total amount of capacity to be added, ξ denotes the random demand, and $F(x,\xi)$ represents the number of unserved requests, specifically,

$$F(x,\xi) = \min_{s,f} \left\{ \sum_i s_i : \begin{array}{ll} \sum_i \sum_{r \in R(i)} A_r f_{ir} \leq x + c & \\ \sum_{r \in R(i)} f_{ir} + s_i = \xi^i, & \forall i \\ f_{ir} \geq 0, s_i \geq 0, & \forall i, r \in R(i) \end{array} \right\}. \tag{2.3.4}$$

Here,

- $R(i)$ is the set of routes used for traffic i (traffic between the source-sink pair of nodes # i),
- ξ^i is the (random) demand for traffic i,
- A_r are the route-arc incidence vectors (so that jth component of A_r is 1 or 0 depending

Table 6: SA vs. SAA on the SSN problem

-		Without LHS		With LHS	
ALG.	N	Obj	CPU	Obj	CPU
N-SA	100	11.0984	1	10.1024	1
	1,000	10.0821	6	10.0313	7
	2,000	9.9812	12	9.9936	12
	4,000	9.9151	23	9.9428	22
E-SA	100	10.9027	1	10.3860	1
	1,000	10.1268	6	10.0984	6
	2,000	10.0304	12	10.0552	12
	4,000	9.9662	23	9.9862	23
SAA	100	11.8915	24	11.0561	23
	1,000	10.0939	215	10.0488	216
	2,000	9.9769	431	9.9872	426
	4,000	9.8773	849	9.9051	853

on whether arc j belongs to the route r),

• c is the vector of current capacities, f_{ir} is the fraction of traffic i transferred via route r, and s is the vector of unsatisfied demands.

In the SSN instance, there are $\dim x = 89$ arcs and $\dim \xi = 86$ source-sink pairs, and components of ξ are independent random variables with known discrete distributions (from 3 to 7 possible values per component), which results in $\approx 10^{70}$ possible demand scenarios.

In the first test with the SSN instance, each of our 3 algorithms was run once without, and once – with the Latin Hyperplane Sampling (LHS) technique; the results are reported in Table 6. We then tested the stability of algorithms by running each of them 100 times, see statistics in Table 7. Note that experiments with the SSN problem were conducted on a more powerful computer: Intel Xeon 1.86GHz with Red Hat Enterprise Linux.

As far as comparison of our three algorithms is concerned, the conclusions are in full agreement with those for the utility and the max-flow problem. We also see that for our particular example, the Latin Hyperplane sampling does not yield much of improvement, especially when a larger sample-size is applied. This result seems to be consistent with the observation in [36].

2.3.5 N-SA vs. E-SA

The data in Tables 3, 4, 6 demonstrate that with the same sample size N, the N-SA somehow outperforms the E-SA in terms of both the quality of approximate solutions and

Table 7: The variability for the SSN problem

-		N-SA			E-SA			SAA		
		Obj		AVG.	Obj		AVG.	Obj		AVG.
N	LHS	MEAN	DEV	CPU	MEAN	DEV	CPU	MEAN	DEV	CPU
1,000	no	10.0624	0.1867	6.03	10.1730	0.1826	6.12	10.1460	0.2825	215.06
1,000	yes	10.0573	0.1830	6.16	10.1237	0.1867	6.14	10.0135	0.2579	216.10
2,000	no	9.9965	0.2058	11.61	10.0853	0.1887	11.68	9.9943	0.2038	432.93
2,000	yes	9.9978	0.2579	11.71	10.0486	0.2066	11.74	9.9830	0.1872	436.94

the running time[2]. The difference in solutions' quality, at the first glance, seems slim, and one could think that adjusting the SA algorithm to the "geometry" of the problem in question (in our case, to minimization over a standard simplex) is of minor importance. We, however, do believe that such a conclusion would be wrong. In order to get a better insight, let us come back to the stochastic utility problem. This test problem has an important advantage – we can easily compute the value of the objective $f(x)$ at a given candidate solution x analytically[3]. Moreover, it is easy to minimize $f(x)$ over the simplex – on a closest inspection, this problem reduces to minimizing an easy-to-compute *univariate* convex function, so that we can approximate the true optimal value f_* to high accuracy by Bisection. Thus, in the case in question we can compare solutions x generated by various algorithms in terms of their "true inaccuracy" $f(x)-f_*$, and this is the rationale behind our "Gaussian setup". We can now exploit this advantage of the stochastic utility problem for comparing properly N-SA and E-SA. In Table 8, we present the true values of the objective $f(\bar{x})$ at the approximate solutions $\bar{x}$ generated by N-SA and E-SA as applied to the instances L1 and L4 of the utility problem (cf. Table 3) along with the inaccuracies $f(\bar{x})-f_*$ and the Monte Carlo estimates $\widehat{f}(\bar{x})$ of $f(\bar{x})$ obtained via 50,000-element samples. We see that the difference in the inaccuracy $f(\bar{x})-f_*$ of the solutions produced by the algorithms is much more significant than it is suggested by the data in Table 3 (where the actual inaccuracy is "obscured" by the estimation error and summation with f_*). Specifically, at the common for both algorithms sample size $N=2,000$, the inaccuracy yielded by N-SA is 3 – 5 times less

[2]The difference in running times can be easily explained: with X being a simplex, the prox-mapping for E-SA takes $O(n\ln n)$ operations vs. $O(n)$ operations for N-SA.

[3]Indeed, $(\xi_1,...,\xi_n)\sim\mathcal{N}(0,I_n)$, so that the random variable $\xi_x=\sum_i(a_i+\xi_i)x_i$ is normal with easily computable mean and variance, and since ϕ is piecewise linear, the expectation $f(x)=\mathbb{E}[\phi(\xi_x)]$ can be immediately expressed via the error function.

Table 8: N-SA vs. E-SA

Method	Problem	$\widehat{f}(\bar{x}), f(\bar{x})$	$f(\bar{x}) - f_*$	Time
N-SA, $N = 2,000$	L2: $n = 1000$	-5.9232/-5.9326	0.0113	5.00
E-SA, $N = 2,000$	L2	-5.8796/-5.8864	0.0575	6.60
E-SA, $N = 10,000$	L2	-5.9059/-5.9058	0.0381	39.80
E-SA, $N = 20,000$	L2	-5.9151/-5.9158	0.0281	74.50
N-SA, $N = 2,000$	L4: $n = 5000$	-5.5855/-5.5867	0.0199	25.00
E-SA, $N = 2,000$	L4	-5.5467/-5.5469	0.0597	44.60
E-SA, $N = 10,000$	L4	-5.5810/-5.5812	0.0254	165.10
E-SA, $N = 20,000$	L4	-5.5901/-5.5902	0.0164	382.00

than the one for E-SA, and in order to compensate for this difference, one should increase the sample size for E-SA (and hence the running time) by factor 5 – 10. It should be added that in light of theoretical complexity analysis carried out in Example 2.2.1, the outlined significant difference in performances of N-SA and E-SA is not surprising; the surprising fact is that E-SA works at all.

2.4 Conclusions of this chapter

It is shown in this chapter that for a certain class of convex stochastic optimization problems, robust versions of the SA approach have similar theoretical estimates of computational complexity, in terms of the required sample size, to the SAA method. Numerical experiments, reported in Section 2.3, confirm this conclusion. These results demonstrate that for considered problems, a properly implemented mirror descent SA algorithm produces solutions of comparable accuracy to the SAA method for the same sample size of generated random points. On the other hand, the implementation (computational) time of the SA method is significantly smaller with a factor of up to $30 - 40$ for considered problems. Thus, both theoretical and numerical results suggest that the mirror descent SA is a viable alternative to the SAA approach, an alternative which at least deserves testing in particular applications.

CHAPTER III

VALIDATION ANALYSIS OF MIRROR DESCENT STOCHASTIC APPROXIMATION

3.1 Overview

In Chapter 2, we introduce the mirror descent SA method applied to problem (1.2.1) and demonstrate that this approach can be competitive and even significantly outperform the SAA method for a certain class of convex stochastic problems, for example, when the set X is a standard simplex. Certain issues related to the mirror descent SA remains to be addressed. One outstanding problem for the SA methods is the absence of a validation procedure to estimate the accuracy of the generated solutions. On the other hand, an important methodological property of the SAA approach is that, with some additional effort, it can provide an estimate of the accuracy of an obtained solution by computing upper and lower (confidence) bounds for the optimal value of the true problem (cf., [40, 75]). The main goal of this chapter is to show that, for a certain class of stochastic convex problems, the mirror descent SA method can also provide similar bounds with considerably less computational effort. More specifically we study in this chapter the following aspects of the mirror descent SA method.

- Investigate different ways to estimate lower and upper bounds for the objective values by the mirror descent SA method, and thus to obtain an accuracy certificate for the attained solutions.

- Adjust the mirror descent SA method to solve two interesting application problems in asset allocation, namely, minimizing[1] the Expected Utility (EU) and minimizing the Conditional Value-at-Risk (CVaR). These models are widely used in practice, for

[1] In order to have a convex rather than concave objective function, we deal here with minimization rather than maximization of the Expected Utility.

example, by investment companies, brokerage firms, mutual funds, and any business that evaluates risks (cf., [62]).

- Understand the performance of the mirror descent SA algorithm for solving stochastic programs with a feasible region more complicated than a simplex. For the EU model, the feasible region is the intersection of a simplex with a box constraint and we will compare two different variants of SA methods for solving it. For the CVaR problem, the feasible region is a polyhedron and we will discuss some techniques to explore its structure.

This chapter is organized as follows. In section 3.2 we give a brief summary to the mirror descent SA method. Section 3.3 is devoted to a derivation and analysis of statistical upper and lower bounds for the optimal value of the true problem. In section 3.4 we discuss an application of the mirror descent SA method to the expected utility and conditional value at risk approaches for the asset allocation problem. A discussion of numerical results is presented in section 3.5. Proofs of the main technical results are given in Section 3.6. Finally, some concluding remarks are made in Section 3.7.

3.1.1 Notation and terminology

For a norm $\|\cdot\|$ on $\mathbb{R}^n$, we denote by $\|x\|_* := \sup\{x^T y : \|y\| \leq 1\}$ the conjugate norm. By $\|x\|_p$ we denote the ℓ_p norm of vector $x \in \mathbb{R}^n$. In particular, $\|x\|_2 = \sqrt{x^T x}$ is the Euclidean norm of $x \in \mathbb{R}^n$. By $\Pi_X(x) := \arg\min_{y \in X} \|x - y\|_2$ we denote metric projection operator onto X. For the process $\xi_1, \xi_2, ...$, we set $\xi^t := (\xi_1, ..., \xi_t)$, and denote by $\mathbb{E}_{|t}$ or by $\mathbb{E}[\cdot|\xi^t]$ the conditional, ξ^t being given, expectation. For a number $a \in \mathbb{R}$ we denote $[a]_+ := \max\{a, 0\}$. By $\partial\phi(x)$ we denote the subdifferential of a convex function $\phi(x)$.

3.2 The mirror descent Stochastic Approximation Method

For the readers' convenience, in this section, we give a brief summary to the mirror descent SA algorithm introduced in Chapter 2. We equip the embedding space $\mathbb{R}^n$, of the feasible domain X of (1.2.1), with a norm $\|\cdot\|$.

Throughout the chapter we assume existence of the following *stochastic oracle*.

- It is possible to generate an iid sample $\xi_1, \xi_2, ...,$ of realizations of random vector ξ, and we have access to a "black box" subroutine (a stochastic oracle): given $x \in X$ and a random realization $\xi \in \Xi$, the oracle returns the quantity $F(x, \xi)$ and a *stochastic subgradient* – a vector $\mathsf{G}(x, \xi)$ such that $\mathsf{g}(x) := \mathbb{E}[\mathsf{G}(x, \xi)]$ is well defined and is a subgradient of $f(\cdot)$ at x, i.e., $\mathsf{g}(x) \in \partial f(x)$.

We also make the following assumption.

(A.3.1) There are positive constants Q and M_* such that for any $x \in X$:

$$\mathbb{E}\left[(F(x, \xi) - f(x))^2\right] \leq Q^2, \tag{3.2.1}$$

$$\mathbb{E}\big[\|\mathsf{G}(x, \xi)\|_*^2\big] \leq M_*^2. \tag{3.2.2}$$

It could be noted that $\mathbb{E}\left[(F(x, \xi) - f(x))^2\right]$ in (3.2.1) is variance of the random variable $F(x, \xi)$.

When speaking about Stochastic Approximation as applied to minimization problem (1.2.1), one usually does not care of how the values of $f(\cdot)$ are observed. All what matters is the observations of the gradient, this is the only information used by the basic SA algorithm (2.2.32), see also (3.2.3) below. We, however, are interested in building upper and lower bounds on the optimal value and/or value of $f(\cdot)$ at a given solution, and in this respect, it does matter how these values are observed. Conditions (3.2.1)–(3.2.2) of assumption (A.3.1) impose restrictions on the magnitudes of noises in the unbiased observations of the values of $f(\cdot)$ and the subgradients of $f(\cdot)$ reported by the stochastic oracle.

The description of the mirror descent SA algorithm is as follows. Starting from point x_1, the algorithm iteratively generates points $x_t \in X^o$ according to the recurrence

$$x_{t+1} := P_{x_t}\big(\gamma_t \mathsf{G}(x_t, \xi_t)\big), \tag{3.2.3}$$

where $\gamma_t > 0$ are deterministic stepsizes and $P_{x_t}(\cdot)$ is the prox-mapping defined in (2.2.28). Note that for $\omega(x) := \frac{1}{2}\|x\|_2^2$, we have that $P_x(y) = \Pi_X(x - y)$ and hence $x_{t+1} = \Pi_X\big(x_t - \gamma_t \mathsf{G}(x_t, \xi_t)\big)$. In that case, the mirror descent SA method is referred to as the *Euclidean SA*.

Now let N be the total number of steps. Let us set

$$\nu_t := \frac{\gamma_t}{\sum_{i=1}^N \gamma_i},\ t = 1, ..., N, \ \text{ and } \ \tilde{x}_N := \sum_{t=1}^N \nu_t x_t. \tag{3.2.4}$$

Note that $\sum_{t=1}^N \nu_t = 1$, and hence $\tilde{x}_N$ is a convex combination of the iterates $x_1, ..., x_N$. Here $\tilde{x}_N$ is considered as the approximate solution generated by the algorithm in course of N steps. The quality of this solution can be quantified as follows (cf., (2.2.42)).

Proposition 3.2.1 *Suppose that condition* (3.2.2) *of assumption* (A.3.1) *holds. Then for the N-step of mirror descent SA algorithm we have that*

$$\mathbb{E}\left[f(\tilde{x}_N) - f^*\right] \leq \frac{D^2_{\omega,X} + (2\alpha)^{-1} M_*^2 \sum_{t=1}^N \gamma_t^2}{\sum_{t=1}^N \gamma_t}. \tag{3.2.5}$$

In implementations of the SA algorithm different stepsize strategies can be applied to (3.2.3) (see Section 2.2.3). We discuss now the *constant stepsize* policy. That is, we assume that the number N of iterations is fixed in advance, and $\gamma_t = \gamma$, $t = 1, ..., N$. In that case

$$\tilde{x}_N = \frac{1}{N} \sum_{t=1}^N x_t. \tag{3.2.6}$$

By choosing the stepsizes as

$$\gamma_t = \gamma := \frac{\theta\sqrt{2\alpha} D_{\omega,X}}{M_* \sqrt{N}}, \quad t = 1, ..., N, \tag{3.2.7}$$

with a (scaling) constant $\theta > 0$, we have in view of (3.2.5) that

$$\mathbb{E}\left[f(\tilde{x}_N) - f^*\right] \leq \max\{\theta, \theta^{-1}\} \Omega_{\omega,X} M_* N^{-1/2}, \tag{3.2.8}$$

with $\Omega_{\omega,X}$ given by (2.2.31). This shows that scaling the stepsizes by the (positive) constant θ results in updating the estimate (3.2.8) by the factor of $\max\{\theta, \theta^{-1}\}$ at most. By Markov inequality it follows from (3.2.8) that for any $\varepsilon > 0$,

$$\mathrm{Prob}\big\{f(\tilde{x}_N) - f^* > \varepsilon\big\} \leq \frac{\sqrt{2}\max\{\theta, \theta^{-1}\} D_{\omega,X} M_*}{\varepsilon\sqrt{\alpha N}}. \tag{3.2.9}$$

It is possible to obtain finer bounds for the probabilities in the left hand side of (3.2.9) when imposing conditions more restrictive than conditions of assumption(A.3.1). Consider the following conditions.

(A.3.2) There are positive constants Q and M_* such that for any $x \in X$:

$$\mathbb{E}\left[\exp\left\{|F(x,\xi) - f(x)|^2/Q^2)\right\}\right] \leq \exp\{1\}, \tag{3.2.10}$$

$$\mathbb{E}\left[\exp\left\{\|\mathsf{G}(x,\xi)\|_*^2/M_*^2\right\}\right] \leq \exp\{1\}. \tag{3.2.11}$$

Note that conditions (3.2.10)–(3.2.11) are stronger than the respective conditions (3.2.1)–(3.2.2). Indeed, if a random variable Y satisfies $\mathbb{E}[\exp\{Y/a\}] \leq \exp\{1\}$ for some $a > 0$, then by Jensen inequality $\exp\{\mathbb{E}[Y/a]\} \leq \mathbb{E}[\exp\{Y/a\}] \leq \exp\{1\}$, and therefore $\mathbb{E}[Y] \leq a$. Of course, conditions (3.2.10)–(3.2.11) hold if for all $(x, \xi) \in X \times \Xi$:

$$|F(x, \xi) - f(x)| \leq Q \text{ and } \|\mathsf{G}(x, \xi)\|_* \leq M_*.$$

The following result has been established in [43, Proposition 2.2].

Proposition 3.2.2 *Suppose that condition* (3.2.11) *of assumption* (A.3.2) *holds. Then for the constant stepsize policy, with the stepsize* (3.2.7)*, the following inequality holds for any* $\Lambda \geq 1$*:*

$$\text{Prob}\left\{f(\tilde{x}_N) - f^* > \max\{\theta, \theta^{-1}\}(12 + 2\Lambda)\Omega_{\omega,X} M_* N^{-1/2}\right\} \leq 2\exp\{-\Lambda\}. \tag{3.2.12}$$

It follows from (3.2.12) that the number N of steps required by the algorithm to solve the problem with accuracy $\varepsilon > 0$, and a (probabilistic) confidence $1 - \beta$, is of order $O\left(\varepsilon^{-2}\log^2(1/\beta)\right)$. Note also that in practice one can modify the mirror descent SA algorithm so that the approximate solution $\tilde{x}_N$ is obtained by averaging over a part of the trajectory (see Section 2.2.3 for details).

3.3 Accuracy certificates for SA solutions

In this section, we discuss several ways to estimate lower and upper bounds for the optimal value of problem (1.2.1), which gives us an accuracy certificate for obtained solutions. Specifically, we distinguish between two types of certificates: the *online certificates* that can be computed quickly when running the SA algorithm, and the *offline certificates* obtained in a more time consuming way at the dedicated *validation step*, after a solution has been obtained.

3.3.1 Online certificate

Consider the numbers ν_t and solution $\tilde{x}_N$, defined in (3.2.4), functions

$$f^N(x) := \sum_{t=1}^N \nu_t \left[f(x_t) + \mathsf{g}(x_t)^T(x - x_t)\right] \text{ and } \hat{f}^N(x) := \sum_{t=1}^N \nu_t[F(x_t, \xi_t) + \mathsf{G}(x_t, \xi_t)^T(x - x_t)],$$

and define

$$f_*^N := \min_{x\in X} f^N(x) \text{ and } f^{*N} := \sum_{t=1}^N \nu_t f(x_t). \tag{3.3.1}$$

Since $\nu_t > 0$ and $\sum_{t=1}^N \nu_t = 1$, it follows by convexity of $f(\cdot)$ that the function $f^N(\cdot)$ underestimates $f(\cdot)$ everywhere on X, and hence $f_*^N \le f^*$. Since $\tilde{x}_N \in X$ we also have that $f^* \le f(\tilde{x}_N)$, and by convexity of $f(\cdot)$ that $f(\tilde{x}_N) \le f^{*N}$. That is, for *any realization* of the random sample $\xi_1, ..., \xi_N$ we have that

$$f_*^N \le f^* \le f(\tilde{x}_N) \le f^{*N}. \tag{3.3.2}$$

It follows from (3.3.2) that $\mathbb{E}[f_*^N] \le f^* \le \mathbb{E}[f^{*N}]$ as well.

Of course, the bounds f_*^N and f^{*N} are unobservable since the values $f(x_t)$ are not known exactly. Therefore we consider their computable counterparts

$$\underline{f}^N = \min_{x\in X} \hat{f}^N(x) \text{ and } \overline{f}^N = \sum_{t=1}^N \nu_t F(x_t, \xi_t). \tag{3.3.3}$$

We refer to $\underline{f}^N$ and $\overline{f}^N$ as *online* bounds. The bound $\overline{f}^N$ can be easily calculated while running the SA procedure. The bound $\underline{f}^N$ involves solving the optimization problem of minimizing a linear in x objective function over set X. If the set X is defined by linear constraints, this is a linear programming problem.

Since x_t is a function of $\xi^{t-1} = (\xi_1, ..., \xi_{t-1})$, and ξ_t is independent of ξ^{t-1}, we have that

$$\mathbb{E}\big[\overline{f}^N\big] = \sum_{t=1}^N \nu_t \mathbb{E}\big\{\mathbb{E}[F(x_t, \xi_t)|\xi^{t-1}]\big\} = \sum_{t=1}^N \nu_t \mathbb{E}\left[f(x_t)\right] = \mathbb{E}[f^{*N}]$$

and

$$\begin{aligned}
\mathbb{E}\big[\underline{f}^N\big] &= \mathbb{E}\left[\mathbb{E}\big\{\min_{x\in X}\big[\textstyle\sum_{t=1}^N \nu_t[F(x_t,\xi_t) + \mathsf{G}(x_t,\xi_t)^T(x-x_t)]\big]\big|\xi^{t-1}\big\}\right] \\
&\le \mathbb{E}\left[\min_{x\in X}\big\{\mathbb{E}\big[\textstyle\sum_{t=1}^N \nu_t[F(x_t,\xi_t) + \mathsf{G}(x_t,\xi_t)^T(x-x_t)]\big]\big|\xi^{t-1}\big\}\right] \\
&= \mathbb{E}\left[\min_{x\in X} f^N(x)\right] = \mathbb{E}\big[f_*^N\big].
\end{aligned}$$

It follows that

$$\mathbb{E}\big[\underline{f}^N\big] \le f^* \le \mathbb{E}\big[\overline{f}^N\big]. \tag{3.3.4}$$

That is, on average $\underline{f}^N$ and $\overline{f}^N$ give, respectively, a lower and an upper bound for the optimal value of problem (1.2.1). In order to see how good are the bounds $\underline{f}^N$ and $\overline{f}^N$

let us estimate expectations and probabilities of the corresponding errors. Proof of the following theorem is given in the Section 3.6.

Theorem 3.3.1 (i) *Suppose that assumption* (A.3.1) *holds. Then*

$$\mathbb{E}\big[f^{*N} - f_*^N\big] \le \frac{2D^2_{\omega,X} + \frac{5}{2}\alpha^{-1}M_*^2\sum_{t=1}^N \gamma_t^2}{\sum_{t=1}^N \gamma_t}, \tag{3.3.5}$$

$$\mathbb{E}\left[\big|\overline{f}^N - f^{*N}\big|\right] \le Q\sqrt{\sum_{t=1}^N \nu_t^2}, \tag{3.3.6}$$

$$\mathbb{E}\left[\big|\underline{f}^N - f_*^N\big|\right] \le \frac{D^2_{\omega,X} + \frac{1}{2}\alpha^{-1}M_*^2\sum_{t=1}^N \gamma_t^2}{\sum_{t=1}^N \gamma_t} + \big(Q + 8\Omega_{\omega,X}M_*\big)\sqrt{\sum_{t=1}^N \nu_t^2}. \tag{3.3.7}$$

In particular, in the case of constant stepsize policy (3.2.7) *we have*

$$\begin{aligned}
&\mathbb{E}\big[f^{*N} - f_*^N\big] \le \big[\theta^{-1} + 5\theta/2\big]\,\Omega_{\omega,X}M_*N^{-1/2},\\
&\mathbb{E}\left[\big|\overline{f}^N - f^{*N}\big|\right] \le QN^{-1/2},\\
&\mathbb{E}\left[\big|\underline{f}^N - f_*^N\big|\right] \le \tfrac{1}{2}\left[\theta^{-1} + \theta\right]\Omega_{\omega,X}M_*N^{-1/2} + \big(Q + 8\Omega_{\omega,X}M_*\big)N^{-1/2},
\end{aligned} \tag{3.3.8}$$

where $\Omega_{\omega,X}$ *is given by* (2.2.31).

(ii) *Moreover, if assumption* (A.3.1) *is strengthened to assumption* (A.3.2), *then in the case of constant*[2] *stepsize policy* (3.2.7) *we have for any* $\Lambda \ge 0$*:*

$$\begin{aligned}
&\mathrm{Prob}\left\{f^{*N} - f_*^N > N^{-1/2}\Omega_{\omega,X}M_*\left(\left[\tfrac{5}{2}\theta + \theta^{-1}\right] + \Lambda\left[4 + \tfrac{5}{2}\theta N^{-1/2}\right]\right)\right\}\\
&\qquad \le 2\exp\{-\Lambda^2/3\} + 2\exp\{-\Lambda^2/12\} + 2\exp\{-3\Lambda\sqrt{N}/4\}\ ,
\end{aligned} \tag{3.3.9}$$

$$\mathrm{Prob}\left\{\big|\overline{f}^N - f^{*N}\big| > \Lambda Q\sqrt{\textstyle\sum_{t=1}^N \nu_t^2}\right\} \le 2\exp\{-\Lambda^2/3\}, \tag{3.3.10}$$

$$\begin{aligned}
&\mathrm{Prob}\left\{|\underline{f}^N - f_*^N| > N^{-1/2}\left(\left[\tfrac{1}{2\theta} + 2\theta\right]\Omega_{\omega,X}M_* + \Lambda\left[Q + [8 + 2\theta N^{-1/2}]\Omega_{\omega,X}M_*\right]\right)\right\}\\
&\qquad \le 6\exp\{-\Lambda^2/3\} + \exp\{-\Lambda^2/12\} + \exp\{-3\Lambda\sqrt{N}/4\}.
\end{aligned} \tag{3.3.11}$$

[2] *The bounds in the Section 3.6 cover the case of general-type stepsizes; here we restrict ourselves with the case of constant stepsizes to avoid less transparent formulas.*

Estimates of the above theorem show that as N grows, the observable quantities $\underline{f}^N$ and $\overline{f}^N$ approach, in a probabilistic sense, their unobservable counterparts, which, in turn, approach each other and thus the optimal value of problem (1.2.1). For the constant stepsize policy (3.2.7), we have that all estimates given in the right hand side of (3.3.8) are of order $O(N^{-1/2})$. It follows that under assumption (A.3.1) and for the constant stepsize policy, difference between the upper $\overline{f}^N$ and lower $\underline{f}^N$ bounds converges on average to zero, with increase of the sample size N, at a rate of $O(N^{-1/2})$.

Note that for the constant stepsize policy (3.2.7) and under assumption (A.3.2), the bounds (3.3.9) – (3.3.11) combine with (3.3.2) to imply that

- Prob $\left\{\overline{f}^N_\Lambda := \overline{f}^N + \Lambda\sigma_+ N^{-1/2} \textit{is not an upper bound on } f(\widetilde{x}^N)\right\} \leq 2\varepsilon^{-\frac{\Lambda^2}{3}}$, with $\sigma_+ = Q$;

- Prob $\left\{\underline{f}^N_\Lambda := \underline{f}^N - [\mu_- + \Lambda\sigma_-]N^{-1/2} \textit{is not a lower bound on } f^*\right\} \leq 6\varepsilon^{-\frac{\Lambda^2}{3}} + \varepsilon^{-\frac{\Lambda^2}{12}} + \varepsilon^{-\frac{3\Lambda\sqrt{N}}{4}}$, with $\Omega_{\omega,X}$ defined by (2.2.31) and

$$\mu_- := \left[\frac{1}{2\theta} + 2\theta\right]\Omega_{\omega,X}M_*,\ \sigma_- := Q + [8 + 2\theta N^{-1/2}]\Omega_{\omega,X}M_*;$$

- Prob $\left\{\overline{f}^N_\Lambda - \underline{f}^N_\Lambda > [\mu + \Lambda\sigma]N^{-1/2}\right\} \leq 10\varepsilon^{-\frac{\Lambda^2}{3}} + 3\varepsilon^{-\frac{\Lambda^2}{12}} + 3\varepsilon^{-\frac{3\Lambda\sqrt{N}}{4}}$, with

$$\mu := \left[\frac{3}{2\theta} + \frac{9\theta}{2}\right]\Omega_{\omega,X}M_*,\ \sigma := 2Q + \left[12 + \frac{9\theta}{2}\right]\Omega_{\omega,X}M_*.$$

Theorem 3.3.1 shows that for large N the online observable random quantities $\overline{f}^N$ and $\underline{f}^N$ are close to the upper bound f^{*N} and lower bound f^N_*, respectively. Besides this, on average, $\overline{f}^N$ indeed overestimates f^*, and $\underline{f}_N$ indeed underestimates f^*. To save words, let us call random estimates which on average under- or overestimate a certain quantity, *on average* lower, respectively, upper bounds on this quantity. From now on, when speaking of "true" lower and upper bounds – those which always (or almost surely) under-, respectively, overestimate the quantity, we add the adjective "valid". Thus, we refer to f^{*N} and f^N_* as *valid* upper and lower bounds on f^*, respectively. Recall that f^{*N} is also a valid upper bound on $f(\tilde{x}_N)$.

Remark 3.3.1 *Recall that the SAA approach also provides a lower on average bound – the random quantity $\hat{f}^N_{SAA}$, which is the optimal value of the sample average problem (cf., [40, 75]). Suppose the same sample ξ_t, $t = 1, \ldots, N$, is applied for both SA and SAA methods. Besides this, assume that the constant stepsize policy is used in the SA method, and hence $\nu_t = 1/N$, $t = 1, .., N$. Finally, assume (as it often is the case) that $\mathsf{G}(x, \xi)$ is a subgradient of $F(x, \xi)$ in x. By convexity of $F(\cdot, \xi)$ and since $\underline{f}^N = \min_{x \in X} \hat{f}^N(x)$, we have*

$$\hat{f}^N_{SAA} := \min_{x \in X} N^{-1} \sum_{t=1}^{N} F(x, \xi_t) \geq \min_{x \in X} \sum_{t=1}^{N} \nu_t \left(F(x_t, \xi_t) + \mathsf{G}(x_t, \xi_t)^T (x - x_t) \right) = \underline{f}^N. \quad (3.3.12)$$

That is, for the same sample the lower bound $\underline{f}^N$ is smaller than the lower bound obtained by the SAA method. However, it should be noted that the lower bound $\underline{f}^N$ is computed much faster than $\hat{f}^N_{SAA}$, since computing the latter one amounts to solving the sample average optimization problem associated with the generated sample. Moreover, we will discuss in the next subsection how to improve the lower bound $\underline{f}^N$. From the computational results, the improved lower bound is comparable to the one obtained by the SAA method. ∎

Remark 3.3.2 *Similar to the SAA method, in order to estimate the variability of the lower bound $\underline{f}^N$, one can run the SA procedure M times, with independent samples, each of size N, and consequently compute the average and sample variance of M realizations of the random quantity $\underline{f}^N$. Alternatively, one can run the SA procedure once but with NM iterations, then partition the obtained trajectory into M consecutive parts, each of size N, for each of these parts calculate the corresponding SA lower bound and consequently compute the average and sample variance of the M obtained numbers.* ∎

3.3.2 Offline certificate

Suppose now that the mirror descent SA method is terminated after N iterations. Given a solution $\tilde{x}_N$ obtained by this method, the objective value $f(\tilde{x}_N)$ can be estimated by Monte Carlo sampling. That is, an iid random sample ξ_j, $j = 1, \ldots, K$, (independent of the random sample used in computing $\tilde{x}_N$) is generated and $f(\tilde{x}_N)$ is estimated by $\hat{f}_K(\tilde{x}_N) := K^{-1} \sum_{j=1}^{K} F(\tilde{x}_N, \xi_j)$. Since this procedure does not require computing prox-mapping and the like, one can use here a large sample size K. Of course, we can expect

that $\hat{f}_K(\tilde{x}_N)$ is a better upper bound on $f(\tilde{x}_N)$ than the online counterpart $\overline{f}^N$ of the valid upper bound f^{*N}.

We now demonstrate that the online lower bound $\underline{f}^N$ can be also improved in the validation step. Given an iid random sample ξ_j, $j = 1, \ldots, S$, we can estimate the (linear in x) form $\ell_S(x; \tilde{x}_N) := f(\tilde{x}_N) + \mathsf{g}(\tilde{x}_N)^T(x - \tilde{x}_N)$ by

$$\hat{\ell}_S(x; \tilde{x}_N) := \frac{1}{S} \sum_{j=1}^{S} \left[F(\tilde{x}_N, \xi_j) + \mathsf{G}(\tilde{x}_N, \xi_j)^T (x - \tilde{x}_N) \right], \tag{3.3.13}$$

and hence construct the following lower bound of f^*:

$$\underline{lb}^N := \min_{x \in X} \left\{ \max \left[\hat{f}^N(x), \hat{\ell}_S(x; \tilde{x}_N) \right] \right\}. \tag{3.3.14}$$

Clearly, by definition we have that $\underline{lb}^N \geq \underline{f}^N$.

It is worth of noting that although $\mathbb{E}\big[\hat{f}^N(x)\big] \leq f(x)$ and $\mathbb{E}\big[\hat{\ell}_S(x; \tilde{x}_N)\big] \leq f(x)$, the expected value of the maximum of these two quantities is not necessarily $\leq f(x)$. Therefore the expected value of $\underline{lb}^N$ is not necessarily $\leq f^*$, i.e., we cannot claim that $\underline{lb}^N$ is a lower on average bound on f^*. However, the following result shows that $\underline{lb}^N$ is "statistically close" to a valid lower bound on f^*, provided that N and S are large. Proof of the following theorem is given in Section 3.6.

Theorem 3.3.2 *Suppose that assumption* (A.3.1) *holds and let the constant stepsizes* (3.2.7) *be used. Then*

$$\sqrt{\mathbb{E}\left\{ \left(\left[\underline{lb}^N - f^* \right]_+ \right)^2 \right\}} \leq \sqrt{2Q^2 + 32\Omega^2_{\omega,X} M^2_*} \left[\frac{1}{\sqrt{N}} + \frac{1}{\sqrt{S}} \right]. \tag{3.3.15}$$

Moreover, under assumption (A.3.2)*, we have that for all* $\Lambda \geq 0$*:*

$$\text{Prob}\left\{ \underline{lb}^N - f^* > [Q + 4\Omega_{\omega,X} M_*] \left[\frac{1}{\sqrt{N}} + \frac{1}{\sqrt{S}} \right] \right\} \leq 4 \exp\{-\Lambda^2/3\}. \tag{3.3.16}$$

3.4 Applications in Asset Allocation

In this section, we discuss an application of the mirror descent SA method to solving asset allocation problems based on the Expected Utility (EU) and the Conditional Value-at-Risk (CVaR) models.

3.4.1 Minimizing the expected utility

We consider the following stochastic utility model:

$$\min_{x\in X}\left\{f(x) := \mathbb{E}\left[\phi\big(\textstyle\sum_{i=1}^n (a_i+\xi_i)x_i\big)\right]\right\}. \tag{3.4.1}$$

Here $X := X' \cap X''$, where

$$X' := \{x \in \mathbb{R}^n : \textstyle\sum_{i=1}^n x_i \le r\} \;\text{ and }\; X'' := \{x \in \mathbb{R}^n : l_i \le x_i \le u_i,\ i = 1,...,n\},$$

$r > 0$, a_i and $0 \le l_i < u_i$, $i = 1,...,n$, are given numbers, $\xi_i \sim \mathcal{N}(0,1)$ are independent random variables having standard normal distribution and $\phi(\cdot)$ is a piecewise linear convex function given by

$$\phi(t) := \max\{c_1 + b_1 t, ..., c_m + b_m t\}, \tag{3.4.2}$$

where c_j and b_j, $j = 1,...,m$, are certain constants. Note that by varying parameters r and l_i, u_i we can change the feasible region from a simplex to a box, or the intersection of a simplex with a box. Note that since the set X is compact and $f(x)$ is continuous, the set of optimal solution of (3.4.1) is nonempty, provided that X is nonempty. A simpler version of problem (3.4.1), in which X is assumed to be a standard simplex, has been considered in Chapter 2.

For solving this problem, we consider two variants of the Mirror Descent SA algorithm: *Non-Euclidean SA (N-SA)* and *Euclidean SA (E-SA)*, which differ from each other in how the norm $\|\cdot\|$ and the distance generating function $\omega(\cdot)$ are chosen.

3.4.1.1 Non-Euclidean SA

In N-SA for solving the EU model, the entropy distance generating function

$$\omega(x) := \sum_{i=1}^n \frac{x_i}{r}\ln\frac{x_i}{r}, \tag{3.4.3}$$

coupled with the $\|\cdot\|_1$ norm is employed. Note that here $X^o = \{x \in X : x > 0\}$ and for $n \ge 3$,

$$D^2_{\omega,X} = \max_{x\in X}\omega(x) - \min_{x\in X}\omega(x) \le \max_{x\in X'}\omega(x) - \min_{x\in X'}\omega(x) \le \ln n.$$

Also observe that for any $x \in X'$, $x > 0$, and $h \in \mathbb{R}^n$,

$$\begin{aligned}\left(\sum_{i=1}^n |h_i|\right)^2 &= \left(\sum_{i=1}^n x_i^{1/2}|h_i|x_i^{-1/2}\right)^2 \le \left(\sum_{i=1}^n x_i\right)\left(\sum_{i=1}^n h_i^2 x_i^{-1}\right) \\ &\le r\left(\sum_{i=1}^n h_i^2 x_i^{-1}\right) = r^2 h^T \nabla^2 \omega(x) h,\end{aligned}$$

where the first inequality follows by Cauchy inequality. Therefore the modulus of ω, with respect to the $\|\cdot\|_1$ norm, satisfies $\alpha \ge r^{-2}$. Note that here $D_{\omega,X}$ can be overestimated while α being underestimated since $X' \subseteq X$, therefore, the stepsizes computed according to (3.2.7) in view of these estimates may not be optimal. Of course, the quantity $D_{\omega,X}$ can be estimated more accurately, for example, by computing $\min_{x\in X} \omega(x)$ explicitly. We will also discuss a few different ways to fine-tune the stepsizes in Section 3.5.

For the entropy distance generating function (3.4.3), the prox-mapping $P_v(z)$ (defined in (2.2.28)) is r times the optimal solution to the optimization problem

$$\begin{array}{ll}\min_x & \sum_{i=1}^n \left(s_i x_i + x_i \ln x_i\right), \\ \text{s.t.} & \sum_{i=1}^n x_i \le 1, \\ & \tilde{l}_i \le x_i \le \tilde{u}_i,\ i = 1, ..., n,\end{array} \tag{3.4.4}$$

where $s_i = rz_i - \ln(v_i/r) - 1$, $\tilde{l}_i = l_i/r$, $\tilde{u}_i = u_i/r$.

In some cases problem (3.4.4) has an explicit solution, e.g., if $l_i = 0$ and $u_i \ge r$, $i = 1, ..., n$ (in that case the constraints $z_i \le u_i$ are redundant). In general, we can solve (3.4.4) as follows. Let $\lambda \ge 0$ denote the Lagrange multiplier associated with the constraint $\sum_{i=1}^n x_i \le 1$ and consider the corresponding Lagrangian relaxation of (3.4.4):

$$\begin{array}{ll}\min_x & \sum_{i=1}^n \left(s_i x_i + x_i \ln x_i\right) + \lambda\left(\sum_{i=1}^n x_i\right), \\ \text{s.t.} & \tilde{l}_i \le x_i \le \tilde{u}_i,\ i = 1, ..., n.\end{array} \tag{3.4.5}$$

This is a separable problem. Since $s_i x_i + x_i \ln x_i + \lambda x_i$ is monotonically decreasing for x_i less than $\exp[-(s_i+1+\lambda)]$ and is monotonically increasing after, we have that the i-th coordinate $\bar{x}_i(\lambda)$ of the optimal solution of (3.4.5) is given by the projection of $\exp[-(s_i + 1 + \lambda)]$ onto the interval $[\tilde{l}_i, \tilde{u}_i]$. Then, to solve problem (3.4.4) is equivalent to find $\lambda \ge 0$ such that

$$\sum_{i=1}^n \bar{x}_i(\lambda) = 1, \quad \text{if } \lambda > 0, \tag{3.4.6}$$

$$\sum_{i=1}^n \bar{x}_i(\lambda) \le 1, \quad \text{if } \lambda = 0. \tag{3.4.7}$$

While inequality (3.4.7) can be easily checked, the root-finding problem (3.4.6) is usually solved to certain precision by using bisection, and each bisection step requires $\mathcal{O}(n)$ operations.

3.4.1.2 Euclidean SA

In the E-SA approach to order to solve the EU model, the Euclidean distance generating function $\omega(x) := \frac{1}{2}x^T x$, coupled with the $\|\cdot\|_2$ norm is employed. Clearly here $X^o = X$ and $\alpha = 1$. We have

$$D^2_{\omega,X} = \max_{x\in X}\omega(x) - \min_{x\in X}\omega(x) \leq \tfrac{1}{2}\left(\min\{r^2, \|u\|_2^2\} - \|l\|_2^2\right).$$

Moreover a procedure similar to the one given in Subsection 3.4.1.1 can be developed for computing the prox mapping $P_x(y)$, which is given here by the metric projection $\Pi_X(x-y)$.

As it was noted in Example 2.2.1, if X is a standard simplex, N-SA can be potentially $\mathcal{O}(\sqrt{n/\log n})$ times faster than E-SA. The same conclusion seems to be applicable to our current situation, although certain caution should be taken since the error estimate (3.2.7) now also depends on l, u and r.

3.4.2 Minimizing the Conditional Value-at-Risk

The idea of minimizing CVaR in place of Value-at-Risk (VaR) is due to Rockafellar and Uryasev [62]. Recall that VaR and CVaR of a random variable Z are defined as

$$\mathrm{VaR}_{1-\beta}(Z) := \inf\left\{\tau : \mathrm{Prob}(Z \leq \tau) \geq 1-\beta\right\}, \tag{3.4.8}$$

$$\mathrm{CVaR}_{1-\beta}(Z) := \inf_{\tau\in\mathbb{R}}\left\{\tau + \beta^{-1}\mathbb{E}[Z-\tau]_+\right\}. \tag{3.4.9}$$

Note that

$$\mathrm{VaR}_{1-\beta}(Z) \in \mathrm{Argmin}_{\tau\in\mathbb{R}}\left\{\tau + \beta^{-1}\mathbb{E}[Z-\tau]_+\right\}, \tag{3.4.10}$$

and hence $\mathrm{VaR}_{1-\beta}(Z) \leq \mathrm{CVaR}_{1-\beta}(Z)$.

The problem of interest in this subsection is:

$$\min_{y\in Y}\mathrm{CVaR}_{1-\beta}\big(-\xi^T y\big), \tag{3.4.11}$$

where ξ is a random vector with mean $\bar{\xi} := \mathbb{E}[\xi]$ and covariance matrix Σ, and

$$Y := \left\{ y \in \mathbb{R}^n_+ : \textstyle\sum_{i=1}^n y_i = 1,\ \bar{\xi}^T y \geq R \right\}.$$

We assume that Y is nonempty and, moreover, contains a positive point. For simplicity we assume in the remaining part of the chapter that ξ has continuous distribution, and hence $\xi^T y$ has continuous distribution for any $y \in Y$.

In view of the definition of CVaR in (3.4.9), our problem becomes:

$$\min_{x \in X} f(x) := \tau + \frac{1}{\beta} \mathbb{E}\left\{ [-\xi^T y - \tau]_+ \right\}, \tag{3.4.12}$$

where $X := Y \times \mathbb{R}$ and $x := (y, \tau)$. Apparently, there exists one difficulty to apply the mirror descent SA for solving the above problem — in (3.4.12), the variables are y *and* τ, so that the feasible domain $Y \times \mathbb{R}$ of the problem is unbounded, while our mirror descent SA requires a bounded feasible domain. However, we will alleviate this problem by showing that the variable τ can actually be restricted into a bounded interval and thus the mirror descent SA method can be applied.

Noting that $\mathrm{VaR}_{1-\beta}(Z) \in \mathrm{Argmin}_{\tau \in \mathbb{R}} \left[\tau + \mathbb{E}\{[Z - \tau]_+\}\right]$, all we need is to find an interval which covers all points $\mathrm{VaR}_{1-\beta}(-\xi^T y)$, $y \in Y$. Now, let Z be a random variable with finite mean μ and variance σ^2. By Cantelli's inequality (also called one-sided Tschebyshev inequality) we have

$$\mathrm{Prob}\{Z \geq t) \leq \frac{\sigma^2}{(t-\mu)^2 + \sigma^2}.$$

Assuming that Z has continuous distribution, we obtain

$$\beta = \mathrm{Prob}\{Z \geq \mathrm{VaR}_{1-\beta}(Z)\} \leq \frac{\sigma^2}{[\mathrm{VaR}_{1-\beta}(Z) - \mu]^2 + \sigma^2},$$

which implies that

$$\mathrm{VaR}_{1-\beta}(Z) \leq \mu + \sqrt{\frac{1-\beta}{\beta}}. \tag{3.4.13}$$

Similarly, if $\mathrm{VaR}_{1-\beta}(Z) \leq \mu$, then

$$1 - \beta = \mathrm{Prob}\{-Z \geq -\mathrm{VaR}_{1-\beta}(Z)\} \leq \frac{\sigma^2}{[-\mathrm{VaR}_{1-\beta}(Z) + \mu]^2 + \sigma^2},$$

which implies that

$$\mathrm{VaR}_{1-\beta}(Z) \geq \mu - \sqrt{\frac{\beta}{1-\beta}}\sigma. \tag{3.4.14}$$

Combing inequality (3.4.13) and (3.4.14) we obtain

$$\mathrm{VaR}_{1-\beta}(Z) \in \left[\mu - \sqrt{\tfrac{\beta}{1-\beta}}\sigma,\ \mu + \sqrt{\tfrac{1-\beta}{\beta}}\sigma\right]. \tag{3.4.15}$$

Note also that if Z is symmetric and $\beta \leq 0.5$, then the previous inclusion can be strengthened to

$$\mathrm{VaR}_{1-\beta}(Z) \in \left[\mu,\ \mu + \sqrt{\tfrac{1-\beta}{\beta}}\sigma\right]. \tag{3.4.16}$$

From this analysis it clearly follows that we lose nothing when restricting τ in (3.4.12) to vary in the segment

$$\tau \in \mathcal{T} := \left[\underline{\mu} - \sqrt{\tfrac{\beta}{1-\beta}}\overline{\sigma},\ \overline{\mu} + \sqrt{\tfrac{1-\beta}{\beta}}\overline{\sigma}\right], \tag{3.4.17}$$

where

$$\underline{\mu} := \min_{y\in Y}\{-\bar{\xi}^T y\},\ \overline{\mu} := \max_{y\in Y}\{-\bar{\xi}^T y\},\ \overline{\sigma}^2 := \max_{y\in Y} y^T \Sigma\, y. \tag{3.4.18}$$

In the case when ξ is symmetric and $\beta \leq 0.5$, this segment can be can be further reduced to:

$$\tau \in \mathcal{T}' := \left[\underline{\mu},\ \overline{\mu} + \sqrt{\tfrac{1-\beta}{\beta}}\overline{\sigma}\right]. \tag{3.4.19}$$

Note that the quantities $\underline{\mu}$ and $\overline{\mu}$ can be easily computed by solving the corresponding linear programs in (3.4.18). Moreover, although $\overline{\sigma}$ can be difficult to compute exactly, it can be replaced with its easily computable upper bound $\max_i \Sigma_{ii}$.

It is worth noting that an alternative upper bound for τ can be obtained in some cases: given an initial point $y_0 \in Y$, we have

$$\mathrm{CVaR}_{1-\beta}(-\xi^T y_0) \geq \mathrm{CVaR}_{1-\beta}(-\xi^T y^*) \geq \mathrm{VaR}_{1-\beta}(-\xi^T y^*),$$

where y^* is an optimal solution of problem (3.4.11). Therefore, in view of (3.4.10), if the value of $\mathrm{CVaR}_{1-\beta}(-\xi^T y_0)$ can be computed or estimated (e.g., by Monte-Carlo simulation), we can restrict the variable τ in (3.4.12) to be $\leq \mathrm{CVaR}_{1-\beta}(-\xi^T y_0)$.

To apply the mirror descent SA to problem (3.4.11), we set $X = Y \times \mathcal{T}$ and define the stochastic oracle by setting

$$F(x,\xi) \equiv F(y,\tau,\xi) = \tau + \tfrac{1}{\beta}\max[-\xi^T y - \tau, 0],$$

$$\mathsf{G}(x,\xi) \equiv [\mathsf{G}_y(y,\tau,\xi); \mathsf{G}_\tau(y,\tau,\xi)] = \begin{cases} [-\beta^{-1}\xi; 1-\beta^{-1}] & , -\xi^T y - \tau > 0 \\ [0;...;0;1] & , \text{otherwise} \end{cases}$$

Further, we choose D_y and D_τ from the relations

$$D_y \geq \max\left[1/2, \sqrt{\max_{y\in Y}\sum_i y_i\ln y_i - \min_{y\in Y}\sum_i y_i\ln y_i}\right],\ D_\tau = \frac{1}{2}\left[\max_{\tau\in\mathcal{T}}\tau^2 - \min_{\tau\in\mathcal{T}}\tau^2\right]$$

(we always can take $D_y = \max[1/2, \sqrt{\ln(n)}]$) and equip X and its embedding space $\mathbb{R}^n_y \times \mathbb{R}_\tau \supset X$ with the distance generating function and the norm as follows:

$$\|(y,\tau)\| = \sqrt{\|y\|_1^2/(2D_y^2) + \tau^2/(2D_\tau^2)} \qquad \left[\Leftrightarrow \|(z,\rho)\|_* = \sqrt{2D_y^2\|z\|_\infty^2 + 2D_\tau^2\rho^2}\right]$$
$$\omega(x) \equiv \omega(y,\tau) = \tfrac{1}{2D_y^2}\textstyle\sum_{i=1}^n y_i\ln y_i + \tfrac{1}{2D_\tau^2}\tau^2$$

Note that with this setup, $X^o = \{(y,\tau)\in X : y > 0\}$. Besides this, it is easily seen that $\sum_{i=1}^n y_i\ln y_i$, restricted on Y, is strongly convex, modulus 1, w.r.t. $\|\cdot\|_1$, whence ω is strongly convex, modulus $\alpha = 1$, on X. An immediate computation shows that $D_{\omega,X} = 1$, and therefore $\Omega_{\omega,X} = \sqrt{2}$. Finally, we set

$$M_* = \sqrt{2D_y^2\beta^{-2}\mathbb{E}\left[\|\xi\|_\infty^2\right] + 2D_\tau^2\max[1,(\beta^{-1}-1)^2]}. \tag{3.4.20}$$

It is easy to verify that with this M_*, our stochastic oracle satisfies (3.2.2).

Indeed, from the formula for $\mathsf{G}(x,\xi)$ we have

$$\mathbb{E}\left[\|\mathsf{G}(x,\xi)\|_*^2\right] = \mathbb{E}\left[2D_y^2\beta^{-2}\|\xi\|_\infty^2 + 2D_\tau^2\max[1,\beta^{-1}-1]^2\right] = M_*^2,$$

as required in (3.2.2). Further, for $x\in X$ we have $|F(x,\xi)-\tau-\beta^{-1}\max[-\tau,0]| \leq \beta^{-1}|\xi^T y| \leq \beta^{-1}|\xi\|_\infty$, whence

$$\mathbb{E}[(F(x,\xi)-f(x))^2] = \mathbb{E}[(F(x,\xi)-\mathbb{E}[F(x,\xi)])^2] \leq \mathbb{E}[(F(x,\xi)-\tau-\beta^{-1}\max[-\tau,0])^2]$$
$$\leq \beta^{-2}\mathbb{E}[\|\xi\|_\infty^2] \leq \Omega_{\omega,X}^2 M_*^2,$$

where the concluding inequality is due to $D_y \geq 1/2$ and $\Omega_{\omega,X} = \sqrt{2}$. We see that assumption (A.3.1) is satisfied with M_* given by (3.4.20) and $Q = \Omega_{\omega,X}M_* = \sqrt{2}M_*$.

3.5 Numerical results

3.5.1 More implementation details

- **Fine-tuning the stepsizes:** In Section 2, we specified the constant stepsize policy for the mirror descent SA method up to the "scaling parameter" θ. In our experiments,

Table 9: The test instances for EU model

NAME	r	u	NAME	r	u
EU-1	100	0.05	EU-6	1	$+\infty$
EU-2	100	0.20	EU-7	10	$+\infty$
EU-3	100	0.40	EU-8	100	$+\infty$
EU-4	100	10.00	EU-9	1,000	$+\infty$
EU-5	100	50.00	EU-10	5,000	$+\infty$

this parameter was chosen by as a result of pilot runs of the mirror descent SA algorithm with several trial values of θ and a very small sample size N (namely, $N = 100$). From these values of θ, we chose for the actual run the one resulting in the smallest online upper bound $\overline{f}^N$ on the optimal value.

- **Bundle-level method for solving SAA problem:** We also compare the results obtained by the mirror descent SA method with those obtained by the SAA coupled with the bundle-level method (SAA-BL) [34]. Note that the SAA problem is to be solved by the Bundle-level method; in our experiments, the SAA problems were solved within relative accuracy 1.e-4 through 1.e-6, depending on the instance.

3.5.2 Computational results for the EU model

In our experiments, we fix $l_i = 0$ and $u_i = u$ for all $1 \leq i \leq n$. The experiments were conducted for ten random instances which have the same dimension $n = 1000$ but differ in the parameters u and r, and the function $\phi(\cdot)$. A detailed description of these instances is shown in Table 9. Observe that for the first five instances, we fix $r = 100$ but change u from 0.05 to 50. For the next five instances, we assume $u = +\infty$ but change r from 1.0 to $5,000.0$.

Here we highlight some interesting findings based on our computational results. More numerical results can be found in Appendix B.

- **The effect of stepsize factor θ:** Our first test is to verify that we can fine-tune the stepsizes by using small pilot run. In this test, we chose between eight different stepsize factors, namely, 0.005, 0.01, 0.05, 0.1, 0.5, 1.0, 5, 10 for both N-SA and E-SA. First, we used short pilot runs ($M = 100$) to select the "most promising" value of the stepsize factor θ, see the beginning of section 3.5.1. Second, we directly tested

Table 10: The stepsize factors

NAME	Best θ	Inferred θ	NAME	Best θ	Inferred θ
EU-1	0.005	0.005	EU-6	5.000	5.000
EU-2	1.000	5.000	EU-7	10.000	10.000
EU-3	1.000	5.000	EU-8	10.000	10.000
EU-4	5.000	10.000	EU-9	10.000	10.000
EU-5	5.000	5.000	EU-10	5.000	5.000

Table 11: Changing u

NAME	N-SA ($\hat{f}(x_*)/f(x_*)$)	E-SA ($\hat{f}(x_*)/f(x_*)$)	SAA ($\hat{f}(x_*)/f(x_*)$)	opt
EU-1	-19.3558/-19.3279	-19.1311/-19.0953	-19.2700/-19.2435	-19.3307
EU-2	-61.4004/-61.3332	-61.7670/-61.6979	-62.8794/-62.7962	-62.9636
EU-3	-81.5215/-81.4339	-80.5735/-80.4873	-83.0845/-82.9732	-83.2145
EU-4	-100.1597/-99.6734	-92.1313/-92.0161	-99.3096/-99.0400	-102.6819
EU-5	-99.5680/-99.2872	-91.2051/-91.0923	-98.5458/-98.2697	-101.9112

which one of the outlined eight values of θ results in the highest quality solution for the sample size $N = 2,000$. The results are presented in the columns "Inferred θ," resp., "Best θ," of Table 10. As we can see from this table, the inferred θ's are very close to the best ones for all test instances and the same conclusion also holds for the E-SA.

- **The effect of changing u:** In Table 11, we report the objective values of EU-1 – EU-5 evaluated at the solutions obtained by N-SA, E-SA and SAA when the sample size is $N = 2,000$. In this table, $\hat{f}(x_*)$ denotes the estimated objective value (using sample size $K = 10,000$) at the obtained solution x_*. Due to the assumption that ξ is normally distributed, the actual objective value $f(x_*)$ can be also computed. Moreover, a close examination reveals that the optimal value of problem (3.4.1) can be computed efficiently (Chapter 2); it is shown in the last column of Table 11.

  One interesting observation from this table is that the performance of N-SA is slightly better than that of E-SA even for EU-1 whose feasible region is actually a box instead of a simplex, so that there are no theoretical reasons to prefer N-SA to E-SA.

  One other observation from this table is that the solution quality of N-SA significantly outperforms that of E-SA for the two largest values of u. The possible explanation is that the feasible region appears more like a simplex when u is big.

Table 12: Changing r

NAME	N-SA($\hat{f}(x_*)/f(x_*)$)	E-SA($\hat{f}(x_*)/f(x_*)$)	SAA($\hat{f}(x_*)/f(x_*)$)	opt
EU-6	-6.2999/-6.2864	-6.2211/-6.2186	-6.3073/ -6.3027	-6.3460
EU-7	-16.2514/-16.2294	-15.3818/-15.3717	-16.1474/-16.1226	-16.4738
EU-8	-97.3613/-97.1581	-89.2032/-89.0897	-96.5163/-96.2450	-99.8824
EU-9	-9.540e+2/-9.513e+2	-8.686e+2/-8.675e+2	-9.419e+2/-9.393e+2	-9.757e+2
EU-10	-4.730e+3/-4.717e+3	-4.322e+3/-4.316e+3	-4.689e+3/-4.675e+3	-4.857e+3

- **The effect of changing r:** Table 12 shows the objective values of EU-6 EU-10 evaluated at the solutions obtained by N-SA, E-SA and SAA when the sample size is $N = 2,000$. In this table, $\hat{f}(x_*)$ and $f(x_*)$, respectively, denote the estimated objective value (using sample size $K = 10,000$) and the actual objective value at the obtained solution x_*, and "opt" denotes the optimal value of problem (3.4.1).

  Recall that the feasible regions for these five instances are simplexes. So, as expected, N-SA consistently outperforms E-SA for all these instances. It is interesting to observe that the objective values achieved by N-SA can be smaller than those by SAA for large r. Note that the SAA problem has been solved to a relatively high accuracy by using the Bundle-level method. For example, for EU-10, the SAA problem was solved to accuracy 0.7e-005.

- **The lower bounds:** Table 13 shows the lower bounds on the objective values of EU-1 – EU-10 obtained by N-SA, E-SA and SAA when the sample size is $N = 2,000$. In Table 13, the lower bounds $\underline{f}^N$ and $\underline{lb}^N$ are the online and offline bounds defined in Section 3. The lower bound for SAA is defined as the optimal value of the corresponding SAA problem. As we can see from this table, the lower bound for SAA is always better than the online lower bound $\underline{f}^N$ for the SA methods (as it should be in the case of constant stepsizes, see Remark 3.3.1). However, the offline lower bound $\underline{lb}^N$ can be close or even better than the lower bound obtained from SAA.

  Moreover, we estimate the variability of the online lower bounds in the way discussed in section 3.3.1 and the results are reported in Table 14. In particular, the second and third column of this table show the mean and the standard deviation obtained from $M = 10$ independent replications of N-SA, each of which has the same sample

Table 13: Lower bounds on optimal values and true optimal values

	N-SA		E-SA		SAA	
NAME	$\underline{f}^N$	$\underline{lb}^N$	$\underline{f}^N$	$\underline{lb}^N$	$\hat{f}^N_{\text{SAA}}$	opt
EU-1	-19.4063	-19.2994	-19.4063	-19.2994	-19.4063	-19.3307
EU-2	-62.9984	-62.8754	-62.9984	-62.8758	-62.9984	-62.9367
EU-3	-83.0039	-82.9730	-83.0039	-82.9730	-83.0039	-83.2145
EU-4	-107.5820	-104.5046	-107.2058	-104.4072	-105.0890	-102.6819
EU-5	-107.5745	-104.0644	-108.4063	-104.3577	-104.3214	-101.9112
EU-6	-6.6111	-6.5288	-6.9171	-6.5849	-6.3658	-6.3460
EU-7	-17.0130	-16.7060	-17.1800	-16.7605	-16.7027	-16.4378
EU-8	-106.7958	-102.6311	-106.5921	-102.2588	-102.2914	-99.8824
EU-9	-1029.0530	-997.7217	-1042.7008	-1000.6626	-999.9114	-9.757e2
EU-10	-5192.0409	-4967.9144	-5192.0409	-4981.8515	-4978.2333	-4.857e3

Table 14: Variability of the lower bounds for N-SA

	Ind. repl.		Dep. repl.		Whole Traj.
NAME	mean	deviation	mean	deviation	$\underline{f}^{NM}$
EU-1	-19.5681	0.0857	-19.5387	0.0842	-19.3461
EU-2	-63.3898	0.2372	-63.3786	0.3502	-63.0444
EU-3	-83.6973	0.3121	-83.7339	0.3098	-83.2649
EU-4	-112.2483	1.5616	-114.1652	2.7470	-105.5543
EU-5	-113.7526	1.5951	-115.3103	2.8232	-104.4565
EU-6	-6.7812	0.0265	-6.8969	0.1374	-6.4522
EU-7	-17.7911	0.2326	-18.3881	0.5519	-16.8022
EU-8	-113.5263	2.1348	-117.4176	4.6588	-102.3509
EU-9	-1091.2836	20.2804	-1140.23774	61.1979	-1006.1846
EU-10	-5466.1266	124.5894	-5553.80221	144.6298	-5048.5643

size $N = 1000$. The third and fourth column read the mean and standard deviation computed for the lower bounds associated with the $M = 10$ consecutive partitions of the trajectory of N-SA with a sample size $NM = 10,000$. The last column reports the online lower bound $\underline{f}^{NM}$. The results indicate that the bounds obtained from independent replications have relatively smaller variability in general.

- **The computation times:** For all instances, the computation times of generating a solution for SA were $10 - 30$ times smaller than that for SAA.

- **The standard deviations:** For the generated solution x_*, we evaluate the corresponding objective value $f(x_*)$ by generating an independent large sample $\xi_1, ..., \xi_K$, of size $K = 10,000$, and computing the estimate $\hat{f}(x_*) = K^{-1} \sum_{j=1}^{K} F(x_*, \xi_j)$ of $f(x_*)$. We also computed an estimate of the standard deviation of $F(x_*, \xi)$:

$$\hat{\sigma} = \sqrt{\sum_{j=1}^{K} \left(F(x_*, \xi_j) - \hat{f}(x_*) \right)^2 / (K - 1)}.$$

Table 15: Standard deviations

	N-SA		SAA	
NAME	$\hat{f}(x_*)$	$\hat{\sigma}$	$\hat{f}(x_*)$	$\hat{\sigma}$
EU-1	-19.3558	3.1487	-19.2700	3.0019
EU-2	-61.4004	8.4178	-62.8749	8.9099
EU-3	-81.5215	11.7493	-83.0845	12.6015
EU-4	-100.1597	38.6309	-99.3096	61.1053
EU-5	-99.5680	35.1278	-98.5458	60.8440
EU-6	-6.2999	0.6798	-6.3073	0.7030
EU-7	-16.2514	3.5233	-16.1474	5.7941
EU-8	-97.3613	36.3939	-96.5163	61.0974
EU-9	-953.9882	383.8223	-941.9854	611.0414
EU-10	-4729.8534	1746.7144	-4688.9239	3053.7409

Note that the standard deviation of $\hat{f}(x_*)$, as an estimate of $f(x_*)$, is estimated by $\frac{\hat{\sigma}}{\sqrt{K}}$. Table 15 compares the deviations for N-SA and SAA computed in the above way. From this table, we observe that for instances with either a larger u or larger r, the values of $\hat{\sigma}$ corresponding to the solutions obtained by N-SA can be significantly smaller (up to 1/2) than those by SAA.

3.5.3 Computational results for the CVaR model

In this subsection, we report some numerical results on applying the mirror descent SA method for the CVaR model (3.4.11). Here the return ξ is assumed to be a normal random vector. In that case random variable $-\xi^T y$ has normal distribution with mean $-\bar{\xi}^T y$ and variance $y^T \Sigma\, y$, and

$$\mathrm{CVaR}_{1-\beta}\{-\xi^T y\} = -\bar{\xi}^T y + \rho\sqrt{y^T \Sigma\, y}, \tag{3.5.1}$$

where $\rho := \frac{\exp(-z_\beta^2/2)}{\beta\sqrt{2\pi}}$ and $z_\beta := \Phi^{-1}(1-\beta)$ with $\Phi(\cdot)$ being the cdf of the standard normal distribution. Consequently the optimal solution for (3.4.11) can be easily obtained by replacing the objective function of (3.4.11) with the right hand side of (3.5.1). Clearly, the resulting problem can be reformulated as a conic-quadratic programming program, and its optimal value thus gives us a benchmark to compare the SA and SAA methods.

Two instances for the CVaR model are considered in our experiments. The first instance (CVaR-1) is obtained from [76]. This instance consists of the 95 stocks from S&P100 (excluding SBC, ATI, GS, LU, and VIA-B) and the mean $\bar{\xi}$ and covariance Σ_ξ were estimated using historical monthly prices from 1996 to 2002. The second one (CVaR-2), which contains

Table 16: The test instances for CVaR model

NAME	n	β	R	opt
CVaR-1	95	0.05	1.0000	-0.9841
CVaR-2	1,000	0.10	1.0500	1.5272

Table 17: Comparing SA and SAA for the CVaR model

		SA					SAA			
NAME	N	$\hat{f}(x_*)$	$f(x_*)$	$\underline{f}_N$	$\underline{lb}^N$	time	$\hat{f}(x_*)$	$f(x_*)$	$\hat{f}^N_{\rm SAA}$	time
CVaR-1	1000	-0.9807	-0.9823	-1.0695	-1.0136	0	-0.9823	-0.9828	-0.9854	15
	2000	-0.9824	-0.9832	-1.0518	-0.9877	1	-0.9832	-0.9835	-0.9852	27
CVaR-2	1000	1.6048	1.5896	1.1301	1.4590	20	1.6396	1.5795	1.3023	928
	2000	1.5766	1.5633	1.3696	1.4973	39	1.5835	1.5557	1.4780	2784

$1,000$ assets, was randomly generated by setting the random return $\xi = \bar{\xi} + Q\zeta$, where ζ is the standard Gaussian vector, $\bar{\xi}_i$ is uniformly distributed in $[0.9, 1.2]$, and Q_{ij} is uniformly distributed in $[0, 0.1]$ for $1 \leq i, j \leq 1,000$. The reliability level β, the bound for expected return R, and the optimal value for these two instances are reported in Table 16.

The computational results for the CVaR model are reported in Table 17, where $\hat{f}(x_*)$ and $f(x_*)$, respectively, denote the estimated objective value (using sample size $K = 10,000$) and the actual objective value at the obtained solution x_*. We conclude from the results in Table 17 that the mirror descent SA method can generate good solutions much faster than SAA. The lower bounds derived for the SA method are also comparable to those for the SAA method.

3.6 Proof of the main results

Our goal in this section is to prove Theorems 3.3.1 and 3.3.2.

We will need the following result (cf., [43, Lemma 6.1]).

Lemma 3.6.1 *Let $\zeta_t \in \mathbb{R}^n$, $v_1 \in X^o$ and $v_{t+1} = P_{v_t}(\zeta_t)$, $t = 1, ..., N$. Then*

$$\sum_{t=1}^{N} \zeta_t^T (v_t - u) \leq V(v_1, u) + (2\alpha)^{-1} \sum_{t=1}^{N} \|\zeta_t\|_*^2, \ \ \forall u \in X. \tag{3.6.1}$$

We denote here $\delta_t := F(x_t, \xi_t) - f(x_t)$ and $\Delta_t := \mathsf{G}(x_t, \xi_t) - \mathsf{g}(x_t)$. Since x_t is a function of ξ^{t-1} and ξ_t is independent of ξ_t, we have that the conditional expectations

$$\mathbb{E}_{|t-1}[\delta_t] = 0 \ \text{ and } \ \mathbb{E}_{|t-1}[\Delta_t] = 0, \tag{3.6.2}$$

and hence the unconditional expectations $\mathbb{E}[\delta_t] = 0$ and $\mathbb{E}[\Delta_t] = 0$ as well.

Part (i) of Theorem 3.3.1: *Proof.* 1.1^0 If in Lemma 3.6.1 we take $v_1 := x_1$ and $\zeta_t := \gamma_t \mathsf{G}(x_t, \xi_t)$, then the corresponding iterates v_t coincide with x_t. Therefore, we have by (3.6.1) and since $V(x_1, u) \leq D^2_{\omega,X}$ that

$$\sum_{t=1}^{N} \gamma_t (x_t - u)^T \mathsf{G}(x_t, \xi_t) \leq D^2_{\omega,X} + (2\alpha)^{-1} \sum_{t=1}^{N} \gamma_t^2 \|\mathsf{G}(x_t, \xi_t)\|_*^2, \ \ \forall u \in X. \tag{3.6.3}$$

It follows that for any $u \in X$:

$$\begin{aligned} &\sum_{t=1}^{N} \nu_t \big[-f(x_t) + (x_t - u)^T \mathsf{g}(x_t) \big] + \sum_{t=1}^{N} \nu_t f(x_t) \\ &\leq \frac{D^2_{\omega,X} + (2\alpha)^{-1} \sum_{t=1}^{N} \gamma_t^2 \|\mathsf{G}(x_t, \xi_t)\|_*^2}{\sum_{t=1}^{N} \gamma_t} + \sum_{t=1}^{N} \nu_t \Delta_t^T (x_t - u). \end{aligned}$$

Since

$$f^{*N} - f_*^N = \sum_{t=1}^{N} \nu_t f(x_t) + \max_{u \in X} \sum_{t=1}^{N} \nu_t \big[-f(x_t) + (x_t - u)^T \mathsf{g}(x_t) \big],$$

it follows that

$$f^{*N} - f_*^N \leq \frac{D^2_{\omega,X} + (2\alpha)^{-1} \sum_{t=1}^{N} \gamma_t^2 \|\mathsf{G}(x_t, \xi_t)\|_*^2}{\sum_{t=1}^{N} \gamma_t} + \max_{u \in X} \sum_{t=1}^{N} \nu_t \Delta_t^T (x_t - u). \tag{3.6.4}$$

Let us estimate the second term in the right hand side of (3.6.4). Let

$$u_1 = v_1 = x_1; u_{t+1} = P_{u_t}(-\gamma_t \Delta_t),\ t = 1, 2, ..., N; v_{t+1} = P_{v_t}(\gamma_t \Delta_t),\ t = 1, 2, ...N. \tag{3.6.5}$$

Observe that Δ_t is a deterministic function of ξ^t, whence u_t and v_t are deterministic functions of ξ^{t-1}. By using Lemma 3.6.1 we obtain

$$\sum_{t=1}^{N} \gamma_t \Delta_t^T (v_t - u) \leq D^2_{\omega,X} + (2\alpha)^{-1} \sum_{t=1}^{N} \gamma_t^2 \|\Delta_t\|_*^2, \ \ \forall u \in X. \tag{3.6.6}$$

Moreover,

$$\Delta_t^T (v_t - u) = \Delta_t^T (x_t - u) + \Delta_t^T (v_t - x_t),$$

and hence it follows by (3.6.6) that

$$\max_{u \in X} \sum_{t=1}^{N} \nu_t \Delta_t^T (x_t - u) \leq \sum_{t=1}^{N} \nu_t \Delta_t^T (x_t - v_t) + \frac{D^2_{\omega,X} + (2\alpha)^{-1} \sum_{t=1}^{N} \gamma_t^2 \|\Delta_t\|_*^2}{\sum_{t=1}^{N} \gamma_t}. \tag{3.6.7}$$

Observe that by similar reasoning applied to $-\Delta_t$ in the role of Δ_t we get

$$\max_{u\in X}\left[-\sum_{t=1}^{N}\nu_t\Delta_t^T(x_t-u)\right]\leq\left[-\sum_{t=1}^{N}\nu_t\Delta_t^T(x_t-u_t)\right]+\frac{D^2_{\omega,X}+(2\alpha)^{-1}\sum_{t=1}^{N}\gamma_t^2\|\Delta_t\|_*^2}{\sum_{t=1}^{N}\gamma_t}. \tag{3.6.8}$$

Moreover, $\mathbb{E}_{|t-1}[\Delta_t]=0$ and u_t,v_t and x_t are functions of ξ^{t-1}, while $\mathbb{E}_{|t-1}\Delta_t=0$ and hence

$$\mathbb{E}_{|t-1}\big[(x_t-v_t)^T\Delta_t\big]=\mathbb{E}_{|t-1}\big[(x_t-u_t)^T\Delta_t\big]=0. \tag{3.6.9}$$

We also have that $\mathbb{E}_{|t-1}\left[\|\Delta_t\|_*^2\right]\leq 4M_*^2$, and hence in view of condition (3.2.2) it follows from (3.6.7) and (3.6.9) that

$$\mathbb{E}\left[\max_{u\in X}\sum_{t=1}^{N}\nu_t\Delta_t^T(x_t-u)\right]\leq\frac{D^2_{\omega,X}+2\alpha^{-1}M_*^2\sum_{t=1}^{N}\gamma_t^2}{\sum_{t=1}^{N}\gamma_t}. \tag{3.6.10}$$

Therefore, by taking expectation of both sides of (3.6.4) and using (3.2.2) together with (3.6.10) we obtain the estimate (3.3.5).

1.2^0 In order to prove (3.3.6) let us observe that $\overline{f}^N-f^{*N}=\sum_{t=1}^{N}\nu_t\delta_t$, and that for $1\leq s<t\leq N$,

$$\mathbb{E}[\delta_s\delta_t]=\mathbb{E}\{\mathbb{E}_{|t-1}[\delta_s\delta_t]\}=\mathbb{E}\{\delta_s\mathbb{E}_{|t-1}[\delta_t]\}=0.$$

Therefore

$$\mathbb{E}\left[\left(\overline{f}^N-f^{*N}\right)^2\right]=\mathbb{E}\left[\left(\sum_{t=1}^{N}\nu_t\delta_t\right)^2\right]=\sum_{t=1}^{N}\nu_t^2\mathbb{E}\left[\delta_t^2\right]=\sum_{t=1}^{N}\nu_t^2\mathbb{E}\left\{\mathbb{E}_{|t-1}\left[\delta_t^2\right]\right\}.$$

Moreover, by condition (3.2.1) of assumption (A.3.1) we have that $\mathbb{E}_{|t-1}\left[\delta_t^2\right]\leq Q^2$, and hence

$$\mathbb{E}\left[\left(\overline{f}^N-f^{*N}\right)^2\right]\leq Q^2\sum_{t=1}^{N}\nu_t^2. \tag{3.6.11}$$

Since $\sqrt{\mathbb{E}[Y^2]}\geq\mathbb{E}|Y|$ for any random variable Y, inequality (3.3.6) follows from (3.6.11).

1.3^0 Let us now look at (3.3.7). We have

$$\begin{array}{rcl}\left|\underline{f}^N-f_*^N\right| & = & \left|\min_{x\in X}\hat{f}^N(x)-\min_{x\in X}f^N(x)\right|\leq\max_{x\in X}\left|\hat{f}^N(x)-f^N(x)\right| \\ & \leq & \left|\sum_{t=1}^{N}\nu_t\delta_t\right|+\max_{x\in X}\left|\sum_{t=1}^{N}\nu_t\Delta_t^T(x_t-x)\right|.\end{array} \tag{3.6.12}$$

We already showed above (see (3.6.11)) that

$$\mathbb{E}\left[\left|\sum_{t=1}^{N}\nu_t\delta_t\right|\right]\leq Q\sqrt{\sum_{t=1}^{N}\nu_t^2}. \tag{3.6.13}$$

Invoking (3.6.7), (3.6.8), we get

$$\max_{x\in X}\left|\sum_{t=1}^N \nu_t\Delta_t^T(x_t-x)\right| \le \left|\sum_{t=1}^N \nu_t\Delta_t^T(x_t-v_t)\right|+\left|\sum_{t=1}^N \nu_t\Delta_t^T(x_t-u_t)\right| + \frac{D^2_{\omega,X}+(2\alpha)^{-1}\sum_{t=1}^N\gamma_t^2\|\Delta_t\|_*^2}{\sum_{t=1}^N\gamma_t}. \tag{3.6.14}$$

Moreover, for $1\le s<t\le N$ we have that $\mathbb{E}\left[\left(\Delta_s^T(x_s-v_s)\right)\left(\Delta_t^T(x_t-v_t)\right)\right]=0$, and hence

$$\begin{aligned}\mathbb{E}\left[\left|\sum_{t=1}^N \nu_t\Delta_t^T(x_t-v_t)\right|^2\right] &= \sum_{t=1}^N \nu_t^2\mathbb{E}\left[\left|\Delta_t^T(x_t-v_t)\right|^2\right] \le 4M_*^2\sum_{t=1}^N\nu_t^2\mathbb{E}\left[\|x_t-v_t\|^2\right]\\ &\le 32M_*^2\alpha^{-1}D^2_{\omega,X}\sum_{t=1}^N\nu_t^2,\end{aligned}$$

where the last inequality follows by (2.2.31). It follows that

$$\mathbb{E}\left[\left|\sum_{t=1}^N \nu_t\Delta_t^T(x_t-v_t)\right|\right] \le 4\sqrt{2\alpha^{-1}}D_{\omega,X}\sqrt{\sum_{t=1}^N\nu_t^2}.$$

By similar reasons,

$$\mathbb{E}\left[\left|\sum_{t=1}^N \nu_t\Delta_t^T(x_t-u_t)\right|\right] \le 4\sqrt{2\alpha^{-1}}D_{\omega,X}\sqrt{\sum_{t=1}^N\nu_t^2}.$$

These two inequalities combine with (3.6.13), (3.6.14) and (3.6.12) to imply (3.3.7). This completes the proof of part (i) of Theorem 3.3.1.

■

Preparing to prove part (ii) of Theorem 3.3.1: To prove part (ii) of Theorem 3.3.1 we need the following known result; we give its proof for the sake of completeness.

Lemma 3.6.2 *Let $\xi_1,\xi_2,...$ be a sequence of iid random variables, $\sigma_t>0$, μ_t, $t=1,...$, be a sequence of deterministic numbers and $\phi_t=\phi_t(\xi^t)$ be deterministic (measurable) functions of $\xi^t=(\xi_1,...,\xi_t)$ such that either*

Case A: $\mathbb{E}_{|t-1}[\phi_t]=0$ w.p.1 and $\mathbb{E}_{|t-1}\left[\exp\{\phi_t^2/\sigma_t^2\}\right]\le\exp\{1\}$ w.p.1 for all t, or

Case B: $\mathbb{E}_{|t-1}\left[\exp\{|\phi_t|/\sigma_t\}\right]\le\exp\{1\}$ for all t.

Then for any $\Lambda\ge 0$ we have the following. In the case of A:

$$\mathrm{Prob}\left\{\sum_{t=1}^N\phi_t>\Lambda\sqrt{\sum_{t=1}^N\sigma_t^2}\right\}\le\exp\{-\Lambda^2/3\}. \tag{3.6.15}$$

In the case of B, setting $\sigma^N:=(\sigma_1,...,\sigma_N)$:

$$\begin{aligned}&\mathrm{Prob}\left\{\sum_{t=1}^N\phi_t>\|\sigma^N\|_1+\Lambda\|\sigma^N\|_2\right\}\le\exp\{-\Lambda^2/12\}+\exp\left\{-\tfrac{3\|\sigma^N\|_2}{4\|\sigma^N\|_\infty}\Lambda\right\}\\ &\le\exp\{-\Lambda^2/12\}+\exp\{-3\Lambda/4\}.\end{aligned} \tag{3.6.16}$$

Proof. Let us set $\bar{\phi}_t := \phi_t/\sigma_t$.

Case A: By the respective assumptions about ϕ_t we have that $\mathbb{E}_{|t-1}[\bar{\phi}_t] = 0$ and $\mathbb{E}_{|t-1}\left[\exp\{\bar{\phi}_t^2\}\right] \leq \exp\{1\}$ w.p.1. By Jensen inequality it follows that for any $a \in [0,1]$:

$$\mathbb{E}_{|t-1}\left[\exp\{a\bar{\phi}_t^2\}\right] = \mathbb{E}_{|t-1}\left[(\exp\{\bar{\phi}_t^2\})^a\right] \leq \left(\mathbb{E}_{|t-1}\left[\exp\{\bar{\phi}_t^2\}\right]\right)^a \leq \exp\{a\}.$$

We also have that $\exp\{x\} \leq x + \exp\{9x^2/16\}$ for all x (this can be verified by direct calculations), and hence

$$\mathbb{E}_{|t-1}\left[\exp\{\lambda\bar{\phi}_t\}\right] \leq \mathbb{E}_{|t-1}\left[\exp\{(9\lambda^2/16)\bar{\phi}_t^2\}\right] \leq \exp\{9\lambda^2/16\}, \ \ \forall \lambda \in [0, 4/3]. \tag{3.6.17}$$

Besides this, we have $\lambda x \leq \frac{3}{8}\lambda^2 + \frac{2}{3}x^2$ for any λ and x, and hence

$$\mathbb{E}_{|t-1}\left[\exp\{\lambda\bar{\phi}_t\}\right] \leq \exp\{3\lambda^2/8\}\mathbb{E}_{|t-1}\left[\exp\{2\bar{\phi}_t^2/3\}\right] \leq \exp\{2/3 + 3\lambda^2/8\}.$$

Combining the latter inequality with (3.6.17), we get

$$\mathbb{E}_{|t-1}\left[\exp\{\lambda\bar{\phi}_t\}\right] \leq \exp\{3\lambda^2/4\}, \ \ \forall \lambda \geq 0.$$

Going back to ϕ_t, the above inequality reads

$$\mathbb{E}_{|t-1}\left[\exp\{\kappa\phi_t\}\right] \leq \exp\{3\kappa^2\sigma_t^2/4\}, \ \ \forall \kappa \geq 0. \tag{3.6.18}$$

Now, since ϕ_τ is a deterministic function of ξ^τ and using (3.6.18), we obtain for any $\kappa \geq 0$:

$$\begin{array}{l}\mathbb{E}\left[\exp\left\{\kappa\sum_{\tau=1}^t \phi_\tau\right\}\right] = \mathbb{E}\left[\exp\left\{\kappa\sum_{\tau=1}^{t-1}\phi_\tau\right\}\mathbb{E}_{|t-1}\exp\{\kappa\phi_t\}\right] \\ \leq \exp\{3\kappa^2\sigma_t^2/4\}\mathbb{E}\left[\exp\{\kappa\sum_{\tau=1}^{t-1}\phi_\tau\}\right],\end{array}$$

and hence

$$\mathbb{E}\left[\exp\left\{\kappa\sum_{t=1}^N \phi_t\right\}\right] \leq \exp\left\{3\kappa^2\sum_{t=1}^N \sigma_t^2/4\right\}. \tag{3.6.19}$$

By Markov inequality, we have for $\kappa > 0$ and $\Lambda \geq 0$:

$$\begin{array}{rcl}\mathrm{Prob}\left\{\sum_{t=1}^N \phi_t > \Lambda\sqrt{\sum_{t=1}^N\sigma_t^2}\right\} & = & \mathrm{Prob}\left\{\exp\left[\kappa\sum_{t=1}^N\phi_t\right] > \exp\left[\kappa\Lambda\sqrt{\sum_{t=1}^N\sigma_t^2}\right]\right\} \\ & \leq & \exp\left[-\kappa\Lambda\sqrt{\sum_{t=1}^N\sigma_t^2}\right]\mathbb{E}\left\{\exp\left[\kappa\sum_{t=1}^N\phi_t\right]\right\}.\end{array}$$

Together with (3.6.19) this implies for $\Lambda \geq 0$:

$$\mathrm{Prob}\left\{\sum_{t=1}^N\phi_t > \Lambda\sqrt{\sum_{t=1}^N\sigma_t^2}\right\} \leq \inf_{\kappa>0}\exp\left\{\tfrac{3}{4}\kappa^2\sum_{t=1}^N\sigma_t^2 - \kappa\Lambda\sqrt{\sum_{t=1}^N\sigma_t^2}\right\} = \exp\left\{-\Lambda^2/3\right\}.$$

Case B: Observe first that if η is a random variable such that $\mathbb{E}[\exp\{|\eta|\}] \le \exp\{1\}$, then

$$0 \le t \le \frac{1}{2} \Rightarrow \mathbb{E}[\exp\{t\eta\}] \le \exp\{t + 3t^2\}. \tag{3.6.20}$$

Indeed, let $f(t) = \mathbb{E}[\exp\{t\eta\}]$. Then $f(0) = 1$, $f'(0) = \mathbb{E}[\eta] \le \ln(\mathbb{E}[\exp\{\eta\}]) \le 1$. Besides this, when $0 \le t \le 1/2$, invoking the Cauchy and the Hölder inequalities we have

$$\begin{aligned} f''(t) &= \mathbb{E}[\exp\{t\eta\}\eta^2] \le [\mathbb{E}[\exp\{2t|\eta|\}]]^{1/2} \left[\mathbb{E}[\eta^4]\right]^{1/2} \le [\mathbb{E}[\exp\{|\eta|\}]]^t \left[\mathbb{E}[\eta^4]\right]^{1/2} \\ &\le \exp\{1/2\} \left[\mathbb{E}[\eta^4]\right]^{1/2}. \end{aligned}$$

It is immediately seen that $s^4 \le (4/\varepsilon)^4 \exp\{|s|\}$ for all s, whence $\left[\mathbb{E}[\eta^4]\right]^{1/2} \le (4/\varepsilon)^2 \varepsilon^{1/2}$ due to $\mathbb{E}[\exp\{|\eta|\}] \le \varepsilon$. Thus, $f''(t) \le 16/\varepsilon$ when $0 \le t \le 1/2$, and thus $f(t) \le 1 + t + (8/\varepsilon)t^2 \le \exp\{t + (8/\varepsilon)t^2\} \le \exp\{t + 3t^2\}$, and (3.6.20) follows.

Let $\gamma \ge 0$ be such that $\gamma\sigma_t \le 1/2$, $1 \le t \le N$. When $t \le N$, we have

$$\begin{aligned} &\mathbb{E}\left[\exp\{\textstyle\sum_{\tau=1}^t \gamma\phi_\tau\}\right] = \mathbb{E}\left[\exp\{\sum_{\tau=1}^t \gamma\sigma_\tau\bar{\phi}_\tau\}\right] = \mathbb{E}\left[\exp\{\sum_{\tau=1}^{t-1} \gamma\sigma_\tau\bar{\phi}_\tau\}\mathbb{E}_{|t-1}\left[\exp\{\gamma\sigma_t\bar{\phi}_t\right]\right] \\ &\le \exp\{\gamma\sigma_t + 3\gamma^2\sigma_t^2\}\mathbb{E}\left[\exp\{\textstyle\sum_{\tau=1}^{t-1} \gamma\sigma_\tau\bar{\phi}_\tau\}\right], \end{aligned}$$

where the concluding inequality is given by (3.6.20) (note that we are in the case when $\mathbb{E}_{|t-1}[\exp\{|\bar{\phi}_t|\}] \le \exp\{1\}$ w.p.1). From the resulting recurrence we get

$$0 \le \gamma\|\sigma^N\|_\infty \le 1/2 \Rightarrow \mathbb{E}\left[\exp\{\sum_{t=1}^N \gamma\phi_t\}\right] \le \exp\{\gamma\|\sigma^N\|_1 + 3\gamma^2\|\sigma^N\|_2^2\}.$$

whence for every $\Lambda \ge 0$, denoting $\beta_s = \|\sigma^N\|_s$,

$$0 \le \gamma\beta_\infty \le 1/2 \Rightarrow p := \mathrm{Prob}\{\sum_{t=1}^N \phi_t > \beta_1 + \Lambda\beta_2\} \le \exp\{3\gamma^2\beta_2^2 - \gamma\Lambda\beta_2\}. \tag{3.6.21}$$

When $\Lambda \le \bar{\Lambda} := 3\beta_2/\beta_\infty$, $\gamma = \Lambda/(6\beta_2)$ satisfies the premise in (3.6.21), and this implication then says that $p \le \exp\{-\Lambda^2/12\}$. When $\Lambda > \bar{\Lambda}$, we can use the implication with $\gamma = (2\beta_\infty)^{-1}$, thus getting

$$p \le \exp\{\frac{\beta_2}{2\beta_\infty}\left[\frac{3\beta_2}{2\beta_\infty} - \Lambda\right]\} \le \exp\{-\frac{3\beta_2}{4\beta_\infty}\Lambda\}.$$

Thus (3.6.16) is proved. ∎

Part (ii) of Theorem 3.3.1: *Proof.* Recall that in part (ii) of Theorem 3.3.1 assumption (A.3.1) is strengthened to assumption (A.3.2). Then, in addition to (3.6.2), we have that

$$\mathbb{E}_{|t-1}\left[\exp\{\delta_t^2/Q^2\}\right] \le \exp\{1\} \ \text{ and } \ \mathbb{E}_{|t-1}\left[\exp\{\|\Delta_t\|_*^2/(2M_*)^2\}\right] \le \exp\{1\}. \tag{3.6.22}$$

Let us also make the following simple observation. If Y_1 and Y_2 are random variables and a_1, a_2, a are numbers such that $a_1 + a_2 \ge a$, then the event $\{Y_1 + Y_2 > a\}$ is included in the union of the events $\{Y_1 > a_1\}$ and $\{Y_2 > a_2\}$, and hence $\mathrm{Prob}\{Y_1 + Y_2 > a\} \le \mathrm{Prob}\{Y_1 > a_1\} + \mathrm{Prob}\{Y_2 > a_2\}$.

2.1^0 Recall that $\overline{f}^N - f^{*N} = \sum_{t=1}^N \nu_t\delta_t$, and hence it follows by case A of Lemma 3.6.2 together with the first equality in (3.6.2) and (3.6.22) that for any $\Lambda \ge 0$:

$$\mathrm{Prob}\left\{\overline{f}^N - f^{*N} > \Lambda Q\sqrt{\textstyle\sum_{t=1}^N \nu_t^2}\right\} \le \exp\{-\Lambda^2/3\}. \tag{3.6.23}$$

In the same way, by considering $-\delta_t$ instead of δ_t, we have that

$$\mathrm{Prob}\left\{f^{*N} - \overline{f}^N > \Lambda Q\sqrt{\textstyle\sum_{t=1}^N \nu_t^2}\right\} \le \exp\{-\Lambda^2/3\}, \tag{3.6.24}$$

The assertion (3.3.10) follows from (3.6.23) and (3.6.24).

2.2^0 Now by (3.6.12) and (3.6.14) we have

$$\begin{aligned}\left|\underline{f}^N - f_*^N\right| \le \left|\textstyle\sum_{t=1}^N \nu_t\delta_t\right| + \left|\textstyle\sum_{t=1}^N \nu_t\Delta_t^T(x_t - v_t)\right| + \left|\textstyle\sum_{t=1}^N \nu_t\Delta_t^T(x_t - u_t)\right| \\ + \frac{D_{\omega,X}^2 + (2\alpha)^{-1}\sum_{t=1}^N \gamma_t^2\|\Delta_t\|_*^2}{\sum_{t=1}^N \gamma_t}.\end{aligned} \tag{3.6.25}$$

As it was shown above (see (3.6.23), (3.6.24)):

$$\mathrm{Prob}\left\{\left|\textstyle\sum_{t=1}^N \nu_t\delta_t\right| > \Lambda Q\sqrt{\textstyle\sum_{t=1}^N \nu_t^2}\right\} \le 2\exp\{-\Lambda^2/3\}. \tag{3.6.26}$$

Moreover, by (2.2.31) we have that $\|x_t - v_t\| \le \|x_t - x_1\| + \|v_t - x_1\| \le 2\sqrt{2\alpha^{-1}}D_{\omega,X}$, and hence

$$\mathbb{E}_{|t-1}\left[\exp\{|\Delta_t^T(x_t - v_t)|^2/(4\sqrt{2\alpha^{-1}}D_{\omega,X}M_*)^2\}\right] \le \exp\{1\}.$$

It follows by case A of Lemma 3.6.2 that

$$\mathrm{Prob}\left\{\left|\textstyle\sum_{t=1}^N \nu_t\Delta_t^T(x_t - v_t)\right| > 4\Lambda\sqrt{2\alpha^{-1}}D_{\omega,X}M_*\sqrt{\textstyle\sum_{t=1}^N \nu_t^2}\right\} \le 2\exp\{-\Lambda^2/3\}. \tag{3.6.27}$$

and similarly

$$\text{Prob}\left\{\left|\sum_{t=1}^N \nu_t\Delta_t^T(x_t-u_t)\right| > 4\Lambda\sqrt{2\alpha^{-1}}D_{\omega,X}M_*\sqrt{\sum_{t=1}^N\nu_t^2}\right\} \le 2\exp\{-\Lambda^2/3\}. \tag{3.6.28}$$

Furthermore, invoking (3.6.22), the random variables $\phi_t = (2\alpha)^{-1}\gamma_t^2\|\Delta_t\|_*^2(\sum_{t=1}^N\gamma_t)^{-1}$ satisfy the premise of case B in Lemma 3.6.2 with $\sigma_t = 2\alpha^{-1}M_*^2\gamma_t^2(\sum_{t=1}^N\gamma_t)^{-1}$. Invoking case B of Lemma, we get

$$\begin{array}{l}\text{Prob}\left\{\frac{(2\alpha)^{-1}\sum_{t=1}^N\gamma_t^2\|\Delta_t\|_*^2}{\sum_{t=1}^N\gamma_t} > \frac{2\alpha^{-1}M_*^2\sum_{t=1}^N\gamma_t^2}{\sum_{t=1}^N\gamma_t} + \Lambda\frac{2\alpha^{-1}M_*^2\sqrt{\sum_{t=1}^N\gamma_t^4}}{\sum_{t=1}^N\gamma_t}\right\}\\ \quad\le \exp\{-\Lambda^2/12\}+\exp\{-\Gamma_N\Lambda\},\quad \Gamma_N = \frac{3\|(\gamma_1^2,...,\gamma_N^2)\|_2}{4\|(\gamma_1^2,...,\gamma_N^2)\|_\infty}\end{array} \tag{3.6.29}$$

Combining this bound with (3.6.27), (3.6.28) and taking into account (3.6.25), we arrive at (3.3.11).

2.3^0 It remains to prove (3.3.9). To this end note by (3.6.4) and (3.6.7) we have

$$f^{*N}-f_*^N \le \frac{2D_{\omega,X}^2+(2\alpha)^{-1}\sum_{t=1}^N\gamma_t^2(\|\mathsf{G}(x_t,\xi_t)\|_*^2+\|\Delta_t\|_*^2)}{\sum_{t=1}^N\gamma_t} + \sum_{t=1}^N\nu_t\Delta_t^T(x_t-v_t), \tag{3.6.30}$$

Completely similar to (3.6.29), we have

$$\begin{array}{l}\text{Prob}\left\{\frac{(2\alpha)^{-1}\sum_{t=1}^N\gamma_t^2\|\mathsf{G}(x_t,\xi_t)t\|_*^2}{\sum_{t=1}^N\gamma_t} > \frac{(2\alpha)^{-1}M_*^2\sum_{t=1}^N\gamma_t^2}{\sum_{t=1}^N\gamma_t} + \Lambda\frac{(2\alpha)^{-1}M_*^2\sqrt{\sum_{t=1}^N\gamma_t^4}}{\sum_{t=1}^N\gamma_t}\right\}\\ \le \exp\{-\Lambda^2/12\}+\exp\{-\Gamma_N\Lambda\}\end{array} \tag{3.6.31}$$

This bound combines with (3.6.29) and (3.6.27) to imply (3.3.9). ∎

Theorem 3.3.2: *Proof.* Let $x_1,...,x_N$ be the trajectory of mirror descent SA, and let $x_{N+t} := \tilde{x}_N$, $t = 1,...,S$. Then we can write

$$\hat{\ell}_S(x;\tilde{x}_N) = \frac{1}{S}\sum_{t=N+1}^{N+S}\left[F(x_t,\xi_t)+\mathsf{G}(x_t,\xi_t)^T(x-x_t)\right].$$

Let x_* be an optimal solution to (1.2.1), and let us set $\eta_t := \Delta_t^T(x_*-x_t)$, $t = 1,...,N+S$. By (2.2.31) we have $\|x_t-x_*\| \le 2\Omega_{\omega,X}$, and since x_t is a deterministic function of ξ^{t-1}, $1 \le t \le N+S$, and the oracle is unbiased, under assumption (A.3.1) we have for $1 \le t \le N+S$,

$$\begin{array}{l}\mathbb{E}_{|t-1}[\delta_t] = 0, \mathbb{E}_{|t-1}\left[\delta_t^2\right] \le Q^2,\\ \mathbb{E}_{|t-1}[\eta_t] = 0, \mathbb{E}_{|t-1}\left[\eta_t^2\right] \le 4\Omega_{\omega,X}^2\mathbb{E}_{|t-1}\left[\|\Delta_t\|_*^2\right] \le 16\Omega_{\omega,X}^2M_*^2.\end{array} \tag{3.6.32}$$

Consequently

$$\hat{f}^N(x_*) = \underbrace{\frac{1}{N}\sum_{t=1}^{N}[f(x_t) + \mathsf{g}(x_t)^T(x_* - x_t)]}_{\leq f(x_*)=\mathrm{Opt}} + \underbrace{\frac{1}{N}\sum_{t=1}^{N}[\delta_t + \eta_t]}_{\zeta_1},$$
$$\hat{\ell}_S(x_*;\tilde{x}_N) = \underbrace{\frac{1}{S}\sum_{t=N+1}^{N+S}[f(x_t) + \mathsf{g}(x_t)^T(x_* - x_t)]}_{\leq f(x_*)=\mathrm{Opt}} + \underbrace{\frac{1}{S}\sum_{t=N+1}^{N+S}[\delta_t + \eta_t]}_{\zeta_2}.$$

It follows that

$$lb^N - f^* \leq \max\left\{\hat{f}^N(x_*), \hat{\ell}_S(x_*;\tilde{x}_N)\right\} - f^* \leq \max\{\zeta_1,\zeta_2\} \leq |\zeta_1| + |\zeta_2|. \tag{3.6.33}$$

From (3.6.32) it follows that

$$\mathbb{E}[\zeta_1^2] \leq N^{-1}\left(2\mathbb{E}\left[\delta_t^2\right] + 2\mathbb{E}\left[\eta_t^2\right]\right) \leq \left(2Q^2 + 32\Omega_{\omega,X}^2 M_*^2\right)N^{-1},$$
$$\mathbb{E}[\zeta_2^2] \leq S^{-1}\left(2\mathbb{E}\left[\delta_t^2\right] + 2\mathbb{E}\left[\eta_t^2\right]\right) \leq \left(2Q^2 + 32\Omega_{\omega,X}^2 M_*^2\right)S^{-1},$$

which combines with (3.6.33) to imply (3.3.15).

Under assumption (A.3.2), along with (3.6.32) we also have that

$$\mathbb{E}_{|t-1}\left[\exp\{\delta_t^2/Q^2\}\right] \leq \exp\{1\},\ \ \mathbb{E}_{|t-1}\left[\exp\{\eta_t^2/(4\Omega_{\omega,X}M_*)^2\}\right] \leq \exp\{1\},$$

and hence

$$\mathbb{E}_{|t-1}\left[\delta_t + \eta_t\right] = 0,\ \ \mathbb{E}_{|t-1}\left[\exp\{[\delta_t + \eta_t]^2/(Q + 4\Omega_{\omega,X}M_*)^2\}\right] \leq \exp\{1\}.$$

Invoking case A of Lemma 3.6.2, we conclude that for all $\Lambda \geq 0$:

$$\mathrm{Prob}\left\{|\zeta_1| > \Lambda[Q + 4\Omega_{\omega,X}M_*]N^{-1/2}\right\} \leq 2\exp\{-\Lambda^2/3\},$$
$$\mathrm{Prob}\left\{|\zeta_2| > [Q + 4\Omega_{\omega,X}M_*]S^{-1/2}\right\} \leq 2\exp\{-\Lambda^2/3\},$$

which combines with (3.6.33) to imply (3.3.16). ∎

3.7 Conclusions of this chapter

In this chapter, we develop accuracy estimates for stochastic programming problems by employing SA type algorithms. We show that while running a mirror descent SA procedure one can compute, with a small additional effort, lower and upper statistical bounds for

the optimal objective value. We demonstrate that for a certain class of convex stochastic programs these bounds are comparable in quality with similar bounds computed by the SAA method, while their computational cost is considerably smaller. Moreover, Extensive numerical experiments were conducted to understand the performance of the mirror descent SA algorithm for solving stochastic programing problems with a feasible set more complicated than a standard simplex.

CHAPTER IV

EFFICIENT METHODS FOR STOCHASTIC COMPOSITE OPTIMIZATION

4.1 Overview

The basic problem of interest in this chapter is the stochastic composite optimization (SCO) given by

$$f^* := \min_{x \in X} \{\Psi(x) := f(x) + h(x)\}, \tag{4.1.34}$$

where X is a convex compact set in Euclidean space $\mathcal{E}$ with inner product $\langle \cdot, \cdot \rangle$, $f : X \to \Re$ is a convex function with Lipschitz continuous gradient, that is,

$$\|\nabla f(x) - \nabla f(x')\|_* \leq L\|x - x'\|, \ \forall x, x' \in X, \tag{4.1.35}$$

($\|\cdot\|$ is a given norm in $\mathcal{E}$, $\|\cdot\|_*$ denotes its conjugate norm, see Subsection 4.1.1), and $h : X \to \Re$ is a convex Lipschitz continuous function such that

$$|h(x) - h(x')| \leq \mathcal{M}\|x - x'\|, \ \forall x, x' \in X. \tag{4.1.36}$$

We assume that problem (4.1.34) is to be solved by iterative algorithms which acquire the subgradients of Ψ via subsequent calls to a stochastic oracle ($\mathcal{SO}$). Specifically, at iteration t of the algorithm, $x_t \in X$ being the input, the $\mathcal{SO}$ outputs a vector $G(x_t, \xi_t)$, where $\{\xi_t\}_{t \geq 1}$ is a sequence of i.i.d. random variables which are also independent of search points x_t. The following assumptions are made for the Borel functions $G(x, \xi_t)$.

A.4.1: For any $x \in X$, we have

$$\text{a)} \quad \mathbb{E}[G(x, \xi_t)] \equiv g(x) \in \partial\Psi(x) \tag{4.1.37}$$

$$\text{b)} \quad \mathbb{E}\left[\|G(x, \xi_t) - g(x)\|_*^2\right] \leq \sigma^2, \tag{4.1.38}$$

where $\partial\Psi(x)$ denotes the subdifferential of Ψ at x (see Subsection 4.1.1).

Observe that problem (4.1.34) covers several important classes of convex programming

problems as certain special cases. For the sake of simplicity, let us consider the situation where the domain X is an Euclidean ball in the following discussion.

Case I: non-smooth convex optimization. Suppose that the smooth component f in Ψ does not exist, or equivalently $f(x) = 0$ for every $x \in X$, and that there is no noise in the $\mathcal{SO}$, i.e., $\sigma = 0$ in (4.1.38). Then, problem (4.1.34) becomes the generic non-smooth convex optimization problem that has been well-studied in the Literature. According to Nemirovski and Yudin [44], if the dimension n is sufficiently large, i.e., $n \geq \mathcal{O}(1)N$, then the rate of convergence for any iterative algorithms to solve nonsmooth convex optimization problems can not be better than $\Psi(\hat{x}_N) - \Psi^* \leq \mathcal{O}(1)(\mathcal{M}/\sqrt{N})$, where N is the number of iterations performed by the algorithm and $\hat{x}_N \in X$ denotes the solution generated by the algorithm after N steps. Moreover, the simple subgradient descent method can achieve, up to a constant factor, the above lower bound. Note that the subgradient descent method is closely related to the gradient projection method of Goldstein and Levitin, Polyak (see [9]). Nemirovski and Yudin [44] also developed the so-called mirror descent algorithm that can be advantageous over the subgradient descent method when X is not an Euclidean ball by using the prox-function (also called Bregman's distance, which was studied by Bregman [10] and many others, see for example, [1, 2, 3, 29, 72] and references therein).

Case II: smooth convex optimization. Suppose that the non-smooth component h in Ψ does not exist, or equivalently $h(x) = 0$ for every $x \in X$, and that there is no noise in the $\mathcal{SO}$, i.e., $\sigma = 0$ in (4.1.38). Then, problem (4.1.34) becomes the smooth convex optimization problem. In [44], Nemirovski and Yudin show that, if the dimension n is sufficiently large, i.e., $n \geq \mathcal{O}(1)N$, then the rate of convergence for any iterative algorithms to solve smooth convex optimization problems can not be better than $\mathcal{O}(1)(L/N^2)$. They also provide a nearly optimal method which can achieve, up to a logrithmic factor, the above lower bound on the rate of convergence. In a series of work ([47, 48]), Nesterov presented novel smooth convex optimization algorithms whose rate of convergence is bounded by $\mathcal{O}(1)(L/N^2)$. Clearly, Nesterov's methods are optimal, up to a constant factor, for smooth convex optimization when $n \geq \mathcal{O}(1)N$. Nesterov's methods were further studied in [49],

[1] and [50] using Bregman's distance and other variants of Nesterov's optimal method can also be found, for example, in [32] and [73].

Case III: Stochastic convex optimization. Suppose that the variance of the $\mathcal{SO}$ is positive, i.e., $\sigma > 0$. Then, problem (4.1.34) becomes the stochastic convex optimization problem. There exist two competitive computational approaches for solving stochastic convex optimization based on Monte Carlo sampling techniques, namely, the *Stochastic Approximation* (SA) and the *Sample Average Approximation* (SAA) methods. Both approaches, the SA and SAA methods, have a long history. The SAA approach was used by many authors in various contexts under different names. Its basic idea is rather simple: generate a (random) sample $\xi_1, ..., \xi_N$, of size N, and approximate the "true" problem (4.1.34) by the so-called sample average problem. Recent theoretical studies (cf., [30, 68, 69]) and numerical experiments (see, e.g., [36, 40, 74]) show that the SAA method coupled with a good (deterministic) algorithm could be reasonably efficient for solving certain classes of two stage stochastic programming problems. The classic SA method mimicks the gradient descent method and goes back to the pioneering paper by Robbins and Monro [61]. Since then stochastic approximation algorithms became widely used in stochastic optimization (see, e.g., [7, 17, 18, 57, 64, 31, 70] and references therein). An important improvement of the SA method was developed by Polyak [58] and Polyak and Juditsky [59], where longer stepsizes were suggested with consequent averaging of the obtained iterates. In these classical SA-type algorithms, it is assumed that the objective function is twice continuously differentiable and strongly convex. Current opinion is that the SAA method can efficiently use a specific (say linear) structure of the considered problem, while the SA approach is a crude subgradient method which often performs poorly in practice. Recently, Nemirovski et. al. [43] (See Chapter 2) considered non-smooth stochastic convex optimization (i.e., $f(x) = 0$ for every $x \in X$ in (4.1.34) and $\sigma > 0$). They demonstrated that a properly modified SA approach can be competitive and even significantly outperform the SAA method for a certain class of stochastic programming problems. The mirror-descent SA presented in [43] exhibits the following rate of convergence $\mathcal{O}(1)(\mathcal{M} + \sigma)/\sqrt{N}$, which is unimprovable

even when the dimension $n = 1$ (this differs from the two above-mentioned deterministic optimization cases where the lower bounds on the rate of convergence are valid only if n is sufficiently large [44]). Close techniques, based on subgradient averaging, have been proposed in Nesterov [51] and used in [24, 26] to solve certain non-smooth stochastic convex optimization problems. It should be noted that the study on general smooth stochastic convex optimization (i.e., $h(x) = 0$ for every $x \in X$ in (4.1.34) and $\sigma > 0$) without the strong convexity assumption on f seems quite limited in the literature.

Since SCO covers these subcases described above, it easily follows that the rate of convergence for any iterative algorithms to solve (4.1.34) can not be better than

$$\mathcal{O}(1)\left[\frac{L}{N^2} + \frac{\mathcal{M}+\sigma}{\sqrt{N}}\right], \tag{4.1.39}$$

where N is the number of iterations performed by the algorithm. This means that, for any algorithms solving problem (4.1.34), one can always point out a "bad" problem instance satisfying (4.1.35), (4.1.36), (4.1.37), and (4.1.38), such that the expected error of the solution generated at the N-step of the algorithm will be, up to a constant factor, greater than the lower bound stated above. However, to the best of our knowledge, none of the existing algorithms achieved this lower bound on the convergence rate. Since the objective function Ψ of (4.1.34) is a non-smooth function, we can directly apply the mirror-descent SA [43] to (4.1.34) and the resulting rate of convergence (cf. (4.2.46)) can no be better than

$$\mathcal{O}(1)\left[\frac{L+\mathcal{M}+\sigma}{\sqrt{N}}\right]. \tag{4.1.40}$$

The best known result so far is given by Juditsky et. al. [25] with the rate of convergence

$$\mathcal{O}(1)\left[\frac{L}{N} + \frac{\mathcal{M}+\sigma}{\sqrt{N}}\right] \tag{4.1.41}$$

by applying an extra-gradient-type algorithm to a variational inequality (v.i.) reformulation of (4.1.34). It is worth noting that this optimal rate of convergence has not been attained even for the deterministic case where $\sigma = 0$. Moreover, with only access to the $\mathcal{SO}$ of the composite function Ψ, it is absolutely unclear whether the lower bound (4.1.39) on the rate of convergence for solving (4.1.34) is achievable or not.

We would provide some motivation to explain why we shall care about the gap between the convergence rates (4.1.40), (4.1.41) and the lower bound (4.1.39). Imagine that we have an algorithm for solving (4.1.34) which achieves the lower bound (4.1.39) on the convergence rate. First of all, this imaginary algorithm will be a universally optimal method for non-smooth, smooth and stochastic convex optimization. Currently different classes of convex optimization problems are being handled by using different (sub)optimal methods. More specifically, mirror descent SA [43] and Nesterov's method [47, 48] are optimal for non-smooth (deterministic or stochastic) and smooth (deterministic) convex optimization respectively, and there does not exist an optimal algorithm for solving smooth stochastic convex optimization problems and general SCO problems. This is partly due to the difficulty that, although either smooth or nonsmooth optimization has been well-studied separately in the literature, a unified treatment for both of them seems highly non-trivial. Secondly, this imaginary optimal algorithm for SCO will allow us to have a very large Lipschitz constant L for problem (4.1.34) without affecting the rate of convergence. Let us have a closer examination of these convergence rates. The convergence rates in (4.1.40) and (4.1.41), will not be affected (up to a constant factor 2), if L is as big as $\mathcal{M}+\sigma$ and $(\mathcal{M}+\sigma)N^{\frac{1}{2}}$, respectively. It can also be easily seen from (4.1.39) that the convergence rate of the imaginary algorithm will not change (up to a constant factor 2) if $L \leq (\mathcal{M}+\sigma)N^{\frac{3}{2}}$. Clearly, the latter range of L that does not affect the rate of convergence for the imaginary algorithm is much bigger than those for the previous two methods and extends much faster as the number of iterations N grows. This fact often has great practical significance and one such example will be given in Section 4.3.2. Thirdly, we would mention some beauty of this algorithm: with only access to the $\mathcal{SO}$ of the composite function Ψ itself, the imaginary method can intelligently tell the difference between the smooth and non-smooth component, and treat them separately in an optimal manner. It is not only of pure mathematical beauty, but also physically meaningful, for example, when one does not have access to the components f and h of the objective function of (4.1.34).

Our contribution mainly consists of the following aspects. Firstly, with a novel analysis, it is demonstrated that a slightly modified mirror descent SA algorithm applied to

(4.1.34) also exhibits the best known so far rate of convergence guaranteed by a more involved stochastic mirror-prox algorithm [25]. Moreover, by properly modifying a variant of Nesterov's optimal method for smooth convex optimization, we propose an accelerated SA (AC-SA), which can achieve the theoretically optimal rate of convergence for solving this class of problems. Clearly, the accelerated SA algorithm is a universally optimal method for non-smooth, smooth and stochastic convex optimization. It should be stressed that Nesterov's optimal method and/or its variants were designed for solving deterministic smooth convex optimization problems. These algorithms, with very aggressive stepsizes employed, were believed to be too sophisticated to solve non-smooth and stochastic convex optimization problems. We, however, substantially extend the analysis of Nesterov's optimal method to non-smooth and stochastic convex optimization, and devise a novel (actually increasing) stepsize policy for solving these problems. Thirdly, we investigate this accelerated SA in more details, for example, derive the exponential bounds for the large deviations of the resulting solution inaccuracy from the expected one, provided the noise from the stochastic oracle is "light-tailed". Finally, the significant advantages of the accelerated SA over the existing algorithms are illustrated in the context of solving a class of stochastic programming problems whose feasible region is a simple compact convex set intersected with an affine manifold. More specifically, if the accelerated SA is applied to solve the quadratic penalization problem where the violation of the affine constraints is penalized, then, surprisingly, the size of the Lagrange multiplier associated with these affine constraints has, asymptotically, no affect on the convergence rate.

We should distinguish the results obtained in this chapter with some related but different development in the literature for solving problems given in the form of (4.1.34). Recently, Nesterov in a very relevant paper [50] presented a first-order method with convergence rate bounded by $\mathcal{O}(1/N)$ to solve convex optimization problems of the form (4.1.34), where the nonsmooth term h is given by

$$h(x) := \sup\{\langle \mathcal{B}y, x\rangle - \phi(y) : y \in Y\},$$

$Y \subseteq \Re^m$ is a compact convex set, $\phi : Y \to \Re$ is a continuous convex function and $\mathcal{B}$ is a

linear operator from $\Re^m$ to $\Re^n$. Nesterov's approach consists of approximating an arbitrary function h from the class by a sufficiently close smooth one with Lipschitz continuous gradient and applying the optimal smooth method in [47, 50] to the resulting problem with h replaced by its approximation function. In a subsequent paper, Nemirovski [42] proposed an extra-gradient type first-order method for solving a slightly more general class of optimization problems than the one considered by Nesterov [50] and also established an $\mathcal{O}(1/N)$ convergence rate for his method. These first-order methods were further studied in, for example, [1, 53, 39, 15, 56, 73, 41], and successfully used in sparse covariance selection, rank reduction in multivariate linear regression and compressed sensing etc. (see, for example, [16, 37, 38, 4]). Another line of investigation is also to consider problems given in the form of (4.1.34) where the non-smooth component h of Ψ in (4.1.34) is sufficiently simple, for example, $h(x) = \|x\|_1$, where $\|\cdot\|_1$ denotes the l_1 norm, so that the non-smooth component can be kept as a part of the prox-step (or projection in the Euclidean case) (Nesterov [52], Tseng [73], Lewis and Wright [35]). Consequently, the convergence rate for solving these problems is the same as that of smooth convex optimization, i.e., $\mathcal{O}(1/N^2)$. These developments clearly differ from ours in the following aspects: (i) those problems considered in [50, 42, 52] and subsequent papers are certain special cases of (4.1.34) in the sense that the nonsmooth term h can either be smoothed or kept in the projection. Therefore, it turned out that stronger convergence results can be obtained for those subcases. We, on the other hand, consider a general non-smooth term h in the objective function of (4.1.34); (ii) the algorithms developed in [50, 52] and related references need to access the smooth and non-smooth component of the composite function Ψ separately. In contrast, our method, in addition to using the structure of the problem, only need to access the composite function itself; iii) in [50, 42, 52] and other references mentioned above, only deterministic optimization problems have been considered. We, however, also focus on the situation where the subgradients of Ψ are contaminated by stochastic noise, i.e., $\sigma > 0$; iv) it should be noted that the development in [50, 52] can be easily incorporated into our method for certain cases where the nonsmooth component h in (4.1.34) consists of the aforementioned special structures.

The chapter is organized as follows. In Section 4.2, we present a slightly modified mirror descent SA algorithm applied to (4.1.34) and describe its convergence properties. Section 4.3 discusses the accelerated stochastic approximation method. More specifically, we present the AC-SA algorithm and its convergence properties in Subsection 4.3.1, and outline an application to demonstrate the advantages of this algorithm in Subsection 4.3.2. Section 4.4 is devoted to proving the main results of this chapter. Finally, some concluding remarks are made in Section 4.5.

4.1.1 Notation and terminology

- For a convex lower semicontinuous function $\phi : X \to \Re$, its subdifferential $\partial f(\cdot)$ is defined as follows: at a point x from the relative interior of X, $\partial\phi$ is comprised of all subgradients g of ϕ at x which are in the linear span of $X - X$. For a point $x \in X \backslash \operatorname{rint} X$, the set $\partial\phi(x)$ consists of all vectors g, if any, such that there exists $x_i \in \operatorname{rint} X$ and $g_i \in \partial\phi(x_i)$, $i = 1, 2, \cdots$, with $x = \lim\limits_{i\to\infty} x_i$, $g = \lim\limits_{i\to\infty} g_i$. Finally, $\partial\phi(x) = \emptyset$ for $x \notin X$. With this definition, it is well-known (see, for example, Ben-Tal and Nemirovksi [5]) that, if a convex function $\phi : X \to \Re$ is Lipschitz continuous, with constant $\mathcal{M}$, with respect to a norm $\|\cdot\|$, then the set $\partial\phi(x)$ is nonempty for any $x \in X$ and

$$g \in \partial\phi(x) \Rightarrow |\langle g, d\rangle| \leq \mathcal{M}\|d\|, \ \forall\, d \in \operatorname{lin}(X - X), \tag{4.1.42}$$

  in other words,

$$g \in \partial\phi(x) \Rightarrow \|g\|_* \leq \mathcal{M}, \tag{4.1.43}$$

  where $\|\cdot\|_*$ denotes the conjugate norm given by $\|g\|_* := \max_{\|d\|\leq 1}\langle g, d\rangle$.

- For the random process $\xi_1, \xi_2, ...$, we set $\xi_{[t]} := (\xi_1, ..., \xi_t)$, and denote by $\mathbb{E}_{|\xi_{[t]}}$ the conditional, $\xi_{[t]}$ being given, expectation.

4.2 Modified mirror-descent stochastic approximation

In this section, we present a slightly modified version of mirror-descent SA method in [43] and demonstrate that it can achieve the best known so far rate of convergence for solving problem (4.1.34).

The mirror descent SA algorithm, as applied to (4.1.34), works with the stochastic oracle of Ψ that satisfies Assumption A.4.1. In some cases, Assumption A.4.1 is augmented by the following "light-tail" assumption.

A.4.2: For any $x \in X$, we have

$$\mathbb{E}\left[\exp\{\|G(x,\xi_t) - g(x)\|_*^2/\sigma^2\}\right] \;\leq\; \exp\{1\}. \tag{4.2.44}$$

It can be easily seen that Assumption A.4.2 implies Assumption A.4.1(b), since by Jensen's inequality,

$$\exp\left(\mathbb{E}[\|G(x,\xi_t) - g(x)\|_*^2/\sigma^2]\right) \;\leq\; \mathbb{E}\left[\exp\{\|G(x,\xi_t) - g(x)\|_*^2/\sigma^2\}\right] \leq \exp\{1\}.$$

We first derive the rate of convergence for a direction application of the mirror descent SA in [43] (see also Section 2.2.3) to problem (4.1.34).

Let $g(x_t) = \mathbb{E}[G(x_t,\xi_t)] = \nabla f(x_t) + h'(x_t)$ for every $t \geq 1$, where $h'(x_t) \in \partial h(x_t)$. Then, in view of assumptions (4.1.35), (4.1.36), (4.1.37) and (4.1.38), and relation (4.1.43), we have

$$\begin{aligned}
\mathbb{E}[\|G(x_t,\xi_t)\|_*^2] &= \|\mathbb{E}[G(x_t,\xi_t)]\|_*^2 + \mathbb{E}[\|G(x_t,\xi_t) - \mathbb{E}[G(x_t,\xi_t)]\|_*^2] \\
&= \|g(x_t)\|_*^2 + \mathbb{E}[\|G(x_t,\xi_t) - g(x_t)\|_*^2] \leq \|\nabla f(x_t) + h'(x_t)\|_*^2 + \sigma^2 \\
&\leq 2\|\nabla f(x_t)\|_*^2 + 2\mathcal{M}^2 + \sigma^2 \\
&\leq 2\left(\|\nabla f(x_1) + \nabla f(x_t) - \nabla f(x_1)\|_*^2\right) + 2\mathcal{M}^2 + \sigma^2 \\
&\leq 4\|\nabla f(x_1)\|_*^2 + 4\|\nabla f(x_t) - \nabla f(x_1)\|_*^2 + 2\mathcal{M}^2 + \sigma^2 \\
&\leq 4\|\nabla f(x_1)\|_*^2 + 4L^2\|x_t - x_1\|^2 + 2\mathcal{M}^2 + \sigma^2 \\
&\leq 4\|\nabla f(x_1)\|_*^2 + 4L^2\Omega_{\omega,X}^2 + 2\mathcal{M}^2 + \sigma^2.
\end{aligned}$$

Now replacing the value of M_* in (2.2.46) or (2.2.55) with

$$M_* = \left(4\|\nabla f(x_1)\|_*^2 + 4L^2\Omega_{\omega,X}^2 + 2\mathcal{M}^2 + \sigma^2\right)^{\frac{1}{2}}, \tag{4.2.45}$$

we can easily see that the rate of convergence for a direct application of the mirror descent SA algorithm presented in Subsection 2.2.3 to problem (4.1.34) is bounded by

$$\mathcal{O}(1)\left[\frac{\Omega_{\omega,X}(\|\nabla f(x_1)\|_* + L\Omega_{\omega,X} + \mathcal{M} + \sigma)}{\sqrt{N}}\right]. \tag{4.2.46}$$

As discussed in Section 4.1, the above rate of convergence is significantly worse than the best known so far result obtained by using the stochastic mirror-prox algorithm [25]. We will show in this section that a slightly modified mirror descent SA algorithm stated below can achieve the best known so far rate of convergence for solving (4.1.34).

The modified mirror descent SA algorithm:

0) Let the initial point x_1 and the step-sizes $\{\gamma_t\}_{t\geq 1}$ be given. Set $t = 1$;

1) Call the $\mathcal{SO}$ for computing $G(x_t, \xi_t)$. Set

$$x_{t+1} := P_{x_t}\left(\gamma_t G(x_t, \xi_t)\right), \tag{4.2.47}$$

$$x_{t+1}^{av} = \left(\sum_{\tau=1}^{t} \gamma_\tau\right)^{-1} \sum_{\tau=1}^{t} \gamma_\tau x_{\tau+1}. \tag{4.2.48}$$

2) Set $t \leftarrow t + 1$ and go to Step 1.

end

We now make a few comments about the above algorithm. Firstly, without loss of generality, we will assume from now on that the initial point x_1 is given by the minimizer of ω over X (see Subsection 2.2.3). Secondly, observe that the above algorithm only slightly differs from the mirror descent SA algorithm in Chapter 2 in the way the averaging step (4.2.48) is defined. More specifically, the sequence $\{x_t^{av}\}_{t\geq 2}$ is obtained by averaging the iterates $x_t, t \geq 2$ with their corresponding weights γ_{t-1}, while the one in [43] is obtained by taking the average of the whole trajectory $x_t, t \geq 1$ with weights γ_t. Note however that, if the constant stepsizes are used, i.e., $\gamma_t = \gamma, \forall\, t \geq 1$, then the averaging step stated above is exactly the same as the one stated in [43] up to shifting one iterate. Thirdly, as we will see later in this section, the improvement on the convergence rate for the mirror descent algorithm described above, as applied to (4.1.34), over the one stated in (4.2.46) follows from a completely different convergence analysis than the one given in [43] and a new stepsize policy which takes into account the structure of the problem.

We start with stating a general convergence result of the above mirror descent SA algorithm without specifying the step-sizes γ_t. The proof of this result will be given in

Subsection 4.4.1.

Theorem 4.2.1 *Assume that the step-sizes γ_t satisfy $0 < \gamma_t \leq \alpha/(2L)$, $\forall\, t \geq 1$. Let $\{x_{t+1}^{av}\}_{t\geq 1}$ be the sequence computed according to (4.2.48) by the modified mirror descent SA algorithm. Then we have*

a) under Assumption A.4.1,

$$\mathbb{E}\left[\Psi(x_{t+1}^{av}) - \Psi^*\right] \leq K_0(t),\, \forall\, t \geq 1, \tag{4.2.49}$$

where

$$K_0(t) := \left(\sum_{\tau=1}^{t}\gamma_\tau\right)^{-1}\left[D_{\omega,X}^2 + \frac{2}{\alpha}(4\mathcal{M}^2+\sigma^2)\sum_{\tau=1}^{t}\gamma_\tau^2\right],$$

$\mathcal{M}$, σ and $D_{\omega,X}$ are given in (4.1.36), (4.1.38) and (2.2.29) respectively;

b) under Assumptions A.4.1 and A.4.2,

$$\text{Prob}\left\{\Psi(x_{t+1}^{av}) - \Psi^* > K_0(t) + \Lambda K_1(t)\right\} \leq \exp\{-\Lambda^2/3\} + \exp\{-\Lambda\},\, \forall\, \Lambda > 0, t \geq 1, \tag{4.2.50}$$

where

$$K_1(t) := \left(\sum_{\tau=1}^{t}\gamma_\tau\right)^{-1}\left[2\Omega_{\omega,X}\sigma\sqrt{\sum_{\tau=1}^{t}\gamma_\tau^2} + \frac{2}{\alpha}\sigma^2\sum_{\tau=1}^{t}\gamma_\tau^2\right],$$

σ and $\Omega_{\omega,X}$ are given in (4.1.38) and (2.2.29) respectively.

It is interesting to compare the results obtained in Theorem 4.2.1 with the corresponding results obtained in [43] for the original mirror descent SA. According to Proposition 3.2.1, for the N-step of the original Mirror Descent SA algorithm applied to (4.1.34), we have that

$$\mathbb{E}\left[\Psi(\tilde{x}_t) - \Psi^*\right] \leq \frac{D_{\omega,X}^2 + (2\alpha)^{-1}M_*^2\sum_{\tau=1}^{t}\gamma_\tau^2}{\sum_{\tau=1}^{t}\gamma_\tau},$$

where M_* is given by (4.2.45). Note that the right hand side of the above inequality differs from $K_0(t)$ defined in Theorem 4.2.1(a) in the second term only. The former one depends on M_*, whence $\|\nabla f(x_1)\| + L\Omega_{\omega,X}$, while such a dependence is removed from $K_0(t)$ for the new result obtained in Theorem 4.2.1(a).

We now describe the selection of the stepsizes for the modified mirror descent SA. For the sake of simplicity, let us suppose that the number of iterations for the above algorithm is fixed in advance, say equal to N, and that the *constant step-size policy* is applied, i.e., $\gamma_t = \gamma$, $t = 1, \cdots, N$, for some $\gamma < \alpha/(2L)$ (note that the assumption of constant step-sizes does not hurt the efficiency estimate). We then conclude from Theorem 4.2.1 that the obtained solution $x_{N+1}^{av} = N^{-1} \sum_{\tau=1}^{N} x_{\tau+1}$ satisfies

$$\mathbb{E}\left[\Psi(x_{N+1}^{av}) - \Psi^*\right] \leq \frac{D_{\omega,X}^2}{N\gamma} + \frac{2\gamma}{\alpha}\left(4\mathcal{M}^2 + \sigma^2\right).$$

Minimizing the right-hand-side of the above inequality with respect to γ over the interval $(0, \alpha/(2L)]$, we conclude that

$$\mathbb{E}\left[\Psi(x_{N+1}^{av}) - \Psi^*\right] \leq K_0^*(N) := \frac{L\Omega_{\omega,X}^2}{N} + \frac{2\Omega_{\omega,X}\sqrt{4\mathcal{M}^2 + \sigma^2}}{\sqrt{N}}, \tag{4.2.51}$$

by choosing γ as

$$\gamma = \min\left\{\frac{\alpha}{2L}, \sqrt{\frac{\alpha D_{\omega,X}^2}{2N(4\mathcal{M}^2 + \sigma^2)}}\right\}.$$

Moreover, with this choice of γ, we have

$$\begin{aligned} K_1(N) &= \frac{2\Omega_{\omega,X}\sigma}{\sqrt{N}} + \frac{2\gamma\sigma^2}{\alpha} \leq \frac{2\Omega_{\omega,X}\sigma}{\sqrt{N}} + \sqrt{\frac{2}{\alpha}} D_{\omega,X} \frac{\sigma^2}{\sqrt{N(4\mathcal{M}^2 + \sigma^2)}} \\ &\leq \frac{2\Omega_{\omega,X}\sigma}{\sqrt{N}} + \sqrt{\frac{2}{\alpha}} D_{\omega,X} \frac{\sigma}{\sqrt{N}} = \frac{3\Omega_{\omega,X}\sigma}{\sqrt{N}}, \end{aligned}$$

hence, bound (4.2.50) implies that

$$\text{Prob}\left\{\Psi(x_{N+1}^{av}) - \Psi^* > K_0^*(N) + \Lambda K_1^*(N)\right\} \leq \exp\{-\Lambda^2/3\} + \exp\{-\Lambda\}, \, \forall\, \Lambda > 0, \tag{4.2.52}$$

where

$$K_1^*(N) := \frac{3\Omega_{\omega,X}\sigma}{\sqrt{N}}.$$

It is interesting to compare the rate of convergence (4.2.51) obtained for the modified mirror descent SA and the one stated in (4.2.46) for the direction application of the mirror descent SA in [43]. Clearly, the latter one is always worse than the former one. Moreover, in the range

$$L \leq \frac{\sqrt{N(4\mathcal{M}^2 + \sigma^2)}}{\Omega_{\omega,X}}, \tag{4.2.53}$$

the first component in (4.2.51) (for abbreviation, the L-component) merely does not affect the error estimate (4.2.51). Note that the range stated in (4.2.53) extends as N increases, meaning that, if N is large, the presence of the smooth component f in the objective function of (4.1.34) does not affect the complexity of finding good approximate solutions. In contrast, this phenomenon does not appear in the error estimate (4.2.46) derived for the mirror descent SA algorithm in [43] which employs certain simple step-size strategies without taking into account the structure of the objective function Ψ.

It should be noted that the mirror descent SA is a direct descendant of the mirror descent algorithm [44]. It is well-known that algorithms of this type are not optimal for smooth convex optimization and hence can not be optimal for stochastic composite optimization. On the other hand, Nesterov's methods [47, 49] and its variants were designed for solving smooth convex optimization problems. These optimal algorithms for smooth convex optimization, with very aggressive stepsizes employed, were believed to be too sophisticated to solve non-smooth and stochastic convex optimization problems. We will investigate a possible extension of Nesterov's method for solving problem (4.1.34) in the next section.

4.3 Accelerated stochastic approximation

In this section, we provide a substantial generalization of Nesterov's methods ([47, 49]) to solve non-smooth and stochastic convex optimization. As a result, we develop a completely new SA-type algorithm, referred to as the accelerated SA (AC-SA) method, which can achieve the theoretically optimal rate of convergence for solving (4.1.34). More specifically, we will state the algorithm and its convergence results in Subsection 4.3.1 and outline an application to illustrate its advantages over the existing SA algorithms in Subsection 4.3.2.

4.3.1 The algorithm and its main convergence properties

The AC-SA algorithm for solving problem (4.1.34) is comprised of the updating of three sequences: $\{x_t\}_{t\geq 1}$, $\{x_t^{ag}\}_{t\geq 1}$, and $\{x_t^{md}\}_{t\geq 1}$. Here, we use the superscript "ag" (which stands for "aggregated") in the sequence obtained by taking a convex combination of all the previous iterates x_t, and the superscript "md" (which stands for "middle") in the sequence obtained by taking a convex combination of the current iterate x_t with the current

aggregated iterate x_t^{ag}. The algorithm is stated as follows.

The AC-SA algorithm:

0) Let the initial points $x_1^{ag} = x_1$, and the step-sizes $\{\beta_t\}_{t\geq 1}$ and $\{\gamma_t\}_{t\geq 1}$ be given. Set $t = 1$.

1) Set $x_t^{md} = \beta_t^{-1} x_t + (1 - \beta_t^{-1}) x_t^{ag}$,

2) Call the $\mathcal{SO}$ for computing $G(x_t^{md}, \xi_t)$. Set

$$x_{t+1} = P_{x_t}(\gamma_t G(x_t^{md}, \xi_t)), \tag{4.3.54}$$

$$x_{t+1}^{ag} = \beta_t^{-1} x_{t+1} + (1 - \beta_t^{-1}) x_t^{ag}, \tag{4.3.55}$$

3) Set $t \leftarrow t + 1$ and go to step 1.

end

We now make a few comments regarding the AC-SA algorithm described above. Firstly, similar to the mirror descent SA algorithm, we assume that the initial point x_1 is the minimizer of ω over X. Secondly, it is worth noting that the major computation cost in each iteration of the AC-SA algorithm is exactly the same as the one of the mirror descent SA algorithm, that is, each iteration of the above algorithm requires only one call to the $\mathcal{SO}$ and one solution of the subproblem (4.3.54). Thirdly, while Nesterov's optimal method and its variants [47, 49, 50, 1, 32, 73] were designed for solving deterministic smooth convex optimization problems, the AC-SA algorithm, as a descendant of Nesterov's optimal method, is capable of solving non-smooth and stochastic convex optimization problems.

Tseng [73] provides a very good summary about different versions of Nesterov's optimal method for deterministic smooth convex optimization. Depending on how much gradient information associated with the previous iterates is used in the prox-mapping, these methods are classified as either 1-memory methods in which only the gradient information from the preceding iterate is used (cf. [47, 1, 32] and [73, Section 3]), or ∞-memory methods where the gradient information from all previous iterates is used (cf. [50, 53] and [73, Section 4]). The 1-memory methods are conceptionally simpler than the ∞-memory methods,

while one possible advantage of the ∞-memory methods is that they also provide a lower estimate on the optimal value during the run of these procedures. Moreover, depending on whether one or two prox-mapping are required in Nesterov's methods and its variants, these methods are classified as either 1-projection methods (cf. [47, 1, 73]) or 2-projection methods (cf. [50, 53, 32]). Note also that Nesterov's method in its initial form was designed for unconstrained convex problems, and later extended to constrained optimization [49] or certain nonsmooth and composite problems with special structures (see Section 4.1 for a summary in this regard). Auslender and Teboulle [1] is among the first to propose, in a deterministic setting, a Bregman regularization of Nesterov method, see also Nesterov [50]. Our AC-SA algorithm is the simplest 1-memory and 1-projection method, while it is also possible to develop SA-type algorithms based on other versions of Nesterov's methods. If the non-smooth component h in the objective function $\Psi(\cdot)$ of (4.1.34) does not exist, or equivalently $h(x) = 0$ for every $x \in X$, and there is no noise in the computed gradient, i.e., $\sigma = 0$, then the AC-SA reduces to a variant of Nesterov's optimal method that is very similar to the ones state in [1, Section 5] and [73, Section 3]. Note that one unique feature of the AC-SA algorithm is that it employs two control parameters β_t and γ_t, which distinguishes itself from other variants of Nesterov's optimal method.

The following theorem states the main convergence results of the AC-SA algorithm applied to stochastic composite optimization, which covers a significantly wider range of problems than those in deterministic smooth convex optimization (see discussions in Section 4.1). The proof of this result will be given in Subsection 4.4.2

Theorem 4.3.1 *Assume that the stepsizes $\beta_t \in [1, \infty)$ and $\gamma_t \in \Re_+$ are chosen such that $\beta_1 = 1$ and the following conditions hold*

$$0 < (\beta_{t+1} - 1)\gamma_{t+1} \leq \beta_t \gamma_t \text{ and } 2L\gamma_t \leq \alpha\beta_t, \ \forall t \geq 1. \tag{4.3.56}$$

Let $\{x_{t+1}^{ag}\}_{t\geq 1}$ be the sequence computed according to (4.3.55) by the AC-SA algorithm. Then we have

a) under Assumption A.4.1,

$$\mathbb{E}[\Psi(x_{t+1}^{ag}) - \Psi^*] \leq \hat{K}_0(t), \, \forall \, t \geq 1, \tag{4.3.57}$$

where

$$\hat{K}_0(t) := \frac{1}{(\beta_{t+1}-1)\gamma_{t+1}} \left[D_{\omega,X}^2 + \frac{2}{\alpha}(4\mathcal{M}^2 + \sigma^2) \sum_{\tau=1}^{t} \gamma_\tau^2 \right],$$

$\mathcal{M}$, σ and $D_{\omega,X}$ are given in (4.1.36), (4.1.38) and (2.2.29) respectively;

b) under Assumptions A.4.1 and A.4.2,

$$\text{Prob}\left\{ \Psi(x_{t+1}^{ag}) - \Psi^* > \hat{K}_0(t) + \Lambda \hat{K}_1(t) \right\} \leq \exp\{-\Lambda^2/3\} + \exp\{-\Lambda\}, \, \forall \, \Lambda > 0, t \geq 1, \tag{4.3.58}$$

where

$$\hat{K}_1(t) := \frac{1}{(\beta_{t+1}-1)\gamma_{t+1}} \left[2\Omega_{\omega,X}\sigma \sqrt{\sum_{\tau=1}^{t} \gamma_\tau^2} + \frac{2}{\alpha}\sigma^2 \sum_{\tau=1}^{t} \gamma_\tau^2 \right],$$

σ and $\Omega_{\omega,X}$ are given in (4.1.38) and (2.2.29) respectively.

It is worth noting the similarity between the results stated in Theorem 4.2.1 for the modified mirror-descent SA and those obtained in Theorem 4.3.1 for the accelerated SA. Comparing (4.2.49) with (4.3.57) (resp., (4.2.50) with (4.3.58)), we can easily that the only difference exists in the factors of $K_0(t)$ and $\hat{K}_0(t)$ (resp., $K_1(t)$ and $\hat{K}_1(t)$). More specifically, the factor $1/\sum_{\tau=1}^{t} \gamma_t$ in $K_0(t)$ and $K_1(t)$ is replaced by $1/[(\beta_{t+1}-1)\gamma_{t+1}]$ in $\hat{K}_0(t)$ and $\hat{K}_1(t)$ while all other terms are the same. This resemblance between the results stated in Theorem 4.2.1 and Theorem 4.3.1 is the outcome of a remarkable unified convergence analysis for both mirror-descent and accelerated SA algorithm (cf. Section 4.4).

We now discuss the determination of the stepsizes β_t and γ_t in the accelerated SA so as to achieve the optimal rate of convergence for solving (4.1.34). Observing that a pair of sequences $\{\beta_t\}_{t\geq 1}$ and $\{\gamma_t\}_{t\geq 1}$ satisfying condition (4.3.56) is given by:

$$\beta_t = \frac{t+1}{2} \text{ and } \gamma_t = \frac{t+1}{2}\gamma \tag{4.3.59}$$

for any $0 < \gamma \leq \alpha/(2L)$, we obtain the following corollary of Theorem 4.3.1 by appropriately choosing this parameter γ.

Corollary 4.3.1 *Suppose that the stepsizes β_t and γ_t in the AC-SA algorithm are set to*

$$\beta_t = \frac{t+1}{2}, \quad \gamma_t = \frac{t+1}{2}\min\left\{\frac{\alpha}{2L}, \frac{\sqrt{6\alpha}D_{\omega,X}}{(N+2)^{\frac{3}{2}}(4\mathcal{M}^2+\sigma^2)^{\frac{1}{2}}}\right\}, \ \forall t \geq 1, \tag{4.3.60}$$

where N is a fixed in advance number of iterations. Then, we have under Assumption A.4.1,

$$\mathbb{E}[\Psi(x_{N+1}^{ag}) - \Psi^*] \leq \hat{K}_0^*(N) := \frac{4L\Omega_{\omega,X}^2}{N(N+2)} + \frac{4\Omega_{\omega,X}\sqrt{4\mathcal{M}^2+\sigma^2}}{\sqrt{N}}, \tag{4.3.61}$$

if in addition, Assumption A.4.2 holds, then

$$\text{Prob}\left\{\Psi(x_{N+1}^{ag}) - \Psi^* > \hat{K}_0^*(N) + \Lambda\hat{K}_1^*(N)\right\} \leq \exp\{-\Lambda^2/3\} + \exp\{-\Lambda\}, \ \forall\, \Lambda > 0, \tag{4.3.62}$$

where

$$\hat{K}_1^*(N) := \frac{10\Omega_{\omega,X}\sigma}{\sqrt{N}}.$$

Proof. Clearly, the stepsizes $\{\beta_t\}_{t\geq 1}$ and $\{\gamma_t\}_{t\geq 1}$ stated in (4.3.60) satisfy the conditions $\beta_1 = 1$, $\beta_t > 1, \forall t \geq 2$, and (4.3.56). Denoting

$$\gamma^* := \min\left\{\frac{\alpha}{2L}, \frac{\sqrt{6\alpha}D_{\omega,X}}{(N+2)^{\frac{3}{2}}(4\mathcal{M}^2+\sigma^2)^{\frac{1}{2}}}\right\},$$

we then conclude from Theorem 4.3.1 that, under Assumption A.4.1,

$$\mathbb{E}[\Psi(x_{N+1}^{ag}) - \Psi^*] \leq \mathcal{T}_0 := \frac{4D_{\omega,X}^2}{N(N+2)\gamma^*} + \frac{8\gamma^*(4\mathcal{M}^2+\sigma^2)}{\alpha N(N+2)}\sum_{\tau=1}^{N}\left(\frac{\tau+1}{2}\right)^2, \tag{4.3.63}$$

and that, under Assumptions A.4.1 and A.4.2,

$$\text{Prob}\left\{\Psi(x_{N+1}^{ag}) - \Psi^* > \mathcal{T}_0 + \Lambda\mathcal{T}_1)\right\} \leq \exp\{-\Lambda^2/3\} + \exp\{-\Lambda\}, \ \forall\, \Lambda > 0, \tag{4.3.64}$$

where

$$\mathcal{T}_1 := \frac{8\Omega_{\omega,X}\sigma}{N(N+2)}\sqrt{\sum_{\tau=1}^{N}\left(\frac{\tau+1}{2}\right)^2} + \frac{8\gamma^*\sigma^2}{N(N+2)\alpha}\sum_{\tau=1}^{N}\left(\frac{\tau+1}{2}\right)^2.$$

Moreover, using the simple observations $\sum_{\tau=1}^{N}(\tau+1)^2 \leq \int_1^{N+1}(u+1)^2du \leq (N+2)^3/3$, $N+2 \leq 3N$ due to $N \geq 1$, and the definition of γ^*, we obtain

$$\begin{aligned}\mathcal{T}_0 &\leq \frac{4D_{\omega,X}^2}{N(N+2)\gamma^*} + \frac{2\gamma^*(4\mathcal{M}^2+\sigma^2)(N+2)^2}{3\alpha N} \leq \frac{8LD_{\omega,X}^2}{N(N+2)\alpha} + \frac{8D_{\omega,X}(4\mathcal{M}^2+\sigma^2)^{\frac{1}{2}}(N+2)^{\frac{1}{2}}}{\sqrt{6\alpha}N} \\ &\leq \frac{8LD_{\omega,X}^2}{N(N+2)\alpha} + \frac{8D_{\omega,X}(4\mathcal{M}^2+\sigma^2)^{\frac{1}{2}}}{\sqrt{2\alpha N}} = \frac{4L\Omega_{\omega,X}^2}{N(N+2)} + \frac{4\Omega_{\omega,X}\sqrt{4\mathcal{M}^2+\sigma^2}}{\sqrt{N}} = \hat{K}_0^*(N),\end{aligned}$$

and

$$\begin{aligned}\mathcal{T}_1 &\leq \frac{8\Omega_{\omega,X}\sigma}{\sqrt{3}N}(N+2)^{\frac{1}{2}} + \frac{2\gamma^*\sigma^2}{3N\alpha}(N+2)^2 \leq \frac{8\Omega_{\omega,X}\sigma}{\sqrt{N}} + \frac{2\sigma^2(N+2)^{\frac{1}{2}}}{3N\sqrt{\alpha}}\frac{\sqrt{6}D_{\omega,X}}{\sqrt{4\mathcal{M}^2+\sigma^2}} \\ &\leq \frac{8\Omega_{\omega,X}\sigma}{\sqrt{N}} + \frac{2\sqrt{2}D_{\omega,X}\sigma}{\sqrt{\alpha N}} = \frac{10\Omega_{\omega,X}\sigma}{\sqrt{N}} = \hat{K}_1^*(N).\end{aligned}$$

Our claim immediately follows by substituting the above bounds of $\mathcal{T}_0$ and $\mathcal{T}_1$ into (4.3.63) and (4.3.64). ∎

We now make a few observations regarding the results obtained in Theorem 4.3.1 and Corollary 4.3.1. Firstly, it is interesting to compare bounds (4.3.61) and (4.2.51) obtained for the AC-SA algorithm and the mirror descent SA algorithm respectively. Clearly, the first one is always better than the latter one up to a constant factor provided that $L > 0$. Moreover, the AC-SA algorithm substantially enlarges the range of L in which the L-component (the first component in (4.3.61)) does not affect the error estimate. Specifically, within the range

$$L \leq \frac{\sqrt{4\mathcal{M}^2+\sigma^2}N^{\frac{3}{2}}}{\Omega_{\omega,X}}, \tag{4.3.65}$$

which extends much faster than (4.2.53) as N increases, the L-component does not change the order of magnitude for the rate of convergence associated with the AC-SA algorithm.

Secondly, observe that the results obtained in Theorem 4.3.1 and Corollary 4.3.1 still hold when the Lipschitz constant $L = 0$. More specifically, we consider the case where $f(x) = 0$ for any $x \in X$. In this case, the stepsizes $\{\beta_t\}_{t\geq 1}$ and $\{\gamma_t\}_{t\geq 1}$ in (4.3.60) become

$$\beta_t = \frac{t+1}{2}, \quad \gamma_t = \frac{\sqrt{6\alpha}D_{\omega,X}(t+1)}{2(N+2)^{\frac{3}{2}}(4\mathcal{M}^2+\sigma^2)^{\frac{1}{2}}}, \ 1 \leq t \leq N+1,$$

and the error estimate (4.3.61) reduces to

$$\mathbb{E}[h(x_{N+1}^{ag}) - h^*] \leq \frac{4\Omega_{\omega,X}\sqrt{4\mathcal{M}^2+\sigma^2}}{\sqrt{N}},$$

where $h^* := \min_{x\in X} h(x)$. Note also that one alternative characterization of x_{N+1}^{ag} is given

by

$$\begin{aligned} x_{N+1}^{ag} &= \frac{2}{N+1}x_{N+1} + \frac{N-1}{N+1}x_N^{ag} = \frac{2}{N+1}x_{N+1} + \frac{2(N-1)}{N(N+1)}x_N + \frac{(N-2)(N-1)}{N(N+1)}x_{N-1}^{ag} \\ &= \frac{2}{N+1}x_{N+1} + \frac{2(N-1)}{N(N+1)}x_N + \frac{2(N-2)}{N(N+1)}x_{N-1} + \cdots + \frac{2}{N(N+1)}x_2 \\ &= \frac{\sum_{t=1}^N (tx_{t+1})}{\sum_{t=1}^N t}. \end{aligned}$$

Hence, in contrast to the usual *constant stepsize* or *decreasing stepsize* strategy (see [43]), the stepsizes γ_t in step (4.3.54) and the weights for taking the average in step (4.3.55) are increasing with the increment of t. To the best of our knowledge, this is the first time that an *increasing stepsize* strategy is introduced in the literature of stochastic approximation or subgradient methods. It is also one of the crucial developments that enable us to have a unified treatment for smooth, non-smooth and stochastic convex optimization.

Finally, note that if there is no stochastic error for the computed subgradient of Ψ, i.e., $\sigma = 0$, then bound (4.3.61) reads

$$\Psi(x_{N+1}^{ag}) - \Psi^* \leq \frac{4L\Omega_{\omega,X}^2}{N(N+2)} + \frac{8\Omega_{\omega,X}\mathcal{M}}{\sqrt{N}},$$

which basically says that the impact of the smooth component on the efficiency estimate vanishes very quickly as N grows. This result also seems to be new in the area of deterministic convex optimization.

4.3.2 Application to stochastic programming

The goal of this subsection is to demonstrate the significant advantages of the AC-SA algorithm over the existing algorithms, for example, the mirror descent SA algorithm, when applied for solving certain class of stochastic programming problems.

Consider the problem of

$$\begin{aligned} \tilde{h}^* := \min_x \quad & \left\{\tilde{h}(x) := \mathbb{E}[\tilde{H}(x,\xi)]\right\} \\ \text{s.t.} \quad & \mathcal{A}x - b = 0, \quad x \in X, \end{aligned} \tag{4.3.66}$$

where $X \subset \mathbb{R}^n$ is a nonempty compact convex set, $\mathcal{A} : \Re^n \to \Re^m$ is a linear operator, $b \in \Re^m$ is given, ξ is a random vector whose probability distribution P is supported on set

$\Xi \subseteq \mathbb{R}^d$ and $H : X \times \Xi \to \mathbb{R}$. We assume that for every $\xi \in \Xi$ the function $\tilde{H}(\cdot, \xi)$ is convex on X, and that the expectation

$$\mathbb{E}[\tilde{H}(x, \xi)] = \int_{\Xi} \tilde{H}(x, \xi) dP(\xi) \tag{4.3.67}$$

is well defined and finite valued for every $x \in X$. It follows that function $\tilde{h}(\cdot)$ is convex and finite valued on X. Moreover, we assume that $\tilde{h}(\cdot)$ is continuous on X. Of course, continuity of $\tilde{h}(\cdot)$ follows from convexity if $\tilde{h}(\cdot)$ is finite valued and convex on a neighborhood of X. With these assumptions, (4.3.66) becomes a convex programming problem. We also make the following assumptions:

A.4.3:

a) It is possible to generate an iid sample $\xi_1, \xi_2, ...$, of realizations of random vector ξ.

b) We have access to a "black box" subroutine (a stochastic oracle). At i-th call, $x \in X$ being the input, the oracle returns a *stochastic subgradient* – a vector $\mathsf{G}(x, \xi_i)$ such that for every $x \in X$, the vector $\mathbb{E}[\mathsf{G}(x, \xi)]$ is well defined and is a subgradient of $\tilde{h}(\cdot)$ at x.

c) There is a constant $\mathcal{M} > 0$ such that

$$\forall x \in X : \mathbb{E}\left[\exp\{\|\mathsf{G}(x, \xi)\|_*^2 / \mathcal{M}^2\}\right] \leq \exp\{1\}. \tag{4.3.68}$$

For the case where the feasible region consists only of the simple convex set X, or equivalently $\mathcal{A} \equiv 0$, Nemirovski et. al. demonstrated in [43] that the mirror descent SA algorithm can substantially outperform the sampling averaging approximation (Shapiro [68]), a widely used approach for stochastic programming in practice. When $\mathcal{A}$ is not identically 0, the mirror descent SA algorithm can still be applied directly to problem (4.3.66) but this approach would require the computation of the prox-mapping onto the feasible region $X \cap \{x : \mathcal{A}x - b = 0\}$, which can be very expensive for many practical problems. Moreover, the selection of the norm $\|\cdot\|$ and the distance generating function ω will be problem dependent. In other words, it is not clear what is the optimal way for choosing these parameter

settings (See Chapters 2 and 3 for more discussions about the parameter settings when the domain is relatively simple).

One alternative approach to alleviate this difficulty is to apply the quadratic penalty approach: instead of solving (4.3.66), we solve certain penalization problem of (4.3.66) obtained by penalizing the violation of the constraint $\mathcal{A}x - b = 0$. In particular, given a penalty parameter $\rho > 0$, we solve

$$\tilde{\Psi}^* = \tilde{\Psi}^*_\rho := \inf_{x \in X} \left\{ \tilde{\Psi}_\rho(x) := \tilde{f}_\rho(x) + \tilde{h}(x) \right\}, \tag{4.3.69}$$

where $\tilde{f}_\rho(x) := \rho\|\mathcal{A}x - b\|^2/2$ and $\|\cdot\|$ denotes the norm induced by the inner product $\langle\cdot,\cdot\rangle$ in $\Re^m$. Define the operator norm $\|\mathcal{A}\| := \max\{\|\mathcal{A}x\|_* : \|x\| \le 1\}$. It can be easily seen that $\nabla\tilde{f}_\rho(x) = \rho\mathcal{A}^*(\mathcal{A}x - b)$ and hence that

$$\|\nabla\tilde{f}_\rho(x) - \nabla\tilde{f}_\rho(x')\|_* = \rho\|\mathcal{A}^*\mathcal{A}(x - x')\|_* \le \rho\|\mathcal{A}^*\|\|\mathcal{A}\|\|x - x'\| = \rho\|\mathcal{A}\|^2\|x - x'\|, \, \forall\, x, x' \in X, \tag{4.3.70}$$

where the last equality follows from the fact that $\|\mathcal{A}\| = \|\mathcal{A}^*\|$. Moreover, in view of Assumption A.4.3 and Jensen's inequality, for any $x \in X$, there exists $\tilde{h}'(x) := \mathbb{E}[\mathsf{G}(x, \xi_t)] \in \partial\tilde{h}(x)$ such that $\mathbb{E}[\|\mathsf{G}(x, \xi_t)\|_*^2] \le \mathcal{M}^2$ and hence that $\|\tilde{h}'(x)\|_* = \|\mathbb{E}[\mathsf{G}(x, \xi_t)]\|_* \le \mathcal{M}$, which together with the fact $\tilde{h}(x) - \tilde{h}(x') \le \langle\tilde{h}'(x), x - x'\rangle, \forall x, x' \in X$ due to the convexity of $\tilde{h}$, clearly imply that

$$|\tilde{h}(x) - \tilde{h}(x')| \le \mathcal{M}\|x - x'\|, \;\; \forall x, x' \in X. \tag{4.3.71}$$

Therefore, the penalization problem (4.3.69) is given in the form of (4.1.34), and can be approximately solved by either the mirror descent SA or the AC-SA algorithm developed in this chapter.

It is well-known that the near-optimal solutions of the penalization problem (4.3.69) also yield near-optimal solutions of (4.3.66) if the penalty parameter ρ is sufficiently large. In this chapter, we are interested in obtaining one particular type of near-optimal solutions of (4.3.66) defined in the following way. First note that x^* is an optimal solution of (4.3.66) if, and only if, $x^* \in X$, $\mathcal{A}x^* - b = 0$ and $\tilde{h}(x^*) \le \tilde{h}^*$. This observation leads us to our definition of a near optimal solution $\tilde{x} \in X$ of (4.3.66), which essentially requires the primal

infeasibility measure $\|\mathcal{A}\tilde{x}-b\|_2$ and the primal optimality gap $[\tilde{h}(\tilde{x})-\tilde{h}^*]^+$ to be both small (Lan and Monteiro [33]).

Definition: Let $\epsilon_p, \epsilon_o > 0$ be given, $\tilde{x} \in X$ is called an (ϵ_p, ϵ_o)-*primal solution* for (4.3.66) if

$$\|\mathcal{A}\tilde{x} - b\| \leq \epsilon_p \text{ and } \tilde{h}(\tilde{x}) - \tilde{h}^* \leq \epsilon_0. \tag{4.3.72}$$

One drawback of the above notion of near optimality of $\tilde{x}$ is that it says nothing about the size of $[\tilde{h}(\tilde{x}) - \tilde{h}^*]^-$. Assume that the set of Lagrange multiplier for (4.3.66)

$$Y^* := \{y \in \Re^m : \tilde{h}^* = \inf\{\tilde{h}(x) + \langle \mathcal{A}x - b, y\rangle : x \in X\}$$

is nonempty. It was observed in [33] that this quantity can be bounded as $[\tilde{h}(\tilde{x}) - \tilde{h}^*]^- \leq \epsilon_p\|y^*\|$, where $y^* \in Y^*$ is an arbitrary Lagrange multiplier for (4.3.66). It is worth noting that some other types of near-optimal solutions of (4.3.66), for example, the primal-dual near-optimal solutions defined in [33], can also be obtained by applying the quadratic penalty approach.

We are now ready to state the iteration-complexity bounds for the modified mirror descent SA and the AC-SA algorithm, applied to the penalization problem (4.3.69), to compute an (ϵ_p, ϵ_o)-primal solution of (4.3.66).

Theorem 4.3.2 *Let y^* be an arbitrary Lagrange multiplier for (4.3.66). Also let the confidence level $\eta \in (0,1)$ and the accuracy tolerance $(\epsilon_p, \epsilon_o) \in \Re_{++} \times \Re_{++}$ be given. If*

$$\rho = \rho(t) := \left(\frac{\sqrt{\epsilon_o + 4\epsilon_p\, t} + \sqrt{\epsilon_o}}{\sqrt{2}\epsilon_p}\right)^2 \tag{4.3.73}$$

for some $t \geq \|y^*\|$, then, with probability greater than $1-\eta$,

a) the RM-SA algorithm applied to (4.3.69) finds an (ϵ_p, ϵ_o)-primal solution of (4.3.66) in at most

$$N_{md}(t) := \left\lceil \max\left\{2R(t)^2, (8\sqrt{2} + 12\lambda)^2 S\right\}\right\rceil \tag{4.3.74}$$

iterations;

b) the AC-SA algorithm applied to (4.3.69) finds an (ϵ_p, ϵ_o)-primal solution of (4.3.66) in at most

$$N_{ac}(t) := \left\lceil \max\left\{ \sqrt{2}R(t), (16\sqrt{2} + 40\lambda)^2 S \right\} \right\rceil \tag{4.3.75}$$

iterations,

where λ satisfies $\exp(-\lambda^2/3) + \exp(-\lambda) \leq \eta$ (clearly $\lambda = \mathcal{O}(1)\log 1/\eta$),

$$R(t) := \frac{\sqrt{\rho(t)}\|\mathcal{A}\|\Omega}{\sqrt{\epsilon_o}}, \quad S := \left(\frac{\Omega M}{\epsilon_o}\right)^2, \tag{4.3.76}$$

Ω and M are given by (2.2.29) and (4.3.68) respectively.

We now make a few observations regarding Theorem 4.3.2. First, the choice of ρ given by (4.3.73) requires that $t \geq \|y^*\|$ and that the iteration-complexity bounds $N_{md}(t)$ and $N_{ac}(t)$ obtained in Theorem 4.3.2 are non-decreasing with respect to t. Second, since the quantity $\|y^*\|$ is not known a priori, it is necessary to guess the value of t. Note however that the influence of t, whence $\|y^*\|$, on the bound $N_{ac}(t)$ is much weaker than that on the bound $N_{md}(t)$. For example, assume that $\epsilon_p = \epsilon_o = \epsilon$. By some straightforward computation, it can be easily seen that the value of $N_{ac}(t)$ does not change when

$$\|y^*\| \leq t \leq \frac{1}{4}\left[\left(\frac{(16\sqrt{2} + 40\lambda)^2 \Omega M^2}{\|A\|\epsilon} - 1\right)^2 - 1\right],$$

while the range of t that does not affect $N_{md}(t)$ is given by

$$\|y^*\| \leq t \leq \frac{1}{4}\left[\left(\frac{(8\sqrt{2} + 12\lambda)M}{\|\mathcal{A}\|} - 1\right)^2 - 1\right].$$

In other words, the AC-SA algorithm allows a big range for t (or $\|y^*\|$), as high as $\mathcal{O}(1/\epsilon^2)$, without affecting the effort to find good approximate solutions of (4.3.66), while the corresponding one for the RM-SA algorithm is much smaller, roughly in $\mathcal{O}(1)$. As a consequence, when $\epsilon \downarrow 0$, the size of Lagrange multiplier asymptotically does not affect the rate of convergence, which seems to be a very important property that has not been observed for the penalty based approaches. Finally, even if t does affect the bounds $N_{ac}(t)$ or $N_{md}(t)$ (i.e., t sits outside the ranges described above), the first bound is in $\mathcal{O}(R(t))$ while the latter one is in $\mathcal{O}(R(t)^2)$.

4.4 Convergence analysis

The goal of this section is to prove the main results of this chapter, namely, Theorems 4.2.1, 4.3.1, and 4.3.2.

4.4.1 Convergence analysis for the mirror descent SA

This subsection is to devoted to the proof of Theorem 4.2.1. Before proving this result, we establish a few technical results from which Theorem 4.2.1 immediately follows.

Let $p(u)$ be a convex function over a convex set $X \in \mathcal{E}$. Assume that $\hat{u}$ is an optimal solution of the problem $\min\{p(u) + \|u - \tilde{x}\|^2 : u \in X\}$ for some $\tilde{x} \in X$. Due to the well-known fact that the sum of a convex and a strongly convex function is also strongly convex, one can easily see that

$$p(u) + \|u - \tilde{x}\|^2 \geq \min\{p(u) + \|u - \tilde{x}\|^2 : u \in X\} + \|u - \hat{u}\|^2.$$

The next lemma generalizes this result to the case where the function $\|u - \tilde{x}\|^2$ is replaced with the prox-function $V(\tilde{x}, u)$ associated with a convex function ω. It is worth noting that the result described below does not assume the strong-convexity of the function ω.

Lemma 4.4.1 *Let X be a convex set in $\mathcal{E}$ and $p, \omega : X \to \Re$ be differentiable convex functions. Assume that $\hat{u}$ is an optimal solution of* $\min\{p(u) + V(\tilde{x}, u) : u \in X\}$. *Then,*

$$\min\{p(u) + V(\tilde{x}, u) : u \in X\} \leq p(u) + V(\tilde{x}, u) - V(\hat{u}, u), \quad \forall u \in X.$$

Proof. The definition of $\hat{u}$ and the fact that $p(\cdot) + V(\tilde{x}, \cdot)$ is a differentiable convex function imply that

$$\langle \nabla p(\hat{u}) + \nabla V(\tilde{x}, \hat{u}), u - \hat{u} \rangle \geq 0, \quad \forall u \in X,$$

where $\nabla V(\tilde{x}, \hat{u})$ denotes the gradient of $V(\tilde{x}, \cdot)$ at $\hat{u}$. Using the definition of the prox-function (2.2.27), it is easy to verify that

$$V(\tilde{x}, u) = V(\tilde{x}, \hat{u}) + \langle \nabla V(\tilde{x}, \hat{u}), u - \hat{u} \rangle + V(\hat{u}, u), \quad \forall u \in X.$$

Using the above two relations and the assumption that p is convex, we then conclude that

$$\begin{aligned} p(u) + V(\tilde{x}, u) &= p(u) + V(\tilde{x}, \hat{u}) + \langle \nabla V(\tilde{x}, \hat{u}), u - \hat{u} \rangle + V(\hat{u}, u)] \\ &\geq p(\hat{u}) + V(\tilde{x}, \hat{u}) + \langle \nabla p(\hat{u}) + \nabla V(\tilde{x}, \hat{u}), u - \hat{u} \rangle + V(\hat{u}, u) \\ &\geq p(\hat{u}) + V(\tilde{x}, \hat{u}) + V(\hat{u}, u), \end{aligned}$$

and hence that the lemma holds. ∎

The following lemma summarizes some properties of the objective function Ψ and f.

Lemma 4.4.2 *Let the functions $\Psi : X \to \Re$ and $f : X \to \Re$ be defined in (4.1.34). We have*

$$0 \leq f(y) - f(x) - \langle \nabla f(x), y - x \rangle \leq \tfrac{L}{2} \|y - x\|^2 \tag{4.4.77}$$

$$0 \leq \Psi(y) - \Psi(x) - \langle \Psi'(x), y - x \rangle \leq \tfrac{L}{2} \|y - x\|^2 + 2\mathcal{M}\|y - x\| \tag{4.4.78}$$

for any $x, y \in X$, where $\Psi'(x) \in \partial \Psi(x)$.

Proof. The first inequalities in both relations (4.4.77) and (4.4.78) follow immediately from the convexity of f and Ψ respectively. The second inequality in (4.4.77) is well-known (see Theorem 2.1.5 of [49] for a proof). This inequality, together with the fact $h(y) - h(x) \leq \mathcal{M}\|y - x\|$ due to the Lipschitz-continuity of h and the identity $\Psi'(x) = \nabla f(x) + h'(x)$ for some $h'(x) \in \partial h(x)$, then imply that

$$\begin{aligned} \Psi(y) &= f(y) + h(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2 + h(x) + \mathcal{M}\|y - x\| \\ &= \Psi(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2 + \mathcal{M}\|y - x\| \\ &= \Psi(x) + \langle \Psi'(x), y - x \rangle + \frac{L}{2}\|y - x\|^2 + \mathcal{M}\|y - x\| - \langle h'(x), y - x \rangle \\ &\leq \Psi(x) + \langle \Psi'(x), y - x \rangle + \frac{L}{2}\|y - x\|^2 + 2\mathcal{M}\|y - x\|, \end{aligned}$$

where the last inequality follows from (4.1.42) with $g = h'(x)$ and $d = x - y$. ∎

The following lemma establishes an important recursion for the mirror descent SA algorithm. Before stating this result, we mention the following simple inequality that will be used more than once in this section:

$$bu - \frac{au^2}{2} \leq \frac{b^2}{2a}, \ \forall a > 0. \tag{4.4.79}$$

Lemma 4.4.3 *Assume that the stepsizes γ_τ satisfy $L\gamma_\tau < \alpha$, $\tau \geq 1$. Let $x_1, \cdots, x_\tau \in X$ be given and $(x_{\tau+1}, x_{\tau+1}^{av}) \in X \times X$ be a pair computed according (4.2.47) and (4.2.48). Also let $\delta_\tau := G(x_\tau, \xi_\tau) - g(x_\tau)$, where $g(x_\tau) = \mathbb{E}[G(x_\tau, \xi_\tau)] \in \partial\Psi(x_\tau)$. Then, we have*

$$\gamma_\tau[\Psi(x_{\tau+1}) - \Psi(x)] + V(x_{\tau+1}, x) \leq V(x_\tau, x) + \Delta_\tau(x), \ \forall\, x \in X, \tag{4.4.80}$$

where

$$\Delta_\tau(x) := \gamma_\tau\langle\delta_\tau, x - x_\tau\rangle + \frac{(2\mathcal{M} + \|\delta_\tau\|_*)^2\gamma_\tau^2}{2(\alpha - L\gamma_\tau)}. \tag{4.4.81}$$

Proof. Denoting $d_\tau := x_{\tau+1} - x_\tau$, due to the strong-convexity of ω, we have $\alpha\|d_\tau\|^2/2 \leq V(x_\tau, x_{\tau+1})$, which together with (4.4.78), then imply that

$$\begin{aligned}
\gamma_\tau\Psi(x_{\tau+1}) &\leq \gamma_\tau[\Psi(x_\tau) + \langle g(x_\tau), d_\tau\rangle + \frac{L}{2}\|d_\tau\|^2 + 2\mathcal{M}\|d_\tau\|] \\
&= \gamma_\tau[\Psi(x_\tau) + \langle g(x_\tau), d_\tau\rangle] + \frac{\alpha}{2}\|d_\tau\|^2 - \frac{\alpha - L\gamma_\tau}{2}\|d_\tau\|^2 + 2\mathcal{M}\gamma_\tau\|d_\tau\| \\
&\leq \gamma_\tau[\Psi(x_\tau) + \langle g(x_\tau), d_\tau\rangle] + V(x_\tau, x_{\tau+1}) - \frac{\alpha - L\gamma_\tau}{2}\|d_\tau\|^2 + 2\mathcal{M}\gamma_\tau\|d_\tau\| \\
&= \gamma_\tau[\Psi(x_\tau) + \langle G(x_\tau, \xi_\tau), d_\tau\rangle] - \gamma_\tau\langle\delta_\tau, d_\tau\rangle + V(x_\tau, x_{\tau+1}) - \frac{\alpha - L\gamma_\tau}{2}\|d_\tau\|^2 + 2\mathcal{M}\gamma_\tau\|d_\tau\| \\
&\leq \gamma_\tau[\Psi(x_\tau) + \langle G(x_\tau, \xi_\tau), d_\tau\rangle] + V(x_\tau, x_{\tau+1}) - \frac{\alpha - L\gamma_\tau}{2}\|d_\tau\|^2 + (2\mathcal{M} + \|\delta_\tau\|_*)\gamma_\tau\|d_\tau\| \\
&\leq \gamma_\tau[\Psi(x_\tau) + \langle G(x_\tau, \xi_\tau), d_\tau\rangle] + V(x_\tau, x_{\tau+1}) + \frac{(2\mathcal{M} + \|\delta_\tau\|_*)^2\gamma_\tau^2}{2(\alpha - L\gamma_\tau)},
\end{aligned}$$

where the last inequality follows from (4.4.79) with $u = \|d_\tau\|$, $b = (2\mathcal{M} + \|\delta\|_*)\gamma_\tau$, and $a = \alpha - L\gamma_\tau$.

Moreover, it follows from the identity (4.2.47), (2.2.28), and Lemma 4.4.1 with $\tilde{x} = x_\tau$, $\hat{u} = x_{\tau+1}$, and $p(\cdot) \equiv \gamma_\tau\langle G(x_\tau, \xi_\tau), \cdot - x_\tau\rangle$ that

$$\begin{aligned}
&\gamma_\tau\Psi(x_\tau) + [\gamma_\tau\langle G(x_\tau, \xi_\tau), x_{\tau+1} - x_\tau\rangle + V(x_\tau, x_{\tau+1})] \\
&\quad\leq \gamma_\tau\Psi(x_\tau) + [\gamma_\tau\langle G(x_\tau, \xi_\tau), x - x_\tau\rangle + V(x_\tau, x) - V(x_{\tau+1}, x)] \\
&\quad= \gamma_\tau[\Psi(x_\tau) + \langle g(x_\tau), x - x_\tau\rangle] + \gamma_\tau\langle\delta_\tau, x - x_\tau\rangle + V(x_\tau, x) - V(x_{\tau+1}, x) \\
&\quad\leq \gamma_\tau\Psi(x) + \gamma_\tau\langle\delta_\tau, x - x_\tau\rangle + V(x_\tau, x) - V(x_{\tau+1}, x),
\end{aligned}$$

where the last inequality follows from the convexity of $\Psi(\cdot)$ and the fact $g(x_\tau) \in \partial\Psi(x_\tau)$.

Combining the above two conclusions and rearranging the terms, we obtain (4.4.80). ∎

Now let us state the following well-known large-deviation result for the martingale sequence (see for example, Lemma 3.6.2 in Chapter 3 for a proof).

Lemma 4.4.4 *Let $\xi_1, \xi_2, ...$ be a sequence of iid random variables, and $\zeta_t = \zeta_t(\xi_{[t]})$ be deterministic Borel functions of $\xi_{[t]}$ such that $\mathbb{E}_{|\xi_{[t-1]}}[\zeta_t] = 0$ a.s. and $\mathbb{E}_{|\xi_{[t-1]}}[\exp\{\zeta_t^2/\sigma_t^2\}] \leq \exp\{1\}$ a.s., where $\sigma_t > 0$ are deterministic. Then*

$$\forall \Lambda \geq 0 : \text{Prob}\left\{\sum_{t=1}^{N} \zeta_t > \Lambda \sqrt{\sum_{t=1}^{N} \sigma_t^2}\right\} \leq \exp\{-\Lambda^2/3\}.$$

We are now ready to prove Theorem 4.2.1.

Proof of Theorem 4.2.1: Let $\bar{x}$ be an optimal solution of (4.1.34). Summing up (4.4.80) from $\tau = 1$ to t, we have

$$\begin{aligned}\sum_{\tau=1}^{t} [\gamma_\tau(\Psi(x_{\tau+1}) - \Psi^*)] &\leq V(x_1, \bar{x}) - V(x_{t+1}, \bar{x}) + \sum_{\tau=1}^{t} \Delta_\tau(\bar{x}) \\ &\leq V(x_1, \bar{x}) + \sum_{\tau=1}^{t} \Delta_\tau(\bar{x}) \leq D_{\omega,X}^2 + \sum_{\tau=1}^{t} \Delta_\tau(\bar{x}),\end{aligned}$$

where the last inequality follows from (2.2.30), which, in view of the fact

$$\Psi(x_{t+1}^{av}) \leq (\sum_{\tau=1}^{t} \gamma_\tau)^{-1} \sum_{\tau=1}^{t} \gamma_\tau \Psi(x_{\tau+1}),$$

then implies that

$$\left(\sum_{\tau=1}^{t} \gamma_\tau\right) \left[\Psi(x_{t+1}^{av}) - \Psi^*\right] \leq D_{\omega,X}^2 + \sum_{\tau=1}^{t} \Delta_\tau(\bar{x}). \tag{4.4.82}$$

Denoting $\zeta_\tau := \gamma_\tau \langle \delta_\tau, \bar{x} - x_\tau \rangle$ and observing that

$$\Delta_\tau(\bar{x}) = \zeta_\tau + \frac{(2\mathcal{M} + \|\delta_\tau\|_*)^2 \gamma_\tau^2}{2(\alpha - L\,\gamma_\tau)} \leq \zeta_\tau + \frac{\gamma_\tau^2}{\alpha - L\gamma_\tau} \left(4\mathcal{M}^2 + \|\delta_\tau\|_*^2\right),$$

we then conclude from (4.4.82) that

$$\begin{aligned}\left(\sum_{\tau=1}^{t} \gamma_\tau\right) \left[\Psi(x_{t+1}^{av}) - \Psi^*\right] &\leq D_{\omega,X}^2 + \sum_{\tau=1}^{t} \left[\zeta_\tau + \frac{\gamma_\tau^2}{\alpha - L\gamma_\tau} \left(4\mathcal{M}^2 + \|\delta_\tau\|_*^2\right)\right] \\ &\leq D_{\omega,X}^2 + \sum_{\tau=1}^{t} \left[\zeta_\tau + \frac{2\gamma_\tau^2}{\alpha} \left(4\mathcal{M}^2 + \|\delta_\tau\|_*^2\right)\right], \end{aligned} \tag{4.4.83}$$

where the last inequality follows from the assumption that $\gamma_t \leq \alpha/(2L)$.

Note that the pair (x_t, x_t^{av}) is a function of the history $\xi_{[t-1]} := (\xi_1, ..., \xi_{t-1})$ of the generated random process and hence is random. Taking expectations of both sides of (4.4.83) and noting that under assumption I, $\mathbb{E}[\|\delta_\tau\|_*^2] \leq \sigma^2$, and

$$\mathbb{E}_{|\xi_{[\tau-1]}}[\zeta_\tau] = 0, \tag{4.4.84}$$

we obtain

$$\left(\sum_{\tau=1}^{t} \gamma_\tau\right) \mathbb{E}\left[\Psi(x_{t+1}^{av}) - \Psi^*\right] \leq D_{\omega,X}^2 + \frac{2}{\alpha}(4\mathcal{M}^2 + \sigma^2) \sum_{\tau=1}^{t} \gamma_\tau^2,$$

which clearly implies part a).

We now show part b) holds. Clearly, by (4.4.84), $\{\zeta_\tau\}_{t\geq 1}$ is a martingale sequence. Moreover, it follows from (2.2.31) and (4.2.44) that

$$\mathbb{E}_{|\xi_{[\tau-1]}}\left[\exp\{\zeta_\tau^2/(2\gamma_\tau\Omega_{\omega,X}\sigma)^2\}\right] \leq \mathbb{E}_{|\xi_{[\tau-1]}}\left[\exp\{(2\gamma_\tau\Omega_{\omega,X}\|\delta_\tau\|_*)^2/(2\gamma_\tau\Omega_{\omega,X}\sigma)^2\}\right] \leq \exp(1),$$

The previous two observations, in view of Lemma 4.4.4, then imply that

$$\forall \Lambda \geq 0 : \text{Prob}\left\{\sum_{\tau=1}^{t} \zeta_\tau > 2\Lambda\Omega_{\omega,X}\sigma\sqrt{\sum_{\tau=1}^{t}\gamma_\tau^2}\right\} \leq \exp\{-\Lambda^2/3\}. \tag{4.4.85}$$

Now observe that under Assumption A.4.2,

$$\mathbb{E}_{|\xi_{\tau-1}}\left[\exp\{\|\delta_\tau\|_*^2/\sigma^2\}\right] \leq \exp\{1\}.$$

Setting $\theta_\tau = \gamma_\tau^2/\sum_{\tau=1}^{t}\gamma_\tau^2$, we have

$$\exp\left\{\sum_{\tau=1}^{t}\theta_t(\|\delta_\tau\|_*^2/\sigma^2)\right\} \leq \sum_{\tau=1}^{t}\theta_\tau\exp\{\|\delta_\tau\|_*^2/\sigma^2\},$$

whence, taking expectations,

$$\mathbb{E}\left[\exp\left\{\sum_{\tau=1}^{t}\gamma_\tau^2\|\delta_\tau\|_*^2/\left(\sigma^2\sum_{\tau=1}^{t}\gamma_\tau^2\right)\right\}\right] \leq \exp\{1\}.$$

It then follows from Markov's inequality that

$$\forall \Lambda \geq 0 : \text{Prob}\left\{\sum_{\tau=1}^{t}\gamma_\tau^2\|\delta_\tau\|_*^2 > (1+\Lambda)\sigma^2\sum_{\tau=1}^{t}\gamma_\tau^2\right\} \leq \exp\{-\Lambda\}. \tag{4.4.86}$$

Combining (4.4.83), (4.4.85), and (4.4.86), and rearranging the terms, we obtain (4.2.50).

■

4.4.2 Convergence analysis for the accelerated SA

The goal of this subsection is to prove Theorem 4.3.1.

In the sequel, with a little abuse of the notation, we use the following entity to denote the error for the computed subgradient at each iteration t of the AC-SA algorithm:

$$\delta_t := G(x_t^{md}, \xi_t) - g(x_t^{md}),$$

where $g'(x_t^{md}) = \mathbb{E}[G(x_t^{md}, \xi_t)] \in \partial\Psi(x_t^{md})$ under Assumption A.4.1.

The following lemma establishes an important recursion for the AC-SA algorithm.

Lemma 4.4.5 *Assume that the stepsizes β_τ and γ_τ satisfy $\beta_\tau \geq 1$ and $L\gamma_\tau < \alpha\beta_\tau$ for all $\tau \geq 1$. Let $(x_\tau, x_\tau^{ag}) \in X \times X$ be given and set $x_\tau^{md} \equiv \beta_\tau^{-1} x_\tau + (1 - \beta_\tau^{-1}) x_\tau^{ag}$. Also let $(x_{\tau+1}, x_{\tau+1}^{ag}) \in X \times X$ be a pair computed according to (4.3.54) and (4.3.55). Then, for every $x \in X$, we have*

$$\beta_\tau\gamma_\tau[\Psi(x_{\tau+1}^{ag}) - \Psi(x)] + V(x_{\tau+1}, x) \leq (\beta_\tau - 1)\gamma_\tau[\Psi(x_\tau^{ag}) - \Psi(x)] + V(x_\tau, x) + \hat{\Delta}_\tau,$$

where

$$\hat{\Delta}_\tau = \hat{\Delta}_\tau(x) \quad := \quad \gamma_\tau\langle\delta_\tau, x - x_\tau\rangle + \frac{(2\mathcal{M} + \|\delta\|_*)^2\beta_\tau\gamma_\tau^2}{2(\alpha\beta_\tau - L\gamma_\tau)}. \tag{4.4.87}$$

Proof. Denoting $d_\tau := x_{\tau+1} - x_\tau$, it can be easily seen that

$$x_{\tau+1}^{ag} - x_\tau^{md} = \beta_\tau^{-1} x_{\tau+1} + (1 - \beta_\tau^{-1}) x_\tau^{ag} - x_\tau^{md} = \beta_\tau^{-1}(x_{\tau+1} - x_\tau) = \beta_\tau^{-1} d_\tau.$$

The above observation together with (4.4.78) and the relation $\alpha\|d_\tau\|^2/2 \leq V(x_\tau, x_{\tau+1})$ then imply that

$$\begin{aligned}
&\beta_\tau\gamma_\tau\Psi(x_{\tau+1}^{ag}) \leq \beta_\tau\gamma_\tau[\Psi(x_\tau^{md}) + \langle g(x_\tau^{md}), x_{\tau+1}^{ag} - x_\tau^{md}\rangle + \tfrac{L}{2}\|x_{\tau+1}^{ag} - x_\tau^{md}\|^2 + 2\mathcal{M}\|x_{\tau+1}^{ag} - x_\tau^{md}\|] \\
&= \beta_\tau\gamma_\tau[\Psi(x_\tau^{md}) + \langle g(x_\tau^{md}), x_{\tau+1}^{ag} - x_\tau^{md}\rangle] + \tfrac{L\gamma_\tau}{2\beta_\tau}\|d_\tau\|^2 + 2\mathcal{M}\gamma_\tau\|d_\tau\| \\
&\leq \beta_\tau\gamma_\tau[\Psi(x_\tau^{md}) + \langle g(x_\tau^{md}), x_{\tau+1}^{ag} - x_\tau^{md}\rangle] + V(x_\tau, x_{\tau+1}) - \tfrac{\alpha\beta_\tau - L\gamma_\tau}{2\beta_\tau}\|d_\tau\|^2 + 2\mathcal{M}\gamma_\tau\|d_\tau\|.
\end{aligned}$$

Noting that

$$\beta_\tau\gamma_\tau[\Psi(x_\tau^{md}) + \langle g(x_\tau^{md}), x_{\tau+1}^{ag} - x_\tau^{md}\rangle] = \beta_\tau\gamma_\tau[\Psi(x_\tau^{md}) + \langle g(x_\tau^{md}), (1-\beta_\tau^{-1})x_\tau^{ag} + \beta_\tau^{-1}x_{\tau+1} - x_\tau^{md}\rangle]$$

$$= (\beta_\tau - 1)\gamma_\tau[\Psi(x_\tau^{md}) + \langle g(x_\tau^{md}), x_\tau^{ag} - x_\tau^{md}\rangle] + \gamma_\tau[\Psi(x_\tau^{md}) + \langle g(x_\tau^{md}), x_{\tau+1} - x_\tau^{md}\rangle]$$

$$\le (\beta_\tau - 1)\gamma_\tau\Psi(x_\tau^{ag}) + \gamma_\tau[\Psi(x_\tau^{md}) + \langle g(x_\tau^{md}), x_{\tau+1} - x_\tau^{md}\rangle]$$

$$= (\beta_\tau - 1)\gamma_\tau\Psi(x_\tau^{ag}) + \gamma_\tau[\Psi(x_\tau^{md}) + \langle G(x_\tau^{md}, \xi_\tau), x_{\tau+1} - x_\tau^{md}\rangle - \langle \delta_\tau, x_{\tau+1} - x_\tau^{md}\rangle]$$

$$= (\beta_\tau - 1)\gamma_\tau\Psi(x_\tau^{ag}) + \gamma_\tau[\Psi(x_\tau^{md}) + \langle G(x_\tau^{md}, \xi_\tau), x_{\tau+1} - x_\tau^{md}\rangle - \langle \delta_\tau, x_\tau - x_\tau^{md}\rangle - \langle \delta_\tau, d_\tau\rangle]$$

$$\le (\beta_\tau - 1)\gamma_\tau\Psi(x_\tau^{ag}) + \gamma_\tau[\Psi(x_\tau^{md}) + \langle G(x_\tau^{md}, \xi_\tau), x_{\tau+1} - x_\tau^{md}\rangle - \langle \delta_\tau, x_\tau - x_\tau^{md}\rangle + \|\delta_\tau\|_*\|d_\tau\|],$$

we conclude from the previous conclusion that

$$\beta_\tau\gamma_\tau\Psi(x_{\tau+1}^{ag}) \le (\beta_\tau - 1)\gamma_\tau\Psi(x_\tau^{ag}) + \gamma_\tau[\Psi(x_\tau^{md}) + \langle G(x_\tau^{md}, \xi_\tau), x_{\tau+1} - x_\tau^{md}\rangle] + V(x_\tau, x_{\tau+1})$$

$$-\gamma_\tau\langle \delta_\tau, x_\tau - x_\tau^{md}\rangle - \tfrac{\alpha\beta_\tau - L\gamma_\tau}{2\beta_\tau}\|d_\tau\|^2 + (2\mathcal{M} + \|\delta\|_*)\gamma_\tau\|d_\tau\|$$

$$\le (\beta_\tau - 1)\gamma_\tau\Psi(x_\tau^{ag}) + \gamma_\tau[\Psi(x_\tau^{md}) + \langle G(x_\tau^{md}, \xi_\tau), x_{\tau+1} - x_\tau^{md}\rangle] + V(x_\tau, x_{\tau+1})$$

$$-\gamma_\tau\langle \delta_\tau, x_\tau - x_\tau^{md}\rangle + \tfrac{(2\mathcal{M}+\|\delta\|_*)^2\beta_\tau\gamma_\tau^2}{2(\alpha\beta_\tau - L\gamma_\tau)},$$

where the last inequality follows from (4.4.79) with $u = \|d_\tau\|$, $b = (2\mathcal{M} + \|\delta\|_*)\gamma_\tau$, and $a = (\alpha\beta_\tau - L\gamma_\tau)/\beta_\tau$.

Moreover, it follows from the identity (4.3.54), (2.2.28), and Lemma 4.4.1 with $\tilde{x} = x_\tau$, $\hat{u} = x_{\tau+1}$, and $p(\cdot) \equiv \gamma_\tau\langle G(x_\tau^{md}, \xi_\tau), \cdot - x_\tau^{md}\rangle$ that

$$\begin{aligned}
&\gamma_\tau\Psi(x_\tau^{md}) + [\gamma_\tau\langle G(x_\tau^{md}, \xi_\tau), x_{\tau+1} - x_\tau^{md}\rangle + V(x_\tau, x_{\tau+1})] \\
&\quad\le\ \gamma_\tau\Psi(x_\tau^{md}) + [\gamma_\tau\langle G(x_\tau^{md}, \xi_\tau), x - x_\tau^{md}\rangle + V(x_\tau, x) - V(x_{\tau+1}, x)] \\
&\quad=\ \gamma_\tau[\Psi(x_\tau^{md}) + \langle g(x_\tau^{md}), x - x_\tau^{md}\rangle] + \gamma_\tau\langle \delta_\tau, x - x_\tau^{md}\rangle + V(x_\tau, x) - V(x_{\tau+1}, x) \\
&\quad\le\ \gamma_\tau\Psi(x) + \gamma_\tau\langle \delta_\tau, x - x_\tau^{md}\rangle + V(x_\tau, x) - V(x_{\tau+1}, x),
\end{aligned}$$

where the last inequality follows from the convexity of $\Psi(\cdot)$ and the fact $g(x_\tau^{md}) \in \partial\Psi(x_\tau^{md})$.

Combining the previous two conclusions, we obtain

$$\beta_\tau\gamma_\tau\Psi(x_{\tau+1}^{ag}) \le (\beta_\tau - 1)\gamma_\tau\Psi(x_\tau^{ag}) + \gamma_\tau\Psi(x) + V(x_\tau, x) - V(x_{\tau+1}, x) +$$
$$\gamma_\tau\langle \delta_\tau, x - x_\tau\rangle + \tfrac{(2\mathcal{M}+\|\delta\|_*)^2\beta_\tau\gamma_\tau^2}{2(\alpha\beta_\tau - L\gamma_\tau)}$$

Our claim immediately follows from the above inequality by subtracting $\beta_\tau\gamma_\tau\Psi(x)$ from both sides and rearranging the terms. ∎

We are now ready to prove Theorem 4.3.1.

Proof of Theorem 4.3.1: Let $\bar{x}$ be an optimal solution of (4.1.34). It follows from the fact that $\Psi(x) \geq \Psi(\bar{x}) = \Psi^*$, $\forall x \in X$, the fact $\beta_\tau \geq 1$, (4.3.56), and Lemma 4.4.5 with $x = \bar{x}$ that, for any $t \geq 1$,

$$\begin{aligned}(\beta_{t+1}-1)\gamma_{t+1}[\Psi(x_{t+1}^{ag})-\Psi^*] &\leq \beta_t\gamma_t[\Psi(x_{t+1}^{ag})-\Psi^*] \\ &\leq (\beta_t-1)\gamma_t[\Psi(x_t^{ag})-\Psi^*]+V(x_t,\bar{x})-V(x_{t+1},\bar{x})+\hat{\Delta}_t(\bar{x}),\end{aligned}$$

from which it follows inductively that

$$\begin{aligned}(\beta_{t+1}-1)\gamma_{t+1}[\Psi(x_{t+1}^{ag})-\Psi^*] &\leq (\beta_1-1)\gamma_1[\Psi(x_1^{ag})-\Psi^*]+V(x_1,\bar{x})-V(x_{t+1},\bar{x})+\sum_{\tau=1}^{t}\hat{\Delta}_\tau(\bar{x}) \\ &= V(x_1,\bar{x})-V(x_{t+1},\bar{x})+\sum_{\tau=1}^{t}\hat{\Delta}_\tau(\bar{x}) \leq D_{\omega,X}^2+\sum_{\tau=1}^{t}\hat{\Delta}_\tau(\bar{x}),\end{aligned}$$

where the first equality follows from the assumption $\beta_1 = 1$ and the last inequality follows from (2.2.30) and the fact $V(x_{t+1},\bar{x}) \geq 0$.

Denoting $\zeta_\tau := \gamma_\tau\langle\delta_\tau, \bar{x}-x_\tau\rangle$ and observing that

$$\begin{aligned}\hat{\Delta}_\tau(\bar{x}) &= \zeta_\tau+\frac{(2\mathcal{M}+\|\delta\|_*)^2\beta_\tau\gamma_\tau^2}{2(\alpha\beta_\tau-L\gamma_\tau)} \leq \zeta_\tau+\frac{\beta_\tau\gamma_\tau^2}{\alpha\beta_\tau-L\gamma_\tau}(4\mathcal{M}^2+\|\delta_\tau\|_*^2) \\ &\leq \zeta_\tau+\frac{2}{\alpha}(4\mathcal{M}^2+\|\delta_\tau\|_*^2)\gamma_\tau^2,\end{aligned}$$

where the last inequality follows from (4.3.56), we then conclude from the previous observation that

$$(\beta_{t+1}-1)\gamma_{t+1}[\Psi(x_{t+1}^{ag})-\Psi(\bar{x})] \leq D_{\omega,X}^2+\sum_{\tau=1}^{t}\left[\zeta_\tau+\frac{2}{\alpha}(4\mathcal{M}^2+\|\delta_\tau\|_*^2)\gamma_\tau^2\right]. \qquad (4.4.88)$$

Note that the triple $(x_t, x_t^{ag}, x_t^{md})$ is a function of the history $\xi_{[t-1]} := (\xi_1, ..., \xi_{t-1})$ of the generated random process and hence is random. Taking expectations of both sides of (4.4.88) and noting that under assumption I, $\mathbb{E}[\|\delta_\tau\|_*^2] \leq \sigma^2$ and $\mathbb{E}_{|\xi_{[\tau-1]}}[\zeta_\tau] = 0$, we obtain

$$(\beta_{t+1}-1)\gamma_{t+1}\mathbb{E}[\Psi(x_{t+1}^{ag})-\Psi^*] \leq D_{\omega,X}^2+\frac{2}{\alpha}(4\mathcal{M}^2+\sigma^2)\sum_{\tau=1}^{t}\gamma_\tau^2,$$

which clearly implies part a).

The proof of part b) is similar to the one of Theorem 4.2.1.b), and hence the details are skipped. ∎

4.4.3 Convergence analysis for quadratic penalty method

The goal of this subsection is to prove Theorem 4.3.2.

Lemma 4.4.6 *If $\tilde{x} \in X$ is an approximate solution of (4.3.69) satisfying*

$$\tilde{\Psi}_\rho(\tilde{x}) - \tilde{\Psi}^* \leq \delta, \tag{4.4.89}$$

then

$$\|\mathcal{A}\tilde{x} - b\| \leq \frac{2}{\rho}\|y^*\| + \sqrt{\frac{2\delta}{\rho}} \tag{4.4.90}$$

$$\tilde{h}(\tilde{x}) - \tilde{h}^* \leq \delta, \tag{4.4.91}$$

where y^* is an arbitrary Lagrange multiplier associated with (4.3.66).

Proof. Let $v(u) := \inf\{\tilde{h}(x) : \mathcal{A}x - b = u,\ x \in X\}$ be the value function associated with (4.3.66). It is well-known that our assumptions imply that v is a convex function such that $-y^* \in \partial v(0)$. Hence,

$$v(u) - v(0) \geq (-y^*)^T u,\ \forall u \in \Re^m.$$

Letting $u := \mathcal{A}\tilde{x} - b$, we conclude from the above observation, the facts that $v(u) \leq \tilde{h}(\tilde{x})$ and $v(0) \geq \tilde{\Psi}^*$, and assumption (4.4.89), that

$$\begin{aligned} -\|y^*\|\|u\| + \rho\|u\|^2/2 &\leq (-y^*)^T u + \rho\|u\|^2/2 \\ &\leq v(u) - v(0) + \rho\|u\|^2/2 \leq \tilde{h}(\tilde{x}) + \rho\|u\|^2/2 - v(0) \\ &\leq \tilde{h}(\tilde{x}) + \rho\|u\|^2/2 - \phi^* = \tilde{\Psi}_\rho(\tilde{x}) - \tilde{\Psi}^* \leq \delta, \end{aligned}$$

which clearly implies (4.4.90). Moreover, the fact that $\tilde{h}^* = v(0) \geq \tilde{\Psi}^*$ implies that

$$\tilde{h}(\tilde{x}) - \tilde{h}^* \leq \tilde{h}(\tilde{x}) + \rho\|u\|^2/2 - \tilde{\Psi}^* = \tilde{\Psi}_\rho(\tilde{x}) - \tilde{\Psi}^* \leq \delta.$$

∎

We are now ready to prove Theorem 4.3.2.

Proof of Theorem 4.3.2: Let $\tilde{x} \in X$ satisfies (4.4.89) with $\delta = \epsilon_o$. Let $\rho_* := \rho(\|y^*\|)$ and observe that $\rho_* \leq \rho(t)$ for every $t \geq \|y^*\|$. It follows from the previous observation and Lemma 4.4.6 that $\tilde{h}(\tilde{x}) - \tilde{h}^* \leq \epsilon_o$ and

$$\begin{aligned}\|A\tilde{x} - b\| &\leq \frac{2}{\rho(t)}\|y^*\| + \sqrt{\frac{2\epsilon_o}{\rho(t)}} \leq \frac{2}{\rho_*}\|y^*\| + \sqrt{\frac{2\epsilon_o}{\rho_*}} = \frac{1}{\sqrt{\rho_*}}\left(\frac{2\sqrt{2}\epsilon_p\|y^*\|}{\sqrt{\epsilon_o + 4\epsilon_p\|y^*\|} + \sqrt{\epsilon_o}} + \sqrt{2\epsilon_o}\right) \\ &= \frac{1}{\sqrt{\rho_*}}\left(\frac{\sqrt{\epsilon_o + 4\epsilon_p\|y^*\|} - \sqrt{\epsilon_o}}{\sqrt{2}} + \sqrt{2\epsilon_o}\right) = \frac{\sqrt{\epsilon_o} + \sqrt{\epsilon_o + 4\epsilon_p\|y^*\|}}{\sqrt{2\rho_*}} = \epsilon_p,\end{aligned}$$

and hence that $\tilde{x}$ is an (ϵ_p, ϵ_o)-primal solution of (4.3.66).

Clearly, by (4.3.70), we have $L = \rho\|\mathcal{A}\|^2$. Observe that the gradient for the smooth component $\tilde{f}_\rho$ in $\tilde{\Psi}_\rho$ (see (4.3.69)) can be computed exactly and hence that the error of approximating the subgradient of $\tilde{\Psi}_\rho$ exists only in the non-smooth component $\tilde{h}$. For any given point $x \in X$, let $\mathsf{G}(x, \xi_t)$ be the output from the stochastic oracle of $\tilde{h}$ and $\tilde{h}'(x) = \mathbb{E}[\mathsf{G}(x, \xi_t)]$. It follows from (4.1.42), Jensen's inequality, and Assumption III.c) that

$$\begin{aligned}&\mathbb{E}\left[\exp\left\{\|\mathsf{G}(x,\xi) - \tilde{h}'(x)\|_*^2/(4M^2)\right\}\right] \leq \mathbb{E}\left[\exp\left\{\left(2\|\mathsf{G}(x,\xi)\|_*^2 + 2\|\tilde{h}'(x)\|_*^2\right)/(4M^2)\right\}\right] \\ &\leq \mathbb{E}\left[\exp\left\{\left(2\|\mathsf{G}(x,\xi)\|_*^2 + 2M^2\right)/(4M^2)\right\}\right] \leq \exp(1/2)\,\mathbb{E}\left[\exp\left\{\|\mathsf{G}(x,\xi)\|_*^2/(2M^2)\right\}\right] \\ &\leq \exp(1/2)\left(\mathbb{E}\left[\exp\left\{\|\mathsf{G}(x,\xi)\|_*^2/M^2\right\}\right]\right)^{\frac{1}{2}} \leq \exp(1),\end{aligned}$$

which then implies Assumption II holds with $Q = 2M$. The previous observations together with (4.2.51) and (4.2.52) then imply that

$$K_0^*(N_{md}) + \lambda K_1^*(N_{md}) = \frac{\rho\|\mathcal{A}\|^2\Omega^2}{N_{md}} + \frac{4\sqrt{2} + 3\lambda}{\sqrt{N_{md}}}\Omega M \leq \frac{\epsilon_o}{2} + \frac{\epsilon_o}{2} \leq \epsilon_o.$$

The previous conclusion, in view of the definition of λ and (4.2.52), clearly imply the claim in part a). Part b) follows similarly from (4.3.62) and the definition of λ, by noting that

$$\begin{aligned}\hat{K}_0^*(N_{ac}) + \lambda\hat{K}_1^*(N_{ac}) &= \frac{4\rho\|\mathcal{A}\|^2\Omega^2}{N_{ac}(N_{ac}+2)} + \frac{8\sqrt{2} + 20\lambda}{\sqrt{N_{ac}}}\Omega M \leq \frac{4\rho\|\mathcal{A}\|^2\Omega^2}{N_{ac}^2} + \frac{8\sqrt{2} + 20\lambda}{\sqrt{N_{ac}}}\Omega M \\ &\leq \frac{\epsilon_o}{2} + \frac{\epsilon_o}{2} \leq \epsilon_o.\end{aligned}$$

■

4.5 Conclusions of this chapter

In this chapter, we consider an important class of convex programming problems whose objective function Ψ is given by the summation of a smooth and non-smooth component. Further, it is assumed that the only information available for the numerical scheme to solve these problems is the subgradients of Ψ contaminated by stochastic noise. Our contribution mainly consists of the following aspects. Firstly, with a novel analysis, it is demonstrated that the simple robust mirror-descent stochastic approximation method applied to the aforementioned problems exhibits the best known so far rate of convergence guaranteed by a more involved stochastic mirror-prox algorithm. Moreover, by incorporating some ideas of the optimal method for smooth minimization, we propose an accelerated scheme, which can achieve, uniformly in dimension, the theoretically optimal rate of convergence for solving this class of problems. Finally, the significant advantages of the accelerated scheme over the existing algorithms are illustrated in the context of solving a class of stochastic programming problems whose feasible region is a simple compact convex set intersected with an affine manifold.

CHAPTER V

FIRST-ORDER AUGMENTED LAGRANGIAN METHODS

5.1 Overview

In Chapters 2, 3 and 4, we focus on convex optimization under a stochastic oracle ($\mathcal{SO}$). More specifically, we study stochastic convex optimizaiton techniques which work with stochastic subgradients of the objective functions of (1.2.1) and (4.1.34) acquired through subsequent calls to the stochastic oracles. In this chapter, we investigate certain interesting deterministic optimizaiton technique, namely, the augmented Lagrangina method, which operates on first-order information of the augmented dual problem. We consider the situation where to obtain the exact first-order information of the dual is time-consuming, i.e., requiring to solve another complicated subproblem, and hence only approximate first-order information is available in reality.

The basic problem of interest in this chapter is the convex programming problem (1.3.1). For the reader's convenience, we re-state this problem as follows.

$$f^* := \min\{f(x) : \mathcal{A}(x) = 0,\, x \in X\}, \tag{5.1.1}$$

where $f : X \to \mathbf{R}$ is a convex function with Lipschitz continuous gradient, $X \subseteq \Re^n$ is a sufficiently simple compact convex set and $\mathcal{A} : \Re^n \to \Re^m$ is an affine function.

For the case where the feasible region consists only of the set X, or equivalently $\mathcal{A} \equiv 0$, Nesterov ([47, 50]) developed methods which can find a point $x \in X$ such that $f(x) - f^* \leq \epsilon$ in at most $\mathcal{O}(\epsilon^{-1/2})$ iterations (see Subsection 1.1.2). Moreover, each iteration of his method requires one gradient evaluation of f and computation of two projections onto X. It is shown that his method achieves, uniformly in the dimension, the lower bound on the number of iterations for minimizing convex functions with Lipschitz continuous gradient over a closed convex set. When $\mathcal{A}$ is not identically 0, Nesterov's optimal method can still be applied

directly to problem (5.1.1) but this approach would require the computation of projections onto the feasible region $X \cap \{x : \mathcal{A}(x) = 0\}$, which for most practical problems is as expensive as solving the original problem itself. An alternative approach for solving (5.1.1) when $\mathcal{A}$ is not identically 0 is to use first-order methods whose iterations require only computation of projections onto the simple set X.

Following this line of investigation, Lan and Monteiro [33] studied two first-order methods for solving (5.1.1) based on two well-known penalization approaches, namely: the quadratic and the exact penalization approaches. Iteration-complexity bounds for these methods are then derived to obtain two types of near optimal solutions of (5.1.1), namely: near primal and near primal-dual optimal solutions. Variants with possibly better iteration-complexity bounds than the aforementioned methods are also discussed. In this work, we still consider another first-order approach for solving (5.1.1) based on the classical augmented Lagrangian approach, where the subproblems are approximately solved by means of Nesterov's optimal method. As a by-product, alternative first-order methods for solving (5.1.1) involving only computation of projections onto the simple set X are obtained.

The augmented Lagrangian method applied to problem (5.1.1) consists of solving a sequence of subproblems of the form

$$d_\rho(\lambda_k) := \min_{x \in X} \left\{ \mathcal{L}_\rho(x, \lambda_k) := f(x) + \langle \lambda_k, \mathcal{A}(x) \rangle + \frac{\rho}{2} \|\mathcal{A}(x)\|^2 \right\}, \tag{5.1.2}$$

where $\rho > 0$ is a given penalty parameter and $\|\cdot\|$ is the norm associated with a given inner product $\langle \cdot, \cdot \rangle$ in $\Re^m$. The multiplier sequence $\{\lambda_k\}$ is generated according to the iterations

$$\lambda_{k+1} = \lambda_k + \rho \mathcal{A}(x_k^*), \tag{5.1.3}$$

where x_k^* is a solution of problem (5.1.2). Since in most cases (5.1.2) can only be solved approximately, x_k^* in (5.1.3) is replaced by an η_k-approximate solution of (5.1.2), i.e., a point $x_k \in X$ such that $\mathcal{L}_\rho(x, \lambda_k) - d_\rho(\lambda_k) \leq \eta_k$. The inexact augmented Lagrangian method obtained in this manner, where the subproblems (5.1.2) are solved by Nesterov's method, is the main focus of our investigation in this chapter. More specifically, we are interested in establishing a bound on the total number of Nesterov's optimal iterations, i.e., the inner iterations, performed throughout the entire inexact AL method.

Several technical issues are addressed in the aforementioned iteration-complexity analysis of the inexact AL method. First, the notion of a near primal-dual optimal solution is introduced and used as a termination criterion by the methods proposed in this chapter. Second, it is well-known that $\mathcal{A}(x_k^*)$ is exactly the gradient of the function d_ρ defined in (5.1.2) at λ_k, and hence that (5.1.3) can be viewed as a steepest ascent iteration with stepsize ρ applied to the function d_ρ. Since, in the inexact AL method, we approximate $d_\rho(\lambda_k) = \mathcal{A}(x_k^*)$ by $\mathcal{A}(x_k)$, where x_k is an approximate solution of (5.1.2), we bound the error of the gradient approximation $\mathcal{A}(x_k)$, namely $\|\mathcal{A}(x_k) - \mathcal{A}(x_k^*)\|$, in terms of the accuracy η_k of the approximate solution x_k, and use this result to derive sufficient conditions on the sequence $\{\eta_k\}$ which guarantee that the corresponding inexact steepest ascent method $\lambda_{k+1} = \lambda_k + \rho\mathcal{A}(x_k)$ has the same rate of convergence as the exact one. Third, as ρ increases, it is well-known that the iteration-complexity of approximately solving each subproblem (5.1.2) increases, while the number of dual iterations (5.1.3), i.e., the outer iterations, decreases. Ways of choosing the parameter ρ so as to balance these two opposing criterions are then proposed. More specifically, ρ is chosen so as to minimize the overall number of inner iterations performed by the inexact AL method.

It turns out that proper selection of the tolerances η_k and the optimal penalty parameter ρ requires knowledge of an upper bound t on $D_\Lambda := \inf_{\lambda \in \Lambda^*} \|\lambda_0 - \lambda^*\|$, where Λ^* is the set of Lagrange multipliers associated with the constraint $\mathcal{A}(x) = 0$. Theoretically, choosing the upper bound t so that $t = \mathcal{O}(D_\Lambda)$ yields the lowest provably iteration-complexity bounds obtained by our analysis. However, since D_Λ is not known a priori, we present a "guess-and-check" procedure which consists of guessing a sequence of estimates for D_Λ and applying the corresponding sequence of inexact AL methods (with pre-specified number of outer-iterations) to (5.1.1) until a near primal-dual solution is eventually obtained. It is shown that the above guess-and-check procedure has the same iteration-complexity as the (ideal) inexact AL method for which the exact value of D_Λ is known in advance. Finally, we present variants with better iteration-complexity bounds than the original inexact AL method and guess-and-check procedure, which consist of directly applying the original approaches to a perturbed problem obtained by adding a strongly convex component to the objective

function of (5.1.1).

This chapter is organized as follows. In Section 5.2, we describe two inexact AL methods and corresponding guess-and-check procedures for solving (5.1.1) and state without proof their iteration-complexity results. More specifically, we discuss the primal-dual termination criterion used in the complexity analysis of the aforementioned methods in Subsection 5.2.1. Results about the augmented dual function, including a key result about how to approximate its gradient, are discussed in Subsection 5.2.2. In Subsection 5.2.3, we describe the first inexact AL method and its corresponding guess-and-check procedure, and present their iteration-complexity results. The second inexact AL method and its corresponding guess-and-check procedure based on applying the above methods to a perturbed problem, obtained by adding a strongly convex component to the objective function of the CP problem (5.1.1), are discussed in Subsection 5.2.4. All technical results of this chapter, which can be skipped by readers interested in the main results only, are presented in Sections 5.3 and 5.4. More specifically, we present some technical results about the projected gradient in Subsection 5.3.1 and about the convergence behavior of the sequence $\{\lambda_k\}$ in Subsection 5.3.2. Subsections 5.4.1 and 5.4.1 give the proofs of the main results in Subsections 5.2.3 and 5.2.4, respectively. In Section 5.5, we compare the results obtained in this chapter for the inexact AL methods with another possible approach for solving variational inequalities (VI) studied in Nemirovski ([42]) for bounded sets, and Monteiro and Svaiter ([41]) for unbounded sets. Finally, we make some concluding remarks in Section 5.6.

5.1.1 Notation and terminology

We denote the set of real numbers by $\mathbf{R}$. Also, $\mathbf{R}_+$ and $\mathbf{R}_{++}$ denote the set of nonnegative and positive real numbers, respectively. In this chapter, we use the notation $\Re^p$ to denote a p-dimensional vector space inherited with a inner product space $\langle \cdot, \cdot \rangle$ and use $\| \cdot \|$ to denote the inner product norm in $\Re^p$, i.e., $\| \cdot \| = \langle \cdot, \cdot \rangle^{1/2}$. Moreover, we define the projection map onto a given closed convex set $\mathcal{C} \in \Re^p$ by

$$\Pi_{\mathcal{C}}(u) := \operatorname{argmin}\{\|u - c\| : c \in \mathcal{C}\}, \quad \forall u \in \Re^p.$$

A function $f : \mathcal{C} \subseteq \Re^p \to \mathbf{R}$ is said to have L-Lipschitz-continuous gradient with respect

to $\|\cdot\|$ if it is differentiable and

$$\|\nabla f(\tilde{u}) - \nabla f(u)\| \leq L\|\tilde{u} - u\|, \quad \forall u, \tilde{u} \in \mathcal{C}. \tag{5.1.4}$$

It is well-known (see Theorem 2.1.5 of [49]) that, for every $u, \tilde{u} \in \mathcal{C}$, we have:

$$\frac{1}{2L}\|\nabla f(\tilde{u}) - \nabla f(u)\|^2 \;\leq\; f(\tilde{u}) - f(u) - \langle \nabla f(u), (\tilde{u} - u)\rangle \;\leq\; \frac{L}{2}\|\tilde{u} - u\|^2, \tag{5.1.5}$$

$$\frac{1}{L}\|\nabla f(\tilde{u}) - \nabla f(u)\|^2 \;\leq\; \langle \nabla f(\tilde{u}) - \nabla f(u), \tilde{u} - u\rangle \;\leq\; L\|\tilde{u} - u\|^2. \tag{5.1.6}$$

5.2 The algorithms and main results

In this section, we present the augmented Lagrangian method applied to (5.1.1) and discuss its computational complexity. Specifically, we discuss the termination criterion for this method in Subsection 5.2.1. We review the augmented dual function and discuss some of its properties in Subsection 5.2.2. In Subsection 5.2.3, we describe a version of the augmented Lagrangian method and discuss its computational complexity. A variant of this method, for which a perturbation term is added into the objective function of (5.1.1), is discussed and analyzed in Subsection 5.2.4.

5.2.1 Termination criterion

The problem of interest in this chapter is the CP problem (5.1.1) where $f : X \to \mathbf{R}$ is a convex function with L_f-Lipschitz-continuous gradient. The Lagrangian dual function and value function associated with (5.1.1) are defined as

$$d(\lambda) \;:=\; \inf\{f(x) + \langle \lambda, \mathcal{A}(x)\rangle : x \in X\}, \quad \forall \lambda \in \Re^m, \tag{5.2.1}$$

$$v(u) \;:=\; \inf\{f(x) : \mathcal{A}(x) = u,\, x \in X\}, \quad \forall u \in \Re^m. \tag{5.2.2}$$

It is well-known that d is always a concave function. Moreover, the assumption we made earlier that f is convex, $\mathcal{A}$ is affine, and X is convex, implies that the function v is convex.

The Lagrangian dual of (5.1.1) is the problem

$$d^* := \sup_{\lambda} d(\lambda). \tag{5.2.3}$$

In addition to the convexity assumptions we made about the data of (5.1.1), we also assume the following conditions throughout the chapter:

A.5.1 The function $v(\cdot)$ is closed and $f^* = v(0)$ is finite.

A.5.2 The set Λ^* of optimal solutions of the dual problem (5.2.3) is nonempty.

It is well-known that $d^* = \overline{\text{co}}\, v(0)$, where $\overline{\text{co}}\, v$ is the closed convex hull of v. Hence, Assumption A.5.1 implies that $f^* = v(0) = \overline{\text{co}}\, v(0) = d^*$, i.e., there is no duality gap for the pair of dual problems (5.1.1) and (5.2.3). Clearly, this implies that $\Lambda^* := \{\lambda^* : d(\lambda^*) = f^*\}$, i.e., Λ^* is the set of Lagrange multipliers. Moreover, it is well-known that latter set is also equal to $-\partial v(0)$. It then follows from Assumption A.5.2 that v is subdifferentiable at 0 and hence that v is proper.

The following result gives a sufficient condition for Assumption A.5.1 and its proof can be found in Appendix A.

Proposition 5.2.1 *If the set of optimal solutions for problem (5.1.1) is nonempty and bounded then Assumption A.5.1 holds.*

As a consequence of Proposition 5.2.1, if X is nonempty and compact, then Assumption A.5.1 holds.

In this chapter, we are interested in obtaining the near-optimal solutions of (5.1.1) defined as follows. Note that $x^* \in X$ is an optimal solution of (5.1.1) and $\lambda^* \in \Re^m$ is a Lagrange multiplier for (5.1.1) if, and only if, $(\tilde{x}, \tilde{\lambda}) = (x^*, \lambda^*)$ satisfies

$$\begin{aligned} &\mathcal{A}(\tilde{x}) = 0, \\ &\nabla f(\tilde{x}) + (\mathcal{A}_0)^* \tilde{\lambda} \in -\mathcal{N}_X(\tilde{x}), \end{aligned} \tag{5.2.4}$$

where $\mathcal{N}_X(\tilde{x}) := \{s \in \Re^n : \langle s, x - \tilde{x}\rangle \leq 0,\ \forall x \in X\}$ denotes the normal cone of X at $\tilde{x}$, and $\mathcal{A}_0$ denotes the linear part of $\mathcal{A}$ defined by $\mathcal{A}_0 := \mathcal{A} - \mathcal{A}(0)$. Based on this observation, we introduce the following notion.

Definition 5.2.1 *For a given pair $(\epsilon_p, \epsilon_d) \in \mathbf{R}_{++} \times \mathbf{R}_{++}$, $(\tilde{x}, \tilde{\lambda}) \in X \times \Re^m$ is called an (ϵ_p, ϵ_d)-primal-dual solution of (5.1.1) if*

$$\|\mathcal{A}(x)\|_* \leq \epsilon_p, \tag{5.2.5}$$

$$\nabla f(\tilde{x}) + (\mathcal{A}_0)^* \tilde{\lambda} \in -\mathcal{N}_X(\tilde{x}) + \mathcal{B}(\epsilon_d), \tag{5.2.6}$$

where $\mathcal{B}(\eta) := \{x \in \Re^n : \|x\| \leq \eta\}$ *for every* $\eta \geq 0$.

The main goal of this chapter is to study the iteration-complexity of the augmented Lagrangian method for computing an (ϵ_p, ϵ_d)-*primal-dual* solution of (5.1.1) defined above.

5.2.2 The augmented dual function

In this subsection, we review the definition of the augmented dual function associated with (5.1.1) and discuss some of its properties.

Given a penalty parameter $\rho > 0$, the augmented dual function $d_\rho : \Re^m \to \mathbf{R}$ associated with (5.1.1) is given by

$$d_\rho(\lambda) := \inf_{x \in X} \left\{ \mathcal{L}_\rho(x, \lambda) := f(x) + \langle \lambda, \mathcal{A}(x) \rangle + \frac{\rho}{2} \|\mathcal{A}(x)\|^2 \right\}, \tag{5.2.7}$$

and the augmented dual with parameter ρ is defined as

$$\sup_{\lambda \in \Re^m} d_\rho(\lambda). \tag{5.2.8}$$

An alternative characterization for the augmented dual function is given by

$$d_\rho(\lambda) = \inf_u \left\{ v_\rho(u, \lambda) := v(u) + \langle \lambda, u \rangle + \frac{\rho}{2} \|u\|^2 \right\}, \tag{5.2.9}$$

where $v(\cdot)$ is the value function given by (5.2.2).

Lemma 5.2.1 *The following statements hold:*

a) problem (5.2.9) has an unique optimal solution u_λ^**;*

b) the (possibly empty) set of optimal solutions of (5.2.7) X_λ^* *is given by*

$$X_\lambda^* = \{x \in X : \mathcal{A}(x) = u_\lambda^* \, and \, f(x) = v(u_\lambda^*)\}; \tag{5.2.10}$$

c) for any $\lambda \in \Re^m$ *and* $\rho > 0$*, we have*

$$v_\rho(u, \lambda) - d_\rho(\lambda) \geq \frac{\rho}{2} \|u - u_\lambda^*\|^2, \;\; \forall \, u \in \Re^m; \tag{5.2.11}$$

d) problem (5.2.8) has the same optimal value and set of optimal solutions as those of (5.2.3).

Proof. We first show a). Observe that convexity of v and Assumption A.5.1 imply that the function $v_\rho(\cdot,\lambda)$ in (5.2.9) is a proper lower-semicontinuous convex function for every $\lambda \in \Re^m$ and $\rho > 0$. Moreover, $v_\rho(\cdot,\lambda)$ is strongly convex with modulus ρ, that is,

$$v_\rho(\alpha u_1 + (1-\alpha)u_2, \lambda) \leq \alpha v_\rho(u_1,\lambda) + (1-\alpha)v_\rho(u_2,\lambda) - \frac{\rho}{2}\alpha(1-\alpha)\|u_1 - u_2\|^2, \quad (5.2.12)$$

for all $(u_1, u_2) \in \Re^m \times \Re^m$ and $\alpha \in (0,1)$. The above two observations clearly imply a). Statement b) follows directly from a), definition (5.2.2), and the equivalence of problems (5.2.7) and (5.2.9). To show c), we let $u_1 = u$ and $u_2 = u^*_\lambda$ in (5.2.12) to obtain

$$\begin{aligned} \frac{\rho}{2}\|u - u^*_\lambda\|^2 &\leq \frac{v_\rho(u,\lambda) - v_\rho(\alpha u + (1-\alpha)u^*_\lambda,\lambda)}{1-\alpha} + \frac{v_\rho(u^*_\lambda,\lambda) - v_\rho(\alpha u + (1-\alpha)u^*_\lambda,\lambda)}{\alpha} \\ &\leq \frac{v_\rho(u,\lambda) - v_\rho(\alpha u + (1-\alpha)u^*_\lambda,\lambda)}{1-\alpha}, \ \ \forall\, \alpha \in (0,1) \end{aligned}$$

where the last inequality follows from the fact that u^*_λ is the optimal solution for problem (5.2.9). Letting α go to zero in the above inequality, and using the lower-semicontinuity of v_ρ and the fact that $d_\rho(\lambda) = v_\rho(u^*_\lambda,\lambda)$, we obtain (5.2.11). Statement d) is a well-known. ■

The following proposition summarizes some important properties of d_ρ.

Proposition 5.2.2 *For any $\rho > 0$, the function d_ρ is concave, differentiable, and*

$$\nabla d_\rho(\lambda) = u^*_\lambda, \ \ \forall\, \lambda \in \Re^m, \quad (5.2.13)$$

where u^*_λ is the unique optimal solution of problem (5.2.9). Moreover, d_ρ has $1/\rho$-Lipschitz-continuous gradient with respect to the inner product norm on $\Re^m$.

Proof. Under Assumption A.5.1, the claim follows immediately from Theorem 1 of [50] applied to the maximization version of (5.2.9), i.e., the problem $\max_u\{-v_\rho(u,\lambda)\}$. ■

In view of Proposition 5.2.2 and Lemma 5.2.1(b), the exact version of the augmented Lagrangian method stated in Section 1.3 can be viewed as a version of the steepest ascent method applied to (5.2.8). Note that one possible drawback of the exact augmented Lagrangian method is that each iteration of this method requires the solution of problem (5.1.2) for computing the gradient $\nabla d_\rho(\lambda_k)$. Since in most applications, problem (5.1.2)

can only be solved approximately, in this chapter we are interested in analyzing the inexact version of the augmented Lagrangian method where the gradient $\nabla d_\rho(\lambda_k)$ is approximated by $\mathcal{A}(x_k)$, where x_k an approximate solution of problem (5.1.2).

The following simple but crucial result gives a bound on the error between $\nabla d_\rho(\lambda_k)$ and its aforementioned approximation.

Proposition 5.2.3 *Assume that $(x, \lambda) \in X \times \Re^m$ is such that $\mathcal{L}_\rho(x, \lambda) - d_\rho(\lambda) \leq \eta$. Then, we have*

$$\|\mathcal{A}(x) - \nabla d_\rho(\lambda)\| = \|\mathcal{A}(\tilde{x}) - u_\lambda^*\| \leq \sqrt{\frac{2\eta}{\rho}}, \tag{5.2.14}$$

where u_λ^* is the unique optimal solution of (5.2.9).

Proof. Letting $u := \mathcal{A}(x)$ and observing that $f(x) \geq v(u)$ due to definition (5.2.2), we conclude that

$$\mathcal{L}_\rho(x, \lambda) = f(x) + \langle \lambda, u \rangle + \frac{\rho}{2}\|u\|^2 \geq v(u) + \langle \lambda, u \rangle + \frac{\rho}{2}\|u\|^2 = v_\rho(u, \lambda). \tag{5.2.15}$$

This inequality, relation (5.2.11), and the assumption that $\mathcal{L}_\rho(x, \lambda) - d_\rho(\lambda) \leq \eta$ then imply that

$$\mathcal{L}_\rho(x, \lambda) - d_\rho(\lambda) \geq v_\rho(u, \lambda) - d_\rho(\lambda) \geq \frac{\rho}{2}\|u - u_\lambda^*\|^2, \tag{5.2.16}$$

and hence that (5.2.14) holds. ∎

5.2.3 The augmented Lagrangian method

In this subsection, we present the augmented Lagrangian method applied to problem (5.1.1) and discuss its convergence behavior.

We start by stating the first inexact AL method that will be studied in this chapter.

The I-AL method:

Input: Initial points $\lambda_0 \in \Re^m$ and $x_{-1} \in X$, penalty parameter $\rho \in \Re_{++}$, outer tolerances $(\epsilon_p, \epsilon_d) \in \mathbf{R}_{++} \times \mathbf{R}_{++}$, iteration limit $\bar{N} \in \mathbb{N} \cup \{+\infty\}$, and inner tolerances $\eta_0, \ldots, \eta_{\bar{N}}$ satisfying

$$0 < \eta_k \leq \frac{\rho \epsilon_p^2}{128}, \quad \forall\, k = 0, \ldots, \bar{K}. \tag{5.2.17}$$

0) Set $k = 0$;

1) Using x_{k-1} as starting point, apply Nesterov's optimal method to find an η_k-approximate solution of problem (5.1.2), i.e., a point $x_k \in X$ such that

$$\mathcal{L}_\rho(x_k, \lambda_k) - d_\rho(\lambda_k) \leq \eta_k; \tag{5.2.18}$$

2) If $\|\mathcal{A}(x_k)\| \leq 3\epsilon_p/4$, then call subroutine Postprocessing with input $(x, \tilde{\lambda}) = (x_k, \lambda_k)$, report **success**, and terminate the algorithm;

3) Otherwise, if $\|\mathcal{A}(x_k)\| > 3\epsilon_p/4$, set $\lambda_{k+1} = \lambda_k + \rho \mathcal{A}(x_k)$ and increment k by 1;

4) If $k = \bar{N}$, report **failure**, and terminate the algorithm; otherwise, go to step 1.

end

We now describe subroutine Postprocessing.

Postprocessing$(x, \tilde{\lambda})$:

Set

$$\zeta = \zeta(\rho) := \min\left\{\frac{\rho\epsilon_p^2}{128}, \frac{\epsilon_d^2}{8M_\rho}\right\}. \tag{5.2.19}$$

P.1) Using $x \in X$ as starting point, apply Nesterov's optimal method to find a ζ-approximate solution $\tilde{x}$ of problem (5.1.2);

P.2) Output a pair $(\tilde{x}^+, \tilde{\lambda}^+)$ given by

$$\tilde{x}^+ := \Pi_X(\tilde{x} - \nabla\mathcal{L}_\rho(\tilde{x}, \tilde{\lambda})/M_\rho) \tag{5.2.20}$$

$$\tilde{\lambda}^+ := \tilde{\lambda} + \rho\mathcal{A}(\tilde{x}^+). \tag{5.2.21}$$

end

We will say that an outer iteration of the I-AL method occurs whenever k is incremented by 1 in Step 3. We will refer to an iteration of Nesterov's optimal method to compute x_k in step 1 or $\tilde{x}$ inside subroutine Postprocessing as an inner iteration of the I-AL method.

We now make a few comments about the I-AL method. First, note that the I-AL method is a generic algorithm in the sense that the parameters ρ and $\{\eta_k\}$ have not been specified. Concrete choices of these parameters will be discussed within the context of the convergence results which will be presented in the remaining part of this subsection. Second, in view of Proposition 5.2.3, an outer iteration of the I-AL method can be viewed as an iteration of a version of the steepest ascent method with inexact gradient with respect to problem (5.2.8). Third, Step 4 ensures that the method terminates in at most $\bar{N}$ outer iterations possibly reporting failure. Fourth, at the beginning of Step 2, the pair (x_k, λ_k) satisfies the primal termination condition (5.2.5), but not necessarily the dual termination criterion (5.2.6). By calling subroutine Postprocessing, the next result, whose proof will be given in Section 5.4.1, guarantees that the output pair $(\tilde{x}^+, \tilde{\lambda}^+)$ of this subroutine satisfies both (5.2.5) and (5.2.6).

Proposition 5.2.4 *Let $\rho > 0$, $(\epsilon_p, \epsilon_d) \in \mathbf{R}_{++} \times \mathbf{R}_{++}$, and $\tilde{\lambda} \in \Re^m$ be given and assume*

that there exists an $x \in X$ satisfying

$$\|\mathcal{A}(x)\| \leq \frac{3\epsilon_p}{4} \quad \textit{and} \quad \mathcal{L}_\rho(x, \tilde{\lambda}) - d_\rho(\tilde{\lambda}) \leq \frac{\rho\epsilon_p^2}{128}.$$

If $\tilde{x} \in X$ is a point satisfying $\mathcal{L}_\rho(\tilde{x}, \tilde{\lambda}) - d_\rho(\tilde{\lambda}) \leq \zeta$, where ζ is given by (5.2.19), then the pair $(\tilde{x}^+, \tilde{\lambda}^+)$ defined by (5.2.20) and (5.2.21) is an (ϵ_p, ϵ_d)-primal-dual solution of (5.1.1).

The following result follows as an immediate consequence of Proposition 5.2.4.

Corollary 5.2.1 *If the I-AL method successfully terminates (i.e., at Step 2), then the output pair of subroutine Postprocessing is an (ϵ_p, ϵ_d)-primal-dual solution of (5.1.1).*

Proof. The result follows from Proposition 5.2.4, (5.2.17), and the fact that at Step 4, conditions (5.2.18) and $\|\mathcal{A}(x_k)\| \leq 3\epsilon_p/4$ hold. ∎

Our next result below describes conditions on the parameters ρ and $\{\eta_k\}$ which guarantee the successful termination of the I-AL method.

Theorem 5.2.1 *Let $\rho \in \mathbf{R}_{++}$ and $(\epsilon_p, \epsilon_d) \in \mathbf{R}_{++} \times \mathbf{R}_{++}$ be given. Assume that the iteration limit $\bar{N}$ of the I-AL method satisfies*

$$\bar{N} \geq N := \left\lceil \frac{16 D_\Lambda^2}{\rho^2 \epsilon_p^2} \right\rceil, \tag{5.2.22}$$

where $D_\Lambda := \inf_{\lambda^* \in \Lambda^*} \|\lambda_0 - \lambda^*\|$, and the sequence $\{\eta_k\}_{k=0}^{\bar{N}-1} \subseteq \mathbf{R}_{++}$ satisfies

$$\sum_{k=0}^{\bar{N}-1} \eta_k \leq \frac{\rho\epsilon_p^2}{128}. \tag{5.2.23}$$

Then, the I-AL method successfully terminates in at most N outer iterations.

We now make a few observations about Theorem 5.2.1. First, we observe that Theorem 5.2.1 holds regardless of the method used to find the approximate solution x_k in step 1 or $\tilde{x}$ in subroutine Postprocessing. Second, although the number of outer iterations of the I-AL method does not depend on ϵ_d, the number of inner iterations will depend on it, since the number of inner iteration inside subroutine Postprocessing clearly depends on ϵ_d in view of (5.2.19). Third, observe that equation (5.2.22) implies that the larger ρ is, the smaller the

bound N on the number of outer iterations will be. On the other hand, since the Lipschitz constant of the objective function of subproblem (5.1.2) is given by

$$M_\rho := L_f + \rho\|\mathcal{A}\|^2, \tag{5.2.24}$$

increasing ρ will increase M_ρ, and as a consequence, will increase the iteration-complexity bound of Nesterov's optimal method for finding an approximate solution of (5.1.2).

The following result provides a bound on the total number of inner iterations, i.e., the iterations performed by Nesterov's optimal method, in the I-AL algorithm.

Proposition 5.2.5 *Let $(\epsilon_p, \epsilon_d) \in \mathbf{R}_{++} \times \mathbf{R}_{++}$, $\rho > 0$, $\bar{N} \in \mathbb{N} \cup \{+\infty\}$ and $\{\eta_k\}_{k=0}^{\bar{N}-1} \subseteq \mathbf{R}_{++}$ be given such that conditions (5.2.22) and (5.2.23) are satisfied. Then, the I-AL method applied to (5.1.1) successfully terminates in N outer iterations, and computes an (ϵ_p, ϵ_d)-primal-dual solution of (5.1.1) in at most $\mathcal{I}_p + \mathcal{I}_d$ inner iterations, where N is defined in Theorem 5.2.1,*

$$\mathcal{I}_p := \left\lfloor \sqrt{2} D_X M_\rho^{\frac{1}{2}} \sum_{k=0}^{N-1} \eta_k^{-\frac{1}{2}} + N \right\rfloor, \quad \mathcal{I}_d := \left\lceil 4 D_X \max\left\{ \frac{4M_\rho^{\frac{1}{2}}}{\rho^{\frac{1}{2}}\epsilon_p}, \frac{M_\rho}{\epsilon_d} \right\} \right\rceil \tag{5.2.25}$$

and

$$D_X := \max_{x_1, x_2 \in X} \|x_1 - x_2\|. \tag{5.2.26}$$

Proof. Clearly, in view of Corollary 1.1.1 and Theorem 5.2.1, the number of inner iterations performed at step 1 of the I-AL method is bounded by

$$\sum_{k=0}^{N-1} \left\lceil D_X \sqrt{\frac{2M_\rho}{\eta_k}} \right\rceil \leq \sqrt{2} D_X M_\rho^{\frac{1}{2}} \sum_{k=0}^{N-1} \eta_k^{-\frac{1}{2}} + N,$$

and hence by $\mathcal{I}_p$. Moreover, by Corollary 1.1.1, the number of inner iterations performed at step 2 (inside subroutine PostProcessing) is bounded by $\lceil D_X \sqrt{2M_\rho/\zeta} \rceil$. Using the definition of ζ in (5.2.19), it follows that the number of inner iterations performed at step 3 is bounded by $\mathcal{I}_d$. The claim then easily follows by combining the previous two observations.

■

We now present a few consequences of the results obtained in Proposition 5.2.5. The first one stated below bounds the total number of inner iterations of the I-AL method when a summable sequence $\{\eta_k\}$ satisfying condition (5.2.23) is chosen.

Theorem 5.2.2 *Let $\rho > 0$ be an arbitrary penalty parameter and $(\epsilon_p, \epsilon_d) \in \Re_{++} \times \Re_{++}$ be given. If, for some $\xi > 0$, the I-AL method is applied to problem (5.1.1) with input $\bar{N} = +\infty$ and*

$$\eta_k = \frac{\xi \rho \epsilon_p^2}{128(1+\xi)(k+1)^{1+\xi}}, \quad \forall\, k \geq 0, \tag{5.2.27}$$

then the I-AL method successfully terminates in N outer iterations and computes an (ϵ_p, ϵ_d)-primal-dual solution of (5.1.1) in at most

$$\mathcal{O}\left(\frac{D_X M_\rho^{\frac{1}{2}}}{\rho^{\frac{1}{2}} \epsilon_p} \left[\left(\frac{D_\Lambda}{\rho \epsilon_p} \right)^{3+\xi} + 1 \right] + \frac{D_X M_\rho}{\epsilon_d} + \frac{D_\Lambda^2}{\rho^2 \epsilon_p^2} + 1 \right) \tag{5.2.28}$$

inner iterations, where N is given by (5.2.22). In particular, if

$$\rho = \frac{4}{\epsilon_p} \left(\frac{(D_\Lambda)^{3+\xi} \epsilon_d}{\|\mathcal{A}\|} \right)^{\frac{1}{4+\xi}} + \frac{L_f}{\|\mathcal{A}\|^2}, \tag{5.2.29}$$

then the I-AL method successfully terminates in

$$\left\lceil \min \left\{ \left(\frac{D_\Lambda \|\mathcal{A}\|}{\epsilon_d} \right)^{\frac{2}{4+\xi}}, \frac{16 D_\Lambda^2 \|\mathcal{A}\|^4}{L_f^2 \epsilon_p^2} \right\} \right\rceil \tag{5.2.30}$$

outer iterations and computes an (ϵ_p, ϵ_d)-primal-dual solution of (5.1.1) in at most

$$\mathcal{O}\left(D_X \left(\frac{\|A\|^{\frac{7+2\xi}{4+\xi}} D_\Lambda^{\frac{3+\xi}{4+\xi}}}{\epsilon_p \epsilon_d^{\frac{3+\xi}{4+\xi}}} + \frac{\|\mathcal{A}\|}{\epsilon_p} + \frac{L_f}{\epsilon_d} \right) + \left(\frac{\|\mathcal{A}\| D_\Lambda}{\epsilon_d} \right)^{\frac{2}{4+\xi}} + 1 \right) \tag{5.2.31}$$

inner iterations.

We now make a few observations about Theorem 5.2.2. First, in contrast to the quadratic penalty method where the penalty parameter should be chosen larger than a certain threshold value in order to derive provable iteration-complexity results (see Lan and Monteiro [33]), the I-AL method has an iteration-complexity bound, namely (5.2.28), which holds regardless of the value of the penalty parameter ρ. Second, it is not difficult to see that the choice of ρ in (5.2.29) gives the best iteration-complexity bound based on (5.2.28) up to a constant factor. Third, a drawback of the above result is that the formula for ρ in (5.2.29) depends on the unknown value D_Λ. This drawback will be remedied by the next two results of this subsection.

Instead of choosing a summable sequence $\{\eta_k\}$, the next result assumes $\bar{N}$ is finite and chooses $\eta_0, \ldots, \eta_{\bar{N}-1}$uniformly, and instead of assuming the exact knowledge of D_Λ, it

assumes that an upper bound $t \geq D_\Lambda$ is given. The motivation for choosing $\eta_0, \ldots, \eta_{\bar{N}-1}$ uniformly is that the minimum of the summation term in the definition of $\mathcal{I}_p$ in (5.2.25) subject to a condition like (5.2.23) occurs exactly when $\eta_0, \ldots, \eta_{N-1}$ is uniformly chosen.

Theorem 5.2.3 *Let $(\epsilon_p, \epsilon_d) \in \Re_{++} \times \Re_{++}$ be given. If, for some $t \geq D_\Lambda$, the I-AL is applied to problem (5.1.1) with input*

$$\rho = \rho(t) := \frac{4\, t^{\frac{3}{4}} \epsilon_d^{\frac{1}{4}}}{\|\mathcal{A}\|^{\frac{1}{4}} \epsilon_p} + \frac{L_f}{\|\mathcal{A}\|^2}, \;\; \bar{N} = \bar{N}(t) := \left\lceil \frac{16 t^2}{\rho(t)^2 \epsilon_p^2} \right\rceil, \tag{5.2.32}$$

$$\eta_k = \eta(t) := \frac{\rho(t) \epsilon_p^2}{128 \bar{N}(t)}, \forall\, k \geq 0, \tag{5.2.33}$$

then the method successfully terminates in

$$\left\lceil \min \left\{ \frac{D_\Lambda^2 \|\mathcal{A}\|^{\frac{1}{2}}}{t^{\frac{3}{2}} \epsilon_d^{\frac{1}{2}}}, \frac{16 D_\Lambda^2 \|\mathcal{A}\|^4}{L_f^2 \epsilon_p^2} \right\} \right\rceil \leq \left\lceil \min \left\{ \frac{D_\Lambda^{\frac{1}{2}} \|\mathcal{A}\|^{\frac{1}{2}}}{\epsilon_d^{\frac{1}{2}}}, \frac{16 D_\Lambda^2 \|\mathcal{A}\|^4}{L_f^2 \epsilon_p^2} \right\} \right\rceil \tag{5.2.34}$$

outer iterations and computes an (ϵ_p, ϵ_d)-primal-dual solution in at most $\mathcal{O}(\mathcal{I}_{pd}(t))$ inner iterations, where

$$\mathcal{I}_{pd}(t) := \left\lceil D_X \left(\frac{\|\mathcal{A}\|^{\frac{7}{4}} t^{\frac{3}{4}}}{\epsilon_p \epsilon_d^{\frac{3}{4}}} + \frac{\|\mathcal{A}\|}{\epsilon_p} + \frac{L_f}{\epsilon_d} \right) + \left(\frac{t \|\mathcal{A}\|}{\epsilon_d} \right)^{\frac{1}{2}} \right\rceil, \tag{5.2.35}$$

and D_X and D_Λ are defined in Theorem 5.2.1 and Proposition 5.2.5, respectively.

Observe that the choice of ρ, $\bar{N}$, and $\{\eta_k\}$ given by (5.2.32) and (5.2.33) requires $t \geq D_\Lambda$ so as to guarantee conditions (5.2.22) and (5.2.23), and hence that the conclusions of Theorem (5.2.1) hold. We now develop a guess-and-check procedure that attempts to find such a constant t while at the same time checks for potentially early termination of the procedure.

I-AL guess-and-check procedure:

Input: Initial points $\lambda_0 \in \Re^m$ and $x_{-1} \in X$, and tolerances $(\epsilon_p, \epsilon_d) \in \mathbf{R}_{++} \times \mathbf{R}_{++}$.

0) Set $t_0 = \min\{(\beta_0/\beta_1)^{\frac{4}{3}}, (\beta_0/\beta_2)^2\}$ and $j = 0$, where

$$\beta_0 := 1 + \frac{32 D_X \|\mathcal{A}\|}{\epsilon_p}, \; \beta_1 := \frac{32 D_X \|\mathcal{A}\|^{\frac{7}{4}}}{\epsilon_p \epsilon_d^{\frac{3}{4}}}, \; \beta_2 := \frac{\|\mathcal{A}\|^{\frac{1}{2}}}{\epsilon_d^{\frac{1}{2}}}; \tag{5.2.36}$$

1) Run the I-AL method with the above input and with $\rho = \rho(t_j)$, $\bar{N} = \bar{N}(t_j)$ and

$$\eta_k = \eta(t_j), \quad k = 0, \ldots, \bar{N}(t_j);$$

2) If the I-AL method successfully terminates, **stop**; Otherwise, if the I-AL method reports failure, set $t_{j+1} = 2t_j$, $j = j + 1$, and go to step 1.

end

The following result gives the iteration-complexity of the above procedure for obtaining an (ϵ_p, ϵ_d)-primal-dual solution of (5.1.1).

Theorem 5.2.4 *Let $(\epsilon_p, \epsilon_d) \in \mathbf{R}_{++} \times \mathbf{R}_{++}$ be given. The I-AL guess-and-check procedure finds an (ϵ_p, ϵ_d)-primal-dual solution of (5.1.1) in at most $\mathcal{O}(\mathcal{I}_{pd}(D_\Lambda))$ inner iterations, where $\mathcal{I}_{pd}(t)$ is defined by (5.2.35).*

It is interesting to compare the iteration-complexity bound obtained in Theorem 5.2.4 with the corresponding one obtained for the quadratic penalty method in [33] to compute an (ϵ_p, ϵ_d)-primal-dual solution of (5.1.1), namely,

$$\mathcal{O}\left(D_X\left(\frac{\|\mathcal{A}\|^2 D_\Lambda}{\epsilon_p \epsilon_d} + \frac{\|\mathcal{A}\|}{\epsilon_p} + \frac{L_f}{\epsilon_d}\right) + 1\right).$$

Clearly, the latter one is worse than $\mathcal{O}(\mathcal{I}_{pd}(D_\Lambda))$ by a factor of $\mathcal{O}((\|\mathcal{A}\| D_\Lambda / \epsilon_d)^{\frac{1}{4}})$.

Finally, we make some observations about the possibility of exploiting the warm-start strategy for solving the augmented Lagrangian subproblems (5.1.2). Even though we already stated the I-AL method with the warm-start strategy included, i.e., the one in which the approximate solution of the previous subproblem is used as a starting point for the solution of next subproblem, the proofs of the results stated in this subsection make no use of this feature. The difficulty in exploiting this feature here is due to the fact that the objective functions of the augmented Lagrangian subproblems are convex, but not necessarily strongly convex. But in next subsection, by adding a small strongly convex perturbation to the objective function of problem (5.1.1), we will be able to guarantee that the objective functions of the corresponding augmented Lagrangian subproblems will be strongly convex, and thereby exploit the warm start strategy for solving the augmented Lagrangian subproblems, and consequently, the original problem (5.1.1).

5.2.4 The I-AL method applied to a perturbation problem

In this subsection, we will exploit the possibility of solving problem (5.1.1) by applying a slightly modified version of the I-AL algorithm to a perturbed problem obtained by adding a small strongly convex perturbation to the objective function of (5.1.1).

We start by introducing the perturbed problem, namely:

$$f_\gamma^* := \min\{f_\gamma(x) := f(x) + \frac{\gamma}{2}\|x - x_0\|^2 : \mathcal{A}(x) = 0, x \in X\}, \tag{5.2.37}$$

where x_0 is a fixed point in X and $\gamma > 0$ is a prespecified perturbation parameter. It is well-known that if γ is sufficiently small, then an approximate solution of (5.2.37) will also be an approximate solution of (5.1.1).

The following simple lemma relates the optimal values of the perturbation problem (5.2.37) and the original problem (5.1.1).

Lemma 5.2.2 *Let f^* and f_γ^* be the optimal values defined in (5.1.1) and (5.2.37), respectively. Then,*

$$0 \leq f_\gamma^* - f^* \leq \gamma D_X^2/2, \tag{5.2.38}$$

where D_X is defined in Proposition 5.2.5.

Proof. The first inequality in (5.2.38) follows immediately from the fact that $f_\gamma \geq f$. Now, let x^* and x_γ^* be optimal solutions of (5.1.1) and (5.2.37), respectively. Then,

$$f_\gamma^* = f(x_\gamma^*) + \frac{\gamma}{2}\|x_\gamma^* - x_0\|^2 \leq f(x^*) + \frac{\gamma}{2}\|x^* - x_0\|^2 \leq f^* + \frac{\gamma D_X^2}{2},$$

from which the second inequality in (5.2.38) follows. ∎

In this section, we will derive an iteration-complexity bound for obtaining an (ϵ_p, ϵ_d)-primal-dual solution of (5.1.1) by applying the I-AL method directly to the perturbed problem (5.2.37) for a conveniently chosen perturbation parameter $\gamma > 0$.

The augmented dual function associated with (5.2.37) is given by

$$d_{\rho,\gamma}(\lambda) := \min_{x \in X} \left\{ \mathcal{L}_{\rho,\gamma}(x, \lambda) := f_\gamma(x) + \lambda^T \mathcal{A}(x) + \frac{\rho}{2}\|\mathcal{A}(x)\|^2 \right\}, \tag{5.2.39}$$

or alternatively, by

$$d_{\rho,\gamma}(\lambda) = \inf_u \left\{ v_{\rho,\gamma}(u,\lambda) := v_\gamma(u) + \langle \lambda, u \rangle + \frac{\rho}{2} \|u\|^2 \right\}, \tag{5.2.40}$$

where $v_\gamma(\cdot)$ is the value function associated with the perturbed problem (5.2.37) (see definition (5.2.2)). We denote the optimal solution of (5.2.40) by $u^*_{\lambda,\gamma}$.

It can be easily seen that the function $\mathcal{L}_{\rho,\gamma}(\cdot,\lambda)$ has $M_{\rho,\gamma}$-Lipschitz continuous gradient where

$$M_{\rho,\gamma} := L_f + \rho\|A\|^2 + \gamma, \tag{5.2.41}$$

and that it is strongly convex with modulus γ with respect to $\|\cdot\|$.

We now describe a modification of the I-AL method.

The Modified I-AL method: This method is the same as I-AL method applied to the perturbed problem (5.2.37) (and hence with M_ρ, $\mathcal{L}_\rho$, and d_ρ replaced by $M_{\rho,\gamma}$, $\mathcal{L}_{\rho,\gamma}$, and $d_{\rho,\gamma}$) except that instead of Nesterov's method, its variant described in Theorem 1.1.5 is used to compute the approximate solutions x_k in step 1 and $\tilde{x}$ in subroutine Postprocessing, and the tolerance ζ in (5.2.19) is replaced by

$$\tilde{\zeta} = \tilde{\zeta}(\rho,\gamma) := \min\left\{ \frac{\rho\epsilon_p^2}{128}, \frac{\epsilon_d^2}{32 M_{\rho,\gamma}} \right\}. \tag{5.2.42}$$

The next results is a corresponding version of Proposition 5.2.4, which guarantees that the output pair $(\tilde{x}^+, \tilde{\lambda}^+)$ of subroutine Postprocessing is an (ϵ_p, ϵ_d)-primal-dual solution of (5.1.1).

Proposition 5.2.6 *Let $\rho > 0$, $(\epsilon_p, \epsilon_d) \in \mathbf{R}_{++} \times \mathbf{R}_{++}$, and $\tilde{\lambda} \in \Re^m$ be given, and define*

$$\gamma := \frac{\epsilon_d}{2D_X}. \tag{5.2.43}$$

Assume that there exists an $x \in X$ satisfying

$$\|\mathcal{A}(x)\| \le \frac{3\epsilon_p}{4} \quad \text{and} \quad \mathcal{L}_{\rho,\gamma}(x,\tilde{\lambda}) - d_{\rho,\gamma}(\tilde{\lambda}) \le \frac{\rho\epsilon_p^2}{128}.$$

If $\tilde{x} \in X$ is a point satisfying $\mathcal{L}_{\rho,\gamma}(\tilde{x},\tilde{\lambda}) - d_{\rho,\gamma}(\tilde{\lambda}) \le \tilde{\zeta}$, where $\tilde{\zeta}$ is given by (5.2.42), then the pair $(\tilde{x}^+, \tilde{\lambda}^+)$ defined by (5.2.20) and (5.2.21) with $\mathcal{L}_\rho$ replaced by $\mathcal{L}_{\rho,\gamma}$ is an (ϵ_p, ϵ_d)-primal-dual solution of (5.1.1).

The following result follows as an immediate consequence Proposition 5.2.4.

Corollary 5.2.2 *If the modified I-AL method successfully terminates (i.e., at Step 2), then the output pair of subroutine Postprocessing is an (ϵ_p, ϵ_d)-primal-dual solution of (5.1.1).*

Proof. The result follows from Proposition 5.2.6, (5.2.17), and the fact that at Step 4, conditions (5.2.18) and $\|\mathcal{A}(x_k)\| \leq 3\epsilon_p/4$ hold. ∎

We now state the corresponding versions of Theorems 5.2.3 and 5.2.4 with respect to the modified I-AL method.

Theorem 5.2.5 *Let $(\epsilon_p, \epsilon_d) \in \Re_{++} \times \Re_{++}$ be given, and let γ be given by (5.2.43). For some $t > 0$, consider the modified I-AL method applied to the perturbed problem (5.2.37) with input*

$$\rho = \rho_\gamma(t) \;\; := \;\; \frac{4t}{\epsilon_p(\log \mathcal{T}(t))^{\frac{1}{2}}} + \frac{L_f + \gamma}{\|\mathcal{A}\|^2}, \tag{5.2.44}$$

$$\bar{N} = \bar{N}_\gamma(t) \;\; := \;\; \left\lceil \frac{16t^2}{\rho_\gamma(t)^2 \epsilon_p^2} \right\rceil, \quad \eta_k = \eta_\gamma(t) := \frac{\rho_\gamma(t)\epsilon_p^2}{128\bar{N}_\gamma(t)}, \quad \forall\ k \geq 0, \tag{5.2.45}$$

where

$$\mathcal{T}(t) := \mathcal{S}_1 t^{\frac{1}{2}} + \mathcal{S}_2 + \mathcal{S}_3, \tag{5.2.46}$$

$$\mathcal{S}_1 := \sqrt{\frac{D_X \|\mathcal{A}\|^2}{\epsilon_p \epsilon_d}}, \;\; \mathcal{S}_2 := \sqrt{\frac{D_X L_f}{\epsilon_d}} + 1 \;\; \textit{and } \mathcal{S}_3 := \sqrt{\frac{D_X \|\mathcal{A}\|}{\epsilon_p}} + 3. \tag{5.2.47}$$

Then the following statements hold:

a) the total number of inner iterations performed by the above method is bounded by

$$\mathcal{O}\left\{ \left(\mathcal{S}_1 t^{\frac{1}{2}} + \mathcal{S}_2 \left[\log \mathcal{T}(t)\right]^{\frac{1}{4}} \right) \left[\log \mathcal{T}(t)\right]^{\frac{3}{4}} \max\left(1, \log \frac{D_\Lambda^\gamma \log \mathcal{T}(t)}{t}\right) \right\}; \tag{5.2.48}$$

b) if $t \geq D_\Lambda^\gamma$, where $D_\Lambda^\gamma := \inf_{\lambda_\gamma \in \Lambda_\gamma^*} \|\lambda_0 - \lambda^*\|$ and Λ_γ^* denotes the set of Lagrange multipliers associated with (5.2.37), then the above method successfully terminates in $\mathcal{O}(\log \mathcal{T}(t))$ outer iterations with an (ϵ_p, ϵ_d)-primal-dual solution of (5.1.1).

Observe that the choice of ρ, $\bar{N}$, and $\{\eta_k\}$ given by (5.2.44) and (5.2.45) requires $t \geq D_\Lambda$ to guarantee the successful termination of the modified I-AL method. We now develop a guess-and-check procedure that attempts to find such a constant t while at the same time checks for potentially early termination of the procedure.

The modified I-AL guess-and-check procedure:

Input: Initial points $\lambda_0 \in \Re^m$ and $x_{-1} \in X$, and tolerances $(\epsilon_p, \epsilon_d) \in \mathsf{R}_{++} \times \mathsf{R}_{++}$.

0) Let scalar $\hat{t}$ and function $\psi : \Re^+ \to \Re$ be defined as

$$\hat{t} := \left[\frac{\mathcal{S}_2^2 + \mathcal{S}_2 \sqrt{\mathcal{S}_2^2 + 4(\mathcal{S}_2 + \mathcal{S}_3)}}{2\mathcal{S}_1} \right]^2, \quad \psi(t) := \mathcal{S}_1 t^{\frac{1}{2}} - \mathcal{S}_2 \left[\log(\mathcal{S}_1 t^{\frac{1}{2}} + \mathcal{S}_2 + \mathcal{S}_3) \right]^{\frac{1}{4}}, \tag{5.2.49}$$

where $\mathcal{S}_1, \mathcal{S}_2$ and $\mathcal{S}_3$ are given by (5.2.47). Find a point $t_0 \in [0, \hat{t}]$ such that $0 \leq \psi(t_0) \leq 1$.

1) Run the modified I-AL method with the above input and with $\rho = \rho_\gamma(t_j)$, $\bar{N} = \bar{N}_\gamma(t_j)$, $\eta_k = \eta_\gamma(t_j)$ for $k \geq 0$, where γ is given by (5.2.43), and $\rho_\gamma(\cdot)$, $\bar{N}_\gamma(\cdot)$ and $\eta_\gamma(\cdot)$ are defined in (5.2.44) and (5.2.45).

2) If the modified I-AL method successfully terminates, **stop**; otherwise, set $t_{j+1} = 2t_j$, $j = j + 1$, and go to step 1.

end

We now discuss the issue about the existence of t_0 satisfying $0 \leq \psi(t_0) \leq 1$. It will be shown in Lemma 5.4.4 that $\psi(0) \leq 0$, $\psi(\hat{t}) \geq 0$, and function ψ is non-decreasing. This clearly implies the existence of the required t_0. Moreover, t_0 can be computed as follows. If $\psi(\hat{t}) \leq 1$, we can take $t_0 = \hat{t}$. Otherwise, a binary search procedure starting with the interval $[0, \hat{t}]$, which must contain the desired scalar t_0, determines such a scalar in $\log \hat{t}$ iterations.

The following result gives the iteration-complexity of the above procedure for obtaining an (ϵ_p, ϵ_d)-primal-dual solution of (5.1.1).

Theorem 5.2.6 *Let $(\epsilon_p, \epsilon_d) \in \mathsf{R}_{++} \times \mathsf{R}_{++}$ be given. The modified I-AL guess-and-check procedure described above finds an (ϵ_p, ϵ_d) -primal-dual solution of (5.1.1) in at most*

$$\mathcal{O}\left\{\mathcal{S}_1[D_\Lambda^\gamma]^{\frac{1}{2}}[\log \mathcal{T}(D_\Lambda^\gamma)]^{\frac{3}{4}} \log\log \mathcal{T}(D_\Lambda^\gamma) + \mathcal{S}_2 \log \mathcal{T}(0) \log\log \mathcal{T}(0)\right\}, \tag{5.2.50}$$

inner iterations, where $\mathcal{S}_1, \mathcal{S}_2$, $\mathcal{T}(\cdot)$ and D_Λ^γ are defined in Theorem 5.2.5.

It is interesting to compare the iteration-complexity bound obtained in Theorem 5.2.6 with the corresponding one obtained for the quadratic penalty method in [33] to compute an (ϵ_p, ϵ_d)-primal-dual solution of (5.1.1), namely, $\mathcal{O}\left(\mathcal{T}(\|\lambda_\gamma^*\|) \log \mathcal{T}(\|\lambda_\gamma^*\|)\right)$, where λ_γ^* is the minimum-norm Lagrange multiplier for the perturbed problem (5.2.37). Clearly, if the initial multiplier $\lambda_0 = 0$, then $\|\lambda_\gamma^*\| = D_\lambda^\gamma$ and the latter complexity bound reduces to $\mathcal{O}\left(\mathcal{T}(D_\Lambda^\gamma) \log \mathcal{T}(D_\Lambda^\gamma)\right)$. Note that for the situation where

$$\mathcal{S}_2 \log \mathcal{T}(0) \log\log \mathcal{T}(0) = \mathcal{O}\left\{\mathcal{S}_1[D_\Lambda^\gamma]^{\frac{1}{2}}[\log \mathcal{T}(D_\Lambda^\gamma)]^{\frac{3}{4}} \log\log \mathcal{T}(D_\Lambda^\gamma)\right\}, \tag{5.2.51}$$

bound (5.2.50) is majorized by $\mathcal{O}\left(\mathcal{T}(D_\Lambda^\gamma)[\log \mathcal{T}(D_\Lambda^\gamma)]^{\frac{3}{4}} \log\log \mathcal{T}(D_\Lambda^\gamma)\right)$. Clearly, inequality (5.2.51) holds if $L_f = 0$. Hence, when $\lambda_0 = 0$ and (5.2.51) holds, the first complexity bound is worse than the latter one in Theorem 5.2.6 by a factor of $(\log \mathcal{T}(D_\Lambda^\gamma))^{\frac{1}{4}} / \log\log \mathcal{T}(D_\Lambda^\gamma)$. It should be mentioned that if a good warm-start λ_0 for problem (5.2.37) is known, i.e., the ratio $D_\Lambda^\gamma / \|\lambda_\gamma^*\|$ is small, then the complexity bound in Theorem 5.2.6 is substantially smaller than the above one.

5.3 Basic Tools

This section discusses some technical results that will be used in our analysis. It consists of two subsections. The first one develops several technical results involving projected gradients. The second subsection develops the convergence results for the steepest descent method with inexact gradient, which will play a crucial role in our analysis for the augmented Lagrangian methods.

5.3.1 Projected gradient and the optimality conditions

In this subsection, we assume that the inner product space $\Re^n$ is endowed with the norm $\|\cdot\|$ associated with its inner product and consider the CP problem (1.1.8).

It is well-known that $x^* \in X$ is an optimal solution of (1.1.8) if and only if $\nabla\phi(x^*) \in -\mathcal{N}_X(x^*)$. Moreover, this optimality condition is in turn related to the projected gradient of the function ϕ over X defined as follows.

Definition 5.3.1 *Given a fixed constant $\tau > 0$, we define the projected gradient of ϕ at $\tilde{x} \in X$ with respect to X as (see, for example, [49])*

$$\nabla\phi(\tilde{x})]_X^\tau := \frac{1}{\tau}\left[\tilde{x} - \Pi_X(\tilde{x} - \tau\nabla\phi(\tilde{x}))\right], \tag{5.3.1}$$

where $\Pi_X(\cdot)$ is the projection map onto X defined in terms of the inner product norm $\|\cdot\|$ (see Subsection 5.1.1).

The following proposition (see Proposition 4 in [33] for the proof) relates the projected gradient to the aforementioned optimality condition.

Proposition 5.3.1 *Let $\tilde{x} \in X$ be given and define $\tilde{x}^+ := \Pi_X(\tilde{x} - \tau\nabla\phi(\tilde{x}))$. Then, for any given $\epsilon \geq 0$, the following statements hold:*

a) $\|\nabla\phi(\tilde{x})]_X^\tau\| \leq \epsilon$ if, and only if, $\nabla\phi(\tilde{x}) \in -\mathcal{N}_X(\tilde{x}^+) + \mathcal{B}(\epsilon)$;

b) $\|\nabla\phi(\tilde{x})]_X^\tau\| \leq \epsilon$ implies that $\nabla\phi(\tilde{x}^+) \in -\mathcal{N}_X(\tilde{x}^+) + \mathcal{B}\left((1 + \tau L_\phi)\epsilon\right)$.

The following result, whose proof is given in Lemma 5 of [33], states some properties of the projected gradient.

Lemma 5.3.1 *Assume that $x^* \in \operatorname{Argmin}_{x\in X}\phi(x)$. Let $\tilde{x} \in X$ be given and define*

$$\tilde{x}^+ := \Pi_X(\tilde{x} - \tau\nabla\phi(\tilde{x})).$$

Then, the following statements hold:

a) $\phi(\tilde{x}^+) - \phi(\tilde{x}) \leq -\tau\|\nabla\phi(\tilde{x})]_X^\tau\|^2/2$ for any $\tau \leq 1/L_\phi$;

b) for any $x \in X$, we have

$$\phi(x) - \phi(x^*) \geq \frac{1}{2L_\phi}\|\nabla\phi(x)]_X^{1/L_\phi}\|^2. \tag{5.3.2}$$

5.3.2 Steepest descent method with inexact gradient

In this subsection, we consider the unconstrained problem

$$p^* := \inf\{p(\lambda) : \lambda \in \Re^m\}, \tag{5.3.3}$$

where $p : \Re^m \to \mathbf{R}$ is convex and has L_p-Lipschitz-continuous gradient. We assume throughout this subsection that p^* is finite and that the set of optimal solutions Γ^* of (5.3.3) is nonempty. We are interested in the situation where the gradient $\nabla p(\lambda)$ at any given $\lambda \in \Re^m$ can only be evaluated approximately. This situation arises for example in the case where $p = -d_\rho$, where the computation of the exact gradient requires finding the exact optimal solution of the nonlinear optimization problem (5.2.9) (see Proposition 5.2.2). The aim is to apply the results obtained here to the function $p = -d_\rho$ in order to prove the main convergence results of the augmented Lagrangian method discussed in Sections 5.2.3 and 5.2.4.

An iterate of the steepest descent method with inexact gradient for solving problem (5.3.3) consists of:

$$\lambda_{k+1} = \lambda_k - \frac{\alpha_k}{L_p} p'_k \tag{5.3.4}$$

where $\alpha_k > 0$ is the stepsize and p'_k is an approximation of the gradient $\nabla p(\lambda_k)$. Define the deviation and the relative deviation between p'_k and $\nabla p(\lambda_k)$ respectively by

$$\delta_k := p'_k - \nabla p(\lambda_k), \qquad e_k := \frac{\|\delta_k\|}{\|p'_k\|}. \tag{5.3.5}$$

Before stating the main result of this subsection about the convergence of the inexact steepest descent method, we first present a few technical results.

Lemma 5.3.2 *If $e_k \leq 1 - \alpha_k/2$, then $p(\lambda_{k+1}) \leq p(\lambda_k)$.*

Proof. Using the second inequality of (5.1.5) with $\lambda = \lambda_k$ and $\tilde{\lambda} = \lambda_{k+1}$, relations (5.3.4) and (5.3.5), and the Cauchy-Schwartz inequality, we conclude that

$$\begin{aligned} p(\lambda_{k+1}) - p(\lambda_k) &\leq \langle \nabla p(\lambda_k), \lambda_{k+1} - \lambda_k \rangle + \frac{L_p}{2} \|\lambda_{k+1} - \lambda_k\|^2 \\ &= -\frac{\alpha_k}{L_p} \langle p'_k - \delta_k, p'_k \rangle + \frac{\alpha_k^2}{2L_p} \|p'_k\|^2 \;=\; -\frac{\alpha_k}{L_p} \|p'_k\|^2 \left(1 - \frac{\alpha_k}{2} - \frac{\|\delta_k\|}{p'_k}\right) \\ &= -\frac{\alpha_k}{L_p} \|p'_k\|^2 \left(1 - \frac{\alpha_k}{2} - e_k\right) \leq 0, \end{aligned}$$

where the last inequality is due to the assumption that $e_k \leq 1 - \alpha_k/2$. ∎

Lemma 5.3.3 *Assume that $e_k < 1$. Then, for every $\lambda^* \in \Lambda^*$, we have*

$$\alpha_k \beta_k \langle \nabla p(\lambda_k), \lambda_k - \lambda^* \rangle \leq \frac{L_p}{2} \left(\|\lambda_k - \lambda^*\|^2 - \|\lambda_{k+1} - \lambda^*\|^2 \right) + \alpha_k \langle \delta_k, \lambda^* - \lambda_k \rangle, \tag{5.3.6}$$

where

$$\beta_k := 1 - \alpha_k/[2(1 - e_k)^2]. \tag{5.3.7}$$

Proof. First note that, by (5.3.5), we have

$$\|\nabla p(\lambda_k)\| = \|p'_k - \delta_k\| \geq \|p'_k\| - \|\delta_k\| = (1 - e_k)\|p'_k\|. \tag{5.3.8}$$

This inequality, the assumption that $e_k < 1$ and relations (5.3.4) and (5.3.5) then imply

$$\begin{aligned}
\|\lambda_{k+1} - \lambda^*\|^2 &= \left\| \lambda_k - \frac{\alpha_k}{L_p} p'_k - \lambda^* \right\|^2 = \|\lambda_k - \lambda^*\|^2 - \frac{2\alpha_k}{L_p} \langle p'_k, \lambda_k - \lambda^* \rangle + \frac{\alpha_k^2}{L_p^2} \|p'_k\|^2 \\
&\leq \|\lambda_k - \lambda^*\|^2 - \frac{2\alpha_k}{L_p} \langle \nabla p(\lambda_k) + \delta_k, \lambda_k - \lambda^* \rangle + \frac{\alpha_k^2}{L_p^2 (1 - e_k)^2} \|\nabla p(\lambda_k)\|^2 \\
&\leq \|\lambda_k - \lambda^*\|^2 + \frac{2\alpha_k}{L_p} \langle \delta_k, \lambda^* - \lambda_k \rangle - \frac{2\alpha_k}{L_p} \left(1 - \frac{\alpha_k}{2(1 - e_k)^2} \right) \langle \nabla p(\lambda_k), \lambda_k - \lambda^* \rangle,
\end{aligned}$$

where the last inequality follows from the first inequality in (5.1.6) and the fact that $\nabla p(\lambda^*) = 0$. Rearranging the later inequality and using the definition of β_k, we obtain (5.3.6). ∎

Lemma 5.3.4 *Assume that, for some constant $c_1 > 0$, we have*

$$e_k \leq 1 - \sqrt{\frac{\alpha_k + c_1}{2}}. \tag{5.3.9}$$

Then, for any $\lambda^* \in \Lambda^*$, we have

$$\alpha_k [p(\lambda_k) - p^*] \leq \frac{L_p}{c_1} \left[\left(1 + \frac{2\alpha_k e_k^2}{c_1} \right) \|\lambda_k - \lambda^*\|^2 - \|\lambda_{k+1} - \lambda^*\|^2 \right]. \tag{5.3.10}$$

Proof. By the Cauchy-Schwartz inequality and relations (5.3.5), (5.1.5), (5.3.6) and

(5.3.8), we have

$$\begin{aligned}
&\frac{L_p}{2}\left(\|\lambda_k-\lambda^*\|^2-\|\lambda_{k+1}-\lambda^*\|^2\right)+\alpha_k e_k\|p_k'\|\,\|\lambda_k-\lambda^*\| \\
&\quad\geq\ \frac{L_p}{2}\left(\|\lambda_k-\lambda^*\|^2-\|\lambda_{k+1}-\lambda^*\|^2\right)+\alpha_k\langle\delta_k,\lambda^*-\lambda_k\rangle \\
&\quad\geq\ \alpha_k\beta_k\langle\nabla p(\lambda_k),\lambda_k-\lambda^*\rangle\ \geq\ \alpha_k\beta_k\left([p(\lambda_k)-p(\lambda^*)]+\frac{1}{2L_p}\|\nabla p(\lambda_k)\|^2\right) \\
&\quad\geq\ \alpha_k\beta_k\left([p(\lambda_k)-p(\lambda^*)]+\frac{1}{2L_p}(1-e_k)^2\|p_k'\|^2\right).
\end{aligned}$$

Letting $x=\|p_k'\|/(L_p\|\lambda_k-\lambda^*\|)$ and rearranging the above inequality, we conclude that

$$\alpha\beta_k[p(\lambda_k)-p(\lambda^*)]\leq\frac{L_p}{2}\left[\left(1+2\alpha_k e_k x-\alpha_k\beta_k(1-e_k)^2x^2\right)\|\lambda_k-\lambda^*\|^2-\|\lambda_{k+1}-\lambda^*\|^2\right].$$

Relation (5.3.10) now follows from the above inequality by noting that (5.3.7) and (5.3.9) imply that

$$\beta_k\geq(1-e_k)^2\beta_k=(1-e_k)^2-\frac{\alpha_k}{2}\geq\frac{c_1}{2} \tag{5.3.11}$$

and that the quadratic function $1+2\alpha_k e_k x-\alpha_k\beta_k(1-e_k)^2x^2$ is bounded above by

$$1+\frac{\alpha_k e_k^2}{\beta_k(1-e_k)^2}\leq 1+\frac{2\alpha_k e_k^2}{c_1}.$$

■

The following theorem states the convergence properties of the inexact steepest descent method described above.

Theorem 5.3.1 *Assume that for some positive constants c_1, we have*

$$e_k\leq 1-\sqrt{\frac{\alpha_k+c_1}{2}} \tag{5.3.12}$$

for every $k\geq 0$. Then, the sequence $\{\lambda_k\}$ generated by the inexact steepest descent method (5.3.4) satisfies

$$p(\lambda_k)-p^*\leq\frac{L_p}{c_1\sum_{i=0}^k\alpha_i}\left[\|\lambda_0-\lambda^*\|^2\exp\left(\sum_{i=0}^k\frac{2\alpha_i e_i^2}{c_1}\right)-\|\lambda_{k+1}-\lambda^*\|^2\right] \tag{5.3.13}$$

for every $\lambda^*\in\Lambda^*$, where p^* is defined in (5.3.4).

Proof. Using Lemma 5.3.4, it is easy to see by induction that

$$\sum_{i=0}^{k} \alpha_i [p(\lambda_i) - p^*] \leq \frac{L_p}{c_1} \left[\|\lambda_0 - \lambda^*\|^2 \prod_{i=0}^{k} \left(1 + \frac{2\alpha_k e_k^2}{c_1} \right) - \|\lambda_{k+1} - \lambda^*\|^2 \right] \tag{5.3.14}$$

for every $k \geq 0$. The above inequality, Lemmas 5.3.2 and 5.3.4, the inequality $\log(1+x) \leq x$ for any $x > -1$ and assumption (5.3.12) then imply that

$$\begin{aligned} \left(\sum_{i=0}^{k} \alpha_i \right) [p(\lambda_k) - p^*] \; &\leq \; \sum_{i=0}^{k} \alpha_i [p(\lambda_i) - p^*] \\ &\leq \; \frac{L_p}{c_1} \left[\|\lambda_0 - \lambda^*\|^2 \exp \left(\sum_{i=0}^{k} \log(1 + 2\alpha_i e_i^2 / c_1) \right) - \|\lambda_{k+1} - \lambda^*\| \right] \\ &\leq \; \frac{L_p}{c_1} \left[\|\lambda_0 - \lambda^*\|^2 \exp \left(\sum_{i=0}^{k} \frac{2\alpha_i e_i^2}{c_1} \right) - \|\lambda_{k+1} - \lambda^*\| \right] \end{aligned}$$

for every $k \geq 0$. ∎

As a consequence of Theorem 5.3.1, we obtain the following result which gives an upper bound on the quantities $\|\nabla p(\lambda_k)\|$ and $\|p'(\lambda_k)\|$.

Corollary 5.3.1 *Assume that, for some positive constant c_1, relation (5.3.12) holds for every $k \geq 0$. Then, the sequence $\{\lambda_k\}$ generated by the inexact steepest descent method (5.3.4) satisfies*

$$\frac{\alpha_k + c_1}{2} \|p'_k\|^2 \leq \|\nabla p(\lambda_k)\|^2 \leq \frac{2L_p^2 \|\lambda_0 - \lambda^*\|^2}{c_1 \sum_{i=0}^{k} \alpha_i} \exp \left(\sum_{i=0}^{k} \frac{2\alpha_i e_i^2}{c_1} \right) \tag{5.3.15}$$

for every $\lambda^* \in \Lambda^*$.

Proof. Clearly by definition of e_k, we have $\|\nabla p(\lambda_k)\| \geq (1 - e_k)\|p'_k\|$, which together with (5.3.12), imply that $\|\nabla p(\lambda_k)\|^2 \geq (\alpha_k + c_1)\|p'_k\|^2/2$. Moreover, using (5.1.5), (5.3.13), and the fact that $\nabla p(\lambda^*) = 0$, we conclude that

$$\|\nabla p(\lambda_k)\|^2 \leq 2L_p(p(\lambda_k) - p^*) \leq \frac{2L_p^2 \|\lambda_0 - \lambda^*\|^2}{c_1 \sum_{i=0}^{k} \alpha_i} \exp \left(\sum_{i=0}^{k} \frac{2\alpha_i e_i^2}{c_1} \right).$$

Our claim clearly follows from the above two observations. ∎

5.4 Convergence Analysis

In this section, we prove the main results presented in Subsections 5.2.3 and 5.2.4.

5.4.1 Convergence analysis for the I-AL method

The goal of this subsection is to prove the convergence results for the I-AL method stated in Subsection 5.2.3, namely: Proposition 5.2.4, Proposition 5.2.5 and Theorems 5.2.1, 5.2.2, 5.2.3 and 5.2.4.

We first give the proof of Proposition 5.2.4 which guarantees that subroutine PostProcessing of the I-AL method outputs an (ϵ_p, ϵ_d)-primal-dual solution of (5.1.1).

Proof of Proposition 5.2.4: Clearly, by Lemma 5.3.1(b) with $\phi(\cdot) = \mathcal{L}_\rho(\cdot, \tilde{\lambda})$ and $L_\phi = M_\rho$, we have

$$\|\nabla \mathcal{L}_\rho(\tilde{x}, \tilde{\lambda})]_X^{1/M_\rho}\| \leq \left\{2M_\rho \left[\mathcal{L}_\rho(\tilde{x}, \tilde{\lambda}) - d_\rho(\tilde{\lambda})\right]\right\}^{\frac{1}{2}} \leq \sqrt{2M_\rho\, \zeta} \leq \frac{\epsilon_d}{2},$$

where the second and last inequalities follow from the assumption that $\mathcal{L}_\rho(\tilde{x}, \tilde{\lambda}) - d_\rho(\tilde{\lambda}) \leq \zeta$ and relation (5.2.19), respectively. The above inequality together with (5.2.7), (5.2.21) and Proposition 5.3.1(b) with $\phi(\cdot) = \mathcal{L}_\rho(\cdot, \tilde{\lambda})$, $L_\phi = M_\rho$ and $\tau = 1/M_\rho$ then imply that

$$\nabla f(\tilde{x}^+) + (\mathcal{A}_0)^* \tilde{\lambda}^+ = \nabla f(\tilde{x}^+) + (\mathcal{A}_0)^*(\tilde{\lambda} + \rho \mathcal{A}(\tilde{x}^+)) = \nabla \mathcal{L}_\rho(\tilde{x}^+, \tilde{\lambda}) \in -\mathcal{N}_X(\tilde{x}^+) + \mathcal{B}(\epsilon_d),$$

where $\tilde{x}^+$ is defined in (5.2.20). Moreover, it follows from Lemma 5.3.1(a) with $\phi(\cdot) = \mathcal{L}_\rho(\cdot, \tilde{\lambda})$, $L_\phi = M_\rho$ and $\tau = 1/M_\rho$ that $\mathcal{L}_\rho(\tilde{x}^+, \tilde{\lambda}) \leq \mathcal{L}_\rho(\tilde{x}, \tilde{\lambda})$. This observation, the assumption that $\mathcal{L}_\rho(\tilde{x}, \tilde{\lambda}) - d_\rho(\tilde{\lambda}) \leq \zeta$ and (5.2.19) then imply that

$$\mathcal{L}_\rho(\tilde{x}^+, \tilde{\lambda}) - d_\rho(\tilde{\lambda}) \leq \mathcal{L}_\rho(\tilde{x}, \tilde{\lambda}) - d_\rho(\tilde{\lambda}) \leq \zeta \leq \frac{\rho \epsilon_p^2}{128}.$$

Using this conclusion, the assumption that $\mathcal{L}_\rho(x, \tilde{\lambda}) - d_\rho(\tilde{\lambda}) \leq \rho\epsilon_p^2/128$ and Proposition 5.2.3, we then obtain

$$\max\{\|\mathcal{A}(\tilde{x}^+) - u_\lambda^*\|, \|\mathcal{A}(x) - u_\lambda^*\|\} \leq \frac{\epsilon_p}{8},$$

which together with the assumption that $\|\mathcal{A}(x)\| \leq 3\epsilon_p/4$ imply

$$\|\mathcal{A}(\tilde{x}^+)\| \leq \|\mathcal{A}(\tilde{x}^+) - u_\lambda^*\| + \|\mathcal{A}(x) - u_\lambda^*\| + \|\mathcal{A}(x)\| \leq \frac{\epsilon_p}{8} + \frac{\epsilon_p}{8} + \frac{3\epsilon_p}{4} = \epsilon_p. \tag{5.4.1}$$

We have thus shown that $(\tilde{x}^+, \tilde{\lambda}^+)$ is an (ϵ_p, ϵ_d)-primal-dual solution of (5.1.1). ∎

Theorem 5.2.1 states certain conditions on the parameters ρ and η_k which guarantee

that the I-AL method will successfully terminate in at most N outer iterations. We now give a proof of this result.

Proof of Theorem 5.2.1: Since $\bar{N} \geq N$ by assumption, the I-AL method does not terminate with failure within the first N outer iterations. Assume for contradiction that the I-AL method does not successfully terminate within the first N outer iterations. This implies that $\|\mathcal{A}(x_k)\| > 3\epsilon_p/4$ for all $0 \leq k \leq N-1$. Letting $\delta_k := \|\mathcal{A}(x_k) - u^*_{\lambda_k}\|$ and $e_k := \delta_k/\|\mathcal{A}(x_k)\|$ for all $k \geq 0$, we conclude from the previous observation, (5.2.18), Proposition 5.2.3 and assumptions (5.2.22) and (5.2.23) that

$$\sum_{k=0}^{N-1} e_k^2 = \sum_{k=0}^{N-1} \frac{\delta_k^2}{\|\mathcal{A}(x_k)\|^2} \leq \frac{16}{9\epsilon_p^2} \sum_{k=0}^{N-1} \|\mathcal{A}(x_k) - u^*_{\lambda_k}\|^2 \leq \frac{32}{9\rho\epsilon_p^2} \sum_{k=0}^{N-1} \eta_k \leq \frac{32}{9\rho\epsilon_p^2} \sum_{k=0}^{\bar{N}-1} \eta_k \leq \frac{1}{36}. \tag{5.4.2}$$

Noting that (5.4.2) implies $e_k \leq 1/6$, and hence that condition (5.3.12) holds with $\alpha_k = 1$ and $c_1 = 7/18$, it follows from (5.4.2) and Corollary 5.3.1 with $p(\cdot) = -d_\rho(\cdot)$, $L_p = 1/\rho$, $p'_k = \mathcal{A}(x_k)$, $c_1 = 7/18$ and $\alpha_k = 1$ that

$$\|\mathcal{A}(x_k)\|^2 \leq \frac{4D_\Lambda^2}{c_1(1+c_1)\,\rho^2(k+1)} \exp\left(\frac{2}{c_1}\sum_{j=0}^{k} e_j^2\right) \leq \frac{1296 D_\Lambda^2}{175\rho^2(k+1)} \exp\left(\frac{1}{7}\right) \leq \frac{9D_\Lambda^2}{\rho^2(k+1)}, \tag{5.4.3}$$

for every $0 \leq k \leq N-1$. The above inequality with $k = N-1$ together with (5.2.22) then imply that

$$\|\mathcal{A}(x_{N-1})\|^2 \leq \frac{9D_\Lambda^2}{\rho^2 N} \leq \frac{9\epsilon_p^2}{16},$$

which clearly contradicts the fact $\|\mathcal{A}(x_{N-1})\| > 3\epsilon_p/4$. ∎

Theorem 5.2.2 bounds the total number of inner iterations of the I-AL method when a summable sequence $\{\eta_k\}$ satisfying (5.2.23) is used. Before proving this theorem, we first state the following two technical results.

The proof of the following result is given in Appendix A.

Proposition 5.4.1 *For some positive integer L, let positive scalars $p_1, p_2, \cdots, p_L$ be given. Then, there exists a constant $C = C(p_1, \cdots, p_L)$ such that for any nonnegative scalars*

$\beta_0, \beta_1, \cdots, \beta_L$, ν, and $\bar{t}$, we have

$$\sum_{k=0}^{K} \left\lceil \beta_0 + \sum_{l=1}^{L} \left(\beta_l \, t_k^{p_l}\right) \right\rceil \max\left\{1, \left\lceil \log \frac{\nu}{t_k} \right\rceil \right\} \leq C \left\lceil \beta_0 + \sum_{l=1}^{L} \left(\beta_l \, \bar{t}^{p_l}\right) \right\rceil \max\left\{1, \left\lceil \log \frac{\nu}{\bar{t}} \right\rceil \right\}, \tag{5.4.4}$$

where

$$K := \max\left\{0, \left\lceil \log\left(\frac{\bar{t}}{t_0}\right) \right\rceil \right\}, \; t_0 := \min_{1 \leq l \leq L} \left(\frac{\max(\beta_0, 1)}{\beta_l}\right)^{1/p_l}, \; t_k = t_0 2^k, \; \forall k = 1, \ldots, K. \tag{5.4.5}$$

In particular, if $\nu = \bar{t}$, then (5.4.4) implies that

$$\sum_{k=0}^{K} \left\lceil \beta_0 + \sum_{l=1}^{L} \left(\beta_l \, t_k^{p_l}\right) \right\rceil \leq C \left\lceil \beta_0 + \sum_{l=1}^{L} \left(\beta_l \, \bar{t}^{p_l}\right) \right\rceil. \tag{5.4.6}$$

Lemma 5.4.1 *The following statements hold:*

a) for every $t \geq 1$ and $a, b \geq 0$, we have $(a+b)^t \leq [(2a)^t + (2b)^t]/2$;

b) for any $K \geq 1$ and $\xi > 0$, we have

$$\sum_{k=0}^{+\infty} (k+1)^{-(1+\xi)} \leq 1 + \int_0^{+\infty} (t+1)^{-(1+\xi)} dt \leq (\xi+1)/\xi, \tag{5.4.7}$$

$$\sum_{k=0}^{K-1} (k+1)^{\xi} \leq \int_0^{K} (t+1)^{\xi} dt \leq \frac{1}{1+\xi} (K+1)^{1+\xi}. \tag{5.4.8}$$

Proof. Statement a) follows directly from the convexity of x^t for any $x \geq 0$ and b) is obvious. ∎

We are now ready to prove Theorem 5.2.2.

Proof of Theorem 5.2.2: We first show that condition (5.2.23) holds. Indeed, by (5.2.27), (5.4.7) and the assumption that $\bar{N} = +\infty$, we have

$$\sum_{k=0}^{\bar{N}-1} \eta_k = \frac{\xi \rho \epsilon_p^2}{128(\xi+1)} \sum_{k=0}^{\infty} \frac{1}{(k+1)^{1+\xi}} \leq \frac{\rho \epsilon_p^2}{128}.$$

It then follows from Proposition 5.2.5 that the method will successfully terminate in N outer iterations and the total number of inner iterations is bounded by $\mathcal{I}_p + \mathcal{I}_d$, where N,

$\mathcal{I}_p$ and $\mathcal{I}_d$ are defined in (5.2.22) and (5.2.25). Observe that by (5.2.27), (5.4.8), (5.2.22) and Lemma 5.4.1(a) with $a = 16D_\Lambda^2/(\rho^2\epsilon_p^2)$, $b = 2$ and $t = (3+\xi)/2$, we have

$$\begin{aligned}
\sum_{k=0}^{N-1} \eta_k^{-\frac{1}{2}} &= \frac{8\sqrt{2}(1+\xi)^{\frac{1}{2}}}{\xi^{\frac{1}{2}}\rho^{\frac{1}{2}}\epsilon_p} \sum_{k=0}^{N-1} (k+1)^{\frac{1+\xi}{2}} \leq \frac{16\sqrt{2}(1+\xi)^{\frac{1}{2}}}{(3+\xi)\xi^{\frac{1}{2}}\rho^{\frac{1}{2}}\epsilon_p} (N+1)^{\frac{3+\xi}{2}} \\
&\leq \frac{16\sqrt{2}(1+\xi)^{\frac{1}{2}}}{(3+\xi)\xi^{\frac{1}{2}}\rho^{\frac{1}{2}}\epsilon_p} \left(\frac{16D_\Lambda^2}{\rho^2\epsilon_p^2} + 2\right)^{\frac{3+\xi}{2}} = \frac{64C(\xi)}{\rho^{\frac{1}{2}}\epsilon_p} \left(\frac{8D_\Lambda^2}{\rho^2\epsilon_p^2} + 1\right)^{\frac{3+\xi}{2}} \\
&\leq \frac{32C(\xi)}{\rho^{\frac{1}{2}}\epsilon_p} \left[\left(\frac{4D_\Lambda}{\rho\epsilon_p}\right)^{3+\xi} + 2^{\frac{3+\xi}{2}}\right],
\end{aligned}$$

where $C(\xi) := (1+\xi)^{\frac{1}{2}} 2^{\frac{\xi}{2}}/[(3+\xi)\xi^{\frac{1}{2}}]$. This relation together with (5.2.22) and (5.2.25) then imply that

$$\begin{aligned}
\mathcal{I}_p &\leq \sqrt{2} D_X M_\rho^{\frac{1}{2}} \sum_{k=0}^{N-1} \eta_k^{-\frac{1}{2}} + N \\
&\leq \frac{32\sqrt{2}C(\xi) D_X M_\rho^{\frac{1}{2}}}{\rho^{\frac{1}{2}}\epsilon_p} \left[\left(\frac{4D_\Lambda}{\rho\epsilon_p}\right)^{3+\xi} + 2^{\frac{3+\xi}{2}}\right] + \frac{16D_\Lambda^2}{\rho^2\epsilon_p^2} + 1.
\end{aligned}$$

Moreover, it can be easily seen from (5.2.25) that

$$\mathcal{I}_d \leq 4D_X \left\{\frac{4M_\rho^{\frac{1}{2}}}{\rho^{\frac{1}{2}}\epsilon_p} + \frac{M_\rho}{\epsilon_d}\right\} + 1.$$

Combining the previous two inequalities, we immediately see that the the total number of inner iterations performed by the I-AL method is bounded by (5.2.28).

Assume now that ρ is chosen as in (5.2.29). Then, bound (5.2.30) follows by combining the definition of N in (5.2.22) with the fact that by (5.2.29),

$$\rho \geq \max\left\{\frac{1}{\epsilon_p}\left(\frac{(D_\Lambda)^{3+\xi}\epsilon_d}{\|\mathcal{A}\|}\right)^{\frac{1}{4+\xi}}, \frac{L_f}{\|\mathcal{A}\|^2}\right\}. \tag{5.4.9}$$

Also, (5.4.9) implies that $\rho \geq L_f/\|\mathcal{A}\|^2$, and hence that

$$M_\rho = L_f + \rho\|\mathcal{A}\|^2 \leq 2\rho\|\mathcal{A}\|^2 = \frac{2\|\mathcal{A}\|^2}{\epsilon_p}\left(\frac{(D_\Lambda)^{3+\xi}\epsilon_d}{\|\mathcal{A}\|}\right)^{\frac{1}{4+\xi}} + 2L_f = 2\epsilon_d\left(\frac{\|A\|^{\frac{7+2\xi}{4+\xi}} D_\Lambda^{\frac{3+\xi}{4+\xi}}}{\epsilon_p\epsilon_d^{\frac{3+\xi}{4+\xi}}} + \frac{L_f}{\epsilon_d}\right). \tag{5.4.10}$$

Hence, bound (5.2.28) is majorized by

$$\mathcal{O}\left(\frac{D_X\|\mathcal{A}\|}{\epsilon_p}\left[\left(\frac{D_\Lambda}{\rho\epsilon_p}\right)^{3+\xi} + 1\right] + D_X\left(\frac{\|A\|^{\frac{7+2\xi}{4+\xi}} D_\Lambda^{\frac{3+\xi}{4+\xi}}}{\epsilon_p\epsilon_d^{\frac{3+\xi}{4+\xi}}} + \frac{L_f}{\epsilon_d}\right) + \frac{D_\Lambda^2}{\rho^2\epsilon_p^2} + 1\right). \tag{5.4.11}$$

Also, by (5.4.9), we have

$$\frac{D_\Lambda}{\rho\epsilon_p} \leq D_\Lambda \left(\frac{\|\mathcal{A}\|}{D_\Lambda^{3+\xi}\epsilon_d} \right)^{\frac{1}{4+\xi}} = \left(\frac{D_\Lambda \|\mathcal{A}\|}{\epsilon_d} \right)^{\frac{1}{4+\xi}}.$$

Substituting the above inequality into (5.4.11), we obtain bound (5.2.31). ∎

Theorem 5.2.3 provides a bound on the total number inner iterations of the I-AL method when a uniform sequence $\{\eta_k\}$ is used, under the assumption that an upper bound t on D_Λ, is known. We will now provide a proof of Theorem 5.2.3.

Proof of Theorem 5.2.3 Using (5.2.32) and the assumption that $t \geq D_\Lambda$, we obtain

$$\bar{N}(t) \geq \left\lceil \frac{16 D_\Lambda^2}{\rho^2 \epsilon_p^2} \right\rceil = N. \tag{5.4.12}$$

Also note that (5.2.32) and (5.2.33) imply that

$$\sum_{k=0}^{\bar{N}-1} \eta_k = \bar{N}\eta(t) = \bar{N}(t)\eta(t) = \frac{\rho\epsilon_p^2}{128}.$$

We have thus shown that conditions (5.2.22) and (5.2.23) hold. It then follows from Proposition 5.2.5 that the total number of outer iterations is bounded by N, where N is defined by (5.2.22). Bound (5.2.34) now follows by combining the definition of N in (5.2.22) with the fact that

$$\rho = \rho(t) \geq \max\left\{ \frac{4t^{\frac{3}{4}}}{\|\mathcal{A}\|^{\frac{1}{4}}\epsilon_p}, \frac{L_f}{\|\mathcal{A}\|^2} \right\} \geq \max\left\{ \frac{4D_\Lambda^{\frac{3}{4}}}{\|\mathcal{A}\|^{\frac{1}{4}}\epsilon_p}, \frac{L_f}{\|\mathcal{A}\|^2} \right\}. \tag{5.4.13}$$

It also follows from Proposition 5.2.5 that the total number of inner iterations is bounded by $\mathcal{I}_p + \mathcal{I}_d$, where $\mathcal{I}_p$ and $\mathcal{I}_d$ are given by (5.2.25). Noting that by (5.2.32), (5.2.33) and Lemma 5.4.1(a) with $a = 16t^2/(\rho^2\epsilon_p^2)$, $b = 1$ and $t = 3/2$, we have

$$\sum_{k=0}^{\bar{N}(t)-1} \eta_k^{-\frac{1}{2}} = \frac{8\sqrt{2}}{\rho(t)^{\frac{1}{2}}\epsilon_p} \bar{N}(t)^{\frac{3}{2}} \leq \frac{8\sqrt{2}}{\rho(t)^{\frac{1}{2}}\epsilon_p} \left(\frac{16t^2}{\rho(t)^2\epsilon_p^2} + 1 \right)^{\frac{3}{2}} \leq \frac{16}{\rho(t)^{\frac{1}{2}}\epsilon_p} \left(\frac{64t^3}{\rho(t)^3\epsilon_p^3} + 1 \right),$$

we then conclude from (5.2.25), (5.2.32) and (5.4.12) that

$$\mathcal{I}_p \leq \sqrt{2} D_X M_\rho^{\frac{1}{2}} \sum_{k=0}^{\bar{N}(t)-1} \eta_k^{-\frac{1}{2}} + \bar{N}(t) \leq \frac{16\sqrt{2} D_X M_\rho^{\frac{1}{2}}}{\rho(t)^{\frac{1}{2}}\epsilon_p} \left(\frac{64t^3}{\rho(t)^3\epsilon_p^3} + 1 \right) + \frac{16t^2}{\rho(t)^2\epsilon_p^2} + 1. \tag{5.4.14}$$

Now, by using the first relation in (5.2.32), we have that $\rho(t) \geq L_f/\|\mathcal{A}\|^2$, and hence that

$$M_\rho = L_f + \rho(t)\|\mathcal{A}\|^2 \leq 2\rho(t)\|\mathcal{A}\|^2. \tag{5.4.15}$$

This conclusion together with (5.4.13) and (5.4.14) then imply that

$$\begin{aligned}\mathcal{I}_p &\leq \frac{32D_X\|\mathcal{A}\|}{\epsilon_p}\left(\frac{64t^3}{\rho(t)^3\epsilon_p^3}+1\right)+\frac{16t^2}{\rho(t)^2\epsilon_p^2}+1 \\ &\leq \frac{32D_X\|\mathcal{A}\|}{\epsilon_p}\left(\frac{\|\mathcal{A}\|^{\frac{3}{4}}t^{\frac{3}{4}}}{\epsilon_d^{\frac{3}{4}}}+1\right)+\frac{\|\mathcal{A}\|^{\frac{1}{2}}t^{\frac{1}{2}}}{\epsilon_d^{\frac{1}{2}}}+1. \qquad (5.4.16)\end{aligned}$$

Moreover, it easily follows from (5.2.25), (5.4.15) and (5.2.32) that

$$\begin{aligned}\mathcal{I}_d &\leq 4D_X\left(\frac{4M_\rho^{\frac{1}{2}}}{\rho(t)^{\frac{1}{2}}\epsilon_p}+\frac{M_\rho}{\epsilon_d}\right)+1 \leq 4D_X\left(\frac{4\sqrt{2}\|\mathcal{A}\|}{\epsilon_p}+\frac{2\rho(t)\|\mathcal{A}\|^2}{\epsilon_d}\right)+1 \\ &= \frac{16\sqrt{2}D_X\|\mathcal{A}\|}{\epsilon_p}+8D_X\left(\frac{4t^{\frac{3}{4}}\|\mathcal{A}\|^{\frac{7}{4}}}{\epsilon_p\epsilon_d^{\frac{3}{4}}}+\frac{L_f}{\epsilon_d}\right)+1. \qquad (5.4.17)\end{aligned}$$

Combining (5.4.16) and (5.4.17), we easily see that the I-AL method computes an (ϵ_p, ϵ_d)-primal-dual solution of (5.1.1) in at most $\mathcal{O}(\mathcal{I}_{pd}(t))$ inner iterations, where $\mathcal{I}_{pd}(t)$ is defined by (5.2.35). ∎

We now give the proof of Theorem 5.2.4, which establishes the iteration-complexity of the I-AL guess-and-check procedure.

Proof of Theorem 5.2.4 Suppose that the I-AL guess-and-check procedure terminates when the iteration count j is equal to J. Letting

$$\bar{J} := \max\{0, \lceil \log(D_\Lambda/t_0)\rceil\} \qquad (5.4.18)$$

and noting that $t_{\bar{J}} = t_0 2^{\bar{J}} \geq D_\Lambda$, we conclude from Theorem 5.2.3 that $J \leq \bar{J}$. Let $\mathcal{I}_{p,j}$, $j = 1, \ldots, J$, denote the number of inner iterations performed at step 1) of the I-AL method during loop j of the I-AL guess-and-check procedure, and let $\mathcal{I}_{d,J}$ denote the number of inner iterations performed by subroutine Postprocessing during loop J of the I-AL guess-and-check procedure. Then, the overall number of inner iterations performed by the I-AL guess-and-check procedure is bounded by

$$\sum_{j=0}^{J}\mathcal{I}_{p,j}+\mathcal{I}_{d,J} \leq \sum_{j=0}^{\bar{J}}\mathcal{I}_{p,j}+\mathcal{I}_{d,J}. \qquad (5.4.19)$$

Since the total number of outer iterations at the jth loop is bounded by $N(t_j)$, it follows

from Corollary 1.1.1 that

$$\mathcal{I}_{p,j} \leq \sum_{k=0}^{\bar{N}(t_j)-1} \left\lceil D_X \sqrt{\frac{2M_\rho}{\eta_k}} \right\rceil \leq \sqrt{2} D_X M_\rho^{\frac{1}{2}} \sum_{k=0}^{\bar{N}(t_j)-1} \eta_k^{-\frac{1}{2}} + \bar{N}(t_j).$$

Hence, similar to the proof of (5.4.14), (5.4.15) and (5.4.16), we can show that for $j = 0, \ldots, J$, we have

$$\mathcal{I}_{p,j} \leq 32 D_X \left[\frac{t_j^{\frac{3}{4}} \|\mathcal{A}\|^{\frac{7}{4}}}{\epsilon_p \epsilon_d^{\frac{3}{4}}} + \frac{\|\mathcal{A}\|}{\epsilon_p} \right] + \frac{t_j^{\frac{1}{2}} \|\mathcal{A}\|^{\frac{1}{2}}}{\epsilon_d^{\frac{1}{2}}} + 1 \leq \left\lceil \beta_0 + \beta_1 t_j^{\frac{3}{4}} + \beta_2 t_j^{\frac{1}{2}} \right\rceil,$$

where β_0, β_1, and β_2 are given by (5.2.36). Noting that $t_j = t_0 2^j$ for every j and the definition of t_0 in step 0) of the I-AL guess-and-check procedure, it follows from the previous inequality and relation (5.4.6) with $L = 2$, $p_1 = 3/4$, $p_2 = 1/2$, $\bar{t} = D_\Lambda$, $J = \bar{J}$, and β_0, β_1, and β_2 as above that

$$\sum_{j=0}^{\bar{J}} \mathcal{I}_{p,j} = \mathcal{O}(1) \left\lceil \beta_0 + \beta_1 D_\Lambda^{\frac{3}{4}} + \beta_2 D_\Lambda^{\frac{1}{2}} \right\rceil. \tag{5.4.20}$$

Now, using (5.4.18), it is easy to see that $t_J \leq t_{\bar{J}} \leq \max\{t_0, 2D_\Lambda\}$ and hence that

$$t_J^{\frac{3}{4}} \leq \max\left\{ t_0^{\frac{3}{4}}, (2D_\Lambda)^{\frac{3}{4}} \right\} \leq \max\left\{ \frac{\beta_0}{\beta_1}, (2D_\Lambda)^{\frac{3}{4}} \right\} \leq \frac{\beta_0}{\beta_1} + (2D_\Lambda)^{\frac{3}{4}}, \tag{5.4.21}$$

where the last inequality is due to the definition of t_0 in Step 0 of the I-AL guess-and-check procedure. Using this inequality, the definition of β_0 and β_1 in (5.2.36), and an argument similar to the proof of (5.4.17), we have

$$\begin{aligned} \mathcal{I}_{d,J} &\leq \frac{16\sqrt{2} D_X \|\mathcal{A}\|}{\epsilon_p} + 8 D_X \left[\frac{4 t_J^{\frac{3}{4}} \|\mathcal{A}\|^{\frac{7}{4}}}{\epsilon_p \epsilon_d^{\frac{3}{4}}} + \frac{L_f}{\epsilon_d} \right] + 1 \\ &\leq \beta_0 + \beta_1 t_J^{\frac{3}{4}} + \frac{8 D_X L_f}{\epsilon_d} \leq 2\beta_0 + \beta_1 (2D_\Lambda)^{\frac{3}{4}} + \frac{8 D_X L_f}{\epsilon_d}. \end{aligned} \tag{5.4.22}$$

Now, using (5.4.20) and (5.4.22), it is easy to see that the right-high-side of (5.4.19) is bounded by $\mathcal{O}(\mathcal{I}_{pd}(D_\Lambda))$, where $\mathcal{I}_{pd}(\cdot)$ is defined in (5.2.35). ∎

5.4.2 Convergence analysis for the I-AL method applied to the perturbed problem

The goal of this subsection is to prove the convergence results stated in Subsection 5.2.4, namely, Proposition 5.2.6 and Theorems 5.2.5 and 5.2.6.

We first prove Proposition 5.2.6 which guarantees that subroutine Postprocessing of the

modified I-AL method outputs an (ϵ_p, ϵ_d)-primal-dual solution of (5.1.1).

Proof of Proposition 5.2.6: As in the proof of Proposition 5.2.4 with ζ replaced by $\tilde{\zeta}$, we can show that

$$\nabla f_\gamma(\tilde{x}^+) + (\mathcal{A}_0)^* \tilde{\lambda}^+ \in -\mathcal{N}_X(\tilde{x}^+) + \mathcal{B}\left(\frac{\epsilon_d}{2}\right),$$

where $\tilde{x}^+$ is defined in (5.2.20) with $\mathcal{L}_\rho$ replaced by $\mathcal{L}_{\rho,\gamma}$. Noting that

$$\nabla f_\gamma(\tilde{x}^+) = \nabla f(\tilde{x}^+) + \gamma(\tilde{x}^+ - x_0)$$

and that (5.2.26) and (5.2.43) imply that

$$\gamma\|\tilde{x}^+ - x_0\| \leq \gamma D_X = \frac{\epsilon_d}{2},$$

we then conclude that

$$\nabla f(\tilde{x}^+) + (\mathcal{A}_0)^* \tilde{\lambda}^+ \in -\mathcal{N}_X(\tilde{x}^+) + \mathcal{B}(\epsilon_d).$$

Moreover, similar to the proof of Proposition 5.2.4, we can show that $\|\mathcal{A}(\tilde{x}^+)\| \leq \epsilon_p$. Thus, $(\tilde{x}^+, \lambda_k^+)$ is an (ϵ_p, ϵ_d)-primal-dual solution for (5.1.1). ∎

Theorems 5.2.5 provides a bound on the total number of inner iterations performed by the modified I-AL method. Before proving this result, we first present two technical lemmas. The first one stated below establishes an important technical result that allows us to take the advantage of the "warm-start" strategy described in the end of Subsection 5.2.3.

Lemma 5.4.2 *Let $(x_k, \lambda_k) \in X \times \Re^m$ be given and let $\lambda_{k+1} = \lambda_k + \rho\mathcal{A}(x_k)$. If $\mathcal{L}_{\rho,\gamma}(x_k, \lambda_k) - d_{\rho,\gamma}(\lambda_k) \leq \eta_k$, then*

$$\frac{\gamma}{2}\|x_k - x_{k+1}^*\|^2 \leq \mathcal{L}_{\rho,\gamma}(x_k, \lambda_{k+1}) - d_{\rho,\gamma}(\lambda_{k+1}) \leq \left(\sqrt{\eta_k} + \sqrt{\frac{\rho}{2}}\|\mathcal{A}(x_k)\|\right)^2, \qquad (5.4.23)$$

where x_{k+1}^* is the unique solution of $\min_{x\in X} \mathcal{L}_{\rho,\gamma}(x, \lambda_{k+1})$.

Proof. The first inequality in (5.4.23) follows immediately from the strong convexity of $\mathcal{L}_{\rho,\gamma}(\cdot, \lambda_{k+1})$. Hence, it suffices to show the second inequality in (5.4.23). Clearly, by definition (5.2.39) and the fact that $\lambda_{k+1} = \lambda_k + \rho\mathcal{A}(x_k)$, we have

$$\mathcal{L}_{\rho,\gamma}(x_k, \lambda_{k+1}) - \mathcal{L}_{\rho,\gamma}(x_k, \lambda_k) = \rho\|\mathcal{A}(x_k)\|^2.$$

The above observation together with the assumption $\mathcal{L}_{\rho,\gamma}(x_k,\lambda_k)-d_{\rho,\gamma}(\lambda_k)\leq\eta_k$ then imply that

$$\begin{aligned}
&\mathcal{L}_{\rho,\gamma}(x_k,\lambda_{k+1})-d_{\rho,\gamma}(\lambda_{k+1})\\
&= [\mathcal{L}_{\rho,\gamma}(x_k,\lambda_{k+1})-\mathcal{L}_{\rho,\gamma}(x_k,\lambda_k)]+[\mathcal{L}_{\rho,\gamma}(x_k,\lambda_k)-d_{\rho,\gamma}(\lambda_{k+1})]\\
&= \rho\|\mathcal{A}(x_k)\|^2+[\mathcal{L}_{\rho,\gamma}(x_k,\lambda_k)-d_{\rho,\gamma}(\lambda_k)]+[d_{\rho,\gamma}(\lambda_k)-d_{\rho,\gamma}(\lambda_{k+1})]\\
&\leq \rho\|\mathcal{A}(x_k)\|^2+\eta_k+[d_{\rho,\gamma}(\lambda_k)-d_{\rho,\gamma}(\lambda_{k+1})]. \qquad (5.4.24)
\end{aligned}$$

Moreover, in view of Proposition 5.2.2 applied to the perturbed problem (5.2.37), the function $d_{\rho,\gamma}(\cdot)$ is concave and has $1/\rho$-Lipschitz-continuous gradient and $\nabla d_{\rho,\gamma}(\lambda)=u^*_{\lambda,\gamma}$. It then follows from (5.1.5) that

$$\begin{aligned}
-d_{\rho,\gamma}(\lambda_{k+1})+d_{\rho,\gamma}(\lambda_k) &\leq \langle -u^*_{\lambda_k,\gamma},\lambda_{k+1}-\lambda_k\rangle+\frac{1}{2\rho}\|\lambda_{k+1}-\lambda_k\|^2\\
&= -\rho\langle u^*_{\lambda_k,\gamma},\mathcal{A}(x_k)\rangle+\frac{\rho}{2}\|\mathcal{A}(x_k)\|^2, \qquad (5.4.25)
\end{aligned}$$

where the last equality follows from the fact that $\lambda_{k+1}-\lambda_k=\rho\mathcal{A}(x_k)$. Combining (5.4.24) and (5.4.25), we obtain

$$\begin{aligned}
&\mathcal{L}_{\rho,\gamma}(x_k,\lambda_{k+1})-d_{\rho,\gamma}(\lambda_{k+1})\leq\eta_k+\rho\langle\mathcal{A}(x_k)-u^*_{\lambda_k,\gamma},\mathcal{A}(x_k)\rangle+\frac{\rho}{2}\|\mathcal{A}(x_k)\|^2\\
&\leq \eta_k+\rho\|\mathcal{A}(x_k)-u^*_{\lambda_k,\gamma}\|\|\mathcal{A}(x_k)\|+\frac{\rho}{2}\|\mathcal{A}(x_k)\|^2\\
&\leq \eta_k+\sqrt{2\rho\,\eta_k}\,\|\mathcal{A}(x_k)\|+\frac{\rho}{2}\|\mathcal{A}(x_k)\|^2=\left(\sqrt{\eta_k}+\sqrt{\frac{\rho}{2}}\|\mathcal{A}(x_k)\|\right)^2,
\end{aligned}$$

where the last inequality follows from Proposition 5.2.3 with $\mathcal{L}_\rho=\mathcal{L}_{\rho,\gamma}$, $d_\rho=d_{\rho,\gamma}$, and $u^*_{\lambda_k}=u^*_{\lambda_k,\gamma}$. ∎

The following technical result states a bound on the number of inner iterations performed by the modified I-AL method applied to (5.2.37) when a constant sequence $\{\eta_k\}$ is applied.

Lemma 5.4.3 *Let $\rho>0$, $(\epsilon_p,\epsilon_d)\in\Re_{++}\times\Re_{++}$ and $\bar{N}\in\mathbb{N}$ be given, and let γ be given by (5.2.43). Consider the modified I-AL method applied to the perturbed problem (5.2.37) with penalty parameter ρ, iteration limit $\bar{N}$ and inner tolerances $\eta_0,\ldots,\eta_{\bar{N}}$ given by*

$$\eta_k=\eta_\gamma:=\frac{\rho\epsilon_p^2}{128\bar{N}},\quad k=0,\ldots,\bar{N}-1. \qquad (5.4.26)$$

Then the following statements hold:

a) the total number of inner iterations performed by the above method is bounded by

$$\left\lceil \sqrt{\tfrac{8M_{\rho,\gamma}}{\gamma}} \right\rceil \left\{ 2\max\left(1, \left\lceil \log \tfrac{64\gamma \bar{N} D_X^2}{\rho \epsilon_p^2} \right\rceil\right) + \min(\bar{N}, N_\gamma) \left\lceil 2\log\left(1 + \tfrac{24 \bar{N}^{\frac{1}{2}} D_\Lambda^\gamma}{\rho \epsilon_p}\right) \right\rceil \right.$$
$$\left. + \max\left(1, \left\lceil \log \tfrac{16\gamma M_{\rho,\gamma} D_X^2}{\epsilon_d^2} \right\rceil\right) \right\}, \tag{5.4.27}$$

where

$$N_\gamma := \left\lceil \frac{16[D_\Lambda^\gamma]^2}{\rho^2 \epsilon_p^2} \right\rceil; \tag{5.4.28}$$

b) if $\bar{N} \geq N_\gamma$, then the above method successfully terminates in N_γ outer iterations with an (ϵ_p, ϵ_d)-primal-dual solution of (5.1.1).

Proof. Statement b) immediately follows from the assumption $\bar{N} \geq N_\gamma$ and Theorem 5.2.1 applied to the perturbed problem (5.2.37). We now show part a). Note that by Statement b), the number of outer iterations of the above method is bounded by $\min\{\bar{N}, N_\gamma\}$. Assume that the method terminates at the K-th outer iteration for some

$$0 \leq K \leq \min\{\bar{N}, N_\gamma\} - 1. \tag{5.4.29}$$

Clearly, $\|\mathcal{A}(x_k)\| > 3\epsilon_p/4$ for all $0 \leq k \leq K-1$. Hence, by using an argument similar to the one preceding (5.4.3), we can show that

$$\|\mathcal{A}(x_k)\|^2 \leq \frac{9[D_\Lambda^\gamma]^2}{\rho^2(k+1)}, \quad k = 1, \ldots, K-1. \tag{5.4.30}$$

For $k = 0, \ldots, K$, let $x_k^* := \operatorname{argmin}_{x \in X} \mathcal{L}_{\rho,\gamma}(x, \lambda_k)$, and l_k denote the number of inner iterations performed at step 1 of the modified I-AL method. By Theorem 1.1.5 with $\phi(\cdot) = \mathcal{L}_{\rho,\gamma}(\cdot, \lambda_0)$, $L_\phi = M_{\rho,\gamma}$, $\mu = \gamma$ and $\epsilon = \eta_\gamma$, (5.2.26) and (5.4.26), we have

$$\begin{aligned} l_0 &\leq \left\lceil \sqrt{\tfrac{8M_{\rho,\gamma}}{\gamma}} \right\rceil \max\left\{1, \left\lceil \log \frac{\gamma \|x_{-1} - x_0^*\|^2}{2\eta_\gamma} \right\rceil\right\} \leq \left\lceil \sqrt{\tfrac{8M_{\rho,\gamma}}{\gamma}} \right\rceil \max\left\{1, \left\lceil \log \frac{\gamma D_X^2}{2\eta_\gamma} \right\rceil\right\} \\ &= \left\lceil \sqrt{\tfrac{8M_{\rho,\gamma}}{\gamma}} \right\rceil \max\left\{1, \left\lceil \log \frac{64\gamma \bar{N} D_X^2}{\rho \epsilon_p^2} \right\rceil\right\}. \end{aligned} \tag{5.4.31}$$

It also follows from Theorem 1.1.5 that

$$l_k \leq \left\lceil \sqrt{\tfrac{8M_{\rho,\gamma}}{\gamma}} \right\rceil \max\left\{1, \left\lceil \log \frac{\gamma \|x_{k-1} - x_k^*\|^2}{2\eta_\gamma} \right\rceil\right\}, \quad \forall\, k = 1, \ldots, K.$$

Now by using (5.4.23) and (5.4.30), we have

$$\frac{\gamma\|x_{k-1}-x_k^*\|^2}{2} \le \left(\sqrt{\eta_\gamma}+\sqrt{\frac{\rho}{2}}\,\|\mathcal{A}(x_{k-1})\|\right)^2 \le \left(\sqrt{\eta_\gamma}+\frac{3D_\Lambda^\gamma}{\sqrt{2\rho k}}\right)^2.$$

We then conclude from the previous two observations and (5.4.26) that

$$\begin{aligned} l_k &\le \left\lceil\sqrt{\frac{8M_{\rho,\gamma}}{\gamma}}\right\rceil \max\left\{1,\left\lceil 2\log\left(1+\frac{3D_\Lambda^\gamma}{\sqrt{2\rho k\eta_\gamma}}\right)\right\rceil\right\} \\ &= \left\lceil\sqrt{\frac{8M_{\rho,\gamma}}{\gamma}}\right\rceil \left\lceil 2\log\left(1+\frac{3D_\Lambda^\gamma}{\sqrt{2\rho k\eta_\gamma}}\right)\right\rceil \le \left\lceil\sqrt{\frac{8M_{\rho,\gamma}}{\gamma}}\right\rceil \left\lceil 2\log\left(1+\frac{3D_\Lambda^\gamma}{\sqrt{2\rho\eta_\gamma}}\right)\right\rceil \\ &= \left\lceil\sqrt{\frac{8M_{\rho,\gamma}}{\gamma}}\right\rceil \left\lceil 2\log\left(1+\frac{24\bar{N}^{\frac{1}{2}}D_\Lambda^\gamma}{\rho\epsilon_p}\right)\right\rceil, \quad \forall\, k=1,\ldots,K. \end{aligned}$$

The above conclusion together with (5.4.27) and (5.4.31) then clearly imply that the total number of inner iterations performed at step 1) of the modified I-AL method is bounded by

$$\begin{aligned} l_0+\textstyle\sum_{k=1}^K l_k &\le l_0 + K\left\lceil\sqrt{\frac{8M_{\rho,\gamma}}{\gamma}}\right\rceil \left\lceil 2\log\left(1+\frac{24\bar{N}^{\frac{1}{2}}D_\Lambda^\gamma}{\rho\epsilon_p}\right)\right\rceil \\ &\le \left\lceil\sqrt{\frac{8M_{\rho,\gamma}}{\gamma}}\right\rceil \left\{2\max\left(1,\left\lceil\log\frac{64\gamma\bar{N}D_X^2}{\rho\epsilon_p^2}\right\rceil\right)+K\left\lceil\log\left(1+\frac{24\bar{N}^{\frac{1}{2}}D_\Lambda^\gamma}{\rho\epsilon_p}\right)\right\rceil\right\}. \end{aligned} \tag{5.4.32}$$

Moreover, let $\tilde{l}_K$ denote the number of inner iterations performed by subroutine PostProcessing. By using Theorem 1.1.5 with $\phi(\cdot)=\mathcal{L}_{\rho,\gamma}(\cdot,\lambda_K)$, $L_\phi=M_{\rho,\gamma}$, $\mu=\gamma$ and $\epsilon=\tilde{\zeta}$ and (5.2.42), we have

$$\begin{aligned} \tilde{l}_K &\le \left\lceil\sqrt{\frac{8M_{\rho,\gamma}}{\gamma}}\right\rceil \max\left\{1,\left\lceil\log\frac{\gamma D_X^2}{2\tilde{\zeta}}\right\rceil\right\} \\ &\le \left\lceil\sqrt{\frac{8M_{\rho,\gamma}}{\gamma}}\right\rceil \left[\max\left\{1,\left\lceil\log\frac{64\gamma D_X^2}{\rho\epsilon_p^2}\right\rceil\right\}+\max\left\{1,\left\lceil\log\frac{16\gamma M_{\rho,\gamma}D_X^2}{\epsilon_d^2}\right\rceil\right\}\right]. \end{aligned} \tag{5.4.33}$$

Combining inequalities (5.4.29), (5.4.32) and (5.4.33), we can easily see that the total number of inner iterations performed by the modified I-AL method is bounded by (5.4.27).

■

We are now ready to prove Theorem 5.2.5.

Proof of Theorem 5.2.5: We first show part a). It immediately follows from Lemma 5.4.3(a) that the total number of inner iterations performed by the modified I-AL method is bounded by (5.4.27) with $\bar{N}=\bar{N}_\gamma(t)$ and $\rho=\rho_\gamma(t)$. Note that by (5.2.44), (5.2.45), (5.4.40) and the fact that, by (5.2.46) and (5.2.47), $\log\mathcal{T}(t)\ge 2$, we have

$$\bar{N}_\gamma(t) \le \frac{16t^2}{\rho_\gamma(t)^2\epsilon_p^2}+1 \le \log\mathcal{T}(t)+1 \le 2\log\mathcal{T}(t). \tag{5.4.34}$$

Also, using definitions (5.2.41) and (5.2.44), we have that

$$\gamma \leq M_{\rho,\gamma} = L_f + \gamma + \rho\|\mathcal{A}\|^2 \leq 2\rho\|\mathcal{A}\|^2. \tag{5.4.35}$$

This observation together with (5.2.43) and (5.2.44) then imply that

$$\begin{aligned}
\left\lceil \sqrt{\frac{8M_{\rho,\gamma}}{\gamma}} \right\rceil &\leq \left\lceil 4\sqrt{\frac{\rho\|\mathcal{A}\|^2}{\gamma}} \right\rceil = \left\lceil 4\left(\frac{4t\|\mathcal{A}\|^2}{\gamma\epsilon_p(\log \mathcal{T}(t))^{\frac{1}{2}}} + \frac{L_f}{\gamma} + 1 \right)^{\frac{1}{2}} \right\rceil \\
&\leq 4\left(\frac{4D_X t\|\mathcal{A}\|^2}{\epsilon_p\epsilon_d(\log \mathcal{T}(t))^{\frac{1}{2}}} + \frac{D_X L_f}{\epsilon_d} + 1 \right)^{\frac{1}{2}} + 1 \\
&\leq 8\sqrt{\frac{D_X t\|\mathcal{A}\|^2}{\epsilon_p\epsilon_d}}(\log \mathcal{T}(t))^{-\frac{1}{4}} + 4\sqrt{\frac{D_X L_f}{\epsilon_d}} + 5.
\end{aligned} \tag{5.4.36}$$

Observe that, by (5.4.34), (5.4.35), (5.2.46) and (5.2.47),

$$\begin{aligned}
\log \frac{64\gamma D_X^2 \bar{N}_\gamma(t)}{\rho_\gamma(t)\epsilon_p^2} &\leq \log \frac{128\gamma D_X^2 \log \mathcal{T}(t)}{\rho_\gamma(t)\epsilon_p^2} \leq \log \frac{256\|\mathcal{A}\|^2 D_X^2 \log \mathcal{T}(t)}{\epsilon_p^2} \\
&= 8 + 4\log\left(\frac{\|\mathcal{A}\|D_X}{\epsilon_p} \right)^{\frac{1}{2}} + \log\log \mathcal{T}(t) = \mathcal{O}\left(\log \mathcal{T}(t)\right),
\end{aligned} \tag{5.4.37}$$

and that, by (5.2.45), the fact that $\log x \leq x$, and (5.4.34),

$$\begin{aligned}
&\min(\bar{N}_\gamma(t), N_\gamma)\left\lceil 2\log\left(1 + \frac{24D_\Lambda^\gamma\,[\bar{N}_\gamma(t)]^{\frac{1}{2}}}{\rho_\gamma(t)\epsilon_p} \right) \right\rceil \\
&\leq \bar{N}_\gamma(t)\left\lceil 2\log\left(1 + \frac{6D_\Lambda^\gamma}{t}\left[\log \bar{N}_\gamma(t)\right]^{\frac{1}{2}}[\bar{N}_\gamma(t)]^{\frac{1}{2}} \right) \right\rceil \\
&\leq \bar{N}_\gamma(t)\left\lceil 2\log\left(1 + \frac{6D_\Lambda^\gamma}{t}\bar{N}_\gamma(t) \right) \right\rceil \\
&\leq \bar{N}_\gamma(t)\left\lceil 2\log\left(1 + \frac{12D_\Lambda^\gamma}{t}\log \mathcal{T}(t) \right) \right\rceil \\
&= \mathcal{O}\left\{ \log \mathcal{T}(t)\max\left(1, \log\frac{D_\Lambda^\gamma \log \mathcal{T}(t)}{t} \right) \right\}.
\end{aligned} \tag{5.4.38}$$

It also follows from (5.4.35), (5.2.43), (5.2.44), (5.2.45), (5.2.46) and (5.2.47) that

$$\begin{aligned}
\log \frac{16\gamma M_{\rho,\gamma} D_X^2}{\epsilon_d^2} &\leq \log \frac{16M_{\rho,\gamma}^2 D_X^2}{\epsilon_d^2} \leq \log\left(\frac{8\rho\|\mathcal{A}\|^2 D_X}{\epsilon_d} \right)^2 \\
&\leq 2\log\left[\frac{8\|\mathcal{A}\|^2 D_X}{\epsilon_d}\left(\frac{4t}{\epsilon_p(\log \mathcal{T}(t))^{\frac{1}{2}}} + \frac{L_f + \gamma}{\|\mathcal{A}\|^2} \right) \right] \\
&= 2\log\left[\frac{8\|\mathcal{A}\|^2 D_X}{\epsilon_d}\left(\frac{4t}{\epsilon_p(\log \mathcal{T}(t))^{\frac{1}{2}}} + \frac{L_f}{\|\mathcal{A}\|^2} + \frac{\epsilon_d}{2D_X\|\mathcal{A}\|^2} \right) \right] \\
&\leq 2\log\left[8D_X\left(\frac{4t\|\mathcal{A}\|^2}{\epsilon_p\epsilon_d} + \frac{L_f}{\epsilon_d} \right) + 4 \right] = \mathcal{O}(\log \mathcal{T}(t)).
\end{aligned} \tag{5.4.39}$$

Now substituting bounds (5.4.36), (5.4.37), (5.4.38), and (5.4.39) into bound (5.4.27), we obtain bound (5.2.48). Statement b) follows immediately from Lemma 5.4.3(b) and the fact that, by (5.2.45), the assumption $t \geq D_\Lambda^\gamma$ and (5.4.34),

$$N_\gamma = \left\lceil \frac{16[D_\Lambda^\gamma]^2}{\rho_\gamma(t)^2 \epsilon_p^2} \right\rceil \leq \left\lceil \frac{16t^2}{\rho_\gamma(t)^2 \epsilon_p^2} \right\rceil = \bar{N}_\gamma(t) \leq 2 \log \mathcal{T}(t). \tag{5.4.40}$$

■

Before proving Theorem 5.2.6, we first state two technical results that summarize some properties of the function ψ defined in (5.2.49).

Lemma 5.4.4 *Let $\psi(t)$ and $\hat{t}$ be defined in (5.2.49). Then, the following statements hold:*

a) $\psi(t)$ is continuous and non-decreasing for $t \geq 0$;

b) $\psi(0) \leq 0$ and $\psi(\hat{t}) \geq 0$.

Proof. Statement a) immediately following from the fact that, by (5.2.49),

$$\begin{aligned} \psi'(t) &= \mathcal{S}_1 \left\{ 1 - \frac{\mathcal{S}_2}{4(\mathcal{S}_1 t^{\frac{1}{2}} + \mathcal{S}_2 + \mathcal{S}_3)} \left[\log(\mathcal{S}_1 t^{\frac{1}{2}} + \mathcal{S}_2 + \mathcal{S}_3) \right]^{-\frac{3}{4}} \right\} \frac{1}{2\sqrt{t}} \\ &\geq \mathcal{S}_1 (1 - 1/4) \frac{1}{2\sqrt{t}} \geq \frac{3\mathcal{S}_1}{8\sqrt{t}} \geq 0, \quad \forall\, t > 0, \end{aligned}$$

where in the first inequality we use the fact that $\log(\mathcal{S}_1 t^{\frac{1}{2}} + \mathcal{S}_2 + \mathcal{S}_3) \geq 2$ in view of (5.2.47). It can be easily seen from (5.2.49) that $\psi(0) \leq 0$. Noting that, by the definition of $\hat{t}$ in (5.2.49),

$$\mathcal{S}_1^2 \hat{t} - \mathcal{S}_2^2 (\mathcal{S}_1 \hat{t}^{\frac{1}{2}} + \mathcal{S}_2 + \mathcal{S}_3) = \mathcal{S}_1^2 \hat{t} - \mathcal{S}_1 \mathcal{S}_2^2 \hat{t}^{\frac{1}{2}} - \mathcal{S}_2^2 (\mathcal{S}_2 + \mathcal{S}_3) = 0,$$

we conclude from (5.2.49) and the fact that $\log \tau \leq \tau \leq \tau^2$ for $\tau \geq 1$ that

$$\psi(\hat{t}) = \mathcal{S}_1 \hat{t}^{\frac{1}{2}} - \mathcal{S}_2 \left[\log(\mathcal{S}_1 \hat{t}^{\frac{1}{2}} + \mathcal{S}_2 + \mathcal{S}_3) \right]^{\frac{1}{4}} \geq \mathcal{S}_1 \hat{t}^{\frac{1}{2}} - \mathcal{S}_2 (\mathcal{S}_1 \hat{t}^{\frac{1}{2}} + \mathcal{S}_2 + \mathcal{S}_3)^{\frac{1}{2}} = 0.$$

We have thus shown that b) holds. ■

Lemma 5.4.5 *Let $\psi(t)$ and $\hat{t}$ be defined in (5.2.49). Then, there exists $t_0 \in [0, \hat{t}]$ such that $0 \leq \psi(t_0) \leq 1$. Moreover, we have*

$$\begin{aligned} \mathcal{S}_1 t_0^{\frac{1}{2}} &\leq \mathcal{S}_2 \left[\log \mathcal{T}(t_0)\right]^{\frac{1}{4}} + 1, & (5.4.41) \\ \mathcal{S}_1 t^{\frac{1}{2}} &\geq \mathcal{S}_2 [\log \mathcal{T}(t)]^{\frac{1}{4}}, \quad \forall\, t \geq t_0, & (5.4.42) \\ \log \mathcal{T}(t_0) &= \mathcal{O}\left(\log \mathcal{T}(0)\right), & (5.4.43) \end{aligned}$$

where $\mathcal{T}(\cdot)$, $\mathcal{S}_1$ and $\mathcal{S}_2$ are defined in (5.2.46) and (5.2.47).

Proof. The existence of $t_0 \in [0, \hat{t}]$ satisfying $0 \leq \psi(t_0) \leq 1$ follows immediately from Lemma 5.4.4. Inequality (5.4.41) follows from (5.2.46), (5.2.49) and the fact $\psi(t_0) \leq 1$. Moreover, we conclude from (5.2.46), (5.2.49), the assumption $\psi(t_0) \geq 0$ and Lemma 5.4.4(a) that

$$\mathcal{S}_1 t^{\frac{1}{2}} - \mathcal{S}_2 \left[\log \mathcal{T}(t)\right]^{\frac{1}{4}} = \mathcal{S}_1 t^{\frac{1}{2}} - \mathcal{S}_2 \left[\log(\mathcal{S}_1 t^{\frac{1}{2}} + \mathcal{S}_2 + \mathcal{S}_3)\right]^{\frac{1}{4}} = \psi(t) \geq \psi(t_0) \geq 0$$

for any $t \geq t_0$, and hence that (5.4.42) holds. Also note that by (5.2.46), (5.2.49) and the fact that $t_0 \leq \hat{t}$, we have

$$\log \mathcal{T}(t_0) = \log(\mathcal{S}_1 t_0^{\frac{1}{2}} + \mathcal{S}_2 + \mathcal{S}_3) \leq \log(\mathcal{S}_1 \hat{t}^{\frac{1}{2}} + \mathcal{S}_2 + \mathcal{S}_3) = \mathcal{O}\left(\log(\mathcal{S}_2 + \mathcal{S}_3)\right) = \mathcal{O}\left(\log \mathcal{T}(0)\right).$$

■

We are now ready to prove Theorem 5.2.6.

Proof of Theorem 5.2.6: Consider parameter t_0 computed in step 0 of the modified I-AL guess-and-check procedure. Assume first that $t_0 \geq D_\Lambda^\gamma$. Using this assumption, Theorem 5.2.5, relations (5.4.41) and (5.4.43), and the fact that, by (5.2.46) and (5.2.47), $\mathcal{T}(t) \geq 4$ for every $t \geq 0$, we conclude that the modified I-AL guess-and-check procedure will successfully terminate after the first loop and that the total number of inner iterations is bounded by

$$\begin{aligned} & \mathcal{O}\left\{\left(\mathcal{S}_1 t_0^{\frac{1}{2}} + \mathcal{S}_2 \left[\log \mathcal{T}(t_0)\right]^{\frac{1}{4}}\right) \left[\log \mathcal{T}(t_0)\right]^{\frac{3}{4}} \max\left(1, \log \frac{D_\Lambda^\gamma \log \mathcal{T}(t_0)}{t_0}\right)\right\} \\ &= \mathcal{O}\left\{\left(\mathcal{S}_1 t_0^{\frac{1}{2}} + \mathcal{S}_2 \left[\log \mathcal{T}(t_0)\right]^{\frac{1}{4}}\right) \left[\log \mathcal{T}(t_0)\right]^{\frac{3}{4}} \log\log \mathcal{T}(t_0)\right\} \\ &= \mathcal{O}\left\{\mathcal{S}_2 \log \mathcal{T}(t_0) \log\log \mathcal{T}(t_0)\right\} = \mathcal{O}\left\{\mathcal{S}_2 \log \mathcal{T}(0) \log\log \mathcal{T}(0)\right\}, \end{aligned}$$

which is clearly bounded by (5.2.50).

Now assume that $t_0 < D_\Lambda^\gamma$. Suppose that the modified I-AL guess-and-check procedure terminates when the iteration count j is equal to J. Let

$$\bar{J} := \max\{0, \lceil \log(D_\Lambda^\gamma / t_0) \rceil\} \tag{5.4.44}$$

and note that

$$2D_\Lambda^\gamma \geq t_{\bar{J}} := t_0 2^{\bar{J}} \geq D_\Lambda^\gamma. \tag{5.4.45}$$

Theorem 5.2.5(b) and the second inequality in (5.4.45) then imply that $J \leq \bar{J}$. Also observe that, by relation (5.4.4) with $L = 1$, $p_1 = 1/2$, $\bar{t} = D_\Lambda^\gamma$, $K = \bar{J}$, $\nu = D_\Lambda^\gamma \log \mathcal{T}(t_{\bar{J}})$, $\beta_0 = 0$ and $\beta_1 = 1/\sqrt{t_0}$, we have

$$\begin{aligned}
&\sum_{j=0}^{\bar{J}} t_j^{\frac{1}{2}} \max\left(1, \log \frac{D_\Lambda^\gamma \log \mathcal{T}(2D_\Lambda^\gamma)}{t_j}\right) \\
&\quad \leq \sqrt{t_0} \sum_{j=0}^{\bar{J}} \left\lceil \frac{1}{\sqrt{t_0}} t_j^{\frac{1}{2}} \right\rceil \max\left(1, \left\lceil \log \frac{D_\Lambda^\gamma \log \mathcal{T}(2D_\Lambda^\gamma)}{t_j} \right\rceil\right) \\
&\quad = \mathcal{O}\left\{ \sqrt{t_0} \left\lceil \frac{1}{\sqrt{t_0}} [D_\Lambda^\gamma]^{\frac{1}{2}} \right\rceil \max\left(1, \left\lceil \log \frac{D_\Lambda^\gamma \log \mathcal{T}(2D_\Lambda^\gamma)}{D_\Lambda^\gamma} \right\rceil\right) \right\} \\
&\quad = \mathcal{O}\left\{ \left([D_\Lambda^\gamma]^{\frac{1}{2}} + \sqrt{t_0} \right) \max\left(1, \lceil \log\log \mathcal{T}(2D_\Lambda^\gamma) \rceil\right) \right\} \\
&\quad = \mathcal{O}\left([D_\Lambda^\gamma]^{\frac{1}{2}} \log\log \mathcal{T}(D_\Lambda^\gamma) \right),
\end{aligned} \tag{5.4.46}$$

where the last identity follows from the facts that $t_0 \leq D_\Lambda^\gamma$ and $\log \mathcal{T}(D_\Lambda^\gamma) \geq 2$. Using the facts that $J \leq \bar{J}$ and the function $\mathcal{T}$ given by (5.2.46) is non-decreasing, Theorem 5.2.5(a), relations (5.4.42) and (5.4.46), and the simple observation that by (5.4.45), we have $t_0 \leq t_j \leq 2D_\Lambda^\gamma$ for every $j = 1, \ldots, \bar{J}$, we conclude that the total number of inner iterations performed by the modified I-AL guess-and-check procedure is bounded by

$$\begin{aligned}
&\mathcal{O}\left\{ \sum_{j=0}^{\bar{J}} \left[\left(\mathcal{S}_1 t_j^{\frac{1}{2}} + \mathcal{S}_2 \left[\log \mathcal{T}(t_j)\right]^{\frac{1}{4}} \right) \left[\log \mathcal{T}(t_j)\right]^{\frac{3}{4}} \max\left(1, \log \frac{D_\Lambda^\gamma \log \mathcal{T}(t_j)}{t_j}\right) \right] \right\} \\
&\quad = \mathcal{O}\left\{ \sum_{j=0}^{\bar{J}} \left[\mathcal{S}_1 t_j^{\frac{1}{2}} [\log \mathcal{T}(t_j)]^{\frac{3}{4}} \max\left(1, \log \frac{D_\Lambda^\gamma \log \mathcal{T}(t_j)}{t_j}\right) \right] \right\} \\
&\quad = \mathcal{O}\left\{ [\log \mathcal{T}(2D_\Lambda^\gamma)]^{\frac{3}{4}} \mathcal{S}_1 \sum_{j=0}^{\bar{J}} \left[t_j^{\frac{1}{2}} \max\left(1, \log \frac{D_\Lambda^\gamma \log \mathcal{T}(2D_\Lambda^\gamma)}{t_j}\right) \right] \right\} \\
&\quad = \mathcal{O}\left\{ [\log \mathcal{T}(D_\Lambda^\gamma)]^{\frac{3}{4}} \mathcal{S}_1 [D_\Lambda^\gamma]^{\frac{1}{2}} \log\log \mathcal{T}(D_\Lambda^\gamma) \right\},
\end{aligned}$$

which is clearly bounded by (5.2.50). ∎

5.5 Comparision with other first-order methods

In this section, we compare the results obtained in this chapter for the inexact AL methods with another possible approach for solving variational inequalities (VI) studied in Nemirovski ([42]) for bounded sets, and Monteiro and Svaiter ([41]) for unbounded sets.

Given a closed convex set $\Omega \in \Re^p$ and a monotone continuous function $F : \Omega \to \Re^p$. The (monotone) VI problem with respect to the pair (F, X), denoted by $VIP(F, \Omega)$, consists of finding w^* such that

$$w^* \in \Omega, \qquad \langle w - w^*, F(w^*)\rangle \geq 0, \quad \forall w \in \Omega. \tag{5.5.1}$$

It is well-known that, under the assumption that F is monotone and continuous, (5.5.1) is equivalent to

$$w^* \in \Omega, \qquad \langle w - w^*, F(w)\rangle \geq 0, \quad \forall w \in \Omega.$$

Relaxing the above two conditions, we obtain the following two notions of approximate solutions of $VIP(F, \Omega)$.

Definition 5.5.1 *A point $\bar{w} \in \Omega$ is a (ϱ, ϵ)-strong (resp., (ϱ, ϵ)-weak) solution of $VIP(F, \Omega)$ if there exists $r \in \Re^n$ such that $\|r\| \leq \varrho$ and, for every $w \in \Omega$, $\langle w - \bar{w}, F(\bar{w}) - r\rangle \geq -\epsilon$ (resp., $\langle w - \bar{w}, F(w) - r\rangle \geq -\epsilon$).*

It is well-known that the CP problem (5.1.1) is equivalent to solving the $VIP(F, \Omega)$, where $\Omega := X \times \Re^m$ and

$$F(w) = F(x, \lambda) := \begin{pmatrix} \nabla f(x) + \mathcal{A}_0^* \lambda \\ -\mathcal{A}(x) \end{pmatrix}.$$

Moreover, defining the norm on $\Re^n \times \Re^m$ as $\|w\| := (\|x\|^2 + \|\lambda\|^2)^{1/2}$, then it is easy to see that an (ϵ_p, ϵ_d)-primal-dual solution $(\bar{x}, \bar{\lambda})$ is a $(\varrho, 0)$-strong solution, where $\varrho = \max\{\epsilon_p, \epsilon_d\}$. Disregarding L_f, $\|\mathcal{A}\|$, D_X, D_Λ and D_Λ^γ, it has been shown in Monteiro and Svaiter ([41]) that, given $(\varrho, \epsilon) \in \Re_{++} \times \Re_{++}$, a variant of the Korpelevich's method can find an (ϱ, ϵ)-strong solution for $VIP(F, \Omega)$ in $\mathcal{O}(\varrho^{-2} + \epsilon^{-1})$. On the other hand, we show in this chapter

that a $(\varrho, 0)$-strong solution, and hence an approximate solution as above, can be found in

$$\mathcal{O}\left(\frac{1}{\varrho}(\log \varrho^{-1})^{3/4} \log\log \varrho^{-1}\right)$$

by applying the modified guess-and-check procedure in Subsection 3.2 with $\epsilon_p = \epsilon_d = \varrho/\sqrt{2}$. Hence, the complexity in this chapter is better than the one in [41] by at least a factor of

$$\varrho(\log \varrho^{-1})^{3/4} \log\log \varrho^{-1}.$$

It should be noted that [41] also shows that an (ϱ, ϵ)-weak solution for $VIP(F, \Omega)$ can be found in

$$\mathcal{O}(\varrho^{-1} + \epsilon^{-1}). \tag{5.5.2}$$

It would be interesting to see whether our analysis in this chapter can be modified to the context of finding a weak solution of $VIP(F, \Omega)$ so as to obtain a better iteration-complexity bound than (5.5.2).

5.6 Conclusions of this chapter

In this chapter, we present first-order methods for solving problem (5.1.1) based on an inexact version of the classical augmented Lagrangian approach, where the subproblems are approximately solved by means of Nesterov's optimal method. We establish a bound on the total number of Nesterov's optimal iterations, i.e., the inner iterations, performed throughout the entire inexact AL method to obtain a near primal-dual optimal solution. We also present variants with better iteration-complexity bounds than the original inexact AL method, which consist of applying the original approach directly to a perturbed problem obtained by adding a strongly convex component to the objective function of the CP problem. We show that the iteration-complexity of the inexact AL methods for obtaining a near primal-dual optimal solution compares favorably with other penalty based approaches, such as the quadratic and exact penaly method studied in [33], and another possible approach for solving variational inequalities studied in Nemirovski ([42]), and Monteiro and Svaiter ([41]).

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

In this thesis we investigate the design and complexity analysis of the algorithms to solve convex optimization problems under inexact first-order information. The main goal is to design efficient algorithms for solving convex programming problems under a stochastic oracle. To this end, we have

- introduced the mirror descent stochastic approximation algorithm and present highly encouraging numerical results for this method applied to a certain class of convex programming problems;

- developed accuracy estimates for stochastic programming problems by employing SA type algorithms. We show that while running a mirror descent SA procedure one can compute, with a small additional effort, lower and upper statistical bounds for the optimal objective value. We demonstrate that for a certain class of convex stochastic programs these bounds are comparable in quality with similar bounds computed by the SAA method, while their computational cost is considerably smaller;

- conducted extensive numerical experiments to understand the performance of the mirror descent SA algorithm for solving stochastic programing problems with a feasible set more complicated than a standard simplex;

- demonstrated that a slightly modified mirror descent SA algorithm exhibits the best known so far rate of convergence for solving stochastic composite optimization which is guaranteed by a more involved stochastic mirror-prox algorithm;

- closed the theoretical gap between the upper and lower bounds on the rate of convergence for solving stochastic composite optimization by developing the accelerated

stochastic approximation algorithm. Notice that the accelerated SA is the first universally optimal method for smooth, non-smooth and stochastic convex optimization;

- suggested viable ways to extend the application of efficient SA algorithms to a certain class of equality constrained stochastic convex programming problems. More specifically, if the accelerated SA is applied to solve the quadratic penalization problem where the violation of the affine constraints is penalized, then the size of the Lagrange multiplier associated with these affine constraints has, asymptotically, no affect on the convergence rate.

We have also investigated certain interesting deterministic optimization technique, namely, the augmented Lagrangian method, which operates on first-order information of the augmented dual problem. We consider the situation where to obtain the exact first-order information of the augmented dual is time-consuming and hence only approximate first-order information is available in reality. Our main contribution consists of the following aspects.

- We establish a bound on the total number of Nesterov's optimal iterations, i.e., the inner iterations, performed throughout the entire inexact AL method to obtain a near primal-dual optimal solution. We also present variants with better iteration-complexity bounds than the original inexact AL method;

- We show that the iteration-complexity of the inexact AL methods for obtaining a near primal-dual optimal solution compares favorably with other penalty based approaches, such as the quadratic and exact penalty method studied, and another possible approach for solving variational inequalities;

- Some theoretical guidelines and guess-and-check procedures to specify certain parameters for the I-AL methods are provided.

A few topics are worth mentioning for future studies:

- to extend the accelerated SA algorithm for solving strongly convex programming problems and investigate its application in stochastic dynamic programming and statistical

learning;

- to conduct more computational studies for first-order methods for convex programming. It will be very rewarding to propose first-order methods with both superb practical performance and optimal rate of convergence;
- to explore the structure of the problems and the employed termination criterions, and to improve convergence results for certain important convex programming problems, such as, the equality constrained CP problems studied in the last two chapters;
- to solve problems arising from real applications such as engineering design, healthcare and energy industry.

APPENDIX A

SOME TECHNICAL PROOFS

Our goal in this chpater is to prove Propositions 5.2.1 and 5.4.1.

Proof of Proposition 5.2.1: Let $\{(b_k, r_k)\}$ be a sequence of epi$f$ converging to (b, r) for $k \to +\infty$. It suffices to show that $v(b) \leq r$. First notice that the fact that $v(b_k) \leq r_k$ implies that there exists $x_k \in \mathcal{F}(b_k)$ such that $f(x_k) = v(b_k) \leq r_k$. Now we claim that the sequence $\{x_k\}$ is bounded. Hence, by using this claim, there exists an accumulation point x of the sequence $\{x_k\}$ such that $x \in \mathcal{F}(b), f(x) \leq r$ as $k \to +\infty$, which clearly implies that $v(b) \leq r$. Now it remains to show that the sequence$\{x_k\}$ is bounded. Indeed, let $f'_\infty(\cdot)$ denote the recession function of f, and $\mathcal{F}(b)_\infty$ denote the recession cone of the set $\mathcal{F}(b)$. Also let $\phi_b(\cdot) := f(\cdot) + \mathrm{I}_{\mathcal{F}(b)}(\cdot)$, using the assumption that the set of optimal solutions for (5.1.1) is nonempty and bounded, we have

$$\{\phi_0\}'_\infty(d) = f'_\infty(d) + \mathrm{I}_{\mathcal{F}(0)_\infty} > 0$$

for all $d \neq 0$ (see Definitions 2.2.2 and 3.2.3, Remark 3.2.8 and Proposition 3.2.9 in [22]). It can also be easily seen that the recession cone $\mathcal{F}(b_k)_\infty \equiv \mathcal{F}(0)_\infty$. It then follows from the above two relations that $\{\phi_{b_k}\}'_\infty(d) > 0$ for all $d \neq 0$, which, by Remark 3.2.8 in [22], implies that $x_k \in \mathrm{Argmin}_x \phi_{b_k}(x)$ is bounded. ∎

Our goal in the remaining part of this section is to prove Proposition 5.4.1. We first start with an easy case of the result. Before stating this easy case, we mention a simple inequality.

Lemma A.0.1 *For any scalars $\tau \geq 0$, $x > 0$, and $\alpha \geq 0$, we have $\tau x + \alpha \leq (\tau + \alpha)\lceil x \rceil$.*

Lemma A.0.2 *For some positive integer L, let positive scalars $p_1, p_2, \cdots, p_L$ be given and define*

$$C_0 := 2 + \max\left\{2, \max_{1 \leq l \leq L}\left(\frac{2}{p_l} + \frac{4^{p_l}}{2^{p_l} - 1}\right)\right\}.$$

Then, for any positive scalars $\beta_0, \beta_1, \cdots, \beta_L$, and $\bar{t} > t_0$, we have

$$\sum_{k=0}^{K} \left\lceil \beta_0 + \sum_{l=1}^{L} \beta_l t_k^{p_l} \right\rceil \leq C_0 \left\lceil \beta_0 + \sum_{l=1}^{L} \beta_l \bar{t}^{p_l} \right\rceil, \tag{A.0.1}$$

where K and $t_0, \ldots, t_K$ are defined in (5.4.5).

Proof. Without loss of generality, assume that $t_0 = (\max(\beta_0, 1)/\beta_1)^{1/p_1}$. Clearly, due to the definition of K in (5.4.5) and the assumption $\bar{t} > t_0$, we have $K < \log(\bar{t}/t_0) + 1$, and hence that $t_0 2^{K+1} < 4\bar{t}$. Using these relations and the inequality $\log x = (\log x^p)/p \leq x^p/p$ for any $x > 0$, $p > 0$, we obtain

$$\begin{aligned}
\sum_{k=0}^{K} \left\lceil \beta_0 + \sum_{l=1}^{L} \beta_l t_k^{p_l} \right\rceil &\leq \sum_{k=0}^{K} \left(1 + \beta_0 + \sum_{l=1}^{L} \beta_l t_0^{p_l} 2^{p_l k} \right) \leq (1+\beta_0)(1+K) + \sum_{l=1}^{L} \beta_l t_0^{p_l} \frac{2^{(K+1)p_l}}{2^{p_l} - 1} \\
&\leq (1+\beta_0) \left[2 + \log\left(\frac{\bar{t}}{t_0} \right) \right] + \sum_{l=1}^{L} \beta_l \frac{(4\bar{t})^{p_l}}{2^{p_l} - 1} \\
&\leq (1+\beta_0) \left[2 + \frac{1}{p_1} \left(\frac{\bar{t}}{t_0} \right)^{p_1} \right] + \sum_{l=1}^{L} \frac{4^{p_l}}{2^{p_l} - 1} \beta_l \bar{t}^{p_l} \\
&\leq (1+\beta_0) \left[2 + \frac{1}{p_1} \left(\frac{\beta_2 \bar{t}^{p_1}}{\max(\beta_1, 1)} \right) \right] + \sum_{l=1}^{L} \frac{4^{p_l}}{2^{p_l} - 1} \beta_l \bar{t}^{p_l} \\
&\leq 2(1+\beta_0) + \frac{2}{p_1} \beta_1 \bar{t}^{p_1} + \sum_{l=1}^{L} \frac{4^{p_l}}{2^{p_l} - 1} \beta_l \bar{t}^{p_l} \\
&\leq 2 + \max\left\{ 2, \frac{2}{p_1} + \frac{4^{p_1}}{2^{p_1} - 1}, \max_{2 \leq l \leq L} \frac{4^{p_l}}{2^{p_l} - 1} \right\} (\beta_0 + \sum_{l=1}^{L} \beta_l \bar{t}^{p_l}).
\end{aligned}$$

Inequality (A.0.1) now clearly follows from the above conclusion, Lemma A.0.1, and some trivial majorization. ∎

The following technical lemma provides an useful inequality to prove a more difficult case of our result.

Lemma A.0.3 *Let the positive scalars a and p be given. For any $0 \leq K \leq a - 1/(p \ln 2) - 1$, we have*

$$\sum_{k=0}^{K} 2^{pk}(a-k) \leq \frac{2^{p(K+1)}}{p \ln 2} \left[a - (K+1) + \frac{1}{p \ln 2} \right].$$

Proof. Noting that the function $\psi(s) := 2^{ps}(a - s)$ is non-decreasing for any $s \leq$

$a - 1/(p \ln 2)$, we obtain

$$\begin{aligned}\sum_{k=0}^{K} 2^{pk}\,(a-k) &\leq \int_0^{K+1} 2^{ps}\,(a-s)\,ds = \frac{2^{ps}}{p\ln 2}\left(a - s + \frac{1}{p_l \ln 2}\right)\Big|_0^{K+1} \\ &\leq \frac{2^{p(K+1)}}{p\ln 2}\left[a - (K+1) + \frac{1}{p_l \ln 2}\right].\end{aligned}$$

■

We now prove a more difficult case of the result.

Lemma A.0.4 *For some positive integer L, let positive scalars $p_1, p_2, \cdots, p_L$ be given and define*

$$C_1 := 1 + \frac{1}{(\ln 2)\,\min_{1\leq l\leq L} p_l}\,, \tag{A.0.2}$$

$$C_2 := 5 + \max\left\{5, \max_{1\leq l\leq L}\left(\frac{4}{p_l^2} + \frac{7}{p_l} + 2C_1 4^{p_l}\right)\right\}. \tag{A.0.3}$$

Then, for any positive scalars $\beta_0, \beta_1, \cdots, \beta_L$, and ν, we have

$$\sum_{k=0}^{K}\left\lceil \beta_0 + \sum_{l=1}^{L}\beta_l t_k^{p_l}\right\rceil \max\left\{1, \left\lceil \log\frac{\nu}{t_k}\right\rceil\right\} \leq C_2\left\lceil \beta_0 + \sum_{l=1}^{L}\beta_l \bar{t}^{p_l}\right\rceil \max\left\{1, \left\lceil \log\frac{\nu}{\bar{t}}\right\rceil\right\} \tag{A.0.4}$$

for every $\bar{t} \in (t_0, \tilde{t}]$, where $\tilde{t} := 2^{-C_1}\nu$ and the scalars K and $t_0, \ldots, t_K$ are defined in (5.4.5).

Proof. Without loss of generality, assume that $t_0 = (\max(\beta_0, 1)/\beta_1)^{1/p_1}$. Clearly, due to the definition of K in (5.4.5) and the assumption $\bar{t} > t_0$, we have $\log(\bar{t}/t_0) \leq K \leq \log(\bar{t}/t_0) + 1$. Using these relations, the inequalities $\log x = (\log x^p)/p \leq x^p/p$ for any $x > 0$, $p > 0$, and $(\log x)^2 = (2\log x^{p/2})/p)^2 \leq 4x^p/p^2$ for any $x \geq 1$, $p > 0$, and the facts

$\log(\nu/\bar{t}) \geq C_1 \geq 1$ and $\bar{t}/t_0 \geq 1$ due to the assumption $t_0 < \bar{t} \leq 2^{-C_1}\nu$, we obtain

$$\begin{aligned}
\sum_{k=0}^{K}\left(\log\frac{\nu}{t_k}+1\right) &= \sum_{k=0}^{K}\left(\log\frac{\nu}{t_0}+1-k\right) = (K+1)\left(\log\frac{\nu}{t_0}+1-\frac{K}{2}\right) \\
&= (K+1)\left(\log\frac{\nu}{t_0}-K+1+\frac{K}{2}\right) \leq (K+1)\left(\log\frac{\nu}{\bar{t}}+1+\frac{K}{2}\right) \\
&= \frac{1}{2}\left[2(K+1)\log\frac{\nu}{\bar{t}}+K^2+3K+2\right] \leq \frac{1}{2}(K^2+5K+4)\left\lceil\log\frac{\nu}{\bar{t}}\right\rceil \\
&\leq \frac{1}{2}\left(\log^2\frac{\bar{t}}{t_0}+7\log\frac{\bar{t}}{t_0}+10\right)\left\lceil\log\frac{\nu}{\bar{t}}\right\rceil \\
&\leq \frac{1}{2}\left[\left(\frac{4}{p_1^2}+\frac{7}{p_1}\right)\left(\frac{\bar{t}}{t_0}\right)^{p_1}+10\right]\left\lceil\log\frac{\nu}{\bar{t}}\right\rceil \\
&= \frac{1}{2}\left[\left(\frac{4}{p_1^2}+\frac{7}{p_1}\right)\frac{\beta_1\bar{t}^{p_1}}{\max\{1,\beta_0\}}+10\right]\left\lceil\log\frac{\nu}{\bar{t}}\right\rceil \\
&\leq \left[\left(\frac{4}{p_1^2}+\frac{7}{p_1}\right)\frac{\beta_1\bar{t}^{p_1}}{1+\beta_0}+5\right]\left\lceil\log\frac{\nu}{\bar{t}}\right\rceil. \qquad \text{(A.0.5)}
\end{aligned}$$

Moreover, it follows from the fact $K \leq \log(\bar{t}/t_0)+1$ and the assumption $t_0 < \bar{t} \leq 2^{-C_1}\nu$ that

$$\begin{aligned}
K+1 &\leq \log\frac{\bar{t}}{t_0}+2 = \log\frac{2\nu}{t_0}+\log\frac{\bar{t}}{2\nu}+2 \leq \log\frac{2\nu}{t_0}-C_1+1 \\
&= \log\frac{2\nu}{t_0}-\frac{1}{(\ln 2)\,\min_{1\leq l\leq L}p_l} \leq \log\frac{2\nu}{t_0}-\frac{1}{(\ln 2)\,p_l}, \ \forall\, 1\leq l\leq L,
\end{aligned}$$

which, together with Lemma A.0.3 (with $a = \log(2\nu/t_0)$ and $p = p_l$) and the fact that $K \geq \log(\bar{t}/t_0)$, then imply that

$$\begin{aligned}
\sum_{k=0}^{K}2^{p_l k}\left(\log\frac{2\nu}{t_0}-k\right) &\leq \frac{2^{p_l(K+1)}}{p_l\ln 2}\left(\log\frac{2\nu}{t_0}-(K+1)+\frac{1}{p_l\ln 2}\right) \\
&\leq \frac{2^{p_l(K+1)}}{p_l\ln 2}\left(\log\frac{2\nu}{t_0}-(\log\frac{\bar{t}}{t_0}+1)+\frac{1}{p_l\ln 2}\right) \\
&= \frac{2^{p_l(K+1)}}{p_l\ln 2}\left(\log\frac{\nu}{\bar{t}}+\frac{1}{p_l\ln 2}\right) \leq \frac{2}{p_l\ln 2}2^{p_l(K+1)}\left\lceil\log\frac{\nu}{\bar{t}}\right\rceil \\
&\leq 2C_1 2^{p_l(K+1)}\left\lceil\log\frac{\nu}{\bar{t}}\right\rceil, \qquad \text{(A.0.6)}
\end{aligned}$$

where the last two inequalities follow from the relation $\log(\nu/\bar{t}) \geq \log(\nu/\tilde{t}) = C_1 \geq 1/(p_l\ln 2)$.

By the definitions of K and $t_k, \forall k = 1,\cdots,K$ in (5.4.5) and the assumption $\bar{t} \leq 2^{-C_0}\nu$, we have $t_k = t_0 2^k \leq t_0 2^{\log(\bar{t}/t_0)+1} = 2\bar{t} \leq 2^{-C_0+1}\nu$, which implies that $\log(\nu/t_k) \geq C_0 - 1 > 0$ and hence that $\max\{1, \lceil\log(\nu/t_k)\rceil\} = \lceil\log(\nu/t_k)\rceil \leq \log(\nu/t_k)+1$. Using these relations,

(A.0.5), (A.0.6), the relation $t_0 2^{K+1} < 4\bar{t}$ due to $K < \log(\bar{t}/t_0) + 1$, we obtain

$$\begin{aligned}
\sum_{k=0}^{K}\left\lceil \beta_0 + \sum_{l=1}^{L}\beta_l t_k^{p_l}\right\rceil \max\left\{1, \left\lceil \log\frac{\nu}{t_k}\right\rceil\right\} &\le \sum_{k=0}^{K}\left(1+\beta_0+\sum_{l=1}^{L}\beta_l t_k^{p_l}\right)\left(\log\frac{\nu}{t_k}+1\right)\\
&= (1+\beta_0)\sum_{k=0}^{K}\left(\log\frac{\nu}{t_k}+1\right) + \sum_{l=1}^{L}\sum_{k=0}^{K}\beta_l t_0^{p_l}2^{kp_l}\left(\log\frac{\nu}{t_k}+1\right)\\
&= (1+\beta_0)\sum_{k=0}^{K}\left(\log\frac{\nu}{t_k}+1\right) + \sum_{l=1}^{L}\sum_{k=0}^{K}\beta_l t_0^{p_l}2^{kp_l}\left(\log\frac{2\nu}{t_0}-k\right)\\
&\le \left\{5(1+\beta_0) + \left(\frac{4}{p_1^2}+\frac{7}{p_1}\right)\beta_1\bar{t}^{p_1} + 2C_0\sum_{l=1}^{L}\beta_l t_0^{p_l}2^{p_l(K+1)}\right\}\left\lceil \log\frac{\nu}{\bar{t}}\right\rceil\\
&\le \left[5(1+\beta_0) + \left(\frac{4}{p_1^2}+\frac{7}{p_1}\right)\beta_1\bar{t}^{p_1} + 2C_0\sum_{l=1}^{L}4^{p_l}\beta_l\bar{t}^{p_l}\right]\left\lceil \log\frac{\nu}{\bar{t}}\right\rceil\\
&\le \left[5+\max\left\{5, \frac{4}{p_1^2}+\frac{7}{p_1}+2C_0 4^{p_1}, 2C_0\max_{2\le l\le L}4^{p_l}\right\}\left(\beta_0+\sum_{l=1}^{L}\beta_l\bar{t}^{p_l}\right)\right]\left\lceil \log\frac{\nu}{\bar{t}}\right\rceil.
\end{aligned}$$

Inequality (A.0.4) now immediately follows from the above conclusion, Lemma A.0.1, the fact that

$\max\{1, \lceil\log(\nu/\bar{t})\rceil\} = \lceil\log(\nu/\bar{t})\rceil$ due to the assumption $\bar{t} \le \nu 2^{-C_1}$, and some trivial majorization. ∎

We are now ready to prove Proposition 5.4.1.

Proof of Proposition 5.4.1. Assume first that $\bar{t} \le t_0$. Due to the definition of K in (5.4.5), we have $K = 0$ in this case, which, in view of the definition of t_0 in (5.4.5) and Lemma A.0.1, then imply that

$$\begin{aligned}
\sum_{k=0}^{K}\left\lceil \beta_0 + \sum_{l=1}^{L}\beta_l t_k^{p_l}\right\rceil \max\left\{1, \left\lceil \log\frac{\nu}{t_k}\right\rceil\right\} &= \left\lceil \beta_0 + \sum_{l=1}^{L}\beta_l t_0^{p_l}\right\rceil \max\left\{1, \left\lceil \log\frac{\nu}{t_0}\right\rceil\right\}\\
&\le \left\lceil \beta_0 + \sum_{l=1}^{L}\beta_l t_0^{p_l}\right\rceil \max\left\{1, \left\lceil \log\frac{\nu}{\bar{t}}\right\rceil\right\} \le \lceil\beta_0 + L\max\{\beta_0, 1\}\rceil \max\left\{1, \left\lceil \log\frac{\nu}{\bar{t}}\right\rceil\right\}\\
&\le \lceil(L+1)\beta_0 + L\rceil \max\left\{1, \left\lceil \log\frac{\nu}{\bar{t}}\right\rceil\right\} \le (2L+1)\lceil\beta_0\rceil \max\left\{1, \left\lceil \log\frac{\nu}{\bar{t}}\right\rceil\right\}\\
&\le (2L+1)\left\lceil \beta_0 + \sum_{l=1}^{L}\beta_l\bar{t}^{p_l}\right\rceil \max\left\{1, \left\lceil \log\frac{\nu}{\bar{t}}\right\rceil\right\}.
\end{aligned}$$

Hence, in the case where $\bar{t} \le t_0$, inequality (5.4.4) holds with $C = 2L + 1$.

Assume now that $\bar{t} > t_0$. Denoting $\tilde{t} := 2^{-C_1}\nu$, where C_1 is given in (A.0.2), we further consider two subcases: a) $\tilde{t} \ge \bar{t}$; b) $\tilde{t} \le \bar{t}$. In subcase a), we have $\tilde{t} \ge \bar{t} > t_0$, which, in view

of Lemma A.0.4, clearly implies that inequality (5.4.4) holds with $C = C_2$.

We now consider the remaining subcase b) where $\tilde{t} \leq \bar{t}$. Denoting $\tilde{K} := \max\{0, \lceil \log(\tilde{t}/t_0) \rceil\}$, we have $t_k \geq \tilde{t}$ for any $k \geq \tilde{K}$ and hence that,

$$\max\left\{1, \left\lceil \log \frac{\nu}{t_k} \right\rceil\right\} \leq \max\left\{1, \left\lceil \log \frac{\nu}{\bar{t}} \right\rceil\right\} = \max\left\{1, \lceil C_1 \rceil\right\} \leq C_1 + 1, \, \forall k \geq \tilde{K},$$

which together with Lemma A.0.2, then imply that

$$\begin{aligned}
&\sum_{k=\tilde{K}}^{K} \left\lceil \beta_0 + \sum_{l=1}^{L} \left(\beta_l \, t_k^{p_l}\right) \right\rceil \max\left\{1, \left\lceil \log \frac{\nu}{t_k} \right\rceil\right\} \\
&\quad \leq \quad (C_1+1) \sum_{k=\tilde{K}}^{K} \left\lceil \beta_0 + \sum_{l=1}^{L} \left(\beta_l \, t_k^{p_l}\right) \right\rceil \leq (C_1+1) \sum_{k=0}^{K} \left\lceil \beta_0 + \sum_{l=1}^{L} \left(\beta_l \, t_k^{p_l}\right) \right\rceil \\
&\quad \leq \quad C_0(C_1+1) \left\lceil \beta_0 + \sum_{l=1}^{L} \beta_l \bar{t}^{p_l} \right\rceil \max\left\{1, \left\lceil \log \frac{\nu}{\bar{t}} \right\rceil\right\}.
\end{aligned}$$

Moreover, if $\tilde{K} \geq 1$, i.e., $\tilde{t} > t_0$, it then follows from Lemma A.0.4 with $\bar{t} = \tilde{t}$ that

$$\begin{aligned}
\sum_{k=0}^{\tilde{K}-1} \left\lceil \beta_0 + \sum_{l=1}^{L} \beta_l t_k^{p_l} \right\rceil \left\lceil \log \frac{\nu}{t_k} \right\rceil \quad &\leq \quad C_2 \left\lceil \beta_0 + \sum_{l=1}^{L} \beta_l \tilde{t}^{p_l} \right\rceil \left\lceil \log \frac{\nu}{\tilde{t}} \right\rceil = C_2 \left\lceil \beta_0 + \sum_{l=1}^{L} \beta_l \tilde{t}^{p_l} \right\rceil \lceil C_1 \rceil \\
&\leq \quad C_2 \left\lceil \beta_0 + \sum_{l=1}^{L} \beta_l \bar{t}^{p_l} \right\rceil \lceil C_1 \rceil \\
&\leq \quad (C_1+1) C_2 \left\lceil \beta_0 + \sum_{l=1}^{L} \beta_l \bar{t}^{p_l} \right\rceil \max\left\{1, \left\lceil \log \frac{\nu}{\bar{t}} \right\rceil\right\},
\end{aligned}$$

where the second inequality follows from the assumption $\tilde{t} \leq \bar{t}$. Combining the previous two conclusions, we conclude that (5.4.4) holds with $C = (C_1+1)(C_0+C_2)$. ∎

APPENDIX B

DETAILED NUMERICAL RESULTS FOR VALIDATION ANALYSIS

Table 18: SA vs SAA for EU-1

-		VERIFICATION				CONSTRUCTION		
ALG.	$step$	obj	dev	$\tilde{lb}^N$	lb^N	$\overline{f}^N$	ub	$time$
N-SA (N=100)	0.005	-19.3503	3.1475	-18.9132	-18.9132	-20.6156	-19.2006	0
	0.010	-19.3420	3.1445	-18.9131	-18.9131	-20.6156	-19.1929	1
	0.050	-19.2762	3.1205	-18.9127	-18.9127	-20.6156	-19.1315	0
	0.100	-19.1952	3.0915	-18.9124	-18.9124	-20.6156	-19.0559	0
	0.500	-18.5939	2.8906	-18.9112	-18.9112	-20.6156	-18.4886	1
	1.000	-17.9408	2.6962	-18.9115	-18.9115	-20.6156	-17.8641	0
	5.000	-14.8128	1.9667	-18.9265	-18.9265	-20.6156	-14.8226	0
	10.000	-13.0173	1.6291	-18.9382	-18.9382	-20.6156	-13.0614	1
N-SA (N=1000)	0.005	-19.3549	3.1484	-19.2891	-19.2863	-19.4403	-19.1874	2
	0.010	-19.3512	3.1463	-19.2891	-19.2863	-19.4403	-19.1838	2
	0.050	-19.3219	3.1293	-19.2891	-19.2862	-19.4403	-19.1552	2
	0.100	-19.2856	3.1088	-19.2891	-19.2862	-19.4403	-19.1197	2
	0.500	-19.0087	2.9659	-19.2891	-19.2861	-19.4403	-18.8495	2
	1.000	-18.6918	2.8258	-19.2892	-19.2862	-19.4403	-18.5398	2
	5.000	-16.8467	2.2564	-19.2909	-19.2876	-19.4403	-16.7191	2
	10.000	-15.3926	1.9396	-19.2923	-19.2888	-19.4403	-15.2802	2
N-SA (N=2000)	0.005	-19.3558	3.1487	-19.2998	-19.2994	-19.4063	-19.2671	3
	0.010	-19.3530	3.1468	-19.2998	-19.2994	-19.4063	-19.2643	3
	0.050	-19.3310	3.1319	-19.2998	-19.2994	-19.4063	-19.2426	3
	0.100	-19.3038	3.1139	-19.2998	-19.2994	-19.4063	-19.2156	3
	0.500	-19.0948	2.9879	-19.2998	-19.2994	-19.4063	-19.0088	3
	1.000	-18.8530	2.8640	-19.2999	-19.2995	-19.4063	-18.7693	4
	5.000	-17.4311	2.3983	-19.3011	-19.3007	-19.4063	-17.3062	3
	10.000	-16.2469	2.1547	-19.3021	-19.3016	-19.4063	-16.0626	3
E-SA (N=100)	0.005	-18.6749	2.9086	-18.9113	-18.9113	-20.6156	-18.5716	0
	0.010	-17.9962	2.6935	-18.9123	-18.9123	-20.6156	-17.9301	0
	0.050	-15.1313	2.0948	-18.9150	-18.9150	-20.6156	-15.1044	0
	0.100	-14.1627	1.9553	-18.9134	-18.9134	-20.6156	-14.1330	0
	0.500	-13.2785	1.8369	-18.9124	-18.9124	-20.6156	-13.3260	1
	1.000	-13.1712	1.8221	-18.9122	-18.9122	-20.6156	-13.2404	0
	5.000	-13.0722	1.8108	-18.9122	-18.9122	-20.6156	-13.1487	1
	10.000	-13.0589	1.8091	-18.9122	-18.9122	-20.6156	-13.1320	0
E-SA (N=1000)	0.005	-19.0549	2.9808	-19.2891	-19.2861	-19.4403	-18.8944	1
	0.010	-18.7532	2.8325	-19.2893	-19.2862	-19.4403	-18.6035	1
	0.050	-16.9969	2.3496	-19.2898	-19.2867	-19.4403	-16.8469	1
	0.100	-15.7783	2.3432	-19.2896	-19.2866	-19.4403	-15.6116	2
	0.500	-13.6786	1.8912	-19.2893	-19.2864	-19.4403	-13.5487	1
	1.000	-13.3306	1.8349	-19.2892	-19.2863	-19.4403	-13.2329	1
	5.000	-13.0408	1.8000	-19.2892	-19.2863	-19.4403	-12.9527	1
	10.000	-13.0024	1.7957	-19.2892	-19.2863	-19.4403	-12.9122	1
E-SA (N=2000)	0.005	-19.1311	3.0016	-19.2998	-19.2994	-19.4063	-19.0447	2
	0.010	-18.9050	2.8701	-19.3000	-19.2995	-19.4063	-18.8221	1
	0.050	-17.5945	2.5185	-19.3003	-19.2999	-19.4063	-17.4574	2
	0.100	-16.5355	2.3273	-19.3003	-19.2999	-19.4063	-16.3353	2
	0.500	-14.4868	2.3222	-19.3000	-19.2996	-19.4063	-13.8779	2
	1.000	-14.1098	2.3486	-19.2999	-19.2996	-19.4063	-13.4123	2
	5.000	-13.5949	2.3780	-19.2999	-19.2995	-19.4063	-13.0214	2
	10.000	-13.4814	2.3808	-19.2999	-19.2995	-19.4063	-12.9633	2
SAA	N=100	-18.5799	2.8127	-	-	-20.6156	-20.6156	2
SAA	N=1000	-19.2104	2.9673	-	-	-19.4403	-19.4403	14
SAA	N=2000	-19.2700	3.0019	-	-	-19.4063	-19.4063	27

Table 19: SA vs SAA for EU-2

-		VERIFICATION				CONSTRUCTION		
ALG.	$step$	obj	dev	$\tilde{lb}^N$	lb^N	$\overline{f}^N$	ub	$time$
N-SA (N=100)	0.005	-38.1358	6.3013	-62.0678	-62.0679	-67.3351	-37.7797	1
	0.010	-38.2777	6.3016	-62.0637	-62.0637	-67.3351	-37.8657	1
	0.050	-39.5633	6.3170	-62.0320	-62.0320	-67.3351	-38.5604	1
	0.100	-41.3122	6.3623	-61.9965	-61.9965	-67.3351	-39.4440	1
	0.500	-51.5592	7.1720	-61.8789	-61.8789	-67.3351	-45.6782	1
	1.000	-53.7202	7.5889	-61.8677	-61.8677	-67.3351	-48.6930	0
	5.000	-50.1970	7.2056	-61.8935	-61.8935	-67.3351	-48.9589	0
	10.000	-44.3844	6.2152	-61.8906	-61.8906	-67.3351	-44.6652	0
N-SA (N=1000)	0.005	-38.5134	6.3022	-62.8773	-62.8591	-63.0584	-37.9218	3
	0.010	-39.0381	6.3052	-62.8761	-62.8583	-63.0584	-38.1864	3
	0.050	-43.4184	6.4096	-62.8675	-62.8524	-63.0584	-40.3681	3
	0.100	-49.1950	6.7643	-62.8607	-62.8477	-63.0584	-43.2200	3
	0.500	-59.5124	7.9965	-62.8492	-62.8396	-63.0584	-53.4417	3
	1.000	-60.5022	8.2671	-62.8475	-62.8383	-63.0584	-56.0828	3
	5.000	-57.4415	8.0493	-62.8512	-62.8410	-63.0584	-55.6279	3
	10.000	-54.4349	7.6609	-62.8532	-62.8425	-63.0584	-53.1233	3
N-SA (N=2000)	0.005	-38.7364	6.3032	-62.8993	-62.8945	-62.9984	-38.1877	6
	0.010	-39.4895	6.3093	-62.8981	-62.8935	-62.9984	-38.5625	5
	0.050	-45.8521	6.5226	-62.8898	-62.8866	-62.9984	-41.6791	5
	0.100	-52.7069	7.0492	-62.8850	-62.8825	-62.9984	-45.5742	6
	0.500	-60.8674	8.2380	-62.8776	-62.8762	-62.9984	-55.6583	5
	1.000	-61.4004	8.4178	-62.8768	-62.8754	-62.9984	-57.9291	6
	5.000	-58.9613	8.1694	-62.8790	-62.8775	-62.9984	-57.5192	5
	10.000	-56.5552	8.1115	-62.8805	-62.8788	-62.9984	-55.3438	5
E-SA (N=100)	0.005	-44.7298	6.5411	-61.9388	-61.9387	-67.3351	-41.2221	0
	0.010	-50.8755	7.1191	-61.8813	-61.8813	-67.3351	-44.6372	0
	0.050	-52.9844	7.7711	-61.8799	-61.8799	-67.3351	-50.4057	1
	0.100	-48.9209	7.0075	-61.9036	-61.9036	-67.3351	-48.3108	1
	0.500	-43.1763	6.4523	-61.9653	-61.9653	-67.3351	-43.2668	0
	1.000	-42.4103	6.4126	-61.9776	-61.9776	-67.3351	-42.6107	0
	5.000	-41.7427	6.3835	-61.9889	-61.9889	-67.3351	-42.1865	1
	10.000	-41.6408	6.3801	-61.9907	-61.9907	-67.3351	-42.0068	1
E-SA (N=1000)	0.005	-56.4912	7.5038	-62.8524	-62.8419	-63.0584	-47.9520	2
	0.010	-60.6441	8.1788	-62.8485	-62.8390	-63.0584	-53.2857	1
	0.050	-59.4372	8.1330	-62.8483	-62.8389	-63.0584	-57.1628	2
	0.100	-56.4664	7.8226	-62.8509	-62.8408	-63.0584	-55.1036	1
	0.500	-47.1630	6.9548	-62.8617	-62.8485	-63.0584	-46.1799	2
	1.000	-44.2730	6.6700	-62.8657	-62.8512	-63.0584	-43.7097	1
	5.000	-42.1794	6.4795	-62.8697	-62.8539	-63.0584	-41.8224	2
	10.000	-41.9037	6.4625	-62.8702	-62.8543	-63.0584	-41.5539	1
E-SA (N=2000)	0.005	-59.5578	7.9151	-62.8792	-62.8776	-62.9984	-51.0444	3
	0.010	-61.7670	8.4029	-62.8772	-62.8758	-62.9984	-55.8117	4
	0.050	-60.3580	8.2543	-62.8773	-62.8759	-62.9984	-58.7962	3
	0.100	-58.0879	7.9671	-62.8788	-62.8773	-62.9984	-57.1121	3
	0.500	-48.9764	7.1088	-62.8862	-62.8836	-62.9984	-48.2204	3
	1.000	-45.8957	7.8486	-62.8898	-62.8865	-62.9984	-45.0630	4
	5.000	-43.3615	8.5239	-62.8937	-62.8898	-62.9984	-42.0979	3
	10.000	-43.1312	8.7238	-62.8943	-62.8903	-62.9984	-41.7328	3
SAA	N=100	-58.9225	8.9102	-	-	-67.3351	-67.3350	2
SAA	N=1000	-62.6459	8.9095	-	-	-63.0584	-63.0584	19
SAA	N=2000	-62.8749	8.9099	-	-	-62.9984	-62.9983	38

Table 20: SA vs SAA for EU-3

-		VERIFICATION				CONSTRUCTION		
ALG.	*step*	*obj*	*dev*	$\bar{lb}^N$	lb^N	$\overline{f}^N$	*ub*	*time*
N-SA (N=100)	0.005	-39.6117	6.3013	-82.3157	-82.3157	-88.4304	-39.2556	1
	0.010	-39.7537	6.3016	-82.3084	-82.3084	-88.4304	-39.3416	1
	0.050	-41.0393	6.3170	-82.2505	-82.2505	-88.4304	-40.0363	0
	0.100	-42.7881	6.3623	-82.1798	-82.1798	-88.4304	-40.9200	0
	0.500	-57.4667	7.8849	-81.8238	-81.8239	-88.4304	-48.3655	0
	1.000	-67.3666	9.5826	-81.7461	-81.7461	-88.4304	-55.4946	0
	5.000	-68.6110	10.8605	-81.7913	-81.7913	-88.4304	-63.1347	1
	10.000	-65.6632	11.2328	-81.8150	-81.8150	-88.4304	-62.5604	1
N-SA (N=1000)	0.005	-39.9893	6.3022	-83.1274	-82.9784	-83.1184	-39.3977	3
	0.010	-40.5141	6.3052	-83.1252	-82.9773	-83.1184	-39.6624	2
	0.050	-44.8943	6.4096	-83.1080	-82.9691	-83.1184	-41.8440	3
	0.100	-50.7953	6.7777	-83.0894	-82.9602	-83.1184	-44.7043	3
	0.500	-77.0042	10.6584	-83.0562	-82.9434	-83.1184	-63.0870	3
	1.000	-80.2141	11.4031	-83.0530	-82.9415	-83.1184	-70.4144	2
	5.000	-77.8238	11.1152	-83.0589	-82.9448	-83.1184	-74.4543	3
	10.000	-75.5959	11.1703	-83.0627	-82.9468	-83.1184	-72.6843	3
N-SA (N=2000)	0.005	-40.2123	6.3032	-83.1493	-82.9893	-83.0039	-39.6637	5
	0.010	-40.9655	6.3093	-83.1470	-82.9888	-83.0039	-40.0384	5
	0.050	-47.3281	6.5226	-83.1294	-82.9845	-83.0039	-43.1550	6
	0.100	-55.7484	7.2340	-83.1132	-82.9798	-83.0039	-47.2583	5
	0.500	-80.1848	11.3168	-83.0944	-82.9737	-83.0039	-67.8061	6
	1.000	-81.5215	11.7493	-83.0932	-82.9730	-83.0039	-73.7930	5
	5.000	-79.2787	11.3618	-83.0972	-82.9747	-83.0039	-76.7361	5
	10.000	-77.1648	10.9266	-83.0998	-82.9755	-83.0039	-75.1356	5
E-SA (N=100)	0.005	-46.2058	6.5411	-82.0663	-82.0663	-88.4304	-42.6980	0
	0.010	-52.9706	7.1919	-81.9096	-81.9096	-88.4304	-46.2106	0
	0.050	-69.0513	10.0283	-81.7446	-81.7446	-88.4304	-59.6579	1
	0.100	-64.4670	9.7883	-81.8014	-81.8014	-88.4304	-60.0157	1
	0.500	-51.4511	10.2614	-81.9668	-81.9668	-88.4304	-49.8357	0
	1.000	-49.3325	11.0861	-82.0287	-82.0287	-88.4304	-46.8866	0
	5.000	-45.5870	6.5358	-82.0956	-82.0956	-88.4304	-43.3252	1
	10.000	-45.3499	6.5216	-82.1024	-82.1024	-88.4304	-43.1719	1
E-SA (N=1000)	0.005	-60.9010	7.8322	-83.0739	-82.9526	-83.1184	-49.9017	2
	0.010	-74.4544	9.9248	-83.0616	-82.9463	-83.1184	-59.1547	2
	0.050	-79.1243	11.3623	-83.0530	-82.9414	-83.1184	-73.2204	2
	0.100	-75.8435	10.8148	-83.0551	-82.9428	-83.1184	-72.6632	2
	0.500	-60.2684	9.7266	-83.0720	-82.9515	-83.1184	-58.7534	1
	1.000	-53.4428	9.7788	-83.0846	-82.9579	-83.1184	-52.3161	2
	5.000	-46.7625	6.4998	-83.1021	-82.9663	-83.1184	-46.4668	2
	10.000	-45.9571	6.4594	-83.1048	-83.1048	-83.9029	-45.5528	2
E-SA (N=2000)	0.005	-68.6529	8.8532	-83.1047	-82.9771	-83.0039	-54.3649	3
	0.010	-78.6518	10.7185	-83.0975	-82.9748	-83.0039	-64.4228	3
	0.050	-80.5735	11.6903	-83.0931	-82.9730	-83.0039	-75.9516	3
	0.100	-78.2839	11.2592	-83.0945	-82.9736	-83.0039	-75.4904	3
	0.500	-64.3056	9.2455	-83.1041	-82.9769	-83.0039	-62.9399	3
	1.000	-56.8833	9.8019	-83.1131	-82.9798	-83.0039	-55.4457	4
	5.000	-48.9097	11.1414	-83.1294	-82.9845	-83.0039	-47.0649	3
	10.000	-47.8454	11.8443	-83.1323	-83.1322	-83.3646	-45.8304	3
SAA	N=100	-77.8239	12.5917	-	-	-88.4304	-88.4304	3
SAA	N=1000	-82.8458	12.6014	-	-	-83.1184	-83.1183	20
SAA	N=2000	-83.0845	12.6015	-	-	-83.0039	-83.0039	38

Table 21: SA vs SAA for EU-4

-		VERIFICATION				CONSTRUCTION		
ALG.	$step$	obj	dev	$\tilde{lb}^N$	lb^N	$\overline{f}^N$	ub	$time$
N-SA (N=100)	0.005	-39.1382	6.3012	-103.8668	-103.8668	-135.1012	-38.8100	1
	0.010	-39.2088	6.3013	-103.8612	-103.8612	-135.1012	-38.8528	0
	0.050	-39.7787	6.3059	-103.8158	-103.8156	-135.1012	-39.1969	0
	0.100	-40.6311	6.3169	-103.7576	-103.7575	-135.1012	-39.6309	0
	0.500	-47.8499	6.6972	-103.2766	-103.2703	-135.1012	-43.2413	0
	1.000	-57.3197	8.0030	-102.8337	-102.7924	-135.1012	-47.9764	1
	5.000	-86.1072	51.7058	-103.4591	-103.4589	-156.8338	-70.8437	1
	10.000	-90.2683	34.5832	-103.7156	-103.7169	-156.8338	-82.7600	1
N-SA (N=1000)	0.005	-39.3257	6.3014	-104.6774	-104.5178	-110.8567	-38.8632	3
	0.010	-39.5851	6.3022	-104.6757	-104.5164	-110.8567	-38.9944	3
	0.050	-41.7089	6.3270	-104.6621	-104.5034	-110.8567	-40.0605	3
	0.100	-44.4731	6.4088	-104.6440	-104.4866	-110.8567	-41.4323	3
	0.500	-67.7814	9.2616	-104.5130	-104.3583	-110.8567	-53.1137	3
	1.000	-86.1106	15.0456	-104.4343	-104.2763	-110.8567	-65.5627	3
	5.000	-99.5512	38.4448	-104.4886	-104.4763	-112.8418	-89.6211	3
	10.000	-98.4015	41.0811	-104.4814	-104.4814	-126.7905	-90.2347	3
N-SA (N=2000)	0.005	-39.4361	6.3017	-104.6990	-104.5673	-107.2058	-39.0746	5
	0.010	-39.8073	6.3032	-104.6976	-104.5658	-107.2058	-39.2600	5
	0.050	-42.8712	6.3533	-104.6831	-104.5530	-107.2058	-40.7748	5
	0.100	-46.8982	6.5210	-104.6635	-104.5357	-107.2058	-42.7393	5
	0.500	-76.9801	11.2353	-104.5415	-104.4233	-107.2058	-58.7447	6
	1.000	-92.4109	18.0166	-104.4804	-104.3618	-107.2058	-72.2851	6
	5.000	-100.1597	38.6309	-104.6123	-104.5046	-107.5820	-92.0804	6
	10.000	-98.7639	33.4377	-104.7341	-104.7341	-115.3887	-92.4497	5
E-SA (N=100)	0.005	-42.3324	6.3614	-103.6555	-103.6555	-135.1012	-40.5109	0
	0.010	-45.7234	6.5356	-103.4658	-103.4658	-135.1012	-42.2545	0
	0.050	-65.4026	9.1650	-102.8399	-102.8251	-135.1012	-54.0087	0
	0.100	-71.9510	11.0358	-102.5784	-102.5595	-135.1012	-60.2342	0
	0.500	-64.6265	11.8169	-103.0449	-102.6408	-135.1012	-61.9645	1
	1.000	-58.5949	13.5084	-102.9164	-102.5881	-135.1012	-57.2348	1
	5.000	-51.8700	7.5180	-103.0982	-103.0799	-135.1012	-51.9177	1
	10.000	-51.4719	8.1276	-103.3523	-103.3523	-244.1578	-46.3071	1
E-SA (N=1000)	0.005	-49.6303	6.7088	-104.6225	-104.4666	-110.8567	-44.0540	2
	0.010	-60.2552	7.8013	-104.5894	-104.4354	-110.8567	-49.3742	2
	0.050	-84.8213	12.8601	-104.5060	-104.3555	-110.8567	-71.1002	2
	0.100	-89.1139	15.4349	-104.4634	-104.3129	-110.8567	-78.0858	2
	0.500	-84.9472	18.8111	-104.3991	-104.2318	-110.8567	-80.3334	1
	1.000	-75.0741	14.9765	-104.4396	-104.2645	-110.8567	-73.2758	2
	5.000	-55.2427	7.3760	-104.5551	-104.4008	-110.8567	-54.8496	2
	10.000	-52.8571	7.0963	-104.5760	-104.5760	-126.2563	-51.0361	2
E-SA (N=2000)	0.005	-54.0474	7.0659	-104.6468	-104.5213	-107.2058	-46.3947	3
	0.010	-68.0061	8.8163	-104.6229	-104.5002	-107.2058	-53.7987	3
	0.050	-88.4424	14.1327	-104.5570	-104.4399	-107.2058	-75.8246	3
	0.100	-92.1313	16.8344	-104.5233	-104.4072	-107.2058	-82.1191	4
	0.500	-89.9439	21.9351	-104.4546	-104.3297	-107.2058	-84.7472	3
	1.000	-81.7297	19.6360	-104.4727	-104.3299	-106.7333	-77.5126	3
	5.000	-57.7183	8.8114	-104.5973	-104.4756	-107.2058	-56.7866	3
	10.000	-55.4649	15.4939	-104.6239	-104.5875	-109.0572	-52.8853	3
SAA	N=100	-72.3744	143.0249	-	-	-134.5648	-134.5647	3
SAA	N=1000	-96.2697	72.8944	-	-	-108.3190	-108.3142	93
SAA	N=2000	-99.3096	61.1053	-	-	-105.0890	-105.0881	163

Table 22: SA vs SAA for EU-5

-		VERIFICATION				CONSTRUCTION		
ALG.	$step$	obj	dev	$\tilde{lb}^N$	lb^N	$\overline{f}^N$	ub	$time$
N-SA (N=100)	0.005	-38.3300	6.3012	-103.8925	-103.8925	-146.7304	-38.0152	1
	0.010	-38.3669	6.3012	-103.8890	-103.8890	-146.7304	-38.0375	1
	0.050	-38.6626	6.3019	-103.8610	-103.8610	-146.7304	-38.2167	1
	0.100	-39.0423	6.3062	-103.8250	-103.8250	-146.7304	-38.4418	0
	0.500	-42.6895	6.4052	-103.5194	-103.5183	-146.7304	-40.2841	0
	1.000	-47.4963	6.7351	-103.2526	-103.2406	-146.7304	-42.6755	0
	5.000	-79.9383	18.7838	-103.1278	-102.9631	-146.7304	-61.4136	0
	10.000	-82.0792	34.7555	-104.2617	-104.2619	-279.7724	-52.0580	0
N-SA (N=1000)	0.005	-38.4278	6.3012	-104.7014	-104.5825	-112.9030	-38.0263	3
	0.010	-38.5629	6.3014	-104.7006	-104.5813	-112.9030	-38.0947	3
	0.050	-39.6565	6.3081	-104.6938	-104.5749	-112.9030	-38.6461	3
	0.100	-41.0560	6.3294	-104.6840	-104.5659	-112.9030	-39.3467	3
	0.500	-53.2736	7.1266	-104.6034	-104.4865	-112.9030	-45.3025	3
	1.000	-68.1682	9.5052	-104.5172	-104.3987	-112.9030	-52.9831	3
	5.000	-97.9097	33.0506	-104.5134	-104.1638	-112.9030	-82.6433	3
	10.000	-97.3159	38.5009	-104.4859	-104.4859	-123.3988	-85.8145	3
N-SA (N=2000)	0.005	-38.4854	6.3013	-104.7198	-104.5355	-108.4063	-38.2120	5
	0.010	-38.6783	6.3017	-104.7228	-104.5320	-108.4063	-38.3085	5
	0.050	-40.2483	6.3150	-104.7155	-104.5281	-108.4063	-39.0894	5
	0.100	-42.2733	6.3583	-104.7058	-104.5187	-108.4063	-40.0870	5
	0.500	-59.6215	7.8891	-104.6267	-104.4384	-108.4063	-48.5978	6
	1.000	-77.4046	11.5699	-104.5631	-104.3676	-108.4063	-58.7659	5
	5.000	-99.5680	35.1278	-104.5241	-104.0644	-107.5745	-86.5396	6
	10.000	-98.9885	34.4152	-104.9583	-104.9291	-109.5558	-90.8342	5
E-SA (N=100)	0.005	-41.4236	6.3568	-103.6469	-103.6468	-146.7304	-39.6674	1
	0.010	-44.6804	6.5178	-103.4361	-103.4354	-146.7304	-41.3419	1
	0.050	-64.1034	9.0602	-102.8722	-102.8181	-146.7304	-52.8315	0
	0.100	-70.9505	10.9285	-102.7239	-102.5757	-146.7304	-59.1516	0
	0.500	-64.2812	11.8766	-103.3224	-102.7288	-146.7304	-61.4639	1
	1.000	-58.1064	13.4753	-103.5058	-102.7123	-146.7304	-56.6712	1
	5.000	-51.1580	7.5291	-103.4436	-103.4436	-146.7304	-51.2048	0
	10.000	-51.2166	13.0679	-103.8111	-103.8111	-261.9004	-45.2895	0
E-SA (N=1000)	0.005	-48.4335	6.6780	-104.6471	-104.5295	-112.9030	-43.0687	2
	0.010	-58.6551	7.6979	-104.6129	-104.4961	-112.9030	-48.1791	2
	0.050	-83.6780	12.7125	-104.5254	-104.4092	-112.9030	-69.8393	1
	0.100	-88.1730	15.2741	-104.4785	-104.3620	-112.9030	-76.9825	2
	0.500	-84.5810	18.8582	-104.4300	-104.2320	-112.9030	-79.8171	1
	1.000	-74.9702	15.3083	-104.4912	-104.2363	-112.9030	-73.0591	2
	5.000	-54.6844	7.4084	-104.5690	-104.4563	-112.9030	-54.2912	1
	10.000	-52.2596	7.1116	-104.4963	-104.4963	-127.0456	-50.5816	2
E-SA (N=2000)	0.005	-52.6794	7.0095	-104.6720	-104.4858	-108.4063	-45.3230	3
	0.010	-66.3654	8.6960	-104.6477	-104.4615	-108.4063	-52.4664	3
	0.050	-87.3763	13.9872	-104.5820	-104.3943	-108.4063	-74.6040	3
	0.100	-91.2051	16.6639	-104.5481	-104.3577	-108.4063	-81.0510	4
	0.500	-89.4774	21.9242	-104.4849	-104.2575	-108.4063	-84.1870	3
	1.000	-81.5608	19.8583	-104.4922	-104.2144	-107.5745	-77.2696	3
	5.000	-57.3049	8.9504	-104.5317	-104.4206	-108.4811	-56.3306	3
	10.000	-54.7480	15.3229	-104.6310	-104.5285	-111.2797	-52.2968	4
SAA	N=100	16.3822	352.8689	-	-	-139.9952	-139.9946	7
SAA	N=1000	-95.5139	72.8589	-	-	-107.5509	-107.5479	94
SAA	N=2000	-98.5458	60.8440	-	-	-104.3214	-104.3189	123

Table 23: SA vs SAA for EU-6

-		VERIFICATION				CONSTRUCTION		
ALG.	$step$	obj	dev	$\tilde{lb}^N$	lb^N	$\overline{f}^N$	ub	$time$
N-SA (N=100)	0.005	-3.2692	0.5659	-9.1974	-9.1929	-13.1477	-3.2371	0
	0.010	-3.2854	0.5654	-9.1828	-9.1782	-13.1154	-3.2470	0
	0.050	-3.4342	0.5608	-9.1341	-9.1268	-13.0708	-3.3265	1
	0.100	-3.6279	0.5575	-9.0201	-9.0084	-12.9881	-3.4259	1
	0.500	-4.8157	0.5394	-7.7299	-7.7046	-11.6856	-4.1219	0
	1.000	-5.5010	0.6512	-7.3008	-7.2726	-10.4938	-4.6646	0
	5.000	-5.8002	1.2173	-7.3728	-7.3052	-8.9175	-5.5545	1
	10.000	-5.5984	1.4642	-7.5619	-7.5619	-9.5508	-4.2212	1
N-SA (N=1000)	0.005	-3.3121	0.5647	-9.2181	-9.2067	-9.9742	-3.2526	3
	0.010	-3.3715	0.5632	-9.1979	-9.1864	-9.9567	-3.2827	3
	0.050	-3.8487	0.5479	-8.8674	-8.8565	-9.8440	-3.5257	3
	0.100	-4.3741	0.5163	-8.1960	-8.1924	-9.5077	-3.8138	2
	0.500	-5.9203	0.5450	-6.9083	-6.9083	-8.0532	-5.0557	3
	1.000	-6.1874	0.5714	-6.7434	-6.7118	-7.3578	-5.5638	3
	5.000	-6.2676	0.7185	-6.7578	-6.6102	-6.7471	-6.0365	3
	10.000	-6.2267	0.8552	-6.8889	-6.7390	-6.8616	-5.9295	2
N-SA (N=2000)	0.005	-3.3373	0.5641	-9.2038	-9.1858	-9.5292	-3.2789	5
	0.010	-3.4221	0.5620	-9.1728	-9.1555	-9.5017	-3.3212	5
	0.050	-4.0807	0.5371	-8.5959	-8.5939	-9.2867	-3.6586	5
	0.100	-4.6990	0.4988	-7.8122	-7.8122	-8.8931	-4.0284	5
	0.500	-6.0973	0.5415	-6.7822	-6.7568	-7.5327	-5.3268	6
	1.000	-6.2570	0.5726	-6.7112	-6.6218	-7.0512	-5.7515	6
	5.000	-6.2999	0.6798	-6.7664	-6.5228	-6.6111	-6.1190	6
	10.000	-6.2680	0.7645	-6.8086	-6.5796	-6.6414	-6.0729	6
E-SA (N=100)	0.005	-4.0352	0.5470	-8.6247	-8.6115	-12.7305	-3.6504	0
	0.010	-4.5783	0.5240	-7.9275	-7.9102	-11.8871	-3.9938	1
	0.050	-5.4874	0.5718	-7.2348	-7.1953	-10.2022	-4.8879	0
	0.100	-5.4693	0.6366	-7.3451	-7.2166	-9.9064	-5.0562	0
	0.500	-4.6187	0.6189	-7.9389	-7.8735	-10.8348	-4.5629	1
	1.000	-4.4133	0.5711	-8.1226	-8.0784	-11.1633	-4.3515	0
	5.000	-4.2599	0.5936	-8.3883	-8.3823	-11.4331	-3.9670	0
	10.000	-4.2737	0.6287	-8.3853	-8.3853	-11.2202	-3.9810	1
E-SA (N=1000)	0.005	-5.0592	0.4849	-7.5206	-7.5206	-8.9054	-4.2862	1
	0.010	-5.6208	0.4926	-7.1603	-7.1603	-8.3405	-4.8291	2
	0.050	-6.1205	0.5189	-6.7480	-6.7248	-7.3575	-5.6980	2
	0.100	-6.1459	0.5583	-6.7191	-6.6559	-7.1219	-5.8464	2
	0.500	-5.6174	0.6491	-7.1390	-7.0398	-7.2743	-5.4976	1
	1.000	-4.9747	0.5283	-7.5679	-7.5189	-8.1052	-4.8908	1
	5.000	-4.3580	0.5235	-8.2042	-8.1930	-8.8407	-4.1480	2
	10.000	-4.3034	0.5275	-8.3010	-8.2835	-8.9120	-3.9828	2
E-SA (N=2000)	0.005	-5.3939	0.4874	-7.3006	-7.3006	-8.2738	-4.5691	3
	0.010	-5.8117	0.4944	-7.0176	-7.0176	-7.8526	-5.0944	3
	0.050	-6.1948	0.5143	-6.6925	-6.6485	-7.1033	-5.8463	3
	0.100	-6.2211	0.5543	-6.6867	-6.5849	-6.9171	-5.9714	3
	0.500	-5.8539	0.7090	-7.0041	-6.7430	-6.9879	-5.7299	3
	1.000	-5.2905	0.7075	-7.3551	-7.2946	-7.5212	-5.1628	3
	5.000	-4.4801	1.1354	-8.1429	-8.1410	-8.5007	-4.2242	3
	10.000	-4.3277	0.5561	-8.2469	-8.2469	-8.6401	-4.0363	3
SAA	N=100	-5.5376	2.1216	-	-	-6.8526	-6.8525	26
SAA	N=1000	-6.2782	0.7413	-	-	-6.4036	-6.4036	102
SAA	N=2000	-6.3073	0.7030	-	-	-6.3658	-6.3657	171

Table 24: SA vs SAA for EU-7

-		VERIFICATION				CONSTRUCTION		
ALG.	*step*	*obj*	*dev*	$\bar{lb}^N$	lb^N	$\overline{f}^N$	*ub*	*time*
N-SA (N=100)	0.005	-10.1487	0.6301	-16.7163	-16.7119	-21.0028	-10.1177	0
	0.010	-10.1512	0.6301	-16.7162	-16.7117	-21.0028	-10.1192	0
	0.050	-10.1711	0.6302	-16.7150	-16.7102	-21.0028	-10.1313	0
	0.100	-10.1961	0.6303	-16.7134	-16.7083	-21.0028	-10.1464	1
	0.500	-10.4349	0.6348	-16.7012	-16.6925	-21.0028	-10.2695	1
	1.000	-10.7510	0.6493	-16.6868	-16.6719	-21.0028	-10.4278	1
	5.000	-13.2835	1.1899	-16.6515	-16.5916	-21.0028	-11.7522	0
	10.000	-14.8802	3.0185	-16.6923	-16.6556	-21.0028	-13.0563	0
N-SA (N=1000)	0.005	-10.1553	0.6301	-16.7971	-16.7844	-17.6115	-10.1173	2
	0.010	-10.1644	0.6301	-16.7971	-16.7844	-17.6115	-10.1219	2
	0.050	-10.2378	0.6304	-16.7966	-16.7839	-17.6115	-10.1590	2
	0.100	-10.3310	0.6314	-16.7961	-16.7833	-17.6115	-10.2058	3
	0.500	-11.1321	0.6664	-16.7917	-16.7778	-17.6115	-10.5988	3
	1.000	-12.1903	0.7818	-16.7871	-16.7707	-17.6115	-11.1174	2
	5.000	-15.7924	2.3172	-16.7884	-16.7433	-17.6115	-13.9232	3
	10.000	-16.0895	3.9575	-16.7985	-16.7517	-17.6916	-14.8794	3
N-SA (N=2000)	0.005	-10.1592	0.6301	-16.7993	-16.7752	-17.1800	-10.1350	5
	0.010	-10.1722	0.6301	-16.7993	-16.7752	-17.1800	-10.1415	5
	0.050	-10.2773	0.6307	-16.7988	-16.7748	-17.1800	-10.1939	5
	0.100	-10.4117	0.6327	-16.7982	-16.7743	-17.1800	-10.2604	5
	0.500	-11.5675	0.7016	-16.7931	-16.7699	-17.1800	-10.8237	5
	1.000	-12.9902	0.9087	-16.7882	-16.7649	-17.1800	-11.5501	5
	5.000	-16.0889	2.6801	-16.7911	-16.7272	-17.1800	-14.4612	5
	10.000	-16.2514	3.5233	-16.8289	-16.7060	-17.0130	-15.2442	5
E-SA (N=100)	0.005	-10.4177	0.6344	-16.7022	-16.6946	-21.0028	-10.2623	1
	0.010	-10.7019	0.6467	-16.6896	-16.6790	-21.0028	-10.4084	0
	0.050	-12.5437	0.8717	-16.6406	-16.6173	-21.0028	-11.4639	0
	0.100	-13.3191	1.0560	-16.6219	-16.5900	-21.0028	-12.1220	1
	0.500	-12.8743	1.2036	-16.6803	-16.6111	-21.0028	-12.5344	0
	1.000	-12.2309	1.3391	-16.7054	-16.6083	-21.0028	-12.0632	0
	5.000	-11.4557	0.7573	-16.7121	-16.6892	-21.0028	-11.4739	1
	10.000	-11.3683	0.8805	-16.7440	-16.7440	-38.9462	-9.4234	0
E-SA (N=1000)	0.005	-11.0297	0.6590	-16.7928	-16.7794	-17.6115	-10.5586	1
	0.010	-11.9230	0.7390	-16.7899	-16.7761	-17.6115	-11.0045	1
	0.050	-14.5576	1.2241	-16.7834	-16.7671	-17.6115	-13.1309	2
	0.100	-15.0710	1.4747	-16.7808	-16.7621	-17.6115	-13.8965	2
	0.500	-14.9006	1.8922	-16.7831	-16.7486	-17.6115	-14.3748	1
	1.000	-14.0335	1.7091	-16.7764	-16.7400	-17.6115	-13.7961	2
	5.000	-11.8605	0.7532	-16.7853	-16.7631	-17.7664	-11.8199	2
	10.000	-11.5749	0.7139	-16.7770	-16.7745	-19.4777	-11.3529	2
E-SA (N=2000)	0.005	-11.4013	0.6847	-16.7948	-16.7716	-17.1800	-10.7573	3
	0.010	-12.6465	0.8260	-16.7925	-16.7696	-17.1800	-11.3853	3
	0.050	-14.9456	1.3482	-16.7868	-16.7642	-17.1800	-13.6209	3
	0.100	-15.3818	1.6120	-16.7841	-16.7605	-17.1800	-14.3156	3
	0.500	-15.3547	2.1793	-16.7834	-16.7375	-17.1800	-14.7953	3
	1.000	-14.6640	2.0545	-16.7838	-16.7280	-17.0917	-14.2104	3
	5.000	-12.2430	1.0414	-16.7730	-16.7523	-17.2239	-12.0750	3
	10.000	-11.8296	1.5142	-16.7796	-16.7683	-17.4557	-11.5982	3
SAA	N=100	-3.3040	37.2588	-	-	-20.2507	-20.2507	8
SAA	N=1000	-15.8481	6.9923	-	-	-17.0154	-17.0154	92
SAA	N=2000	-16.1474	5.7941	-	-	-16.7027	-16.7027	162

Table 25: SA vs SAA for EU-8

-		VERIFICATION				CONSTRUCTION		
ALG.	$step$	obj	dev	$\tilde{lb}^N$	lb^N	$\overline{f}^N$	ub	$time$
N-SA (N=100)	0.005	-36.2835	6.3012	-101.9556	-101.9114	-144.8201	-35.9718	0
	0.010	-36.3124	6.3012	-101.9538	-101.9092	-144.8201	-35.9893	1
	0.050	-36.5442	6.3016	-101.9395	-101.8916	-144.8201	-36.1298	1
	0.100	-36.8355	6.3030	-101.9215	-101.8694	-144.8201	-36.3061	0
	0.500	-39.6483	6.3649	-101.7797	-101.6844	-144.8201	-37.7425	0
	1.000	-43.3600	6.5633	-101.6175	-101.4440	-144.8201	-39.5960	1
	5.000	-71.6724	13.9475	-101.3197	-100.7563	-144.8201	-54.8455	1
	10.000	-84.9435	40.5205	-101.8959	-101.5494	-144.8201	-68.2916	0
N-SA (N=1000)	0.005	-36.3602	6.3012	-102.7634	-102.6365	-110.9079	-35.9730	2
	0.010	-36.4661	6.3013	-102.7628	-102.6358	-110.9079	-36.0266	3
	0.050	-37.3210	6.3054	-102.7580	-102.6303	-110.9079	-36.4582	3
	0.100	-38.4101	6.3184	-102.7518	-102.6230	-110.9079	-37.0048	3
	0.500	-47.8370	6.7992	-102.7009	-102.5581	-110.9079	-41.6132	2
	1.000	-60.0317	8.3413	-102.6522	-102.4779	-110.9079	-47.6696	3
	5.000	-94.2198	26.2742	-102.7009	-102.2027	-110.9079	-76.7166	3
	10.000	-96.0841	35.6518	-102.7998	-102.7069	-113.2295	-84.5569	3
N-SA (N=2000)	0.005	-36.4053	6.3013	-102.7854	-102.5447	-106.5921	-36.1526	5
	0.010	-36.5565	6.3015	-102.7847	-102.5441	-106.5921	-36.2283	5
	0.050	-37.7823	6.3096	-102.7793	-102.5397	-106.5921	-36.8390	5
	0.100	-39.3542	6.3358	-102.7722	-102.5338	-106.5921	-37.6158	5
	0.500	-52.8930	7.2748	-102.7137	-102.4821	-106.5921	-44.2204	5
	1.000	-68.7131	9.9199	-102.6639	-102.4285	-106.5921	-52.5656	5
	5.000	-96.6775	30.0392	-102.7187	-102.0942	-106.5921	-81.6017	5
	10.000	-97.3613	36.3939	-103.1348	-102.6311	-106.7958	-88.4846	6
E-SA (N=100)	0.005	-39.4169	6.3579	-101.7935	-101.7127	-144.8201	-37.6452	0
	0.010	-42.7055	6.5220	-101.6526	-101.5378	-144.8201	-39.3361	1
	0.050	-62.1922	9.0853	-101.1522	-100.9075	-144.8201	-50.8899	0
	0.100	-68.9682	10.9544	-100.9908	-100.6405	-144.8201	-57.1876	0
	0.500	-62.1416	11.8625	-101.6391	-100.8601	-144.8201	-59.3630	1
	1.000	-55.9963	13.4815	-101.8781	-100.9130	-144.8201	-54.5801	0
	5.000	-49.1046	7.5264	-101.9411	-101.7109	-144.8201	-49.1509	0
	10.000	-49.1742	13.0490	-102.2136	-102.2136	-264.0121	-43.3696	1
E-SA (N=1000)	0.005	-46.4951	6.6852	-102.7147	-102.5803	-110.9079	-41.0802	2
	0.010	-56.8137	7.7224	-102.6853	-102.5454	-110.9079	-46.2404	2
	0.050	-81.7285	12.7478	-102.6202	-102.4536	-110.9079	-67.9185	2
	0.100	-86.1748	15.3124	-102.5958	-102.4023	-110.9079	-75.0241	1
	0.500	-82.4456	18.8482	-102.6128	-102.2557	-110.9079	-77.7174	2
	1.000	-72.7705	15.2295	-102.5510	-102.2047	-110.9079	-70.8866	2
	5.000	-52.5930	7.4005	-102.6446	-102.5023	-110.9079	-52.1998	2
	10.000	-50.1320	7.1088	-102.4849	-102.4849	-129.2836	-48.5130	1
E-SA (N=2000)	0.005	-50.7816	7.0227	-102.7354	-102.5038	-106.5921	-43.3549	3
	0.010	-64.5382	8.7273	-102.7125	-102.4846	-106.5921	-50.5611	3
	0.050	-85.4090	14.0222	-102.6544	-102.4285	-106.5921	-72.6737	3
	0.100	-89.2032	16.7044	-102.6282	-102.3879	-106.5921	-79.0840	3
	0.500	-87.3661	21.9272	-102.6191	-102.2588	-106.5921	-82.0989	3
	1.000	-79.3765	19.8054	-102.5945	-102.1901	-105.7417	-75.1063	3
	5.000	-55.1786	8.9174	-102.5391	-102.4659	-106.5921	-54.2121	3
	10.000	-52.6952	15.3574	-102.5993	-102.5416	-109.3439	-50.2180	3
SAA	N=100	17.9771	352.1680	-	-	-137.9611	-137.9610	6
SAA	N=1000	-93.4177	73.2566	-	-	-105.5243	-105.5203	87
SAA	N=2000	-96.5163	61.0974	-	-	-102.2914	-102.2906	160

Table 26: SA vs SAA for EU-9

-		VERIFICATION				CONSTRUCTION		
ALG.	$step$	obj	dev	$\bar{lb}^N$	lb^N	$\overline{f}^N$	ub	$time$
N-SA (N=100)	0.005	-339.6061	63.0118	-996.3371	-995.8950	-1424.9808	-336.4921	1
	0.010	-339.8855	63.0119	-996.3197	-995.8738	-1424.9808	-336.6617	1
	0.050	-342.1271	63.0159	-996.1800	-995.7035	-1424.9808	-338.0205	0
	0.100	-344.9437	63.0286	-996.0070	-995.4879	-1424.9808	-339.7255	0
	0.500	-372.0744	63.6080	-994.6335	-993.7017	-1424.9808	-353.6077	1
	1.000	-407.8978	65.4583	-993.0530	-991.3779	-1424.9808	-371.5070	1
	5.000	-684.5668	134.4153	-989.9175	-984.1916	-1424.9808	-519.4333	0
	10.000	-825.7714	371.5833	-995.3352	-991.8104	-1424.9808	-653.3679	0
N-SA (N=1000)	0.005	-340.3479	63.0121	-1004.4143	-1003.1457	-1085.8590	-336.4929	2
	0.010	-341.3713	63.0133	-1004.4085	-1003.1391	-1085.8590	-337.0111	2
	0.050	-349.6371	63.0511	-1004.3618	-1003.0844	-1085.8590	-341.1842	2
	0.100	-360.1603	63.1723	-1004.3020	-1003.0151	-1085.8590	-346.4668	3
	0.500	-451.1080	67.6560	-1003.8092	-1002.3886	-1085.8590	-390.9568	3
	1.000	-569.4040	82.1386	-1003.3266	-1001.6071	-1085.8590	-449.5039	2
	5.000	-915.9993	255.1959	-1003.7318	-998.8554	-1085.8590	-738.1731	3
	10.000	-939.0872	363.5217	-1004.6796	-1001.2663	-1110.0806	-824.2031	3
N-SA (N=2000)	0.005	-340.7840	63.0125	-1004.6342	-1002.2271	-1042.7008	-338.2817	5
	0.010	-342.2456	63.0148	-1004.6273	-1002.2205	-1042.7008	-339.0132	4
	0.050	-354.0951	63.0906	-1004.5751	-1002.1788	-1042.7008	-344.9174	5
	0.100	-369.2778	63.3355	-1004.5067	-1002.1223	-1042.7008	-352.4237	5
	0.500	-500.0473	72.1099	-1003.9385	-1001.6221	-1042.7008	-416.1931	5
	1.000	-654.8810	97.2201	-1003.4420	-1001.0944	-1042.7008	-497.1990	6
	5.000	-941.6887	291.9360	-1003.9342	-997.7147	-1042.7008	-788.1257	5
	10.000	-953.9882	383.8223	-1008.5464	-997.7217	-1029.0530	-858.7257	5
E-SA (N=100)	0.005	-370.7854	63.5736	-994.7223	-993.9163	-1424.9808	-353.1477	0
	0.010	-403.5081	65.1983	-993.3187	-992.1737	-1424.9808	-369.9728	0
	0.050	-598.0518	90.7247	-988.3173	-985.8744	-1424.9808	-485.1822	1
	0.100	-666.1747	109.4112	-986.6931	-983.2019	-1424.9808	-548.2724	0
	0.500	-598.7146	118.6970	-993.1622	-985.3957	-1424.9808	-570.7319	0
	1.000	-537.1122	134.7827	-995.5442	-985.8843	-1424.9808	-522.8490	1
	5.000	-467.9027	75.2778	-996.1817	-993.8806	-1424.9808	-468.3696	0
	10.000	-468.5466	130.5814	-998.9136	-998.9137	-2618.0620	-411.5473	0
E-SA (N=1000)	0.005	-441.2167	66.8149	-1003.9286	-1002.5858	-1085.8590	-387.3251	2
	0.010	-543.9050	77.0978	-1003.6355	-1002.2361	-1085.8590	-438.6718	1
	0.050	-793.6082	127.2962	-1002.9834	-1001.3190	-1085.8590	-655.3626	2
	0.100	-838.3218	152.9272	-1002.7401	-1000.8066	-1085.8590	-726.6114	2
	0.500	-801.7337	188.5334	-1002.9137	-999.3462	-1085.8590	-754.2690	2
	1.000	-705.3121	152.7011	-1002.2903	-998.8188	-1085.8590	-686.3337	1
	5.000	-502.9805	74.0455	-1003.2255	-1001.8013	-1085.8590	-499.0478	1
	10.000	-480.5189	126.7385	-1001.5706	-1001.5706	-1202.6516	-462.7362	2
E-SA (N=2000)	0.005	-483.8734	70.1595	-1004.1355	-1001.8192	-1042.7008	-409.9674	3
	0.010	-621.0829	87.1127	-1003.9064	-1001.6274	-1042.7008	-481.7079	3
	0.050	-830.5069	140.0433	-1003.3258	-1001.0670	-1042.7008	-702.9633	3
	0.100	-868.6242	166.8363	-1003.0636	-1000.6626	-1042.7008	-767.2545	3
	0.500	-850.8188	219.2586	-1002.9677	-999.3737	-1042.7008	-798.0241	3
	1.000	-771.2904	198.3226	-1002.7299	-998.6815	-1034.2088	-728.4992	3
	5.000	-529.0234	89.3518	-1002.1681	-1001.4363	-1042.7008	-519.2999	4
	10.000	-503.8182	153.3248	-1002.7689	-1002.1950	-1070.2659	-479.2331	3
SAA	N=100	201.2264	3521.2462	-	-	-1356.8687	-1356.8677	7
SAA	N=1000	-910.8297	734.8461	-	-	-1032.3006	-1032.2738	91
SAA	N=2000	-941.9854	611.0414	-	-	-999.9114	-999.8982	161

Table 27: SA vs SAA for EU-10

-		VERIFICATION				CONSTRUCTION		
ALG.	*step*	*obj*	*dev*	$\tilde{lb}^N$	lb^N	$\overline{f}^N$	*ub*	*time*
N-SA (N=100)	0.005	-1676.8591	315.0591	-4960.2040	-4957.9885	-7103.4410	-1661.1747	0
	0.010	-1678.5487	315.0600	-4960.0990	-4957.8610	-7103.4410	-1662.1996	0
	0.050	-1692.1085	315.0894	-4959.2590	-4956.8288	-7103.4410	-1670.4180	1
	0.100	-1709.1651	315.1817	-4958.2084	-4955.5264	-7103.4410	-1680.7371	1
	0.500	-1875.6770	319.4277	-4949.9686	-4944.6357	-7103.4410	-1765.0413	0
	1.000	-2094.9050	333.1882	-4940.8992	-4930.5815	-7103.4410	-1874.2415	0
	5.000	-3650.9872	840.3725	-4931.2576	-4905.6257	-7103.4410	-2747.9567	0
	10.000	-3800.7619	1630.9513	-4977.4536	-4977.4725	-13023.9623	-2090.9446	1
N-SA (N=1000)	0.005	-1681.3447	315.0616	-5000.6024	-4994.2585	-5407.8319	-1661.5425	3
	0.010	-1687.5350	315.0701	-5000.5675	-4994.2185	-5407.8319	-1664.6766	3
	0.050	-1737.6309	315.3476	-5000.2840	-4993.8904	-5407.8319	-1689.9472	3
	0.100	-1801.6284	316.2411	-4999.9201	-4993.4581	-5407.8319	-1722.0148	2
	0.500	-2358.9495	349.5722	-4996.9862	-4989.6317	-5407.8319	-1993.8515	3
	1.000	-3055.9686	452.0546	-4994.5041	-4985.1447	-5407.8319	-2347.5158	3
	5.000	-4632.4882	1521.6509	-4998.8894	-4971.5507	-5407.8319	-3826.7989	2
	10.000	-4647.9744	1923.8632	-5001.4221	-4999.3838	-5761.7873	-4164.6643	3
N-SA (N=2000)	0.005	-1683.9821	315.0644	-5001.7014	-4989.6669	-5192.0409	-1670.7092	5
	0.010	-1692.8258	315.0813	-5001.6535	-4989.6352	-5192.0409	-1675.1336	5
	0.050	-1764.7102	315.6381	-5001.3410	-4989.3671	-5192.0409	-1710.9096	5
	0.100	-1857.2203	317.4494	-5000.9237	-4989.0187	-5192.0409	-1756.5379	5
	0.500	-2652.4518	382.0028	-4997.5860	-4986.0201	-5192.0409	-2145.5733	5
	1.000	-3512.5867	546.2880	-4995.1150	-4982.9658	-5192.0409	-2621.5074	6
	5.000	-4729.8534	1746.7144	-4999.2312	-4967.9144	-5192.0409	-4045.2517	5
	10.000	-4713.8299	1666.9798	-5015.0647	-5015.0645	-5282.8438	-4313.8041	5
E-SA (N=100)	0.005	-1832.4637	317.8680	-4952.1495	-4948.1207	-7103.4410	-1744.2755	0
	0.010	-1996.0775	325.9916	-4945.1242	-4939.4039	-7103.4410	-1828.4007	0
	0.050	-2968.7958	453.6234	-4920.1242	-4907.9057	-7103.4410	-2404.4480	1
	0.100	-3309.4105	547.0562	-4912.0031	-4894.5463	-7103.4410	-2719.8988	0
	0.500	-2972.1098	593.4851	-4944.3479	-4905.5160	-7103.4410	-2832.1963	0
	1.000	-2664.0980	673.9133	-4956.2551	-4907.9598	-7103.4410	-2592.7819	1
	5.000	-2318.0505	376.3888	-4959.4452	-4947.9391	-7103.4410	-2320.3848	0
	10.000	-2321.2721	652.8988	-4973.0991	-4973.1006	-13069.9200	-2037.3464	0
E-SA (N=1000)	0.005	-2184.6204	334.0744	-4998.1803	-4991.4658	-5407.8319	-1915.1624	2
	0.010	-2698.0620	385.4891	-4996.7147	-4989.7212	-5407.8319	-2171.8957	2
	0.050	-3946.5778	636.4811	-4993.4538	-4985.1330	-5407.8319	-3255.3498	1
	0.100	-4170.1457	764.6360	-4992.2371	-4982.5620	-5407.8319	-3611.5939	2
	0.500	-3987.2053	942.6669	-4993.1057	-4975.2666	-5407.8319	-3749.8819	2
	1.000	-3505.0974	763.5056	-4989.9914	-4972.6317	-5407.8319	-3410.2052	1
	5.000	-2493.4395	370.2274	-4994.6645	-4987.5433	-5407.8319	-2473.7761	2
	10.000	-2381.1315	633.6926	-4984.7356	-4984.7356	-6029.5468	-2289.8039	2
E-SA (N=2000)	0.005	-2397.9041	350.7973	-4999.2141	-4987.6338	-5192.0409	-2028.3740	4
	0.010	-3083.9515	435.5634	-4998.0716	-4986.6741	-5192.0409	-2387.0764	3
	0.050	-4131.0714	700.2164	-4995.1679	-4983.8716	-5192.0409	-3493.3532	3
	0.100	-4321.6580	834.1813	-4993.8571	-4981.8515	-5192.0409	-3814.8095	3
	0.500	-4232.6307	1096.2932	-4993.3757	-4975.4059	-5192.0409	-3968.6575	4
	1.000	-3834.9890	991.6129	-4992.1858	-4971.9463	-5149.5918	-3621.0437	3
	5.000	-2623.6539	446.7588	-4989.3751	-4985.7188	-5192.0409	-2575.0362	3
	10.000	-2497.6344	766.5842	-4992.3817	-4989.5130	-5329.9094	-2374.7451	3
SAA	N=100	1039.0135	17631.6927	-	-	-6763.1581	-6763.1456	7
SAA	N=1000	-4530.8206	3687.4747	-	-	-5140.1782	-5140.1360	89
SAA	N=2000	-4688.9239	3053.7409	-	-	-4978.2333	-4978.1967	161

REFERENCES

[1] Auslender, A. and Teboulle, M., "Interior gradient and proximal methods for convex and conic optimization," *SIAM Journal on Optimization*, vol. 16, pp. 697–725, 2006.

[2] Bauschke, H., Borwein, J., and Combettes, P., "Bregman monotone optimization algorithms," *SIAM Journal on Controal and Optimization*, vol. 42, pp. 596–636, 2003.

[3] Beck, A. and Teboulle, M., "Mirror-descent and nonlinear projected subgradient methods for convex optimization," *Operations Research Letters*, vol. 31, pp. 167–175, 2003.

[4] Becker, S., Bobin, J., and Candes, E., "Nesta: A fast and accurate first-order method for sparse recovery," manuscript, California Institute of Technology, 2009.

[5] Ben-Tal, A. and A.Nemirovski, "Non-euclidean restricted memory level method for large-scale convex optimization," *Mathematical Programming*, vol. 102, pp. 407–456, 2005.

[6] Ben-Tal, A. and Nemirovski, A., *Lectures on Modern Convex Optimization: Analysis, Algorithms, Engineering Applications*. MPS-SIAM Series on Optimization, Philadelphia: SIAM, 2000.

[7] Benveniste, A., Métivier, M., and Priouret, P., *Algorithmes adaptatifs et approximations stochastiques*. Masson, 1987. English translation: *Adaptive Algorithms and Stochastic Approximations*, Springer Verlag (1993).

[8] Bertsekas, D., *Constrained Optimization and Lagrange Multiplier Methods*. New York: Academic Press, first ed., 1982.

[9] Bertsekas, D., *Nonlinear Programming*. New York: Athena Scientific, second ed., 1999.

[10] Bregman, L., "The relaxation method of finding the common point convex sets and its application to the solution of problems in convex programming," *USSR Comput. Math. Phys.*, vol. 7, pp. 200–217, 1967.

[11] Burer, S. and Monteiro, R. D. C., "A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization," *Mathematical Programming, Series B*, vol. 95, pp. 329–357, 2003.

[12] Burer, S. and Monteiro, R. D. C., "Local minima and convergence in low-rank semidefinite programming," *Mathematical Programming*, vol. 103, pp. 427–444, 2005.

[13] Chung, K., "On a stochastic approximation method," *Annals of Mathematical Statistics*, pp. 463–483, 1954.

[14] Dantzig, G. B., *Linear Programming and Extensions*. Princeton, NJ: Princeton University Press, 1963.

[15] d'Aspremont, A., "Smooth optimization with approximate gradient," *SIAM Journal on Optimization*, vol. 19, pp. 1171–1183, 2008.

[16] d'Aspremont, A., Banerjee, O., and Ghaoue, L. E., "First-order methods for sparse covariance selection," *SIAM Journal on Matrix Analysis and its Applications*, vol. 30, pp. 56–66, 2008.

[17] Ermoliev, Y., "Stochastic quasigradient methods and their application to system optimization," *Stochastics*, vol. 9, pp. 1–36, 1983.

[18] Gaivoronski, A., "Nonstationary stochastic programming problems," *Kybernetika*, vol. 4, pp. 89–92, 1978.

[19] Goldberg, A. and Tarjan, R.

[20] Golshtein, E. and Tretyakov, N., *Modified Lagrangians and monotone maps in optimization*. New York, USA: Springer-Verlag, 1996.

[21] Hestenes, M. R., "Multiplier and gradient methods," *Journal of Optimization and Application*, vol. 4, pp. 303–320, 1969.

[22] Hiriart-Urruty, J.-B. and Lemaréchal, C., *Convex Analysis and Minimization algorithms I*, vol. 305 of *Comprehensive Study in Mathematics*. New York: Springer-Verlag, 1993.

[23] Jarre, F. and Rendl, F., "An augmented primal-dual method for linear conic programs," manuscript, Institut fur Mathematik, Universit at Dusseldorf, Germany, Austria, April 2007.

[24] Juditsky, A., Nazin, A., Tsybakov, A. B., and Vayatis, N., "Recursive aggregation of estimators via the mirror descent algorithm with average," *Problems of Information Transmission*, vol. 41, p. n.4, 2005.

[25] Juditsky, A., Nemirovski, A., and Tauvel, C., "Solving variational inequalities with stochastic mirror-prox algorithm," manuscript, Georgia Institute of Technology, Atlanta, GA, 2008. submitted to SIAM Journal on Control and Optimization.

[26] Juditsky, A., Rigollet, P., and Tsybakov., A. B., "Learning by mirror averaging," *Annals of Statistics*, vol. 36, pp. 2183–2206, 2008.

[27] Karmarkar, N. K., "A new polynomial–time algorithm for linear programming," *Combinatorica*, vol. 4, pp. 373–395, 1984.

[28] Khachian, L. G., "A polynomial algorithm in linear programming," *Doklady Akademiia Nauk SSSR*, vol. 224, pp. 1093–1096, 1979. English translation: *Soviet Mathematics Loklady* 20, 191-194.

[29] Kiwiel, K., "Proximal minimization methods with generalized bregman functions," *SIAM Journal on Controal and Optimization*, vol. 35, pp. 1142–1168, 1997.

[30] Kleywegt, A. J., Shapiro, A., and de Mello, T. H., "The sample average approximation method for stochastic discrete optimization," *SIAM Journal on Optimization*, vol. 12, pp. 479–502, 2001.

[31] Kushner, H. J. and Yin, G., *Stochastic Approximation and Recursive Algorithms and Applications*, vol. 35 of *Applications of Mathematics*. New York: Springer-Verlag, 2003.

[32] Lan, G., Lu, Z., and Monteiro, R. D. C., "Primal-dual first-order methods with $\mathcal{O}(1/\epsilon)$ iteration-complexity for cone programming," *Mathematical Programming*, 2009. to appear.

[33] Lan, G. and Monteiro, R. D. C., "Iteration-complexity of first-order penalty methods for convex programming," manuscript, School of Industrial Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA, June 2008.

[34] Lemarechal, C., Nemirovski, A., and Nesterov, Y., "New variants of bundle methods," *Mathematical Programming*, vol. 69, pp. 111–148, 1995.

[35] Lewis, A. and Wright, S., "A proximal method for composite minimization," manuscript, Cornell University, Ithaca, NY, 2009.

[36] Linderoth, J., Shapiro, A., and Wright, S., "The empirical behavior of sampling methods for stochastic programming," *Annals of Operations Research*, vol. 142, pp. 215–241, 2006.

[37] Lu, Z., "Smooth optimization approach for sparse covariance selection," *SIAM Journal on Optimization*, vol. 19, pp. 1807–1827, 2009.

[38] Lu, Z., Monteiro, R., and Yuan, M., "Convex optimization methods for dimension reduction and coefficient estimation in multivariate linear regression," manuscript, School of ISyE, Georgia Tech, Atlanta, GA, 30332, USA, January 2008.

[39] Lu, Z., Nemirovski, A., and Monteiro, R. D. C., "Large-scale semidefinite programming via saddle point mirror-prox algorithm," *Mathematical programming*, vol. 109, pp. 211–237, 2007.

[40] Mak, W. K., Morton, D., and Wood, R., "Monte carlo bounding techniques for determining solution quality in stochastic programs," *Operations Research Letters*, vol. 24, pp. 47–56, 1999.

[41] Monteiro, R. and Svaiter, B., "On the complexity of the hybrid proximal extragradient method for the iterates and the ergodic mean," manuscript, School of ISyE, Georgia Tech, Atlanta, GA, 30332, USA, March 2009.

[42] Nemirovski, A., "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, pp. 229–251, 2004.

[43] Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A., "Robust stochastic approximation approach to stochastic programming," *SIAM Journal on Optimization*, vol. 19, pp. 1574–1609, 2009.

[44] Nemirovski, A. and Yudin, D., *Problem complexity and method efficiency in optimization*. Wiley-Interscience Series in Discrete Mathematics, John Wiley, XV, 1983.

[45] Nemirovski, A. S. and Yudin, D. B., "Informational complexity and efficient methods for the solution of convex extremal problems (in Russian)," *Ékonomika i Mathematicheskie Metody*, vol. 12, pp. 357–369, 1976. English translation: *Maketon* 13(2), 3-25.

[46] Nemirovskii, A. and Yudin, D., "On cezari's convergence of the steepest descent method for approximating saddle point of convex-concave functions," *Doklady Akademiia Nauk SSSR*, vol. 239, p. No. 5, 1978. English translation: *Soviet Mathematics Loklady* 19, No. 2.

[47] Nesterov, Y. E., "A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$," *Doklady AN SSSR*, vol. 269, pp. 543–547, 1983. translated as Soviet Math. Docl.

[48] Nesterov, Y. E., "On an approach to the construction of optimal methods of minimization of smooth convex functions," *Ekonomo. i. Mat. Metody*, vol. 24, pp. 509–517, 1988.

[49] Nesterov, Y. E., *Introductory Lectures on Convex Optimization: a basic course*. Massachusetts: Kluwer Academic Publishers, 2004.

[50] Nesterov, Y. E., "Smooth minimization of nonsmooth functions," *Mathematical Programming*, vol. 103, pp. 127–152, 2005.

[51] Nesterov, Y. E., "Primal-dual subgradient methods for convex problems," *Mathematical Programming*, vol. 120, pp. 221–259, 2006.

[52] Nesterov, Y. E., "Gradient methods for minimizing composite objective functions," tech. rep., Center for Operations Research and Econometrics (CORE), Catholic University of Louvain, September 2007.

[53] Nesterov, Y. E., "Smoothing technique and its applications in semidefinite optimization," *Mathematical Programming*, vol. 110, pp. 245–259, 2007.

[54] Nesterov, Y. E. and Nemirovski, A. S., *Interior point Polynomial algorithms in Convex Programming: Theory and Applications*. Philadelphia: Society for Industrial and Applied Mathematics, 1994.

[55] Nocedal, J. and Wright, S. J., *Numerical optimization*. New York, USA: Springer-Verlag, 1999.

[56] Peña, J., "Nash equilibria computation via smoothing techniques," *Optima*, vol. 78, pp. 12–13, 2008.

[57] Pflug, G., "Optimization of stochastic models," in *The Interface Between Simulation and Optimization*, Boston: Kluwer, 1996.

[58] Polyak, B., "New stochastic approximation type procedures," *Automat. i Telemekh.*, vol. 7, pp. 98–107, 1990.

[59] Polyak, B. and Juditsky, A., "Acceleration of stochastic approximation by averaging," *SIAM J. Control and Optimization*, vol. 30, pp. 838–855, 1992.

[60] Powell, M., "An efficient method for nonlinear constraints in minimization problems," in *Optimization* (R. Fletcher, e., ed.), pp. 283–298, Academic Press, 1969.

[61] Robbins, H. and Monro, S., "A stochastic approximation method," *Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 1951.

[62] Rockafellar, R. and Uryasev, S., "Optimization of conditional value-at-risk," *The Journal of Risk*, vol. 2, pp. 21–41, 2000.

[63] Ruszczyński, *Nonlinear Optimization*. Princeton University Press, first ed., 2006.

[64] Ruszczyński, A. and Sysk, W., "A method of aggregate stochastic subgradients with on-line stepsize rules for convex stochastic programming problems," *Mathematical Programming Study*, vol. 28, pp. 113–131, 1986.

[65] Sacks, J., "Asymptotic distribution of stochastic approximation," *Annals of Mathematical Statistics*, vol. 29, pp. 373–409, 1958.

[66] Saigal, R., Vandenberghe, L., and Wolkowicz, H., *Handbook of Semidefinite Programming*. Boston-Dordrecht-London: Kluwer Academic Publishers, 2000.

[67] Sen, S., R.D. Doverspike, R., and Cosares, S., "Network planning with random demand," *Telecommunication Systems*, vol. 3, pp. 11–30, 1994.

[68] Shapiro, A., "Monte carlo sampling methods," in *Stochastic Programming* (Ruszczyński, A. and Shapiro, A., eds.), Amsterdam: North-Holland Publishing Company, 2003.

[69] Shapiro, A. and Nemirovski, A., "On complexity of stochastic programming problems," in *Continuous Optimization: Current Trends and Applications* (Jeyakumar, V. and Rubinov, A., eds.), pp. 111–144, Springer, 2005.

[70] Spall, J., *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. Hoboken, NJ: John Wiley, 2003.

[71] Strassen, V., "The existence of probability measures with given marginals," *Annals of Mathematical Statistics*, vol. 30, pp. 423–439, 1965.

[72] Teboulle, M., "Convergence of proximal-like algorithms," *SIAM Journal on Optimization*, vol. 7, pp. 1069–1083, 1997.

[73] Tseng, P., "On accelerated proximal gradient methods for convex-concave optimization," manuscript, University of Washington, Seattle, May 2008.

[74] Verweij, B., Ahmed, S., Kleywegt, A. J., Nemhauser, G., and Shapiro, A., "The sample average approximation method applied to stochastic routing problems: a computational study," *Computational Optimization and Applications*, vol. 24, pp. 289–333, 2003.

[75] V.I.Norkin, Pflug, G., and Ruszczyński, A., "A branch and bound method for stochastic global optimization," *Mathematical Programming*, vol. 83, pp. 425–450, 1998.

[76] Wang, W. and Ahmed, S., "Sample average approximation of expected value constrained stochastic programs," *Operations Research Letters*, vol. 36, pp. 515–519, 2008.

[77] Ye, Y., *Interior Point Algorithms: Theory and Analysis*. Hoboken, NJ: John Wiley, 1997.

[78] Zhao, X., Sun, D., and Toh, K., "A newton-cg augmented lagrangian method for semidefinite programming," manuscript, National University of Singapore, Singapore, March 2008.

VITA

Guanghui (George) Lan was born on August 9, 1976 in Pingjiang County, Hunan, China. He obtained his B.S. and M.S. degree in Mechanical Engineering from Xiangtan University and Shanghai Jiao Tong University in 1996 and 1999, respectively. He then worked as a software engineer in industry for about three years in China. In August 2002, he enrolled at the University of Louisville and earned a M.S. degree in Industrial Engineering in June 2004. In August 2004, he enrolled at Georgia Institute of Technology and completed a Ph.D. degree in the School of Industrial and Systems Engineering in early August 2009. He became an assistant professor in the Department of Industrial and Systems Engineering at the University of Florida since August 2009.