2012

Rate of Convergence Analysis of Discretization and Smoothing Algorithms for Semi-Infinite Minimax Problems

Royset, J.O.; Pee, E.Y.

J.O. Royset and E.Y. Pee, 2012, "Rate of Convergence Analysis of Discretization and Smoothing Algorithms for Semi-Infinite Minimax Problems," Journal of Optimization Theory and Applications, Vol. 155, No. 3, pp. 855-882.
http://hdl.handle.net/10945/38226

Rate of Convergence Analysis of Discretization and Smoothing Algorithms for Semi-Infinite Minimax Problems*

J. O. Royset† and E. Y. Pee‡

June 1, 2012

Abstract: Discretization algorithms for semi-infinite minimax problems replace the original problem, containing an infinite number of functions, by an approximation involving a finite number, and then solve the resulting approximate problem. The approximation gives rise to a discretization error, and suboptimal solution of the approximate problem gives rise to an optimization error. Accounting for both discretization and optimization errors, we determine the rate of convergence of discretization algorithms, as a computing budget tends to infinity. We find that the rate of convergence depends on the class of optimization algorithms used to solve the approximate problem as well as the policy for selecting discretization level and number of optimization iterations. We construct optimal policies that achieve the best possible rate of convergence and find that, under certain circumstances, the better rate is obtained by inexpensive gradient methods.

Key Words. Semi-infinite minimax problems, robust optimization, discretization algorithms, rate of convergence, exponential smoothing technique.

Mathematics Subject Classification. 90C34, 90C47

---

*The first author acknowledges support from AFOSR Young Investigator and Optimization & Discrete Math. Programs.

†Associate Professor, Operations Research Department, Naval Postgraduate School, Monterey, CA, USA. Corresponding author, email: joroyset@nps.edu

‡Principal Analyst, Operations Analysis and Simulation, Defence Science and Technology Agency, Singapore

1 Introduction

In many applications, such as investment portfolio allocation, engineering design, and policy optimization, decision makers need to determine a best course of action in the presence of uncertain parameters. One possibility for handling these situations is to formulate and solve "robust" optimization models, where a decision is selected that optimizes the performance under worst-case parameter values. We refer to [1–3] for an overview of recent developments. In this paper, we consider discretization algorithms for solving robust optimization models in the form of a *semi-infinite minimax problem* (SIP). SIP aims to minimize $\psi(\cdot)$ on $X \subset \mathbb{R}^d$, where, for every $x \in X$, $\psi(x)$ is the optimal value for the "inner problem" of maximizing $\phi(x, \cdot)$ on $Y \subset \mathbb{R}^m$. While discretization algorithms apply more broadly, we here assume that $\phi(\cdot, y)$ is convex for all $y \in Y$ and X is convex to enable a rate of convergence analysis. We refer to m as the *uncertainty dimension*.

There are numerous algorithms for solving SIP, such as exchange algorithms, local reduction algorithms, smoothing methods, bundle and (sub)gradient methods, and discretization algorithms; see, for example, [4–7, 9, 10], [8, Chapter 3], and [1, Chapter 2]. Discretization algorithms are an attractive class of algorithms due to their simplicity, sound theory, and the need for few assumptions. These algorithms construct an approximation of SIP by replacing Y by a subset of finite cardinality, and then (approximately) solving the resulting *finite minimax problem* using a suitable optimization algorithm, such as one based on nonlinear programming [11], smoothing [12], or an optimality function [8, pp. 242-244]. Since the maximization on Y is replaced by maximization over a set of finite cardinality, restrictive assumptions such as concavity of $\phi(x, \cdot)$ for all $x \in X$ and convexity of Y are avoided. Of course, if the uncertainty dimension is high, discretization may be impractical. Discretization algorithms are therefore mainly applied to problem instances with small uncertainty dimensions as often encountered in engineering design, where the uncertain parameter(s) may be time, frequency, and/or temperature; see, for example, [11] and references therein. Some discretization algorithms involve constructing and solving a sequence of finite minimax problems with increasing level of discretization (see, for instance, [8, Section 3.4]), but, in this paper, we focus on algorithms based on the solution of a single finite minimax problem.

It is well-known that, given a suitable discretization of Y and relatively mild assumptions, global and local minimizers, as well as stationary points of the finite minimax problem, converge to corresponding points of SIP, as the level of discretization grows to infinity; see, for example, [8, Chapter 3] and [13]. The rate of convergence of global minimizers is of order $O(\rho_N^{1/p})$, where ρ_N is the meshsize of a discretization of Y using N discretization points and p is a growth parameter [13]; see also [14]. The rate is improved under additional assumptions on the set of maximizers of $\phi(x, \cdot)$ on Y at an optimal solution x of SIP [13]. The importance of including boundary points of Y in the discretization and the resulting rate of convergence, as N tends to infinity, is discussed in [14]. While these results provide important insight, they do not consider the computational work required to solve the finite minimax problem.

The apparent simplicity of discretization algorithm hides a fundamental trade-off between the level of discretization of Y and the computational work required to approximately solve the resulting finite minimax problem. One would typically require a fine discretization of Y to guarantee that the finite minimax problem approximates SIP, in some sense, with high accuracy. However, in that case, the finite minimax problem becomes large scale (in the number of functions to maximize over) and the computational work to solve it may be high [11, 12]. A coarser discretization saves in the solution time of the correspondingly smaller finite minimax problem at the expense of a poorer approximation of SIP. It is often difficult, in practice, to construct discretizations of Y that balances this trade-off effectively.

In this paper, we examine the rate of convergence of a class of discretization algorithms as a *computing budget* tends to infinity. While one in practice needs to consider a finite computing budget, the paper provides fundamental insight about the trade-off between the level of discretization and the amount of optimization that may guide the implementation of algorithms. We show that the policy for selecting discretization level of Y relative to the size of the available computing budget influences the rate of convergence of discretization algorithms. We identify optimal discretization policies, in a precisely defined sense, for discretization algorithms based on finitely, superlinearly, linearly, and sublinearly convergent optimization algorithms for solving the resulting finite minimax problems, under the assumption that iterations from which these optimization algorithms attain their rates do not tend to infinite as the discretization is refined. We also construct an optimal discretization policy for the case when the finite minimax problem is solved by an exponential smoothing algorithm, where the level of smoothing must be determined too.

Other than [13, 14], there are few studies dealing with rate of convergence of discretization algorithms. For a class of adaptive discretization algorithms, where a sequence of finite minimax problems are solved with gradually higher and adaptively determined levels of discretization, [15, 16] show that suitable rules for selecting the levels of discretization lead to a rate of convergence, as the number of iterations tends to infinity, that is identical to the rate of convergence of the algorithm used to solve the finite minimax problems. Consequently, loosely speaking, the number of iterations required to achieve a certain tolerance when solving SIP is the same as that when solving a finite minimax problem obtained from SIP by discretization. The computational work *in each iteration*, however, may grow rapidly as successively finer discretization levels, and, consequently, larger finite minimax problems must be considered in the adaptive discretization algorithm. To our knowledge, there are no studies that attempt to quantify the rate of convergence of discretization algorithms for semi-infinite minimax problems in terms of a computing budget, accounting for both the number of iterations and the work in each iteration.

An alternative to discretization algorithms is an approach based on algorithm implementation. Here, an existing optimization algorithm, which, when applied to SIP may involve conceptual step such as finding a maximizer of $\phi(x, \cdot)$ on Y, is "implemented" by replacing the conceptual steps with approximations. The ϵ-subgradient method for SIP is an example of an algorithm implementation of the subgradient method under convexity-concavity assumptions. The implementation of (fast) gradient methods for problem instances, where function and gradient evaluations cannot be carried out exactly, is discussed in [10]. That study identifies the "best" gradient method for SIP under assumptions about the computational cost of reducing the evaluation error, the convexity in the first and concavity in second argument of $\phi(\cdot, \cdot)$, convexity of X and Y, and the use of specific gradient methods.

Rate of convergence analysis, in terms of a computing budget, is common in other areas such as Monte Carlo simulation and simulation optimization; see [17] for a review. In those areas, given a computing budget, the goal is to optimally allocate it across different task within the simulation, and to determine the resulting rate of convergence of an estimator as the computing budget tends to infinity. The allocation may be between exploration of new points and estimation of objective function values at known points, as in global optimization [18, 19] and stochastic programming [20, 21], between estimation of different random variables nested by conditioning [22], or between performance estimation of different systems, as in ranking and selection [23]. Even though these studies deal with rather different applications than semi-infinite minimax problems, they motivate the present paper. The paper is most closely related to the recent paper [21], where the authors consider the sample average approximation approach to solving stochastic programs. That approach replaces an expectation in the objective function of the stochastic program by a sample average and then proceeds by solving the sample average problem using an optimization algorithm. They consider sublinearly, linearly, and superlinearly convergent optimization algorithms for solving the sample average problem, determine optimal policies for allocating a computing budget between sampling and optimization, and quantify the associated rate of convergence of the sample average approximation approach as the computing budget tends to infinity. The present paper has the same goals, but in the context of semi-infinite minimax problems. Our treatment of sublinear, linear, and superlinear optimization algorithms for solving the finite minimax problems is similar to the parallel development in [21], but is carried out with different assumptions. The conclusions are naturally somewhat different. We also deal with exponential smoothing algorithms for solving the finite minimax problem, a topic not relevant in the case of stochastic programming.

Next section formally defines SIP, presents the corresponding finite minimax problem, and gives assumptions. Section 3 considers finite, superlinear, linear, and sublinear algorithms for solving the finite minimax problem and determines optimal discretization policies with corresponding rates of convergence, as the computing budget tends to infinity. Section 4 deals with the solution of the finite minimax problem by exponential smoothing algorithms, constructs an optimal discretization and smoothing policy, and determines the corresponding rate of convergence as the computing budget tends to infinity. Section 5 illustrates selected results numerically. The paper ends with concluding remarks in Section 6.

2 Problem Definitions and Assumptions

We start by defining the semi-infinite optimization problem under consideration. Let

$$(P) \qquad \min_{x \in X} \psi(x),$$

where $X \subset \mathbb{R}^d$ is convex, $\psi : \mathbb{R}^d \to \mathbb{R}$ is defined by

$$\psi(x) := \max_{y \in Y} \phi(x, y), \tag{1}$$

Y is a compact subset of $\mathbb{R}^m$, $\phi : \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}$, and $\phi(\cdot, y)$ is convex and as smooth as required by the applied algorithm for all $y \in Y$. Discretization algorithms apply also to nonconvex problems, but our rate analysis below relies on the rates of convergence of "standard" optimization algorithms, which typically require convexity.

Discretization algorithms for solving (P) replace Y by a finite subset $Y_N \subset Y$ of cardinality $N \in \mathbb{N} := \{1, 2, 3, ...\}$ and approximately solve the resulting *finite minimax problem*

$$(P_N) \qquad \min_{x \in X} \psi_N(x),$$

where $\psi_N : \mathbb{R}^d \to \mathbb{R}$ is defined by

$$\psi_N(x) := \max_{y \in Y_N} \phi(x, y). \tag{2}$$

Clearly, when $\phi(\cdot, y)$ is smooth for all $y \in Y_N$ and $X = \mathbb{R}^d$, or given by a finite number of continuously differentiable constraint functions, (P_N) is solvable by numerous nonlinear programming and finite minimax algorithms; see, for example, [11, 12].

The relationship between $\psi(\cdot)$ and $\psi_N(\cdot)$ depends on the properties of $\phi(\cdot, \cdot)$ and Y_N. We adopt the following assumption.

Assumption 2.1. *We assume that the following hold:*

(i) The set of optimal solutions X^* of (P) is nonempty.

(ii) There exists a constant $L \in [0, \infty[$ such that

$$|\phi(x, y) - \phi(x, y')| \leq L\|y - y'\|,$$

for all $x \in X$ and $y, y' \in Y$.

(iii) There exist constants $\overline{N} \in \mathbb{N}$ and $K \in [0, \infty[$ such that (a) the set of optimal solutions X_N^* of (P_N) is nonempty for all $N \geq \overline{N}$, $N \in \mathbb{N}$, and (b) for every $N \geq \overline{N}$, $N \in \mathbb{N}$, and $y \in Y$, there exists a $y' \in Y_N$ with $\|y - y'\| \leq K/N^{1/m}$. □

Part (b) of item (iii) holds, for example, when Y is the unit hypercube in m dimensions and the discretization scheme is uniform across Y, in which case $\overline{N} = 2^m$ and $K = m^{1/2}$; see [24]. The next result is a simple extension of Lemma 3.4.3 in [8], where we use the notation ψ^* and ψ_N^* to denote the optimal values of (P) and (P_N), respectively.

Proposition 2.1. *Suppose that Assumption 2.1 holds. Then,*

$$0 \leq \psi(x) - \psi_N(x) \leq LK/N^{1/m}, \tag{3}$$

for all $x \in X$, $N \in \mathbb{N}, N \geq \overline{N}$, where L, K, and $\overline{N}$ are as in Assumption 2.1.

Moreover,

$$0 \leq \psi^* - \psi_N^* \leq LK/N^{1/m},$$

for all $N \in \mathbb{N}, N \geq \overline{N}$. □

We refer to

$$\psi(x) - \psi_N(x)$$

as the *discretization error*. While the error, due to discretization, may be smaller in specific problem instances, the error bounds in Proposition 2.1 are the tightest possible without further assumptions.

Unless X and $\phi(\cdot, y)$, $y \in Y_N$ have special structures, one cannot expect to obtain a globally optimal solution of (P_N) in finite computing time. Hence, after a finite number of iterations of an optimization algorithm applied to (P_N), there is typically a remaining optimization error. Specifically, given an optimization algorithm $\mathcal{A}$ for (P_N), let $x_N^n \in X$ be the iterate[1] obtained by $\mathcal{A}$ after n iterations when applied to (P_N). Then the *optimization error* is defined as

$$\psi_N(x_N^n) - \psi_N^*.$$

[1]Iterates may depend on quantities, such as algorithm parameters and the initial point used. In this paper, we view the specification of such quantities as part of the algorithm and therefore do not reference them directly.

The rate with which the optimization error decays as n grows depends on the rate of convergence of $\mathcal{A}$ when applied to (P_N). Here and throughout the paper, we only consider algorithms that generate iterates in X exclusively, which is stated in the next assumption.

Assumption 2.2. *For all $N \in \mathbb{N}$, we assume that, if $\{x_N^n\}_{n=0}^\infty$ is generated by a given optimization algorithm when applied to (P_N), then $x_N^n \in X$ for all $n = 0, 1, 2, \ldots$.* □

Assumption 2.2 is satisfied by feasible direction methods. We define the *total error* as

$$\psi(x_N^n) - \psi^*,$$

which measures the quality of the obtained solution after n iteration of the given optimization algorithm applied to (P_N). In view of Assumptions 2.1 and 2.2 and Proposition 2.1,

$$\begin{aligned} 0 \leq \psi(x_N^n) - \psi^* &= \psi(x_N^n) - \psi_N(x_N^n) + \psi_N(x_N^n) - \psi_N^* - \psi^* + \psi_N^* \\ &\leq LK/N^{1/m} + \Delta_N^n(\mathcal{A}), \end{aligned} \tag{4}$$

where $\Delta_N^n(\mathcal{A})$ is an upper bound on the optimization error after n iterations of optimization algorithm $\mathcal{A}$ applied to (P_N). Below, we discuss several different expressions for $\Delta_N^n(\mathcal{A})$ under various assumptions about the optimization algorithm, and effectively also about (P_N). Since it appears difficult to quantify the rate of convergence of the total error, we focus on the rate of convergence of its upper bound in (4), as described next. The rate of convergence of that bound provides a guaranteed minimum rate of convergence of the total error.

We see from (4) that different choices of N and n may result in different bounds on the total error. Let $b \in \mathbb{N}$ be the computing budget available for executing n iterations of the selected optimization algorithm on (P_N). Clearly, the choice of N and n would typically depend on b, and we write N_b and n_b to emphasize this dependence. We refer to $\{(n_b, N_b)\}_{b=1}^\infty$, with $n_b, N_b \in \mathbb{N}$ for all $b \in \mathbb{N}$, as a *discretization policy.* A discretization policy specifies the level of discretization of Y and the number of iterations of the optimization algorithm to execute for any computing budget. If $n_b, N_b \to \infty$, as $b \to \infty$, then the bound on the discretization error vanishes; see Proposition 2.1. Assuming a convergent optimization algorithm to a global minimizer of (P_N), the optimization error and, presumably, the corresponding bound vanish too. For a given optimization algorithm $\mathcal{A}$ and $n, N \in \mathbb{N}$, we define the *total error bound*, denoted by $e(\mathcal{A}, N, n)$, as the right-hand side of (4), i.e.,

$$e(\mathcal{A}, n, N) := LK/N^{1/m} + \Delta_N^n(\mathcal{A}). \tag{5}$$

In this paper, we examine the rate at which the total error bound $e(\mathcal{A}, n_b, N_b)$ vanishes as b tends to infinity for different discretization policies $\{(n_b, N_b)\}_{b=1}^{\infty}$ and optimization algorithms $\mathcal{A}$. We identify optimal discretization policies, which, as precisely stated below, attain the highest possible rate of convergence of the total error bound as the computing budget tends to infinity for a given class of optimization algorithms.

Our analysis relies on the following assumption about the computational work needed by an optimization algorithm to carry out n iterations on (P_N).

Assumption 2.3. *There exist constants $M = M(\mathcal{A}, d) \in]0, \infty[$ and $\nu = \nu(\mathcal{A}) \in [1, \infty[$ such that the computational work required by a given optimization algorithm $\mathcal{A}$ to carry out $n \in \mathbb{N}$ iterations on (P_N) (of dimension d), $N \in \mathbb{N}$, is no larger than nMN^{ν}.* □

Assumption 2.3 holds with $\nu = 1$ if X is $\mathbb{R}^d$, or is a polyhedron, and the optimization algorithm $\mathcal{A}$ is a subgradient or smoothing algorithm (see [24]). In this case, each iteration of the optimization algorithm requires the calculation of $\psi_N(x)$ at the current iterate $x \in X$ (which involves finding the maximum over N scalars) and the evaluation of gradients $\nabla_x \phi(x, y)$ for one $y \in Y_N$ in the subgradient method, and for all $y \in Y_N$ in a smoothing algorithm. Other optimization algorithms for (P_N) tend to result in larger values of ν. For example, the sequential quadratic programming (SQP) algorithm in [11] and the Pshenichnyi-Pironneau-Polak (PPP) algorithm [8, Section 2.4] for solving finite minimax problems require the solution of one or two convex quadratic programs (QPs) with $d+1$ variables and N linear inequality constraints in each iteration. A QP solver based on an interior point method may need $O(d^2N)$ operations per iteration when $N \geq d$ [25]. The number of iterations required by an interior point method on such QPs could be of order $O(\sqrt{d+N})$ [26], or even less, in practice, when using a good method. Hence, ν may be 1.5. In Section 5, we find empirically that the PPP algorithm, with the active-set quadratic program solver LSSOL [27], follows Assumption 2.3 with $\nu = 2$. The constant M generally depends on d. However, as d is assumed to be fixed and the value of M is immaterial in our rate analysis, further consideration of that constant is unnecessary.

We note that computational savings have been observed empirically with the use of active-set strategies when solving (P_N), as well as any QP encountered in the process; see [11, 12, 25, 28]. While of practical importance, in this paper we ignore this possibility, as the effect of active-set strategies in worst-case rate analysis is unclear.

In view of Assumption 2.3, we refer to a discretization policy $\{(n_b, N_b)\}_{b=1}^\infty$ as *asymptotically admissible* if $n_b M N_b^\nu / b \to 1$, as $b \to \infty$. Clearly, an asymptotically admissible discretization policy satisfies the computing budget in the limit as b tends to infinity. We often specify an asymptotically admissible discretization policy in terms of $\{n_b\}_{b=1}^\infty$ only, as the rate of growth of $\{N_b\}_{b=1}^\infty$ then follows by the condition $n_b M N_b^\nu / b \to 1$, as $b \to \infty$. In the next two sections, we determine optimal asymptotically admissible discretization policies and corresponding rates of convergence of the total error bound under different assumptions about the optimization algorithm and, consequently, the optimization error bound $\Delta_{N_b}^{n_b}(\mathcal{A})$.

3 Finite, Superlinear, Linear, and Sublinear Algorithms

We see from (5) that the total error bound consists of discretization and optimization error bounds. The discretization error bound depends on the discretization level N, but not on the optimization algorithm used; see Proposition 2.1. The optimization error bound depends on the rate of convergence of the optimization algorithm used to solve (P_N). In this section, we consider four cases: First, we assume that the optimization algorithm solves (P_N) in a finite number of iterations. Second, we consider optimization algorithms with a superlinear rate of convergence towards an optimal solution of (P_N). Third, we deal with linearly convergent optimization algorithms. Fourth, we assume a sublinearly convergent algorithm.

3.1 Finite Optimization Algorithm

Suppose that the optimization algorithm for solving (P_N) is guaranteed to obtain an optimal solution in a finite number of iterations independently of N, as defined precisely next.

Definition 3.1. *An optimization algorithm $\mathcal{A}$ converges finitely on $\{(P_N)\}_{N=\overline{N}}^\infty$ when X_N^* is nonempty for $N \geq \overline{N}$ and there exist a constant $\overline{n} \in \mathbb{N}$ such that, for all $N \geq \overline{N}$, $N \in \mathbb{N}$, a sequence $\{x_N^n\}_{n=0}^\infty$ generated by $\mathcal{A}$ when applied to (P_N) satisfies $x_N^n \in X_N^*$ for all $n \geq \overline{n}$.* $\square$

No optimization algorithm converges finitely on $\{(P_N)\}_{N=\overline{N}}^{\infty}$ without strong structural assumptions on X, $\phi(\cdot,\cdot)$, and Y such as linearity. In this paper, we are not interested in instance of (P_N) in the form of linear programs, for which finite convergence may be possible, but include this case here as an "ideal" case. As we see below, the case provides an upper bound on the rate of convergence of the total error bound using any optimization algorithm. In view of Definition 3.1, a finitely convergent optimization algorithm $\mathcal{A}^{\text{finite}}$ on $\{(P_N)\}_{N=\overline{N}}^{\infty}$ has no optimization error after a sufficiently large number of iterations. Hence, we define $\Delta_N^n(\mathcal{A}^{\text{finite}}) := 0$ and $e(\mathcal{A}^{\text{finite}}, n, N) := LK/N^{1/m}$ for $n \geq \overline{n}$ and $N \geq \overline{N}$, where L and K are as in Assumption 2.1, and $\overline{n}$ and $\overline{N}$ are as in Definition 3.1. Naturally, one can in this case let the portion of the computing budget allocated to discretization tends to 1, as $b \to \infty$. The next theorem states the rate of convergence of the total error bound in this case.

Theorem 3.1. *Suppose that Assumption 2.1 holds and that $\mathcal{A}^{\text{finite}}$ is a finitely convergent algorithm on $\{(P_N)\}_{N=\overline{N}}^{\infty}$, with $\overline{N}$ as in Assumption 2.1 and number of required iterations $\overline{n}$ as in Definition 3.1. Suppose also that $\mathcal{A}^{\text{finite}}$ satisfies Assumptions 2.2 and 2.3. If $\{(n_b, N_b)\}_{b=1}^{\infty}$ is an asymptotically admissible discretization policy with $n_b = \overline{n}$ for all $b \in \mathbb{N}$, then*

$$\lim_{b\to\infty} \frac{\log e(\mathcal{A}^{\text{finite}}, n_b, N_b)}{\log b} = -\frac{1}{m\nu},$$

where ν is as in Assumption 2.3 and m is the uncertainty dimension.

Proof. Since $\{(n_b, N_b)\}_{b=1}^{\infty}$ is asymptotically admissible, $n_b M N_b^{\nu}/b = \overline{n} M N_b^{\nu}/b \to 1$, as $b \to \infty$, and we have that $N_b \to \infty$, as $b \to \infty$. Here, M is as in Assumption 2.3. Hence, for sufficiently large b, $\Delta_{N_b}^{n_b}(\mathcal{A}^{\text{finite}}) = 0$ and $e(\mathcal{A}^{\text{finite}}, n_b, N_b) = LK/N_b^{1/m}$, where L and K are as in Assumption 2.1. Consequently, for sufficiently large b,

$$\begin{aligned}
\log e(\mathcal{A}^{\text{finite}}, n_b, N_b) &= \log \frac{LK}{\left(\frac{b}{\overline{n}M}\right)^{1/\nu m} \left(\frac{\overline{n}MN_b^{\nu}}{b}\right)^{1/\nu m}} \\
&= \log LK - \frac{1}{\nu m}\log b + \frac{1}{\nu m}\log \overline{n}M - \frac{1}{\nu m}\log\frac{\overline{n}MN_b^{\nu}}{b}.
\end{aligned}$$

Since $\overline{n}MN_b^{\nu}/b \to 1$ as $b \to \infty$, the conclusion follows after dividing by $\log b$ and taking limits. □

Theorem 3.1 gives the asymptotic rate of decay of $e(\mathcal{A}^{\text{finite}}, n_b, N_b)$ on a logarithmic scale as b tends to infinity. We say in this case that the discretization algorithm and its total error bound $e(\mathcal{A}^{\text{finite}}, n_b, N_b)$ converge at rate $b^{-1/(m\nu)}$. Similar statements below are referenced likewise.

For any discretization policy satisfying $n_b M N_b^\nu \leq b$ for all $b \in \mathbb{N}$ and $M \geq 1$, $N_b \leq b^{1/\nu}$ for all $b \in \mathbb{N}$. Hence, in view of Proposition 2.1, the optimal value of (P_N) and the discretization error converge at rate $N_b^{-1/m} \geq b^{-1/(m\nu)}$, as $b \to \infty$. Hence, the discretization error cannot converge at a faster rate than that stipulated in Theorem 3.1. Since the total error bound includes the discretization error bound (see (5)), the total error bound cannot converge faster than the rate $b^{-1/(m\nu)}$ regardless of the optimization algorithm used to solve (P_N). The asymptotically admissible discretization policy stated in Theorem 3.1 is problematic to implement as $\overline{n}$ may be unknown. Still, the resulting rate is an upper bound on the rate that can be obtained by any optimization algorithm, and therefore provides a benchmark for comparison.

3.2 Superlinear Optimization Algorithm

We next consider superlinearly convergent optimization algorithms as defined as follows.

Definition 3.2. *An optimization algorithm $\mathcal{A}$ converges superlinearly with order $\gamma \in]1, \infty[$ on $\{(P_N)\}_{N=\overline{N}}^\infty$ when X_N^* is nonempty for $N \geq \overline{N}$ and there exist constants $\overline{n} \in \mathbb{N}$, $\overline{c} \in [0, \infty[$, and $\rho \in [0, 1[$ such that $\overline{c}^{1/(\gamma-1)}(\psi_N(x_N^n) - \psi_N^*) \leq \rho$ and*

$$\frac{\psi_N(x_N^{n+1}) - \psi_N^*}{(\psi_N(x_N^n) - \psi_N^*)^\gamma} \leq \overline{c} \tag{6}$$

for all $n \geq \overline{n}$, $n \in \mathbb{N}$, and $N \geq \overline{N}$, $N \in \mathbb{N}$. □

Definition 3.2 requires the optimization algorithm to attain a superlinear rate of convergence for sufficiently large n, which is typically the case for Newtonian methods applied to strongly convex instance of (P_N) with twice Lipschitz continuously differentiable functions and $X = \mathbb{R}^d$. For example, the Polak-Mayne-Higgins Algorithm (see Algorithm 2.5.10 of [8]) attains a superlinear rate of order $\gamma = 3/2$. The SQP algorithm of [11] also achieves a superlinear rate of convergence, but its order appears unknown. Definition 3.2 requires that the superlinear regime starts no later than an iteration number independent of N. Assuming that the algorithm is initiated at a point independent of N, this is obtained in the Polak-Mayne-Higgins Algorithm if the Lipschitz constant of $\nabla^2_{xx}\phi(\cdot, \cdot)$ with respect to its first argument is bounded on $X \times Y$, and the eigenvalues of $\nabla^2_{xx}\phi(x,y)$ for all $x \in X$ and $y \in Y$ are positive, bounded from above, and away from zero.

The next lemma identifies a total error bound for a superlinearly convergent algorithm.

Lemma 3.1. *Suppose that Assumption 2.1 holds and that $\mathcal{A}^{\text{super}}$ is a superlinearly convergent algorithm with order $\gamma \in]1,\infty[$ on $\{(P_N)\}_{N=\overline{N}}^{\infty}$, with $\overline{N}$ as in Assumption 2.1. Let $\{x_N^n\}_{n=0}^{\infty}$ be the iterates generated by $\mathcal{A}^{\text{super}}$ when applied to (P_N), $N \in \mathbb{N}$, $N \geq \overline{N}$. Suppose also that $\mathcal{A}^{\text{super}}$ satisfies Assumption 2.2. Then, there exist constants $c \in [0,1[$, $\kappa \in [0,\infty[$, and $\overline{n} \in \mathbb{N}$ such that*

$$\psi(x_N^n) - \psi^* \leq c^{\gamma^n}\kappa + LK/N^{1/m}$$

for all $n \geq \overline{n}$, $n \in \mathbb{N}$, and $N \geq \overline{N}$, $N \in \mathbb{N}$, where L and K are as in Assumption 2.1 and m is the uncertainty dimension.

Proof. Based on Proposition 2.1, Definition 3.2, and recursive application of (6), there exists an $\overline{n} \in \mathbb{N}$ such that

$$\begin{aligned}
&\psi(x_N^n) - \psi^* \\
\leq\ & \psi_N(x_N^n) + LK/N^{1/m} - \psi_N^* \\
\leq\ & \overline{c}^{-1/(\gamma-1)}(\overline{c}^{1/(\gamma-1)}(\psi_N(x_N^{\overline{n}}) - \psi_N^*))^{\gamma^{n-\overline{n}}} + LK/N^{1/m} \\
=\ & \overline{c}^{-1/(\gamma-1)}(\overline{c}^{1/(\gamma-1)}(\psi_N(x_N^{\overline{n}}) - \psi_N^*))^{\gamma^{-\overline{n}}}(\overline{c}^{1/(\gamma-1)}(\psi_N(x_N^{\overline{n}}) - \psi_N^*))^{\gamma^{n}} + LK/N^{1/m} \\
\leq\ & \overline{c}^{-1/(\gamma-1)}\rho^{\gamma^{-\overline{n}}}\rho^{\gamma^n} + LK/N^{1/m}
\end{aligned}$$

for $N \geq \overline{N}$, $N \in \mathbb{N}$, and $n \geq \overline{n}$, $n \in \mathbb{N}$, with ρ as in Definition 3.2. Consequently, the conclusion holds with $c = \rho$ and $\kappa = \overline{c}^{-1/(\gamma-1)}\rho^{\gamma^{-\overline{n}}}$. □

In view of Lemma 3.1, we adopt the upper bound on the optimization error

$$\Delta_N^n(\mathcal{A}^{\text{super}}) := c^{\gamma^n}\kappa$$

for a superlinearly convergent optimization algorithm $\mathcal{A}^{\text{super}}$ on $\{(P_N)\}_{N=\overline{N}}^{\infty}$, where c and κ are as in Lemma 3.1. Consequently, for $n, N \in \mathbb{N}$, we define the total error bound

$$e(\mathcal{A}^{\text{super}}, n, N) := c^{\gamma^n}\kappa + KL/N^{1/m}.$$

The next result states that, if we choose a particular discretization policy, then a superlinearly convergent optimization algorithm results in the same rate of convergence of the total error bound as a finitely convergent algorithm. Hence, the policy stipulated next is optimal in the sense that no other policy guarantees a better rate of convergence.

Theorem 3.2. *Suppose that $\mathcal{A}^{\text{super}}$ satisfies the assumptions of Lemma 3.1 and, in addition, Assumption 2.3 holds. If $\{(n_b, N_b)\}_{b=1}^{\infty}$ is an asymptotically admissible discretization policy with $n_b/\log\log b \to a \in]1/\log\gamma, \infty[$, then*

$$\lim_{b\to\infty} \frac{\log e(\mathcal{A}^{\text{super}}, n_b, N_b)}{\log b} = -\frac{1}{m\nu},$$

where ν is as defined in Assumption 2.3 and m is the uncertainty dimension.

Proof. Straightforward algebraic manipulation gives that

$$\begin{aligned}
&\frac{KL}{N^{1/m}} \\
&= \exp\left(\log KL - \frac{1}{m\nu}\log\left(\frac{nMN^{\nu}}{b}\right) - \frac{1}{m\nu}\log\left(\frac{b}{\log\log b}\right) - \frac{1}{m\nu}\log\left(\frac{\log\log b}{nM}\right)\right) \\
&= \exp\left(\log KL - \frac{1}{m\nu}\log\left(\frac{nMN^{\nu}}{b}\right) - \frac{1}{m\nu}\log b + \frac{1}{m\nu}\log\log\log b - \frac{1}{m\nu}\log\left(\frac{\log\log b}{nM}\right)\right),
\end{aligned}$$

where M is as in Assumption 2.3, and

$$\begin{aligned}
\kappa c^{\gamma^n} &= \exp\left(\log\kappa + \gamma^{\frac{n}{\log\log b}\log\log b}\log c\right) \\
&= \exp\left(\log\kappa + \log c(\log b)^{\frac{n}{\log\log b}\log\gamma}\right).
\end{aligned}$$

Hence,

$$\begin{aligned}
&e(\mathcal{A}^{\text{super}}, n, N) \\
&= \exp\left(\frac{-1}{m\nu}(\log b - \log\log b)\right)\Bigg[\exp\left(\log KL - \frac{1}{m\nu}\log\left(\frac{nMN^{\nu}}{b}\right) - \frac{1}{m\nu}\log\left(\frac{\log\log b}{nM}\right)\right) \\
&\quad + \exp\left(\log\kappa + \log b\left(\frac{1}{m\nu} + \log c(\log b)^{\frac{n}{\log\log b}\log\gamma - 1}\right) - \frac{1}{m\nu}\log\log\log b\right)\Bigg].
\end{aligned}$$

Consequently,

$$\begin{aligned}
&\frac{\log e(\mathcal{A}^{\text{super}}, n, N)}{\log b} = -\frac{1}{m\nu} + \frac{1}{m\nu}\frac{\log\log b}{\log b} \qquad (7) \\
&+ \log\Bigg[\exp\left(\log KL - \frac{1}{m\nu}\log\left(\frac{nMN^{\nu}}{b}\right) - \frac{1}{m\nu}\log\left(\frac{\log\log b}{nM}\right)\right) \\
&\quad + \exp\left(\log\kappa + \log b\left(\frac{1}{m\nu} + \log c(\log b)^{\frac{n}{\log\log b}\log\gamma - 1}\right) - \frac{1}{m\nu}\log\log\log b\right)\Bigg]/\log b.
\end{aligned}$$

Since $n_b M N_b^\nu / b \to 1$, $\log\log b / n_b \to 1/a$, and, due to the facts that $a \log\gamma - 1 > 0$ and $\log c < 0$, $\log c (\log b)^{\frac{n_b}{\log\log b}\log\gamma - 1} \to -\infty$, as $b \to \infty$, we obtain that the expression in brackets in (7), with n and N replaced by n_b and N_b, respectively, tends to a constant as $b \to \infty$. The conclusion then follows from taking limits of the other terms as well. $\square$

Clearly, from Theorem 3.2 and its proof, other choices of discretization policy than the one recommended may result in significant slower rate of convergence of the total error bound, as the computing budget tends to infinity. We observe that the recommended policy is actually a family of policies as there are numerous choices that satisfy the required conditions.

3.3 Linear Optimization Algorithm

We next consider a linearly convergent optimization algorithm defined as follows.

Definition 3.3. *An optimization algorithm $\mathcal{A}$ converges linearly on $\{(P_N)\}_{N=\overline{N}}^\infty$ when X_N^* is nonempty for $N \geq \overline{N}$, and there exist constants $\overline{n} \in \mathbb{N}$ and $\overline{c} \in [0, 1[$ such that*

$$\frac{\psi_N(x_N^{n+1}) - \psi_N^*}{\psi_N(x_N^n) - \psi_N^*} \leq \overline{c}$$

for all $n \geq \overline{n}$, $n \in \mathbb{N}$, and $N \geq \overline{N}$, $N \in \mathbb{N}$. $\square$

The definition requires that the rate of convergence coefficient $\overline{c}$ holds for all N sufficiently large. This is satisfied, for example, in the PPP algorithm, with $\overline{n} = 1$, when the eigenvalues of $\nabla^2_{xx}\phi(x, y)$ for all $x \in X$ and $y \in Y$ are positive, bounded from above, and away from zero, and $X = \mathbb{R}^d$ or is a polyhedron [8, Section 2.4].

Lemma 3.2. *Suppose that Assumption 2.1 holds and that $\mathcal{A}^{\text{linear}}$ is a linearly convergent algorithm on $\{(P_N)\}_{N=\overline{N}}^\infty$, with $\overline{N}$ as in Assumption 2.1. Let $\{x_N^n\}_{n=0}^\infty$ be the iterates generated by $\mathcal{A}^{\text{linear}}$ when applied to (P_N), $N \in \mathbb{N}$, $N \geq \overline{N}$. Suppose also that there exists a constant $C \in \mathbb{R}$ such that $\psi_N(x_N^n) \leq C$ for all $n \in \mathbb{N}$ and $N \geq \overline{N}$, $N \in \mathbb{N}$, and that $\mathcal{A}^{\text{linear}}$ satisfies Assumption 2.2. Then, there exists a constant $\kappa \in [0, \infty[$ such that*

$$\psi(x_N^n) - \psi^* \leq \overline{c}^n \kappa + LK/N^{1/m}$$

for all $n \geq \overline{n}$ and $N \geq \overline{N}$, where $\overline{c}$ and $\overline{n}$ are as in Definition 3.3, and K and L are as in Assumption 2.1.

Proof. Based on Proposition 2.1 and the fact that $\mathcal{A}^{\text{linear}}$ is linearly convergent, we obtain that

$$\begin{aligned}\psi(x_N^n)-\psi^* &\leq \psi_N(x_N^n)+LK/N^{1/m}-\psi_N^* \\ &\leq \overline{c}^{n-\overline{n}}[\psi_N(x_N^{\overline{n}})-\psi_N^*]+LK/N^{1/m} \\ &\leq \overline{c}^{n}(\overline{c}^{-\overline{n}}(C-\psi^*+LK/\overline{N}^{1/m}))+LK/N^{1/m}.\end{aligned}$$

Hence, the results hold with $\kappa=\overline{c}^{-\overline{n}}(C-\psi^*+LK/\overline{N}^{1/m})$. □

We note that the assumption $\psi_N(x_N^n)\leq C$ for all $n\in\mathbb{N}$ and $N\in\mathbb{N}$, $N\geq\overline{N}$, in Lemma 3.2 is rather weak and is satisfied, for example, if the optimization algorithm starts with $x^0\in X$ regardless of N and is a descent algorithm because then $\psi_N(x_N^n)\leq\psi_N(x^0)\leq\psi(x^0)$. In view of Lemma 3.2, we define the optimization error bound for a linearly convergence optimization algorithm $\mathcal{A}^{\text{linear}}$ to be

$$\Delta_N^n(\mathcal{A}^{\text{linear}}):=\overline{c}^n\kappa,$$

where $\overline{c}$ and κ are as in Lemma 3.2, and the total error bound of algorithm $\mathcal{A}^{\text{linear}}$ for $n,N\in\mathbb{N}$ to be

$$e(\mathcal{A}^{\text{linear}},n,N):=\overline{c}^n\kappa+LK/N^{1/m}.$$

The next result states that a linearly convergent optimization algorithm also attains the best possible rate of convergence of the total error bound given in Theorems 3.1 and 3.2 under a suitable choice of $\{(n_b,N_b)\}_{b=1}^{\infty}$.

Theorem 3.3. *Suppose that $\mathcal{A}^{\text{linear}}$ satisfies the assumptions of Lemma 3.2 and, in addition, Assumption 2.3 holds. If $\{(n_b,N_b)\}_{b=1}^{\infty}$ is an asymptotically admissible discretization policy with $n_b/\log b\to a\in]-1/(m\nu\log\overline{c}),\infty[$, where $\overline{c}$ and ν are as in Definition 3.3 and Assumption 2.3, respectively, then*

$$\lim_{b\to\infty}\frac{\log e(\mathcal{A}^{\text{linear}},n_b,N_b)}{\log b}=-\frac{1}{m\nu}.$$

Proof. Algebraic manipulations give that

$$\frac{KL}{N^{1/m}} = \exp\left(\log KL-\frac{1}{m\nu}\log\left(\frac{nMN^\nu}{b}\right)-\frac{1}{m\nu}\log\left(\frac{\log b}{nM}\right)-\frac{1}{m\nu}\log b+\frac{1}{m\nu}\log\log b\right),$$

where M is as in Assumption 2.3, and

$$\overline{c}^n\kappa=\exp\left(\log\kappa+n\log\overline{c}\right)=\exp\left(\log\kappa+\log b(n/\log b)\log\overline{c}\right).$$

Hence,

$$
\begin{aligned}
& e(\mathcal{A}^{\text{linear}}, n, N) \\
= \; & \exp\left(\frac{-1}{m\nu}(\log b - \log\log b)\right)\Bigg[\exp\Bigg(\log\kappa + \left(\frac{n}{\log b}\log\bar{c} + \frac{1}{m\nu}\right)\log b \\
& \quad - \frac{1}{m\nu}\log\log b\Bigg) + \exp\left(\log KL - \frac{1}{m\nu}\log\left(\frac{nMN^{\nu}}{b}\right) - \frac{1}{m\nu}\log\left(\frac{\log b}{nM}\right)\right)\Bigg].
\end{aligned} \tag{8}
$$

Since $a > -1/(m\nu\log\bar{c})$, $n_b\log\bar{c}/\log b + 1/(m\nu) \to a\log\bar{c} + 1/(m\nu) < 0$, as $b \to \infty$. Consequently, the expression in the brackets in (8), with n and N replaced by n_b and N_b, respectively, tends to $\exp(\log KL - (1/(m\nu))\log(1/(aM)))$, as $b \to \infty$. The conclusion then follows from (8) after taking logarithms, dividing by $\log b$, and taking limits. □

3.4 Sublinear Optimization Algorithm

We next consider the situation when the optimization algorithm for solving (P_N) is sublinearly convergent, as given in the following definition.

Definition 3.4. *An optimization algorithm $\mathcal{A}$ converges sublinearly with degree $\gamma \in]0,\infty[$ on $\{(P_N)\}_{N=\overline{N}}^{\infty}$ when X_N^* is nonempty for $N \geq \overline{N}$, and there exists a constant $C \in [0,\infty[$ such that*

$$\psi_N(x_N^n) - \psi_N^* \leq C/n^{\gamma}$$

for all $n \in \mathbb{N}$ and $N \geq \overline{N}$, $N \in \mathbb{N}$. □

The subgradient method is sublinearly convergent in the sense of Definition 3.4 with $\gamma = 1/2$ and $C = D_X L_\phi$ when (P_N) is convex and X is a polyhedron, where D_X is the diameter of X, and L_ϕ is a Lipschitz constant of $\phi(\cdot, y)$ on X independent of $y \in Y$; see [29, pp. 142-143]. Consequently, we define the optimization error bound for a sublinearly convergence optimization algorithm $\mathcal{A}^{\text{sublin}}$ to be

$$\Delta_N^n(\mathcal{A}^{\text{sublin}}) := C/n^{\gamma},$$

and the total error bound for $n, N \in \mathbb{N}$ to be

$$e(\mathcal{A}^{\text{sublin}}, n, N) := C/n^{\gamma} + LK/N^{1/m}.$$

The next result gives an optimal discretization policy for a sublinearly convergent optimization algorithm and also shows the corresponding rate of convergence of the total error bound.

Theorem 3.4. *Suppose that Assumption 2.1 holds and that $\mathcal{A}^{\text{sublin}}$ is a sublinearly convergent algorithm with degree $\gamma \in]0, \infty[$ on $\{(P_N)\}_{N=\overline{N}}^{\infty}$, with $\overline{N}$ as in Assumption 2.1. Suppose also that $\mathcal{A}^{\text{sublin}}$ satisfies Assumptions 2.2 and 2.3, and that $\{(n_b, N_b)\}_{b=1}^{\infty}$ is an asymptotically admissible discretization policy. Then,*

$$\liminf_{b\to\infty} \frac{\log e(\mathcal{A}^{\text{sublin}}, n_b, N_b)}{\log b} \geq -\frac{1}{m\nu + 1/\gamma},$$

where ν is as in Assumption 2.3 and m is the uncertainty dimension.

Moreover, if $n_b/b^{1/(m\nu\gamma+1)} \to a \in]0, \infty[$, as $b \to \infty$, then

$$\lim_{b\to\infty} \frac{\log e(\mathcal{A}^{\text{sublin}}, n_b, N_b)}{\log b} = -\frac{1}{m\nu + 1/\gamma}.$$

Proof. For any $n, N \in \mathbb{N}$,

$$\begin{aligned}
\log e(\mathcal{A}^{\text{sublin}}, n, N) &= \log(C/n^\gamma + KL/N^{1/m}) \\
&\geq \log(\max\{C/n^\gamma, KL/N^{1/m}\}) \\
&= \max\{\log C - \gamma \log n, \log KL - (1/m)\log N\}.
\end{aligned}$$

Let $\{(n_b, N_b)\}_{b=1}^{\infty}$ be an arbitrary asymptotically admissible discretization policy. If $n_b \geq b^{1/(m\nu\gamma+1)}$, then

$$\begin{aligned}
\frac{\log e(\mathcal{A}^{\text{sublin}}, n_b, N_b)}{\log b} &\geq \frac{\log KL - \frac{1}{m}\log N_b}{\log b} \\
&= \frac{\log KL - \frac{1}{m}\log\left(\frac{N_b^\nu n_b}{b}\frac{b}{n_b}\right)^{1/\nu}}{\log b} \\
&\geq \frac{\log KL - \frac{1}{m\nu}\log\left(\frac{N_b^\nu n_b}{b} b^{m\nu\gamma/(m\nu\gamma+1)}\right)}{\log b} \\
&= \frac{\log KL - \frac{1}{m\nu}\log\left(\frac{N_b^\nu n_b}{b}\right) - \frac{1}{m\nu}\log b^{m\nu\gamma/(m\nu\gamma+1)}}{\log b} \\
&= \frac{\log KL - \frac{1}{m\nu}\log\left(\frac{N_b^\nu n_b}{b}\right)}{\log b} - \frac{1}{m\nu + 1/\gamma}.
\end{aligned}$$

If $n_b < b^{1/(m\nu\gamma+1)}$, then

$$\begin{aligned}
\frac{\log e(\mathcal{A}^{\text{sublin}}, n_b, N_b)}{\log b} &\geq \frac{\log C - \gamma \log n_b}{\log b} \\
&> \frac{\log C - \gamma \log b^{1/(m\nu\gamma+1)}}{\log b} \\
&= \frac{\log C}{\log b} - \frac{1}{m\nu + 1/\gamma}.
\end{aligned}$$

Hence, for any $b \in \mathbb{N}$,

$$\frac{\log e(\mathcal{A}^{\text{sublin}}, n_b, N_b)}{\log b} \geq \min \left\{ \frac{\log KL - \frac{1}{m\nu} \log \left(\frac{N_b^\nu n_b}{b} \right)}{\log b}, \frac{\log C}{\log b} \right\} - \frac{1}{m\nu + 1/\gamma}.$$

The first result then follows by taking limits as $b \to \infty$, utilizing the fact that $N_b^\nu n_b/b \to 1/M$, as $b \to \infty$, where M is as in Assumption 2.3.

Next, let $\{(n_b, N_b)\}_{b=1}^\infty$ be an asymptotically admissible discretization policy satisfying $n_b/b^{1/(m\nu\gamma+1)} \to a \in (0, \infty)$, as $b \to \infty$. Then, by algebraic manipulation,

$$\begin{aligned}
e(\mathcal{A}^{\text{sublin}}, n_b, N_b) &= \frac{C}{n_b^\gamma} + \frac{KL}{N_b^{1/m}} \\
&= \left(C \frac{b^{1/(m\nu\gamma+1)}}{n_b} + KL \left(\frac{b}{N_b^\nu n_b} \right)^{1/(m\nu)} \left(\frac{n_b}{b^{1/(m\nu\gamma+1)}} \right)^{1/(m\nu)} \right) b^{-1/(m\nu+1/\gamma)}.
\end{aligned}$$

Since

$$C \frac{b^{1/(m\nu\gamma+1)}}{n_b} + KL \left(\frac{b}{N_b^\nu n_b} \right)^{1/(m\nu)} \left(\frac{n_b}{b^{1/(m\nu\gamma+1)}} \right)^{1/(m\nu)} \to C/a + KL(Ma)^{1/(m\nu)},$$

as $b \to \infty$, where M as in Assumption 2.3, and

$$\begin{aligned}
\frac{\log e(\mathcal{A}^{\text{sublin}}, n_b, N_b)}{\log b} &= \log \left(C \frac{b^{1/(m\nu\gamma+1)}}{n_b} + KL \left(\frac{b}{N_b^\nu n_b} \right)^{1/(m\nu)} \left(\frac{n_b}{b^{1/(m\nu\gamma+1)}} \right)^{1/(m\nu)} \right) / \log b \\
&+ \quad -1/(m\nu + 1/\gamma),
\end{aligned}$$

the second part of the theorem follows after taking limits as $b \to \infty$. □

The first result in Theorem 3.4 states that no asymptotically admissible discretization policy results in a faster rate of convergence of the total error bound than $b^{-1/(m\nu+1/\gamma)}$. The second result states that this optimal rate is attained using a specific policy. We see from Theorem 3.4 that the rate of convergence of the total error bound in the case of a sublinearly convergent optimization algorithm is apparently worse than the best possible achievable by finite, superlinear, and linear algorithms (see Theorems 3.1, 3.2, and 3.3), even for the optimal choice of discretization policy given by the second part of the theorem. Hence, there is a nontrivial computational cost of optimization in this case. As expected, if γ tends to infinity, then the rate in the sublinear case, under the optimal discretization policy, tends to that of the finite, superlinear, and linear cases. We note, however, that ν is typically smaller in the case of a sublinear algorithm than for superlinear and linear algorithms; see the discussion after Assumption 2.3. For example, in the case of the subgradient method, $\nu = 1$, and, since $\gamma = 1/2$ in that case, we obtain from Theorem 3.4 a rate of convergence of the total error bound of $b^{-1/(m+2)}$. In contrast, for a linearly convergent optimization algorithm with $\nu = 1.5$, we obtain a rate of convergence of the total error bound of $b^{-2/(3m)}$. Hence, for all uncertainty dimensions $m < 4$, the linear optimization algorithm results in a better rate of convergence than the sublinear algorithm. For $m = 4$, the rates are the same. For larger m, the sublinear algorithm obtains the better rate. Consequently, the results of this section indicate that the intuitive inclination of using a superlinear or linear algorithm instead of a sublinear one within a discretization algorithm may not always be supported by the above analysis. The next section examines one particular optimization algorithm based on exponential smoothing that behaves similarly to a sublinear algorithm.

Interestingly, the dimension d of x does not influence the above results. This is in dramatic contrast to the uncertainty dimension m, which causes severe degradation in the rate as it grows. Of course, a larger d results in more effort needed in each iteration as discussed after Assumption 2.3. However, in view of the analysis of the present paper, the rate with which one should increase the number of iterations n and level of discretization N, as more computing budget becomes available, remains unchanged with d.

4 Smoothing Optimization Algorithm

In this section, we consider an optimization algorithm for solving (P_N) based on exponential smoothing of $\psi_N(\cdot)$. Instead of solving (P_N) directly using a finite minimax algorithm, as discussed in the previous section, the exponential smoothing algorithm solves (P_N) by solving the smooth approximate problem

$$(P_{Np}) \qquad \min_{x \in X} \psi_{Np}(x),$$

where $p > 0$ is a smoothing parameter and

$$\psi_{Np}(x) \quad := \quad \frac{1}{p} \log \left(\sum_{y \in Y_N} \exp\left(p\phi(x, y)\right) \right). \tag{9}$$

The function $\psi_{Np}(\cdot)$ is a smooth approximation of $\psi_N(\cdot)$ first proposed in [30] and examined in [12, 28, 31–34] for solving finite minimax problem. It is well-known that

$$0 \leq \psi_{Np}(x) - \psi_N(x) \leq \log N/p, \tag{10}$$

for all $x \in \mathbb{R}^d$, $N \in \mathbb{N}$, and $p > 0$; see, for example, [12]. Consequently, a near-optimal solution of (P_N) can be obtained by solving (P_{Np}) for a sufficiently large p. A main advantage of the smoothing algorithm is that, when $\phi(\cdot, y)$ is smooth for all $y \in Y_N$, $\psi_{Np}(\cdot)$ is smooth and (P_{Np}) is solvable by unconstrained smooth optimization algorithms (if $X = \mathbb{R}^d$) or by projection-based smooth optimization algorithm (if X is polyhedral). Hence, the smoothing algorithm avoids solving large-scale quadratic programs as in the case of SQP and PPP minimax algorithms (see, for example, [11] and [8, Section 2.4]). In fact, each iteration of a gradient-based smoothing algorithm only require the evaluation of $\phi(\cdot, y)$ and $\nabla_x \phi(\cdot, y)$, $y \in Y_N$, at the current iterate (and at line search points), which imposes a computational cost proportional to N per iteration. Hence, $\nu = 1$ in Assumption 2.3 for the smoothing algorithm.

Specifically, for a given $N \in \mathbb{N}$, we consider the following smoothing algorithm for solving (P_N):

Optimization Algorithm $\mathcal{A}^{\text{smooth}}$ for Solving (P_N).

Data. $n \in \mathbb{N}$ and $p > 0$.

Step 1. Construct iterates $\{x^i_{Np}\}_{i=0}^n \subset \mathbb{R}^d$ by applying n iterations of an optimization algorithm to (P_{Np}). □

This simple smoothing algorithm $\mathcal{A}^{\text{smooth}}$ can be extended to include adaptive adjustment of the smoothing parameter p (see, for example, [12]), but we here focus on $\mathcal{A}^{\text{smooth}}$.

Discretization of Y, combined with exponential smoothing for the solution (P), is proposed in [35], where proof of convergence is provided, but without an analysis of rate of convergence. In this section, we determine the rate of convergence of this approach. Specifically, we consider the solution of (P) by discretization of Y, as in the previous sections, followed by the application of $\mathcal{A}^{\text{smooth}}$ to (P_N). While we above consider discretization policies of the form $\{(n_b, N_b)\}_{b=1}^{\infty}$, we now also need to determine a smoothing policy $\{p_b\}_{b=1}^{\infty}$, with $p_b > 0$, for all $b \in \mathbb{N}$. A smoothing policy specifies the smoothing parameter to be used in $\mathcal{A}^{\text{smooth}}$, given a particular computing budget b. The discretization policy gives the number of iterations to carry out in $\mathcal{A}^{\text{smooth}}$, as well as the level of discretization.

We assume that Assumption 2.3 holds for $\mathcal{A}^{\text{smooth}}$ regardless of p, i.e., the computational work to carry out n iteration of $\mathcal{A}^{\text{smooth}}$ is independent of p. In view of (9), the value of p does not influence the work to compute $\psi_{Np}(x)$ and its gradient, and hence this assumption is reasonable. However, as shown empirically in [33] and analytically in [12], a large value of p results in ill-conditioning of (P_{Np}) and slow rate of convergence of optimization algorithms applied to that problem. We adopt the following assumption, which, in part, is motivated by results in [12], as discussed subsequently.

Assumption 4.1. *Suppose that there exists an $\overline{N} \in \mathbb{N}$ such that, if $\{x_{Np}^i\}_{i=0}^n$ is constructed by optimization algorithm $\mathcal{A}^{\text{smooth}}$ with data $n \in \mathbb{N}$ and $p > 0$ when applied to (P_N), $N \in \mathbb{N}, N \geq \overline{N}$; then the following holds:*

(i) $x_{Np}^i \in X$ for all $i = 0, 1, ..., n$, $N \in \mathbb{N}$, $N \geq \overline{N}$, and $p > 0$,

(ii) X_N^* is nonempty for $N \in \mathbb{N}, N \geq \overline{N}$, and

(iii) there exist constants $k \in (0, 1)$ and $\kappa \in [0, \infty)$ such that

$$\psi_N(x_{Np}^n) - \psi_N^* \leq \left(1 - \frac{k}{p}\right)^n \kappa + \frac{2 \log N}{p} \tag{11}$$

for any $n, N \in \mathbb{N}, N \geq \overline{N}$ and $p \geq 1$. □

Part (i) of Assumption 4.1 requires that Algorithm $\mathcal{A}^{\text{smooth}}$ generates feasible iterates, which is easily achieved when X is either $\mathbb{R}^d$ or is a polyhedron. Part (iii) is stronger and stipulates that the "optimization error," after executing Algorithm $\mathcal{A}^{\text{smooth}}$, is bounded by the sum of two terms. The first term bounds the error caused by "incomplete" optimization and vanishes as $n \to \infty$. The second term bounds the smoothing error and tends to zero as $p \to \infty$; see (10). For a fixed $p \geq 1$, the first term indicates a linear rate of convergence as $n \to \infty$. However, the rate of convergence coefficient $(1 - k/p)$ tends to 1 as p grows, reflecting the increasing ill-conditioning of (P_{Np}). Hence, Algorithm $\mathcal{A}^{\text{smooth}}$ may converge only sublinearly if $p \to \infty$. If Step 1 of Algorithm $\mathcal{A}^{\text{smooth}}$ utilizes the steepest descent or projected gradient methods to solve (P_{Np}), then Assumption 4.1 holds under standard assumptions, as stated next.

Proposition 4.1. *Suppose that (i) $\phi(\cdot,\cdot)$ is twice continuously differentiable on $X \times Y$, (ii) $X = \mathbb{R}^d$ or is a polyhedron, (iii) there exists a constant $\lambda \in]0, \infty[$ such that*

$$\lambda \|z\|^2 \leq \langle z, \nabla^2_{xx}\phi(x, y)z \rangle,$$

*for all $x \in X$, $z \in \mathbb{R}^d$, and $y \in Y$, (iv) Step 1 of Algorithm $\mathcal{A}^{\text{smooth}}$ utilizes either the steepest descent method with Armijo step size rule (*see Algorithm 1.3.3 in [8]*) if $X = \mathbb{R}^d$ or, otherwise, the projected gradient method with Armijo step size rule (*see Algorithm 1.3.16 in [8]*), (v) there exists a constant $C \in [0, \infty[$ such that the initial iterate $x^0_{Np} \in X$ of Step 1 of Algorithm $\mathcal{A}^{\text{smooth}}$ satisfies $\psi(x^0_{Np}) \leq C$ for all $N \in \mathbb{N}$ and $p > 0$, and (vi) Assumption 2.1 holds. Then, Assumption 4.1 holds with $\overline{N}$ as in Assumption 2.1.*

Proof. Part (i) of Assumption 4.1 follows trivially by the choice of optimization algorithm in Step 1 of Algorithm $\mathcal{A}^{\text{smooth}}$. Part (ii) of Assumption 4.1 is a direct consequence of Assumption 2.1. We next consider part (iii).

Using the same arguments as in Lemma 3.1 of [12], we obtain that $\psi_{Np}(\cdot)$ is twice continuously differentiable and

$$\lambda \|z\|^2 \leq \left\langle z, \nabla^2 \psi_{Np}(x) z \right\rangle, \tag{12}$$

for any $x \in X$, $\in \mathbb{R}^d$, $N \in \mathbb{N}$, and $p > 0$. Moreover, a slight generalization of Lemma 3.2 in [12] yields that, for every bounded set $S \subseteq X$, there exists an $M_S < \infty$ such that

$$\langle z, \nabla^2 \psi_{Np}(x) z \rangle \leq p M_S \|z\|^2, \tag{13}$$

for all $x \in S, z \in \mathbb{R}^d$, $N \in \mathbb{N}$, and $p \geq 1$.

The steepest descent method with Armijo step size rule and the projected gradient method with Armijo step size rule have linear rate of convergence in function values under strong convexity. Its rate coefficient is $1-\xi\lambda_{\min}/\lambda_{\max}$, where $\xi \in]0,1[$ (which depends on the method) and $\lambda_{\max} \geq \lambda_{\min} > 0$ are upper and lower bounds on the eigenvalues of the Hessian of the objective function on a sufficiently large subset of $\mathbb{R}^d$; see Theorems 1.3.7 and 1.3.18 in [8]. Hence, in view of (12) and (13), pM_S and λ provide these upper and lower bounds in the case of (P_{Np}) and, therefore,

$$\psi_{Np}(x_{Np}^{n+1}) - \psi_{Np}^* \leq \left(1 - \frac{k}{p}\right)(\psi_{Np}(x_{Np}^n) - \psi_{Np}^*)$$

for all $n, N \in \mathbb{N}$ and $p \geq 1$, with $k = \xi\lambda/M_S \in (0,1)$. From (10), we then obtain that

$$\begin{aligned}
\psi_N(x_{Np}^n) - \psi_N^* &\leq \psi_{Np}(x_{Np}^n) - \psi_{Np}^* + \log N/p \\
&\leq \left(1 - \frac{k}{p}\right)^n (\psi_{Np}(x_{Np}^0) - \psi_{Np}^*) + \frac{\log N}{p} \\
&\leq \left(1 - \frac{k}{p}\right)^n (\psi_N(x_{Np}^0) - \psi_N^*) + \frac{2\log N}{p} \\
&\leq \left(1 - \frac{k}{p}\right)^n (\psi(x_{Np}^0) - \psi^* + LK) + \frac{2\log N}{p}
\end{aligned}$$

for all $n, N \in \mathbb{N}$ and $p \geq 1$, where we use the fact that $-\psi_N^* \leq -\psi^* + LK$ for all $N \geq \overline{N}$, $N \in \mathbb{N}$, in view of Proposition 2.1. Since we assume that $\psi(x_{Np}^0) \leq C$ for all $N \in \mathbb{N}$ and $p > 0$, the conclusion follows with $\kappa = C - \psi^* + LK$. □

We note that assumption (v) in Proposition 4.1 is rather weak and is satisfied, for example, if the optimization algorithm used to solve (P_{Np}) in Step 1 of Algorithm $\mathcal{A}^{\text{smooth}}$ is initialized with the same iterate regardless of $N \in \mathbb{N}$ and $p > 0$. Next result gives a total error bound for Algorithm $\mathcal{A}^{\text{smooth}}$ under Assumption 4.1.

Lemma 4.1. *Suppose that Assumptions 2.1 and 4.1 hold. If $\{x_{Np}^n\}_{n=0}^\infty$ is generated by Algorithm $\mathcal{A}^{\text{smooth}}$, then*

$$\psi(x_N^n) - \psi^* \leq \left(1 - \frac{k}{p}\right)^n \kappa + \frac{LK}{N^{1/m}} + \frac{2\log N}{p}$$

for all $n, N \in \mathbb{N}, N \geq \overline{N}$ and $p \geq 1$, where $\overline{N}$, k, and κ are as in Assumption 4.1, and L and K as in Assumption 2.1.

Proof. The conclusion follows directly from Proposition 2.1 and Assumption 4.1. □

In view of Lemma 4.1, we define the optimization error bound for Algorithm $\mathcal{A}^{\text{smooth}}$ to be

$$\Delta_{Np}^n(\mathcal{A}^{\text{smooth}}) := \left(1 - \frac{k}{p}\right)^n \kappa + \frac{2\log N}{p}, \tag{14}$$

and the total error bound for $n, N \in \mathbb{N}$ and $p > 0$ to be

$$e(\mathcal{A}^{\text{smooth}}, n, N, p) := \left(1 - \frac{k}{p}\right)^n \kappa + \frac{LK}{N^{1/m}} + \frac{2 \log N}{p}.$$

Before we proceed with the main result of this section, we need the following trivial fact.

Lemma 4.2. *For $x \in [0, 1/2]$, $-2x \le \log(1 - x) \le -x$.* □

Theorem 4.1. *Suppose that Assumptions 2.1, 2.3, and 4.1 hold, that $\{(n_b, N_b)\}_{b=1}^\infty$ is an asymptotically admissible discretization policy, and $\{p_b\}_{b=1}^\infty$ is a smoothing policy with $p_b \ge 1$ for all $b \in \mathbb{N}$. Then,*

$$\liminf_{b\to\infty} \frac{\log e(\mathcal{A}^{\text{smooth}}, n_b, N_b, p_b)}{\log b} \ge -\frac{1}{m\nu + 1},$$

where ν is as defined in Assumption 2.3 and m is the uncertainty dimension.

Moreover, if $p_b/b^{\delta\alpha} \to a \in]0, \infty[$, with $\delta \in]0, 1[$ and $\alpha = 1/(\delta m\nu + 1)$, and $n_b/b^\alpha \to a' \in]0, \infty[$, then

$$\lim_{b\to\infty} \frac{\log e(\mathcal{A}^{\text{smooth}}, n_b, N_b, p_b)}{\log b} = -\frac{1}{m\nu + 1/\delta}.$$

Proof. We first consider part one. If N_b is bounded as $b \to \infty$, then $e(\mathcal{A}^{\text{smooth}}, n_b, N_b, p_b)$ does no vanish as $b \to \infty$, and the conclusion of part one follows trivially. Hence, suppose there exists a $b_0 \in \mathbb{N}$ such that $N_b \ge 3$ for all $b \ge b_0$. Then, algebraic manipulations and Lemma 4.2 give that, for $b \ge b_0$,

$$\begin{aligned}
& \log e(\mathcal{A}^{\text{smooth}}, n_b, N_b, p_b) \\
= \;& \log(e^{n_b \log(1-k/p_b) + \log\kappa} + e^{-(1/m)\log N_b + \log LK} + e^{-\log p_b + \log\log N_b + \log 2}) \qquad (15) \\
\ge \;& \log(e^{-2kn_b/p_b + \log\kappa} + e^{-(1/m)\log N_b + \log LK} + e^{-\log p_b + \log\log N_b + \log 2}) \\
\ge \;& \log(\max\{e^{-2kn_b/p_b + \log\kappa}, e^{-(1/m)\log N_b + \log LK}, e^{-\log p_b + \log\log N_b + \log 2}\}) \\
= \;& \max\{-2kn_b/p_b + \log\kappa, -(1/m)\log N_b + \log LK, \; -\log p_b + \log\log N_b + \log 2\}.
\end{aligned}$$

We consider three cases. First, if $n_b \ge b^{1/(m\nu+1)}$, $b \ge b_0$, then

$$\begin{aligned}
\frac{\log e(\mathcal{A}^{\text{smooth}}, n_b, N_b, p_b)}{\log b} &\ge \frac{-(1/m)\log N_b + \log LK}{\log b} \\
&= \frac{-(1/(m\nu))\log(N_b^\nu n_b/b) - (1/(m\nu))\log(b/n_b) + \log LK}{\log b} \\
&\ge \frac{-(1/(m\nu))\log(N_b^\nu n_b/b) - (1/(m\nu))\log b^{m\nu/(m\nu+1)} + \log LK}{\log b} \\
&= \frac{-(1/(m\nu))\log(N_b^\nu n_b/b)}{\log b} - \frac{1}{m\nu + 1} + \frac{\log LK}{\log b}.
\end{aligned}$$

Second, if $n < b^{1/(m\nu+1)}$ and $p \le b^{1/(m\nu+1)}$, $b \ge b_0$, then

$$\begin{aligned}\frac{\log e(\mathcal{A}^{\text{smooth}}, n_b, N_b, p_b)}{\log b} &\ge \frac{-\log p_b + \log\log N_b + \log 2}{\log b}\\ &\ge \frac{-\log p_b}{\log b}\\ &\ge -\frac{1}{m\nu+1}.\end{aligned}$$

Third, if $n < b^{1/(m\nu+1)}$ and $p > b^{1/(m\nu+1)}$, $b \ge b_0$, then

$$\begin{aligned}\frac{\log e(\mathcal{A}^{\text{smooth}}, n_b, N_b, p_b)}{\log b} &\ge \frac{-2kn_b/p_b + \log\kappa}{\log b}\\ &> \frac{-2k + \log\kappa}{\log b}.\end{aligned}$$

Hence, for any $\epsilon > 0$, there exists a $b_1 \ge b_0$ such that

$$\frac{\log e(\mathcal{A}^{\text{smooth}}, n_b, N_b, p_b)}{\log b} \ge -\frac{1}{m\nu+1} - \epsilon$$

for all $b \in \mathbb{N}$, $b \ge b_1$. Since ϵ is arbitrary, the conclusion of part one follows.

We next consider part two. Let $b_0 \in \mathbb{N}$ be such that $N_b \ge 3$ for all $b \in \mathbb{N}, b \ge b_0$. For $b \ge b_0$, we define

$$\begin{aligned}\underline{e}(\mathcal{A}^{\text{smooth}}, n_b, N_b, p_b) \;&:=\; \exp\left(-\frac{2kn_b}{p_b} + \log\kappa\right) + \exp\left(-\frac{1}{m}\log N_b + \log LK\right)\\ &\quad + \exp\Bigg(-\log p_b + \log\log N_b + \log 2\Bigg).\end{aligned}$$

We define $\overline{e}(\mathcal{A}^{\text{smooth}}, n_b, N_b, p_b)$ identically except with $2k$ replaced by k. Then, using Lemma 4.2 and similar arguments as in (15), we obtain that

$$\underline{e}(\mathcal{A}^{\text{smooth}}, n_b, N_b, p_b) \le e(\mathcal{A}^{\text{smooth}}, n_b, N_b, p_b) \le \overline{e}(\mathcal{A}^{\text{smooth}}, n_b, N_b, p_b) \tag{16}$$

for all $b \in \mathbb{N}$, $b \ge b_0$. We next consider $\overline{e}(\mathcal{A}^{\text{smooth}}, n_b, N_b, p_b)$ and find that

$$-\frac{kn_b}{p_b} = -k\frac{n_b}{b^\alpha}\frac{b^{\delta\alpha}}{p_b}b^{\alpha(1-\delta)},$$

$$-\frac{1}{m}\log N_b = -\frac{1-\alpha}{mv}\log b - \frac{1}{m}\log\left(\frac{N_b^\nu n_b}{b}\right)^{1/\nu} - \frac{1}{m}\log\left(\frac{b^\alpha}{n_b}\right)^{1/\nu},$$

$$-\log p_b = -\delta\alpha\log b - \log\frac{p_b}{b^{\delta\alpha}},$$

and

$$\log\log N_b = \log\log b + \log\left(\frac{\log(N_b^\nu n_b/b)^{1/\nu}}{\log b} + \frac{\log(b^\alpha/n_b)^{1/\nu}}{\log b} + \frac{1-\alpha}{\nu}\right),$$

for all $b \in \mathbb{N}, b \geq b_0$. Using the above expressions, we obtain that, for all $b \in \mathbb{N}$, $b \geq b_0$,

$$\overline{e}(\mathcal{A}^{\text{smooth}}, n_b, N_b, p_b) = \exp\bigg(- \frac{\delta}{\delta m\nu + 1} \log b \bigg) (T_1(b) + T_2(b) + T_3(b)), \tag{17}$$

where

$$T_1(b) := \exp\bigg(- k \frac{n_b}{b^\alpha} \frac{b^{\delta\alpha}}{p_b} b^{\alpha(1-\delta)} + \frac{\delta}{\delta m\nu + 1} \log b + \log \kappa \bigg),$$

$$T_2(b) := \exp\bigg(\frac{\alpha - 1}{m\nu} \log b - \frac{1}{m\nu} \log \frac{N_b^\nu n_b}{b} - \frac{1}{m\nu} \log \frac{b^\alpha}{n_b} + \log LK + \frac{\delta}{\delta m\nu + 1} \log b \bigg),$$

and

$$\begin{aligned} T_3(b) \;\; := \;\; \exp \bigg(& - \delta\alpha \log b + \frac{\delta}{\delta m\nu + 1} \log b - \log \frac{p_b}{b^{\delta\alpha}} + \log\log b \\ & + \log \bigg(\frac{\log(N_b^\nu n_b/b)^{1/\nu}}{\log b} + \frac{\log(b^\alpha/n_b)^{1/\nu}}{\log b} + \frac{1-\alpha}{\nu} \bigg) + \log 2 \bigg) \bigg]. \end{aligned}$$

Since $n_b/b^\alpha \to a'$, $b^{\delta\alpha}/p_b \to 1/a$, as $b \to \infty$, $\alpha = 1/(\delta m\nu + 1)$, and $\delta \in]0,1[$, we obtain that $T_1(b) \to 0$ as $b \to \infty$. We also obtain that

$$\begin{aligned} T_2(b) \;\; &= \;\; \exp\bigg(- \frac{1}{m\nu} \log \frac{N_b^\nu n_b}{b} - \frac{1}{m\nu} \log \frac{b^\alpha}{n_b} + \log LK \bigg) \\ &\to \;\; \exp\bigg(- \frac{1}{m\nu} \log \frac{1}{M} - \frac{1}{m\nu} \log \frac{1}{a'} + \log LK \bigg), \end{aligned}$$

as $b \to \infty$, where M is as in Assumption 2.3. Moreover, we find that there exist constants $b_1 \geq b_0$ and $C \in [0, \infty[$ such that

$$\begin{aligned} & T_3(b) \\ = \;\; & \exp\bigg(- \log \frac{p_b}{b^{\delta\alpha}} + \log\log b + \log \bigg(\frac{\log(N_b^\nu n_b/b)^{1/\nu}}{\log b} + \frac{\log(b^\alpha/n_b)^{1/\nu}}{\log b} + \frac{1-\alpha}{\nu} \bigg) + \log 2 \bigg) \\ \leq \;\; & C e^{\log\log b} = C \log b \end{aligned}$$

for all $b \geq b_1$, $b \in \mathbb{N}$. Consequently, there exist constants $C' \in]C, \infty[$ and $b_2 \in \mathbb{N}, b_2 \geq b_1$, such that, for all $b > b_2$,

$$T_1(b) + T_2(b) + T_3(b) \leq C' \log b$$

for all $b \in \mathbb{N}, b \geq b_2$. Hence, for $b \geq b_2$,

$$\begin{aligned} \frac{\log \overline{e}(\mathcal{A}^{\text{smooth}}, n_b, N_b, p_b)}{\log b} \;\; &\leq \;\; \frac{\log(e^{-\delta/(\delta m\nu + 1) \log b} C' \log b)}{\log b} \\ &= \;\; -\frac{\delta}{\delta m\nu + 1} + \frac{C'}{\log b} + \frac{\log\log b}{\log b} \to -\frac{1}{m\nu + 1/\delta}, \end{aligned}$$

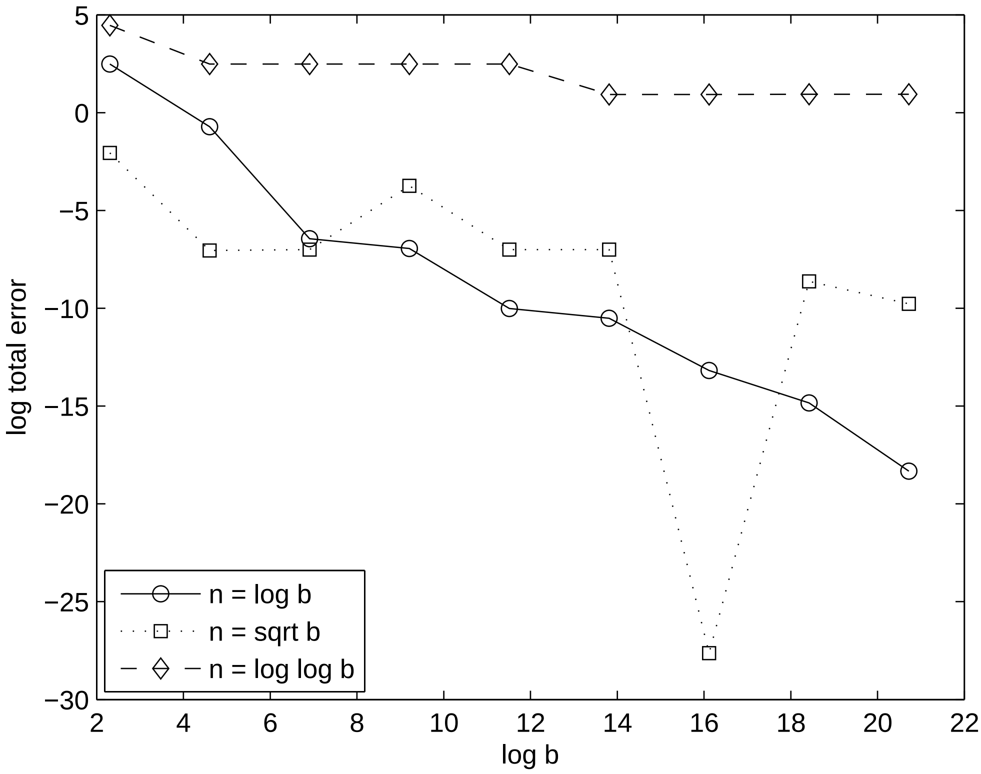

Figure 1: Total error $\psi(x_{N_b}^{n_b}) - \psi^*$ as a function of computing budget b on a logarithmic scale for the policies $n_b = \log b$, $n_b = \sqrt{b}$, and $n_b = \log\log b$.

as $b \to \infty$. Repeating the same argument for $\underline{e}(\mathcal{A}^{\text{smooth}}, n_b, N_b, p_b)$, we obtain that

$$\liminf_{b\to\infty} \frac{\log \underline{e}(\mathcal{A}^{\text{smooth}}, n_b, N_b, p_b)}{\log b} \geq -\frac{1}{m\nu + 1/\delta}.$$

Hence, the conclusion of part two of the theorem follows from (16). □

We see from Theorem 4.1 that Algorithm $\mathcal{A}^{\text{smooth}}$ is competitive with any sublinear optimization algorithm of degree $\gamma \in]0,1]$ (such as the subgradient method with $\gamma = 1/2$) as $\delta \in]0,1[$ can be selected arbitrarily close to one. While the best possible rate of $b^{-1/(m\nu)}$ is not attainable even for the optimal discretization and smoothing policy specified in Theorem 4.1, Algorithm $\mathcal{A}^{\text{smooth}}$ has $\nu = 1$ and, therefore, may still be competitive under certain circumstances.

5 Numerical Example

We illustrate the theoretical results of the paper by considering a problem instance with $X = \mathbb{R}^2$, $Y = [-5, 5]$, and $\phi(x, y) = 5(x_1^2 + x_2^2) - y^2 + x_1(-y + 5) + x_2(y + 3)$, which is Problem 1 in [1], p. 100, except that Y is reduced from two to one dimension. Consequently, $m = 1$. We use initial point $x = (10, -10)$ and the optimal solution, as reported in [24], is $(-0.49090909, -0.30909091)$, with optimal value $\psi^* = -1.69090909$. We let $Y_1 = \{5\}$ and $Y_N = \{-5, -5 + 1/(N-1), -5 + 2/(N-1), ..., , -5 + (N-2)/(N-1), 5\}$ for $N \geq 2$, $N \in \mathbb{N}$.

Focusing on Section 3.3, we adopt the PPP algorithm (see Section 2.4 in [8]), with the LSSOL quadratic program solver [27], for solving P_N, which is linearly convergent with rate of convergence coefficient $1 - \alpha\beta\lambda_{\min}/\lambda_{\max}$ for all iterations and any $N \in \mathbb{N}$. The quantities α and β are Armijo step size parameters, and $\lambda_{\min}$ and $\lambda_{\max}$ are lower and upper bounds on the smallest and largest eigenvalue of $\nabla_x \phi(x, y)$ on $X \times Y$, respectively. We use $\alpha = 0.5$, $\beta = 0.8$, and find that $\lambda_{\min} = \lambda_{\max} = 10$. Consequently, the PPP algorithm satisfies Definition 3.3 with $\overline{n} = 1$ and $\bar{c} = 0.6$. Fitting of empirical run time data from preliminary tests of the PPP algorithm using log-linear regression indicates that $\nu = 2$ in Assumption 2.3. This fact, and the use of a normalized computing budget (i.e., one that incorporates the constant M of Assumption 2.3) yield that an asymptotically admissible discretization policy must satisfy $n_b N_b^2/b \to 1$, as $b \to \infty$. Theorem 3.3 prescribes that a discretization policy with $n_b/\log b \to a$, with $a > -1/(m\nu \log \bar{c})$, attains the best possible rate of convergence of the total error bound. Consequently, we consider policies of the form $n_b = a \log b$, rounded to the nearest integer no smaller than 1, and $N_b = (b/n_b)^{1/2}$, rounded down to the nearest integer no smaller than 1. The critical value $-1/(m\nu \log \bar{c}) \approx 1.3$, and we therefore examine $a = 1$, 2, and 5. We consider computing budgets of $b = 10^1$, 10^2, ..., 10^9, which yield values of n_b and N_b in the ranges 2-104 and 1-6900, respectively. The solid lines in Figures 1-3 show the total error $\psi(x_{N_b}^{n_b}) - \psi^*$ as a function of computing budget b on a logarithmic scale for $a = 1$, 2, and 5, respectively. While the lines are not straight, they have a downward trend with slope of approximately -1 resulting in an empirical rate of convergence of order b^{-1}. As expected, this rate is somewhat better than the guaranteed rate of convergence of the total error bound, which Theorem 3.3 stipulates to be of order $b^{-1/(m\nu)} = b^{-1/2}$. Further

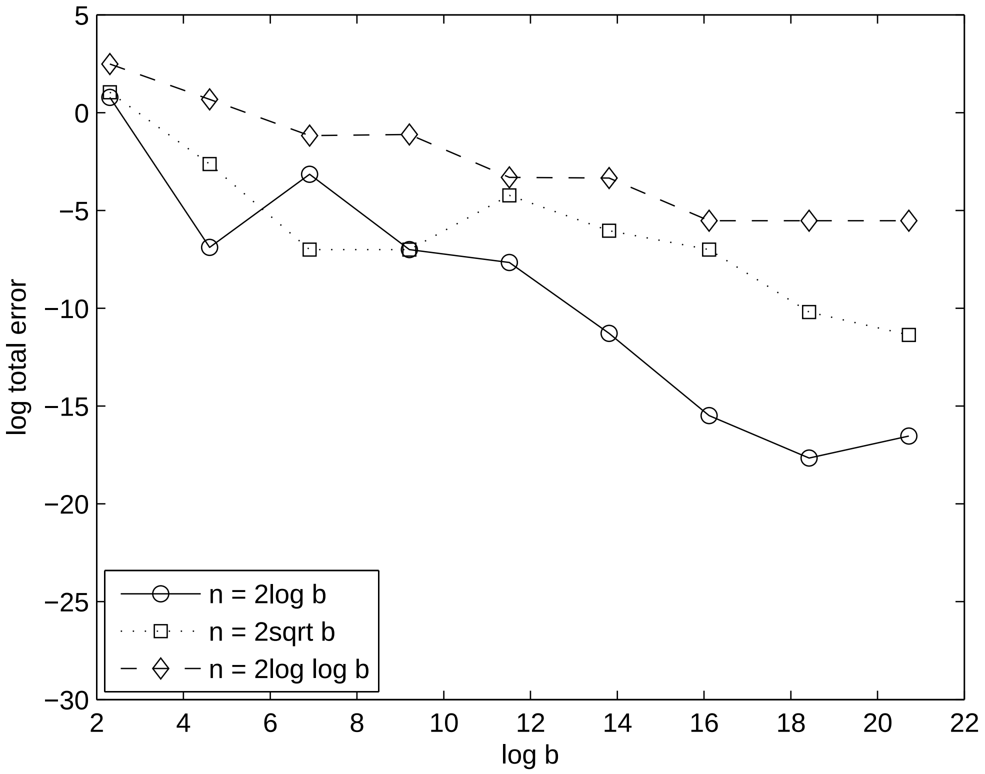

Figure 2: Total error as in Figure 1, but with $n_b = 2\log b$, $n_b = 2\sqrt{b}$, and $n_b = 2\log\log b$.

examination reveals that both the optimization and discretization errors tend to be smaller than their upper bounds of Definition 3.3 and Proposition 2.1 in this problem instance. Still, the rate appears somewhat insensitive to the value of a, as predicted by Theorem 3.3.

For the sake of comparison, we also consider the discretization policies $n_b = ab^{1/2}$ and $n_b = a\log\log b$, with N_b as above. The dotted and dashed lines in Figures 1-3 show the resulting total errors for $a = 1$, 2, and 5. The square-root policy $n_b = ab^{1/2}$ (dotted lines) has generally a smaller downward trend than that of the logarithm policy $n_b = a\log b$, with slopes of about -0.55 resulting in an empirical rate of convergence of $b^{-0.55}$. The slower rate, compared with that of the "optimal" logarithm policy, is consistent with Theorem 3.3. (In Figure 1, we observe an exceptionally small error at the data point near $\log b = 16$, which is explained by the fact that x_N^n, even for small n and/or N, may be close to an optimal solution of P by coincidence.) The iterated logarithm policy $n_b = a\log\log b$ (dashed lines in Figures 1-3) is poor for $a = 1$, with a slope of about -0.1692, but gradually improves as a increases. For $a = 5$, the slope is -0.9 and its empirical rate of convergence is near that of the logarithm

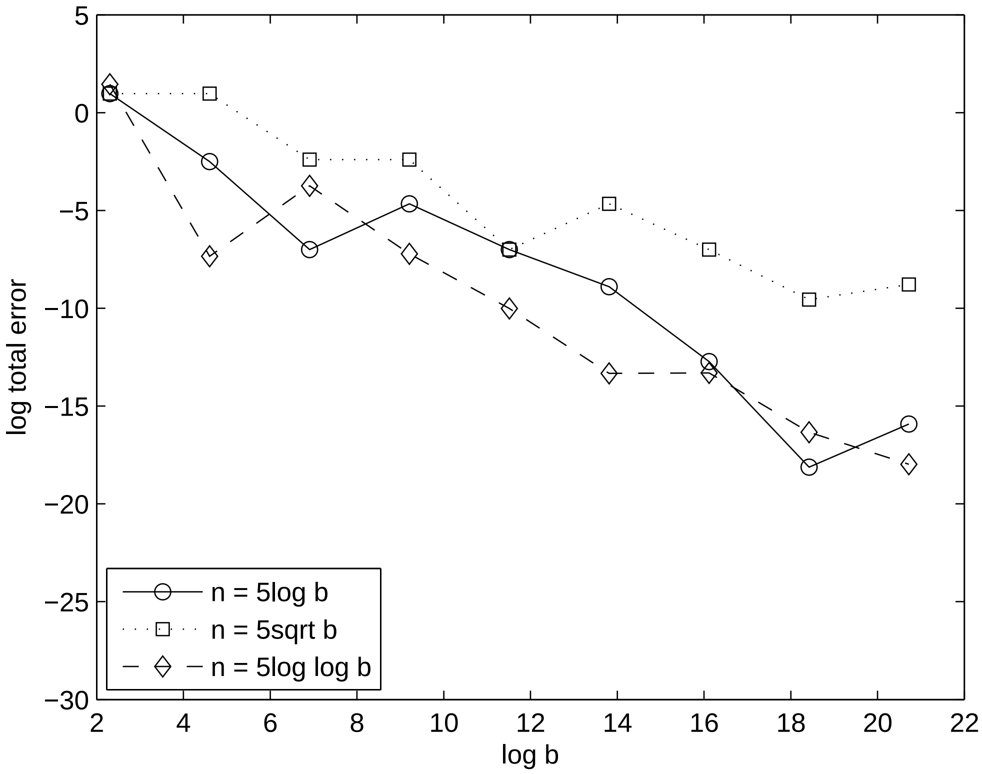

Figure 3: Total error as in Figure 1, but with $n_b = 5\log b$, $n_b = 5\sqrt{b}$, and $n_b = 5\log\log b$.

policy. The improvement stems from the fact that a large a conceals the slower-than-ideal growth of n_b under the iterated logarithm policy for the values of b considered. However, we expect that, as b grows further, n_b will tend to be too small resulting in large optimization errors relative to the discretization errors and, consequently, slower rate of convergence.

We also implement optimization algorithm $\mathcal{A}^{\text{smooth}}$ of Section 4, utilizing the steepest descent method with the Armijo step size rule for Step 1 and parameters as above. We conclude from Proposition 4.1 that Assumption 4.1 holds on the given problem instance and, consequently, also Theorem 4.1, with $\nu = 1$ as discussed after Assumption 2.3. Figure 4 displays the total errors for a range of $\delta \in]0,1[$ under the smoothing policy $p_b = b^{\delta\alpha}$ and discretization policy $n_b = b^{\alpha}$, rounded to the nearest integer no smaller than 1, and $N_b = (b/n_b)$, rounded down to the nearest integer no smaller than 1, where $\alpha = 1/(\delta + 1)$. Clearly, these policies are "optimal" in the sense of Theorem 4.1, and result in a guaranteed rate of convergence of the total error bound of $b^{-1/(1+1/\delta)}$. This theoretical result indicates an improving rate of convergence as $\delta \uparrow 1$, which is consistent with the empirical data of

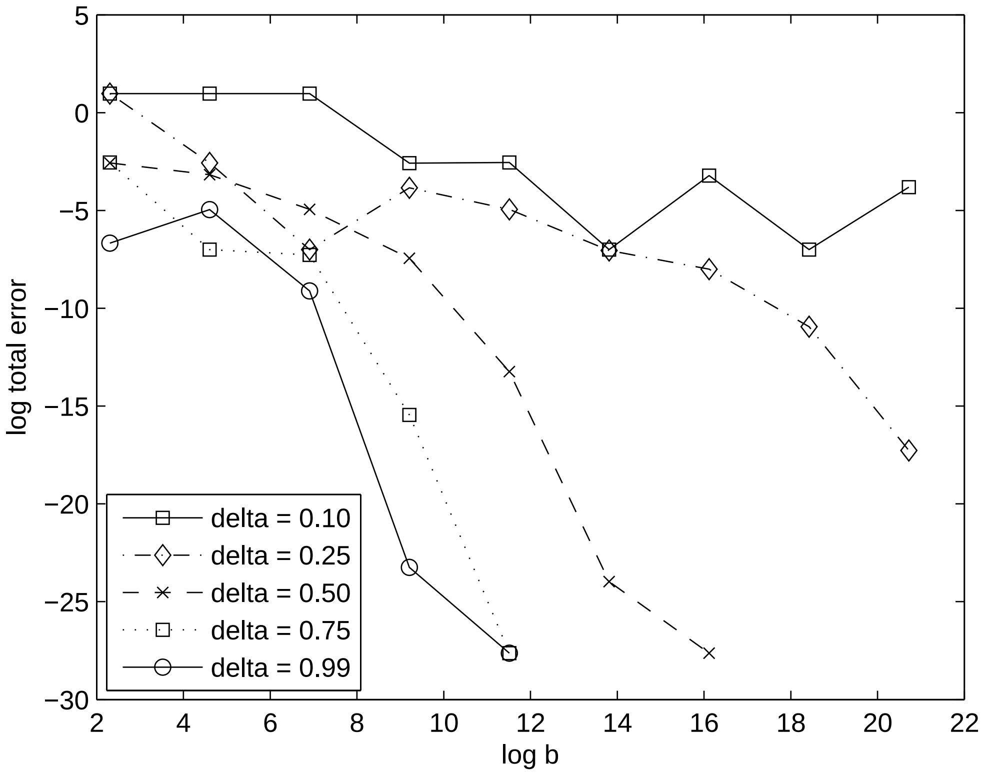

Figure 4: Total error $\psi(x^{n_b}_{N_b}) - \psi^*$ as a function of computing budget b on a logarithmic scale using a smoothing algorithm.

Figure 4. The slope of the line corresponding to $\delta = 0.1$ is approximately -0.4, but the slope gradually improves to -2.6 for $\delta = 0.99$. Again, the empirical rates are better than the guaranteed rates of Theorem 4.1, which are approximately $b^{-0.09}$ and $b^{-0.5}$ for $\delta = 0.1$ and 0.99, respectively. We note that the empirical rates, for large δ, are better than those attained with the PPP algorithm.

6 Conclusions

In this paper, we examined the rate of convergence of discretization algorithms for semi-infinite minimax problems as a computing budget b tends to infinity. These algorithms approximately solve finite minimax problems as subproblems, and we study the rates resulting from the use of various classes of optimization algorithms for this purpose.

We find that in the case of superlinear and linear optimization algorithms, the best possible rate of convergence is $b^{-1/(m\nu)}$, where m is the uncertainty dimension in the semi-infinite minimax problem, and ν is a positive parameter related to the computational work per iteration in the optimization algorithms. The best rate is attained with a particular optimal discretization policy identified in the paper and cannot be improved upon due to the unavoidable discretization error. Other policies may result in substantially slower rates. In the case of sublinear optimization algorithms, with optimization error of order $O(1/n^{\gamma})$, $\gamma > 0$, after n iterations, the best possible rate of convergence is $b^{-1/(m\nu+1/\gamma)}$, which is attained using an optimal discretization policy constructed in the paper. If a smoothing optimization algorithm solves the finite minimax problems, then the best possible rate of convergence is $b^{-1/(m\nu+1)}$, which one can get arbitrarily close to using a specific discretization and smoothing policy.

The algorithm parameter ν varies; there exist sublinear and smoothing algorithms with $\nu = 1$, and superlinear and linear finite minimax algorithms may have $\nu = 2$, as observed empirically. Consequently, under these assumptions, a sublinear algorithm with $\gamma = 1/2$, as in the case of the subgradient method, obtains a rate of convergence of $b^{-1/(m+2)}$. This is better than $b^{-1/(2m)}$ obtained by superlinear and linear algorithms for $m > 2$. The smoothing algorithm obtains essentially $b^{-1/(m+1)}$, which is better than superlinear and linear algorithms for $m > 1$. For $m = 1$ the rates are identical, but we observe empirically the superiority of the smoothing algorithm even in this case. The analysis and computational tests of this paper therefore indicate that inexpensive smoothing algorithms may be preferred.

The rates results of the paper provide guaranteed performance of discretization algorithms, as well as recommendations for the choice of algorithms, parameters, and policies. As expected, we find that the empirical performance of discretization algorithms is better than the worst-case guarantee, but recommendations remain valuable in algorithm selection and tuning, as we see in numerical examples.

References

[1] B. Rustem and M. Howe. *Algorithms for worst-case design and applications to risk management*. Princeton University Press, Princeton, New Jersey, 2002.

[2] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust optimization.* Princeton University Press, Princeton, New Jersey, 2009.

[3] D. Bertsimas, D. B. Brown, and C. Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53:464–501, 2011.

[4] R. Hettich and K. O. Kortanek. Semi-infinite programming: Theory, methods, and applications. *SIAM Review*, 35(3):380–429, 1993.

[5] R. Reemtsen and S. Gorner. Numerical methods for semi-infinite programming: A survey. In R. Reemtsen and J.-J. Ruckmann, editors, *Semi-infinite programming*, pages 195–275. Kluwer Academic Publishers, Dordrecht, Netherlands, 1998.

[6] M. Lopez and G. Still. Semi-infinite programming. *European J. Operational Research*, 180(2):491–518, 2007.

[7] L. Qi, C. Ling, X. Tong, and G. Zhou. A smoothing projected Newton-type algorithm for semi-infinite programming. *Computational Optimization and Applications*, 42:1–30, 2009.

[8] E. Polak. *Optimization. Algorithms and consistent approximations.* Springer, New York, NY, 1997.

[9] K. C. Kiwiel. Convergence of approximate and incremental subgradient methods for convex optimization. *SIAM J. Optimization*, 14(3):807–840, 2004.

[10] O. Devolder, F. Glineur, and Y. Nesterov. First order methods of smooth convex optimization with inexact oracle. *Optimization Online*, 2011.

[11] J. L. Zhou and A. L. Tits. An SQP algorithm for finely discretized continuous minimax problems and other minimax problems with many objective functions. *SIAM J. Optimization*, 6(2):461–487, 1996.

[12] E. Y. Pee and J. O. Royset. On solving large-scale finite minimax problems using exponential smoothing. *J. Optimization Theory and Applications*, 148(2):390–421, 2011.

[13] A. Shapiro. Semi-infinite programming, duality, discretization and optimality conditions. *Optimization: A Journal of Mathematical Programming and Operations Research*, 58(2):133–161, 2009.

[14] G. Still. Discretization in semi-infinite programming: the rate of convergence. *Mathematical Programming*, 91:53–69, 2001.

[15] E. Polak and L. He. Rate-preserving discretization strategies for semi-infinite programming and optimal control. *SIAM J. Control and Optimization*, 30(3):548–572, 1992.

[16] E. Polak, D. Q. Mayne, and J. Higgins. On the extension of Newton's method to semi-infinite minimax problems. *SIAM J. Control and Optimization*, 30(2):376–389, 1992.

[17] C.H. Chen, M. Fu, and L. Shi. Simulation and optimization. In *Tutorials in Operations Research*, pages 247–260, Hanover, Maryland, 2008. INFORMS.

[18] Y. L. Chia and P.W. Glynn. Optimal convergence rate for random search. In *Proceedings of the 2007 INFORMS Simulation Society Research Workshop*. Available at www.informs-sim.org/2007informs-csworkshop/2007workshop.html, 2007.

[19] D. He, L. H. Lee, C.-H. Chen, M.C. Fu, and S. Wasserkrug. Simulation optimization using the cross-entropy method with optimal computing budget allocation. *ACM Transactions on Modeling and Computer Simulation*, 20(1):133–161, 2010.

[20] R. Pasupathy. On choosing parameters in retrospective-approximation algorithms for stochastic root finding and simulation optimization. *Operations Research*, 58:889–901, 2010.

[21] J.O. Royset and R. Szechtman. Optimal budget allocation for sample average approximation. *Available at: http://faculty.nps.edu/joroyset/docs/RoysetSzechtman.pdf*, 2011.

[22] S.-H. Lee and P.W. Glynn. Computing the distribution function of a conditional expectation via Monte Carlo: Discrete conditioning spaces. *ACM Transactions on Modeling and Computer Simulation*, 13(3):238–258, 2003.

[23] C.H. Chen, D. He, M. Fu, and L. H. Lee. Efficient simulation budget allocation for selecting an optimal subset. *INFORMS J. Computing*, 20(4):579–595, 2008.

[24] E.Y. Pee. *On algorithms for nonlinear minimax and min-max-min problems and their efficiency.* Phd thesis, Naval Postgraduate School, Monterey, California, 2011.

[25] J.H. Jung, D.P. O'Leary, and A.L. Tits. Adaptive constraint reduction for convex quadratic programming. *Computational Optimization and Applications*, to appear, 2011.

[26] R. D. C. Monteiro and I. Adler. Interior path following primal-dual algorithms. Part II: Convex quadratic programming. *Mathematical Programming*, 44(1):43–66, 1989.

[27] P. E. Gill, S. J. Hammarling, W. Murray, M. A. Saunders, and M. H. Wright. *User's guide for LSSOL version 1.0: a Fortran package for constrained linear least-squares and convex quadratic programming.* Systems Optimization Laboratory - University of Stanford, Stanford, CA, 1986.

[28] E. Polak, R. S. Womersley, and H. X. Yin. An algorithm based on active sets and smoothing for discretized semi-infinite minimax problems. *J. Optimization Theory and Applications*, 138:311–328, 2008.

[29] Y. Nesterov. *Introductory lectures on convex optimization.* Kluwer Academic, Boston, Massachusetts, 2004.

[30] B. W. Kort and D. P. Bertsekas. A new penalty function algorithm for constrained minimization. In *Proceedings 1972 IEEE Conference Decision and Control*, pages 343–362, 1972.

[31] X. Li. An entropy-based aggregate method for minimax optimization. *Engineering Optimization*, 18:277–285, 1992.

[32] S. Xu. Smoothing method for minimax problems. *Computational Optimization and Applications*, 20:267–279, 2001.

[33] E. Polak, J. O. Royset, and R. S. Womersley. Algorithms with adaptive smoothing for finite minimax problems. *J. Optimization Theory and Applications*, 119(3):459–484, 2003.

[34] F. Ye, H. Liu, S. Zhou, and S. Liu. A smoothing trust-region Newton-CG method for minimax problem. *Applied Mathematics and Computation*, 199(2):581–589, 2008.

[35] E. Polak. Smoothing techniques for the solution of finite and semi-infinite min-max-min problems. In G. D. Pillo and A. Murli, editors, *High performance algorithms and software for nonlinear optimization*. Kluwer Academic Publishers, Dordrecht, Netherlands, 2003.