NOVEL GRADIENT-TYPE OPTIMIZATION ALGORITHMS FOR EXTREMELY LARGE-SCALE NONSMOOTH CONVEX OPTIMIZATION

Research Thesis

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Elena Olvovsky

SUBMITTED TO THE SENATE OF THE TECHNION - ISRAEL INSTITUTE OF TECHNOLOGY

TAMUZ 5765 HAIFA JANUARY 2005

The Research Thesis Was Done under the Supervision of Prof. Alexander Ioffe, Arie Leizarowitz (Faculty of Mathematics) and Prof. Arkadi Nemirovski (Faculty of Industrial Engineering and Management).

THE GENEROUS FINANCIAL HELP OF THE TECHNION IS GRATEFULLY ACKNOWLEDGED.

Contents

Abstract

In this work we develop new gradient-type methods for solving extremely large-scale convex optimization problems. This type of problems arises in many applications, e.g., medical imaging, design of mechanical structures, hard combinatorial problems, etc.

Up to the present moment, the common belief is that the best tools for solving large-scale "well-structured" convex problems are Polynomial Time Interior Point methods. The most attractive feature of these algorithms is their computational efficiency : the computational effort sufficient to find an ϵ-solution is proportional to the "number of accuracy digits" with the proportionality coefficient growing polynomially with the design dimension of the problem. This property means rapid convergence in terms of the number of calculations of the solution approximation (iterations). This provides for the possibility to get high-accuracy solutions. However, all known polynomial time algorithms share a common drawback: the computational effort per iteration grows nonlinearly with the design dimension of the problem. The modern computational facilities rule out processing of nonlinear convex algorithms employing the Interrior Point Methods for design dimensions of 10^5 order of magnitude and above.

In our research we address very large scale optimisation problems. The above mentioned limitation of Interrior Point methods requires exploration of alternatives. A less common approach is to use the so called "Black-Box" represented techniques. These techniques exhibit *dimension-independent* (or nearly dimension-independent) and nearly optimal, in the sense of Information-Based Complexity Theory, rate of convergence. Work on these simple gradient-type methods for convex optimization was inspired by the Unconstrained Gradient Descent algorithm of N. Shor and B. Polyak. Recently Non-Euclidean Restricted Memory method (NERML) was introduced by A. Ben-Tal and A. Nemirovskii. This novel subgradient-type technique is adjustable for the "geometry" of the problem to be solved and also capable to utilize information gathered about the function in previous iterations.

We address the two problems which arise in the application of NERML.

First, we provide an incremental implementation of NERML. This implementation has an added benefit (over incremental implementations of other algorithms)

of addressing a wide range of practical problems. In real life applications the objective function possesses certain specific structure. For instance, in Tomography Image Reconstruction, the objective is the sum of relatively simple functions, and the number of these functions is very large. In Shape Design, the objective function takes the form of the maximum of several other functions. The existing incremental methods adressed only the "sum of functions" case, specific to image reconstruction. We address the more general case of composed function. To be precise, in our work we develop Incremental implementation of Non-Euclidean Restricted Memory Level Method aimed to solve optimization problems of minimizing composed nondecreasing convex function of several convex functions. This implementation is based on the idea of dividing the set of inner functions into subsets and processing these subsets incrementally, one at a time.

Second, we broaden Non-Euclidean Restricted Memory Level Method to solve problems with functional constraints in such a manner that it can be used in the conjunction with the Incremental implementation.

For both Incremental and Constrained algorithms, we prove convergence and estimate efficiency.

Notation

List of Symbols

$\mathbf{R}^n$	n-dimensional Euclidean space
X	convex compact set, domain of the problem
x, y	general elements in the space
$O(\cdot)$	order of complexity
f, g	general symbols for function
$\mathcal{B}$	general method
x_i	search point
f'	gradient or subgradient of f
x^i	approximate solution generated in course of i steps
$\mathcal{P}$	family of optimization problems
$Compl_{\mathcal{P}}(\epsilon)$	ϵ-complexity of $\mathcal{P}$
$\|\cdot\|_p$	standard p-norm on n-dimensional coordinate space
$\triangle_n^{(+)}$	(full) n-dimensional simplex
$\prod_X(\cdot)$	projector onto X
γ	stepsize in the Subgradient Descent and Mirror Descent algorithms
$\langle\cdot,\cdot\rangle$	inner product
E	Euclidean space

$\operatorname{rint} X$	relative boundary of X
$\omega(\cdot)$	distance generating function
α	constant of strong convexity
$\omega_x(\cdot)$	distance function associated with x
$\|\cdot\|_*$	a norm conjugate to $\|\cdot\|$
Ω	is equal to $\max\limits_{x,y\in X}[\omega(y)-\omega(x)-\langle y-x,\omega'(x)\rangle]$
L	Lipschitz constant
$L_{\|\cdot\|}(f)$	Lipschitz constant of f with respect to the norm $\|\cdot\|$
$D_{\|\cdot\|}(X)$	diameter of set X with respect to the norm $\|\cdot\|$
σ	regularization parameter used in the entropy function
Tr	trace of matrix
$\mathbf{S^n}$	space of real symmetric matrices with Frobenius inner product
s	phase
f^s	best found till phase s feasible solution
f_s	a valid lower bound on the optimal value of f, found till phase s
f_*	optimal value of f
c_s	s-th prox-center
l_s	s-th level
F	model of f
t	step
$\mathcal{F}$	"outer" function in incremental implementation of NERML
x^*	optimal solution

List of Abbreviations

IPM	Interior Point Methods
IBCT	Information Based Complexity Theory
MD	Mirror Descent
NERML	Non-Euclidean Restricted Memory Level method
INERML	Incremental NERML

Chapter 1

Introduction

1.1 Large-Scale Convex Optimization via polynomial time methods: advantages and limitations

The primary goal of our research is to develop new methods for solving extremely large-scale convex optimization problems of the form

$$\min_x \{f(x) : x \in X\}, \tag{1.1}$$

where X is a convex compact set in $\mathbf{R}^n$ with a nonempty interior and f is a continuous convex function on X. Such problems arise in many applications, e.g., Medical Imaging, design of mechanical structures, relaxations of difficult combinatorial problems, etc.

In order to outline the scientific background of our research and to motivate the approach we have developed, we start with a brief overview of the state-of-the-art in large-scale convex optimization. For the time being, the common belief is that the best tools for solving large-scale "well-structured" convex problems are *Polynomial Time Interior Point methods* (IPM's) (see [34, 36, 7, 26, 35] and references therein), and we start with addressing strong points and limitations of these techniques. The most attractive feature of Interior Point algorithms is their *polynomiality* (theoretical equivalent of "computational efficiency"): the computational effort sufficient to find an ϵ-solution is proportional to the "number of accuracy digits" $\ln(1/\epsilon)$, with the proportionality coefficient growing polynomially with the sizes of the problem. From the practical standpoint, polynomiality means rapid convergence in terms of the iteration count and thus – the possibility to get high-accuracy solutions. However, all known polynomial time algorithms share a common drawback: the computational

effort per iteration grows *nonlinearly* with the design dimension n of the problem. For example, in an IPM as applied to a typical nonlinear problem (1.1), the cost of an iteration is $O(n^3)$. This phenomenon imposes limitations on the sizes of problems which can be processed, in a realistic time, by polynomial time algorithms: with n of order of 10^5 or more and the cost of an iteration like $O(n^3)$, just a single iteration will last "forever"...

At the present state of our knowledge, design dimensions like 10^5 and more rule out the possibility to process a nonlinear convex program by advanced polynomial-time techniques and leave us, essentially, with just two options:

- techniques for *unconstrained* minimization of *smooth* convex functions (Conjugate Gradients, Quasi-Newton, etc.),

  and

- subgradient-type techniques for *constrained* and/or *nonsmooth* convex optimization.

In our research, we focused on constrained nonsmooth optimization and thus – on subgradient-type techniques.

1.2 Gradient-type methods for large-scale convex optimization: advantages and limitations

The primary motivation for our emphasis on gradient-type methods as a promising tool for extremely large-scale optimization is given by the results of *Information-Based Complexity Theory*, and we start with outline of these results.

1.2.1 Black-box-represented convex problems and their information-based complexity

As applied to nonlinear problems in the form of (1.1), all traditional optimization techniques, including the gradient descent ones, assume complete a priori knowledge of the domain X and thus – possibilities of "global" w.r.t. X operations, like projecting on X. In contrast to this, the objective is assumed "black-box-represented" – the only source of information on f is given by a "First Order oracle" – a black-box routine which, given on input a point $x \in X$, returns on output the value and a subgradient of f at x. The main advantage of this assumption is its generality: it, in a sense, is the weakest assumption on the "information base" of the solution process under which large-scale convex optimization is still possible. In reality, when solving (1.1), we usually possess more a priori information on f than the minimum required

to mimic the First Order oracle, and in this sense the assumption in question is too pessimistic. However, the traditional optimization techniques are unable to utilize "rich" a priory information on f, so that for these techniques the disadvantages of the "black box" model of f are, essentially, irrelevant.

The Information-Based Complexity Theory (IBCT) postulates the black-box representation of the objective in (1.1) and is aimed at finding limits of performance of the corresponding optimization methods. Such a method $\mathcal{B}$ is defined as a collection of *search rules* which specify the subsequent *search points* $x_1 \in X$, $x_2 \in X$,... where $\mathcal{B}$ asks the oracle to compute the values $f(x_t)$ and subgradients $f'(x_t)$ of the objective, along with a collection of *generating rules* which define the subsequent *approximate solutions* $x^1 \in X$, $x^2 \in X$,... generated by the method. It is assumed that t-th search rule can be an arbitrary deterministic function, taking values in X, of the "information" $\mathcal{I}^{t-1} = (f(x_1), f'(x_1), ..., f(x_{t-1}), f'(x_{t-1}))$ accumulated prior to the t-th step, while t-th generating rule can be an arbitrary deterministic function of $\mathcal{I}^t$.

In the IBCT, we fix a *family* $\mathcal{P}$ of optimization problems (1.1), all sharing a common feasible domain X, so that a particular problem from the family can be identify with the corresponding objective f, and ask what is the *Information-Based complexity* of the family, where the central notion of the *complexity* is defined as follows. Let $\epsilon > 0$. We say that the ϵ-complexity $\mathrm{Compl}_{\mathcal{P}}(\epsilon)$ of $\mathcal{P}$ is equal to N, if

> – first, that there exists an optimization method $\mathcal{B}$ such that N-th approximate solution $x^N_{\mathcal{B}}(f)$ generated by $\mathcal{B}$ as applied to instance $f \in \mathcal{P}$ is, for every instance $f \in \mathcal{P}$, ϵ-solution of the instance:
>
> $$f(x^N_{\mathcal{B}}(f)) - \min_X f \leq \epsilon;$$
>
> – second, no solution method can solve *all* instances from $\mathcal{P}$ within accuracy ϵ in less than N steps, i.e., for every integer $M < N$ and every optimization method $\mathcal{B}$ there exists an instance $f \in \mathcal{P}$ such that the M-th approximate solution generated by $\mathcal{B}$ as applied to f is *not* an ϵ-solution to the instance.

The function $\mathrm{Compl}_{\mathcal{P}}(\epsilon)$ we have defined is called the information-based complexity of $\mathcal{P}$.

Informally speaking, $\mathrm{Compl}_{\mathcal{P}}(\epsilon)$ can be treated as a *lower bound* on the computational effort allowed to *guarantee* solving *all* instances from $\mathcal{P}$ within accuracy ϵ by a black-box oriented optimization method. The information-based complexity measures this effort in the number of required oracle calls (computations of f and f') and pays no attention on the amount of computations required to process the answers of the oracle (i.e., to compute the search and the generating rules).

Thus, the information-based complexity provides us with *limits of performance* of black-box-oriented methods.

Remark 1.1 When defining the complexity, we have restricted ourselves with *first order* black-box-oriented methods – those using only the values and the first order derivatives of the objective. In principle, black-box-oriented methods can use derivatives of the objective of higher order as well, which might reduce the complexity. It turns out, however, that in the situations we are interested in this reduction does not occur. In fact, the complexity bounds to follow remain valid when the First Order oracle is replaced with an arbitrary *local* oracle reporting *at least* the first order information. Here the notion of *locality* is defined as follows: an oracle is called local, if for every query point $x \in X$ and every pair of objectives f, g which coincide in a neighbourhood of x, the answers of the oracle when asked at x about f and about g are identical to each other.

1.2.2 Information-Based Complexity of Convex Optimization

Main results on Information-based complexity of Convex Programming can be summarized as follows [23]. Let X be a solid in $\mathbf{R}^n$ (a convex compact set with a nonempty interior), and let $\mathcal{P}$ be the family of all convex functions on $\mathbf{R}^n$ normalized by the condition

$$\max_X f - \min_X f \leq 1. \tag{1.2}$$

For this family,

C.1. Complexity of finding high-accuracy solutions in fixed dimension is independent of the geometry of X. Specifically,

$$\begin{array}{ll} \forall(\epsilon \leq \epsilon(X)): & O(1)n\ln\left(2+\frac{1}{\epsilon}\right) \leq \mathrm{Compl}(\epsilon); \\ \forall(\epsilon > 0): & \mathrm{Compl}(\epsilon) \leq O(1)n\ln\left(2+\frac{1}{\epsilon}\right), \end{array} \tag{1.3}$$

where

- $O(1)$ are appropriately chosen positive absolute constants,
- $\epsilon(X)$ depends on the geometry of X, but never is less than $\frac{1}{n^2}$, where n is the dimension of X.

Note that the quantity $\ln(1/\epsilon)$ can be though of as the number of accuracy digits in an ϵ-solution; with this interpretation, (1.3) says that the "price" of an accuracy digit is $O(n)$ oracle calls (except, perhaps, for $O(\ln n)$ initial accuracy digits which may be cheaper).

C.2. Complexity of finding solutions of fixed accuracy in high dimensions does depend on the geometry of X. Here are 3 typical results:

(a) Let X be an n-dimensional box: $X = \{x \in \mathbf{R}^n : \|x\|_\infty \leq 1\}$. Then

$$\epsilon \leq \frac{1}{2} \Rightarrow O(1)n\ln(\frac{1}{\epsilon}) \leq \mathrm{Compl}(\epsilon) \leq O(1)n\ln(\frac{1}{\epsilon}). \tag{1.4}$$

(b) Let X be an n-dimensional ball: $X = \{x \in \mathbf{R}^n : \|x\|_2 \leq 1\}$. Then

$$n \geq \frac{1}{\epsilon^2} \Rightarrow \frac{O(1)}{\epsilon^2} \leq \mathrm{Compl}(\epsilon) \leq \frac{O(1)}{\epsilon^2}. \tag{1.5}$$

(c) Let X be an n-dimensional hyperoctahedron: $X = \{x \in \mathbf{R}^n : \|x\|_1 \leq 1\}$. Then

$$n \geq \frac{1}{\epsilon^2} \Rightarrow \frac{O(1)}{\epsilon^2} \leq \mathrm{Compl}(\epsilon) \leq \frac{O(\ln n)}{\epsilon^2} \tag{1.6}$$

(in fact, $O(1)$ in the lower bound can be replaced with $O(\ln n)$, provided that $n >> \frac{1}{\epsilon^2}$).

Since we are interested in extremely large-scale problems, the conclusions which we can extract from the outlined results are as follows:

• C.1 is discouraging: it says that we have no hope to guarantee high accuracy, like $\epsilon = 10^{-6}$, when solving large-scale problems with black-box-oriented methods; indeed, with $O(n)$ steps per accuracy digit and *at least* $O(n)$ operations per step (this many operations are required already to input a search point to the oracle), the arithmetic cost per accuracy digit is at least $O(n^2)$, which is prohibitively large for really large n. Note also that the above $O(n^2)$ is just a *lower bound* given by the "over-optimistic" assumption that the computational effort per step is $O(n)$. For all known methods, a "remote" accuracy digit costs *at least* $O(n)$ oracle calls and *at least* $O(n^4)$ operations to process the answers of the oracle.

• C.2 is partly discouraging, partly encouraging. A bad news reported by C.2 is that when X is a box (which is the most typical situation in applications), we have no hope to solve extremely large-scale problems in a reasonable time to a *guaranteed*, even low, accuracy, since the required number of steps should be at least of order of n. A good news reported by C.2 is that *there exist situations where the complexity of minimizing a convex function to a fixed accuracy is independent, or nearly independent, of the design dimension*. Of course, the dependence of the complexity bounds in (1.5) and (1.6) on ϵ is very far from being polynomial in $\ln(1/\epsilon)$; however, this drawback is tolerable when we do not intend to get high accuracy. Another drawback is that there are not that many applications where the feasible set is a ball or a hyperoctahedron. Note, however, that in fact we can

save the most important for us *upper* complexity bounds in (1.5) and (1.6) when requiring from X to be a *subset* of a ball, respectively, of a hyperoctahedron, rather than to be the entire ball/hyperoctahedron. This extension is not costless: we should simultaneously strengthen the normalization condition (1.2). Specifically, it turns out [23, 7] that

C.3. The upper complexity bound in (1.5) remains valid when $X \subset \{x : \|x\|_2 \leq 1\}$ and

$$\mathcal{P} = \mathcal{P}^2_{\rm Lip} = \{f : f \ \text{ is convex and } |f(x) - f(y)| \leq \|x - y\|_2 \ \forall x, y \in X\}.$$

When X is the unit Euclidean ball, or the intersection of this ball with the nonnegative orthant, the lower complexity bound in (1.5) remains valid when the family of all convex objectives satisfying (1.2) is replaced with $\mathcal{P}^2_{\rm Lip}$.

C.4. The upper complexity bound in (1.6) remains valid when $X \subset \{x : \|x\|_1 \leq 1\}$ and

$$\mathcal{P} = \mathcal{P}^1_{\rm Lip}\{f : f \ \text{ is convex and } |f(x) - f(y)| \leq \|x - y\|_1 \ \forall x, y \in X\}.$$

When X is the hyperoctahedron, or the intersection of this set with the nonnegative orthant (which is the simplex $\Delta_n^+ = \{x \in \mathbf{R}^n : x \geq 0, \sum\limits_i x_i \leq 1\}$, or the simplex $\Delta_n = \{x \in \mathbf{R}^n : x \geq 0, \sum\limits_i x_i = 1\}$, the lower complexity bound in (1.6) remains valid when the family of all convex objectives satisfying (1.2) is replaced with $\mathcal{P}^1_{\rm Lip}$.

A crucial good news which makes the complexity results C.3-4 worthy of *practical* interest is that *if a convex problem* (1.1) *is as required in C.3-4 and the domain X of this problem is a simple set, then the (nearly) dimension-independent complexity bounds in C.3-4 can be achieved by "cheap" gradient-type optimization techniques* – with computational effort per step reducing to a single oracle call plus $O(n)$ operations to process the answer of the oracle. In light of this fact combined with C.3-4, we have reasons to hope that under favourable circumstances cheap gradient-type techniques are *the* techniques of choice for extremely large-scale nonsmooth/constrained convex optimization.

We should understand, of course, how realistic are "favourable circumstances" as stated in C.3-4. In this respect, the "ball-like" case mentioned in C.3 seems to be rather artificial: the Euclidean norm associated with this case is a very natural mathematical entity, but this is all one can say in its favour. For example, the normalization of the objective in C.3 is that the Lipschitz constant of f w.r.t. $\|\cdot\|_2$ is ≤ 1, or, which is the same, that the vector of the first order partial derivatives of

f should, at every point, be of $\|\cdot\|_2$-norm not exceeding 1. In other words, "typical" magnitudes of the partial derivatives of f should become smaller and smaller as the number of variables grows; what could be the reasons for such a strange behaviour? In contrast to this, the normalization condition imposed on f in C.4 is that the Lipschitz constant of f w.r.t. $\|\cdot\|_1$ is ≤ 1, or, which is the same, that the $\|\cdot\|_\infty$-norm of the vector of partial derivatives of f is ≤ 1. In other words, the normalization is that the magnitudes of the first order partial derivatives of f should be ≤ 1, and this normalization is "dimension-independent". Of course, in C.3 we deal with minimization over subsets of the unit ball, while in C.4 we deal with minimization over the subsets of the unit hyperoctahedron, a set which is much smaller than the unit ball. However, there do exist problems in reality where we should minimize over the standard simplex

$$\Delta_n = \{x \in \mathbf{R}^n : x \geq 0, \sum_x x_i = 1\},$$

which indeed is a subset of the unit hyperoctahedron, or over much more complicated sets ("spectahedrons") allowing for complexity bounds similar to (1.6).

1.2.3 Intermediate conclusions

The discussion above suggests that when solving extremely large-scale convex programs, it makes sense to look for simple gradient-type techniques which, as applied to convex programs (1.1) with feasible sets of appropriate geometry, exhibit *dimension-independent* (or nearly dimension-independent) and nearly optimal, in the sense of Information-Based Complexity Theory, rate of convergence. This is the general approach we intend to undertake in our research.

1.3 Gradient-type methods for large-scale optimization: state of the art

Gradient-type methods for *nonsmooth* convex optimization originate from the *Subgradient Descent* algorithm originating from N. Shor [31] and B. Polyak [25]. As applied to (1.1), the algorithm becomes

$$x_{t+1} = \Pi_X(x_t - \gamma f'(x_t)), \tag{1.7}$$

where $\Pi_X(y) = \mathrm{argmin}_{y \in X} \|x - y\|_2$ is the projector onto X and $\gamma_t > 0$ are "stepsizes". With properly chosen stepsizes, this algorithm satisfies C.3 (see below).

Subgradient Descent was extensively studied in the literature (see, e.g., [15, 17] and references therein) and was the starting point of numerous extensions. This includes:

1. Extensions from convex problems (1.1) with exact First Order oracle to other problems with convex structure (see [23] and references therein), specifically,

   – problems with convex functional constraints

   $$\min_x \{f(x) : f_i(x) \leq 0,\, i = 1, ..., m, x \in X\},$$

   – saddle point problems

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

   with convex-concave functions F and convex X, Y

   – variational inequalities with monotone operators,

   in the cases of both exact and *noisy* oracles;

2. Inventing *bundle* versions of Subgradient Descent.

   As a practical matter, a severe drawback of Subgradient Descent is that the method is "memoryless": all the information accumulated at the first $t-1$ steps of the method is "summarized" in the corresponding iterate x_t, and this "summary" (which is far from being complete) is all what influences the subsequent computations. As a result of this poor utilization of information, in practical computations the method typically rapidly progresses at the first few tens of iterations and then, in full accordance with the complexity bound (1.5), "gets stuck", with no significant progress in accuracy during thousands of subsequent iterations. The *bundle* versions of Subgradient Descent (originating from C. Lemarechal [18]; for further developments, see [21, 29, 20, 16]) memorize, completely or partly, the information accumulated so far and, as a result, exhibit much more attractive practical behaviour than the Subgradient Descent. Although the theoretical convergence properties of bundle methods in the situation C.3 are not better than those of the Subgradient Descent (see (1.5)), an *empirical* fact is that the "complete memory" bundle methods obey the complexity bound (1.3) and thus are capable to find high-accuracy solutions, provided that the sizes of the problem allow to carry out $O(n \ln(1/\epsilon))$ steps. Besides this advantage (which is of no importance in the extremely large-scale case, where just n steps is "too much"), the bundle methods usually outperform the Subgradient one already at the initial phase of the solution process.

3. Developing "non-Euclidean" versions of Subgradient Descent and bundle methods.

   It turns out that Subgradient Descent and its extensions we have mentioned so far are intrinsically related to problems with "Euclidean geometry", like those

described in C.3. For example, no known Subgradient Descent/Bundle methods satisfy C.4. The "non-Euclidean" extensions of gradient-type methods – *Mirror Descent methods* – originate from [23]; for more comprehensive representation, see [6, 1]. These methods allow to adjust, to some extent, a method to the geometry of problems in question. For the time being, only "memoryless" Mirror Descent algorithms were known; only recently a bundle-type version of this scheme, called NERML (Non-Euclidean Restricted Memory Level method), and aimed at solving extremely large-scale convex programs, was developed [8].

The starting points for our research are the general Mirror Descent scheme and its bundle version – the NERML algorithm. We are about to present a detailed overview of the corresponding results.

1.3.1 The Mirror Descent scheme

The presentation below follows the one of [1]. We focus on problem (1.1) and make the following assumption:

> A. (i) The feasible domain X in (1.1) is a convex compact subset of a finite-dimensional Euclidean space E with inner product $\langle \cdot, \cdot \rangle$.
>
> (ii) The objective f is convex and Lipschitz continuous on X. The subgradients $f'(x)$, $x \in X$, reported by the First Order oracle are parallel to the affine span of X. Besides this, when x belongs to the relative boundary of X, $f'(x)$ belongs to the closure of the set $\bigcup_{y \in \mathrm{rint}\, X} \partial f(y)$.

The general Mirror Descent scheme

The general MD algorithm for solving problems of the form (1.1) is specified by a *distance-generating function* $\omega(x)$ which should be a continuously differentiable and strongly convex function on X:

$$\forall (x, y \in X) : \langle \omega'(x) - \omega'(y), x - y \rangle \geq \alpha \|x - y\|^2, \tag{1.8}$$

where $\|\cdot\|$ is a once for ever fixed norm on E (not necessary the Euclidean norm $\|u\|_2 \equiv \sqrt{\langle u, u \rangle}$ associated with the inner product), and $\alpha > 0$[1)]. Note that (1.8) is

[1)] Note that strong convexity of $\omega(\cdot)$ implies inequality of the type (1.8) w.r.t *whatever* norm on E (recall that E is finite-dimensional). What is important for us, however, is the value of the constant of strong convexity α, which does depend on the choice of the norm, this is why we consider *both* $\omega(\cdot)$ and $\|\cdot\|$ as "setup elements" of our construction.

equivalent to the fact that all *distance functions*

$$\omega_x(y) = \omega(y) - \omega(x) - \langle y - x, \omega'(x)\rangle \tag{1.9}$$

(other names: "prox-terms", "Bregman distances") associated with $x \in X$ satisfy the relation

$$\forall (x, y \in X) : \omega_x(y) \geq \frac{\alpha}{2}\|x - y\|^2. \tag{1.10}$$

The generic MD algorithm associated with $X, \omega(\cdot)$ generates the search points $x_t \in X$ according to the rule

$$x_{t+1} = \operatorname*{argmin}_{y \in X} \left[\omega_{x_t}(y) + \gamma_t \langle f'(x_t), y - x_t\rangle\right], \tag{1.11}$$

where $\gamma_t > 0$ are stepsizes. Informally speaking, at a step we are minimizing over X a linear approximation, taken at x_t, of the objective and augmented by the "prox term" $\frac{1}{\gamma_t}\omega_{x_t}(y)$ which, according to (1.10), penalizes the deviation of a point from x_t and therefore "tries" to keep the next iterate close to the previous one, thus preventing too long steps (since a too long step may lead to a point where the local linear model of f is very far from f).

The MD algorithm starts with an arbitrary point $x_1 \in X$; the approximate solution x^t generated in course of t steps is the best (with the smallest value of the objective) of the search points $x_1, ..., x_t$:

$$x^t \in \operatorname*{Argmin}_{x \in \{x_1, ..., x_t\}} f(x). \tag{1.12}$$

Convergence analysis

Convergence properties of the general MD algorithm can be summarized in the following

Theorem 1.1 [6, 1] *Under assumption* A*, one has*

$$f(x^t) - \min_X f \leq \min_{1 \leq p \leq q \leq t} \left[\frac{\Omega + \alpha^{-1}\sum_{\tau=p}^{q} \gamma_\tau^2 \|f'(x_\tau)\|_*^2}{\sum_{\tau=p}^{q} \gamma_\tau}\right], \tag{1.13}$$

where

- α *is the constant of strong convexity of* $\omega(\cdot)$ *with respect to the norm* $\|\cdot\|$ *(see* (1.8)*),*

- $\Omega = \max\limits_{x,y\in X}[\omega(y) - \omega(x) - \langle y - x, \omega'(x)\rangle]$ *($\Omega < \infty$, since $\omega(\cdot)$ is continuously differentiable on X and X is compact),*
- $\|x\|_* = \max\{\langle x, \xi\rangle : \|\xi\| \le 1$ *is the norm conjugate to* $\|\cdot\|$.

In particular,

(i) *Whenever $\gamma_t \to 0$, $t \to \infty$, and $\sum\limits_t \gamma_t = \infty$, one has $f(x^t) \to \min\limits_X f$;*

(ii) *With the "optimal" stepsizes*

$$\gamma_t = \frac{\sqrt{\Omega\alpha}}{\|f'(x_t)\|_*\sqrt{t}} \tag{1.14}$$

one has

$$f(x^t) - \min_X f \le \frac{2L\sqrt{\Omega}}{\sqrt{\alpha t}}, \tag{1.15}$$

where

$$L = L_{\|\cdot\|}(f) = \sup_{\substack{x,y\in X\\ x\ne y}} \frac{|f(x) - f(y)|}{\|x - y\|}$$

is the Lipschitz constant of f w.r.t. the norm $\|\cdot\|$.

Proof. For $u, x \in X$, let us set

$$H_u(x) = \langle x - u, \omega'(x)\rangle - \omega(x).$$

For τ fixed, let us set $x = x_t$, $\gamma = \gamma_t$ and $x_+ = x_{t+1}$. We have

$$\begin{array}{rrcl} & x_+ & = & \operatorname{argmin}\limits_{y\in X}[\gamma\langle f'(x), y - x\rangle + \omega_x(y)] \qquad\qquad \Rightarrow \\ (a) & 0 & \le & \langle \omega'(x_+) - \omega'(x) + \gamma f'(x), v - x_+\rangle \quad \forall v \in X \end{array}$$

Further,

$$
\begin{array}{rrcl}
 & H_u(x_+) & = & \langle x_+ - u, \omega'(x_+)\rangle - \omega(x_+) \\
 & & = & \langle (x_+ - x) + (x - u), (\omega'(x_+) - \omega'(x)) + \omega'(x)\rangle \\
 & & & -\omega(x_+) \\
 & & = & [\langle x - u, \omega'(x)\rangle - \omega(x)] \\
 & & & +\langle x_+ - x, \omega'(x)\rangle + \langle x_+ - x, \omega'(x_+) - \omega'(x)\rangle \\
 & & & +\langle x - u, \omega'(x_+) - \omega'(x)\rangle \\
 & & & +[\omega(x) - \omega(x_+)] \\
 & & = & H_u(x) + [\omega(x) + \langle x_+ - x, \omega'(x)\rangle - \omega(x_+)] \\
 & & & +\langle x_+ - u, \omega'(x_+) - \omega'(x)\rangle \\
\Rightarrow & H_u(x_+) & \leq & H_u(x) + [\omega(x) + \langle x_+ - x, \omega'(x)\rangle - \omega(x_+)] \\
 & & & +\gamma\langle u - x_+, \gamma f'(x)\rangle \\
 & & & \hfill [\text{by } (a)] \\
\Rightarrow & & & \\
(b) & \gamma\langle f'(x), x - u\rangle & \leq & [H_u(x) - H_u(x_+)] \\
 & & & +\underbrace{[\omega(x) + \langle x_+ - x, \omega'(x)\rangle - \omega(x_+)]}_{\leq 0} \\
 & & & +\gamma\langle f'(x), x - x_+\rangle
\end{array}
$$

Besides this,

$$
\begin{array}{rrcl}
 & 0 & \leq & \langle \omega'(x_+) - \omega'(x) + \gamma f'(x), u - x_+\rangle \\
 & & & \hfill [\text{by } (a)] \\
\Rightarrow & 0 & \leq & \langle \omega'(x_+) - \omega'(x) + \gamma f'(x), x - x_+\rangle \\
\Rightarrow & \langle \omega'(x_+) - \omega'(x), x_+ - x\rangle & \leq & \gamma\langle f'(x), x - x_+\rangle \\
\Rightarrow & \alpha\|x_+ - x\|^2 & \leq & \gamma\langle f'(x), x - x_+\rangle \\
\Rightarrow & & & \\
(c) & \|x_+ - x\| & \leq & \alpha^{-1}\gamma\|f'(x)\|_*,
\end{array}
$$

It follows from (b, c) that

$$\gamma(f(x) - f(u)) \leq \gamma\langle f'(x), x - u\rangle \leq [H_u(x) - H_u(x_+)] + \alpha^{-1}\gamma^2\|f'(x)\|_*^2.$$

Recalling that with $x = x_\tau$, $\gamma = \gamma_\tau$ one has $x_+ = x_{\tau+1}$, we see that for $\tau \leq t$ and every $u \in X$ one has

$$\gamma_\tau(f(x_\tau) - f(u)) \leq [H_u(x_\tau) - H_u(x_{\tau+1})] + \alpha^{-1}\gamma_\tau^2\|f'(x_\tau)\|_*^2. \tag{1.16}$$

Specifying u as the minimizer x_* of f on X, summing up the resulting inequalities (1.16) over $\tau = p, p+1, ..., q$ and taking into account that $f(x_\tau) - f(x_*) \geq f(x^t) -$

$f(x_*)$ for $\tau \le t$, we arrive at the relation

$$\begin{aligned}
&\left(\sum_{\tau=p}^{q}\gamma_\tau\right)(f(x^t)-f(x_*))\\
\le\ & H_{x_*}(x_p)-H_{x_*}(x_{q+1})+\alpha^{-1}\sum_{\tau=p}^{q}\gamma_\tau^2\|f'(x_\tau)\|_*^2\\
=\ & \underbrace{[\omega(x_{q+1})+\langle x_*-x_{q+1},\omega'(x_{q+1})\rangle]}_{\le\omega(x_*)}\\
&-[\omega(x_p)+\langle x_*-x_p,\omega'(x_p)\rangle]\\
&+\alpha^{-1}\sum_{\tau=p}^{q}\gamma_\tau^2\|f'(x_\tau)\|_*^2\\
\le\ & \underbrace{[\omega(x_*)-[\omega(x_p)+\langle x_*-x_p,\omega'(x_p)\rangle]]}_{\le\Omega}+\alpha^{-1}\sum_{\tau=p}^{q}\gamma_\tau^2\|f'(x_\tau)\|_*^2
\end{aligned}$$

and (1.13) follows.

(i), (ii) are straightforward corollaries of (1.13). ∎

Implications

Playing with the "setup parameters" $\omega(\cdot)$, $\|\cdot\|$, one can adjust, to some extent, the MD algorithm to the geometry of the problem to be solved. Let us look at three instructive examples:

Ball setup. Here $\omega(x)=\frac{1}{2}\langle x,x\rangle$ and $\|\cdot\|=\|\cdot\|_2$, which results in

$$\alpha=1,\ \Omega=\frac{1}{2}D^2_{\|\cdot\|_2}(X),$$

where $D_{\|\cdot\|_2}(X)=\max\limits_{x,y\in X}\|x-y\|_2$ is the $\|\cdot\|_2$-diameter of X. Theorem 1.1.(ii) reads

$$\begin{gathered}
\gamma_t=\frac{D_{\|\cdot\|_2}(X)}{\|f'(x_t)\|_*\sqrt{2t}}\Rightarrow f(x^t)-\min_X f\le\frac{L_{\|\cdot\|_2}(f)D_{\|\cdot\|_2}(X)}{\sqrt{2t}}\\
\left[L_{\|\cdot\|_2}(f)=\sup_{x\in\mathrm{rint}\,X}\|f'(x)\|_2\right]
\end{gathered}\tag{1.17}$$

which implies the result announced in C.3. Note that with the Ball setup, the MD method becomes exactly the Subgradient Descent method (1.7).

Simplex setup. Here E is the space $\mathbf{R}^n$, $n>1$, with the standard Euclidean structure, X is a convex compact subset of the standard simplex

$$\Delta_n^+=\{x\in\mathbf{R}^n: x\ge 0,\sum_i x_i\le 1\},$$

the distance-generating function is the "regularized entropy"

$$\omega(x) = \sum_{i=1}^{n} (x_i + n^{-1}\sigma) \ln(x_i + n^{-1}\sigma),$$

where the $\sigma \in [10^{-20}, 1]$ is a regularization parameter, and $\|u\| = \|u\|_1 \equiv \sum_i |u_i|$, so that $\|u\|_* = \|u\|_\infty \equiv \max_i |u_i|$. It is easily seen [8] that with this setup one has

$$\alpha = O(1),\ \Omega \le O(1)\ln(n) \tag{1.18}$$

(from now on, all $O(1)$'s are positive absolute constants). With this setup, Theorem 1.1.(ii) reads

$$\gamma_t = \frac{\sqrt{\ln(n)}}{\|f'(x_t)\|_* \sqrt{t}} \Rightarrow f(x^t) - \min_X f \le O(1) \frac{L_{\|cdot\|_1}(f)\sqrt{\ln(n)}}{\sqrt{t}}$$
$$\left[L_{\|\cdot\|_1}(f) = \sup_{x \in \mathrm{rint}\, X} \|f'(x)\|_\infty \right] \tag{1.19}$$

which implies the result announced in C.4.

Spectahedron setup. Here E is the space $\mathbf{S}^n$, $n > 1$, of real symmetric $n \times n$ matrices with the Frobenius inner product $\langle A, B\rangle = \mathrm{Tr}(AB)$, X is a convex compact subset of the *spectahedron*

$$\Sigma_n^+ = \{x \in \mathbf{S}^n : x \succeq 0, \mathrm{Tr}(x) \le 1\},$$

the distance-generating function is the "regularized matrix entropy"

$$\omega(x) = \mathrm{Tr}((x + n^{-1}\sigma I)\ln(x + n^{-1}\sigma I)) = \sum_i (\lambda_i(x) + n^{-1}\sigma)\ln(\lambda_i(x) + n^{-1}\sigma),$$

where the $\lambda_1(x) \ge \lambda_2(x) \ge ... \ge \lambda_n(x)$ are the eigenvalues of x, $\sigma \in [10^{-20}, , 1]$ is a regularization parameter, and $\|u\| = |u|_1 \equiv \sum_i |\lambda_i(u)|$, so that $\|u\|_* = |u|_\infty \equiv \max_i |\lambda_i(u)|$ is the standard matrix norm of u. It can be shown [8] that for this setup relations (1.18) are valid, so that the rate of convergence of the associated MD method satisfies (1.19).

1.3.2 Bundle-Mirror algorithm NERML

The presentation to follows is a slight modification of the original description of the NERML method as given in [8].

Setup for NERML is identical to the setup $\omega(\cdot), \|\cdot\|$ of the general Mirror Descent scheme. And, to make NERML algorithm implementable, the pair $(X, \omega(\cdot))$ should be simple enough to allow for rapid solving of auxiliary problems of the form

$$x[p] = \operatorname*{argmin}_{x \in X}[\omega(x) + p^T x] \tag{1.20}$$

Execution of NERML as applied to (1.1) is partitioned into subsequent *phases*. At the beginning of phase s ($s = 1, 2, ...$) we have in our disposal

- the best found so far, in terms of the objective, feasible solution, let the corresponding objective value be f^s;
- a valid lower bound $f_s < f^s$ on the optimal value f_* in (1.1);
- a *prox-center* $c_s \in X$ (which can be an arbitrary point of X). We associate with this point the distance function

$$\omega_s(x) = \omega(x) - \langle x, \omega'(c_s)\rangle$$

To initiate the very first phase, we choose somehow the first prox-center $c_1 \in X$, compute $f(c_1), f'(c_1)$ and set

$$f^1 = f(c_1), \quad f_1 = \min_{x \in X}[f(c_1) + \langle x - c_1, f'(c_1)\rangle.$$

The outlined data define *s-th level*

$$\ell_s = f_s + \lambda(f^s - f_s),$$

where $\lambda \in (0, 1)$ is a parameter of the method.

Phase s is comprised of subsequent steps; to simplify notation, we mark all entities related to a step by index t of the step, skipping the phase index s.

Step t **of phase** s **is as follows.** At the beginning of step t, we have in our disposal

- t-th *search point* x_t of the phase,
- t-th *model* $F_t(x)$ of the objective, which is a Lipschitz continuous, with constant $L_{\|\cdot\|}(f)$ w.r.t. $\|\cdot\|$, piecewise linear convex function satisfying the relation

$$\forall (x \in X) : f(x) \geq F_t(x); \tag{a_t}$$

- t-th *localizer* X_t – a set cut off X by a system of finitely many linear inequalities and intersecting the relative interior of X;
- t-th *best found value of the objective* $f^{s,t} \le f^s$ – the minimum of values of the objective at the search points of the phases preceding phase t and the search points of phase s preceding x_t;
- t-th *lower bound* $f_{s,t} \ge f_s$ on f_*

The outlined entities satisfy the relations

$$\begin{array}{rl} x_t = \operatorname*{argmin}_{x \in X_t} \omega_s(x) & (b_t) \\ x \in X \backslash X_t \Rightarrow f(x) > \ell_s & (c_t) \end{array}$$

To initialize the first step of phase s, we can set, e.g.,

$$x_1 = c_s, \quad X_1 = X, \quad F_1(x) = f(c_1) + \langle x - x_1, f'(c_1) \rangle, \quad f^{s,1} = f^s, \quad f_{s,1} = f_s,$$

thus ensuring $(a_1 - c_1)$.

Our actions at step t are as follows:

1. [calling oracle, updating the upper bound, enriching the model] We compute $f(x_t)$, $f'(x_t)$ and set
$$f^{s,t+1} = \min[f^{s,t}, f(x_t)].$$
If
$$f^{s,t+1} - \ell_s \le \theta(f^s - \ell_s), \tag{1.21}$$
where $\theta \in (0,1)$ is a parameter of the method, we terminate phase s ("termination due to essential progress in the objective") and pass to phase $s+1$, setting
$$f^{s+1} = f^{s,t+1}, \ f_{s+1} = f_{s,t},$$
otherwise we enrich the model by setting
$$\begin{array}{rcl} g_t(x) & = & f(x_t) + \langle x - x_t, f'(x_t) \rangle, \\ F_t^+(x) & = & \max[F_t(x), g_t(x)]. \end{array}$$

Remark 1.2 *Note that by construction and in view of* (a_t) *the function* $F_t^+(\cdot)$ *is a Lipschitz continuous, with constant* $L_{\|\cdot\|}(f)$ *w.r.t.* $\|\cdot\|$*, piecewise linear convex function satisfying the relation*

$$\forall (x \in X) : f(x) \ge F_t^+(x). \tag{1.22}$$

2. [updating the lower bound] We solve the auxiliary optimization problem

$$\tilde{f}_t = \min_{x \in X_t} F_t^+(x) \qquad (L_t)$$

and set

$$f_{s,t+1} = \max[\min[\ell_s, \tilde{f}_t], f_{s,t}].$$

Remark 1.3 *Note that* $\min[\ell_s, \tilde{f}_t]$ *(and thus – $f_{s,t+1}$) is a lower bound on f_*; indeed, on $X \backslash X_t$ we have $f(x) \geq \ell_t$ by (c_t), while on X_t, by* (1.22)*, we have $f(x) \geq F_t^+(x) \geq \tilde{f}_t$.*

In the case of

$$f_{s,t+1} \geq \ell_s - \theta(\ell_s - f_s) \qquad (1.23)$$

("significant progress in the lower bound") we terminate the phase s and pass to the phase $s+1$, setting

$$f^{s+1} = f^{s,t+1}, \quad f_{s+1} = f_{s,t+1}.$$

3. [updating the search point, the localizer and the model] We solve the auxiliary problem

$$x_{t+1} = \operatorname{argmin} \left\{ \omega_s(x) : x \in X_{t+1}^+ \equiv X_t \cap \{x : F_t^+(x) \leq \ell_s\} \right\}. \qquad (N_t)$$

Remark 1.4 *Note that the feasible set X_{t+1}^+ of (N_t) is cut off X by finitely many linear inequalities (since X_t is so and F_t^+ is piecewise linear) and intersects the relative interior of X (since X_t is so and the minimum of F_t^+ on X_t is $< \ell_s$ – otherwise (1.23) would be satisfied, which is not the case); in particular, (N_t) is solvable, so that the new search point x_{t+1} is well-defined.*

Finally, we

- Choose, as X_{t+1}, any set, cut off X by finitely many linear inequalities, which is in-between the sets X_{t+1}^+ and

$$X_{t+1}^- = \{x \in X : \langle x - x_{t+1}, \omega_s'(x_{t+1}) \rangle \geq 0\},$$

so that

$$X_{t+1}^- \supset X_{t+1} \supset X_{t+1}^+. \qquad (1.24)$$

Remark 1.5 *Note that*

– *the rule makes sense, since $X_{t+1}^- \supset X_{t+1}^+$ due to the fact that x_{t+1} is the minimizer of $\omega_s(\cdot)$ on X_{t+1}^+,*

- X_{t+1}, *by construction, is cut off* X *by finitely many linear inequalities and intersects the relative interior of* X *(since* X^+_{t+1} *is so),*
- x_{t+1}, X_{t+1} *satisfy* (b_{t+1}) *(since* x_{t+1} *is the minimizer of* f *on both* X^+_{t+1} *and* X^-_{t+1}*) and* (c_{t+1}) *(since* $f(x) \geq F^+_t(x) > \ell_s$ *for* $x \in X_t \backslash X^+_{t+1} \supset X_t \backslash X_{t+1}$ *by (1.22), and* $f(x) > \ell_s$ *for* $x \in X \backslash X_t$ *by* (c_t)*).*

- Update the model $F^+_t(\cdot)$ into the model $F_{t+1}(\cdot)$ in a way which ensure that $F_{t+1}(\cdot)$ is a convex piecewise linear Lipschitz continuous, with constant $L_{\|cdot\|}(f)$, w.r.t. $\|\cdot\|$, function satisfying (a_{t+1}).

  Remark 1.6 *To implement the latter rule, we can set* $F_{t+1} \equiv F^+_t$*, or can delete several components from the representation of* F^+_t *as the maximum of finitely many affine functions, or can replace these components with a number of their convex combinations.*

Step t of phase s is completed, and we loop to step $t+1$.

Approximate solution x^s found in course of the first s phases is the best – with the smallest value of the objective – of the search points generated when running the phases $1, ..., s$.

Convergence analysis. We shall introduce here the main result on convergence properties of NERML. In the next sections the reader will find extended versions of this algorithm along with the corresponding convergence analysis, and the proof of the convergence and complexity properties will be carried out there.

Let us define *s-th gap* as the quantity $\epsilon_s = f^s - f_s$ By its origin, the gap is nonnegative, nonincreasing in s, and is a valid upper bound on the inaccuracy, in terms of the objective, of the approximate solution z^s we have at the beginning of phase s (i.e., $f(z^s)$ is the smallest value of the objective found so far).

The convergence and the complexity properties of the basic NERML algorithm are given by the following result.

Theorem 1.2 [8] (i) *The number* N_s *of oracle calls at a phase* s *is bounded from above as follows:*

$$N_s \leq \frac{4\Omega L^2_{\|\cdot\|}(f)}{\theta^2(1-\lambda)^2\alpha\epsilon_s^2}, \tag{1.25}$$

where

$$\Omega = \Omega[\omega(\cdot)] = \max_{x,y \in X}[\omega(y) - \omega(x) - (y-x)^T\nabla\omega(x)]. \tag{1.26}$$

(ii) *Consequently, for every $\epsilon > 0$, the total number of oracle calls before the first phase s with $\epsilon_s \leq \epsilon$ is started (i.e., before an ϵ-solution to the problem is built) does not exceed*

$$N(\epsilon) = c(\theta, \lambda)\frac{\Omega L^2_{\|\cdot\|}(f)}{\alpha\epsilon^2} \tag{1.27}$$

with an appropriate $c(\theta, \lambda)$ depending solely and continuously on $\theta, \lambda \in (0, 1)$.

NERML: Major implementation issues

Solving auxiliary problems ($\mathbf{L}_t$), ($\mathbf{N}_t$). The major issue in the implementation of the NERML algorithm is *how to solve efficiently the auxiliary problems* (L_t), (N_t). Formally, these problems are of the same design dimension as the problem of interest; what then is gained by reducing the solution of a *single* large-scale problem (1.1) to a long series of auxiliary problems of the same dimension? To answer this question, observe that set X_t is something between sets X_- and X^+, where X_- is cut off X by finite number of linear inequalities, and X^+ is cut off X by only one linear inequality. So, we a priori can choose integer m and insure X_t to be cut off X by at most m linear inequalities for all t (for instance by combining some inequalities in a convex manner). Consequently, we may assume that *the feasible set of* (P_t) *is cut off X by $m + 1$ linear inequalities.*

The crucial point is that with this approach, *applying Lagrange duality, we can reduce* (L_t), (N_t) *to black-box-represented convex programs with at most $m + 1$ decision variables* and thus can solve them at a relatively low computational cost, provided that m (which is in our full control) is not too big (for details, see [8]).

When the standard setups are implementable? As we have seen, the possibility to implement the NERML algorithm depends on the ability to solve rapidly optimization problems of the form (1.20). Let us look at several important cases when this indeed is possible.

Ball setup. Here problem (1.20) becomes $\min\limits_{x\in X}\left[\frac{1}{2}x^Tx - p^Tx\right]$, or, equivalently, $\min\limits_{s\in X}\left[\frac{1}{2}\|x - p\|_2^2\right]$. We see that to solve (1.20) is the same as to *project* on X - to find the point in X which is as close as possible, in the usual $\|\cdot\|_2$-norm, to a given point p. This problem is easy to solve for several simple solids X, e.g.,

- a ball $\{x : \|x - a\|_2 \leq r\}$,
- a box $\{x : a \leq x \leq b\}$,
- the simplex $\Delta_n = \{x : x \geq 0, \sum\limits_i x_i = 1\}$.

In all these cases, it takes $O(n)$ operations to compute the solution.

Simplex setup. Consider the two simplest cases:
S.A: X is the standard simplex Δ_n;
S.B: X is the standard full-dimensional simplex Δ_n^+.
Case S.A. When $X = \Delta_n$, problem (1.20) becomes

$$\min\left\{\sum_i (x_i+\sigma)\ln(x_i+\sigma) - p^T x : x \geq 0, \sum_i x_i = 1\right\} \qquad [\sigma = \sigma n^{-1}] \qquad (1.28)$$

Case S.B. Analogously, when $X = \Delta_n^+$, problem (1.20) becomes

$$\min\left\{\sum_i (x_i+\sigma)\ln(x_i+\sigma) - p^T x : x \geq 0, \sum_i x_i \leq 1\right\} \qquad [\sigma = \sigma n^{-1}] \qquad (1.29)$$

It is easy to verify (see [8]) that the solutions to (1.28), (1.29) can be found, within machine precision, in $O(n)$ operations.

Spectahedron setup. Consider two simple cases of the spectahedron setup:
Sp.A: X is comprised of all block-diagonal matrices of a given block-diagonal structure belonging to Σ_n,
or
Sp.B: X is comprised of all block-diagonal matrices of a given block-diagonal structure belonging to Σ_n^+.
Case Sp.A. Here problem (1.20) becomes

$$\min_{x\in X}\left\{\mathrm{Tr}((x+\sigma I_n)\ln(x+\sigma I_n)) + \mathrm{Tr}(px)\right\} \qquad [\sigma = \sigma n^{-1}].$$

It turns out that we can convert this problem to

$$\min_{\xi\in X}\left\{\mathrm{Tr}((\xi+\sigma I_n)\ln(\xi+\sigma I_n)) + \mathrm{Tr}(\pi\xi)\right\}, \qquad (1.30)$$

where $p = U\pi U^T$ is the eigenvalue decomposition of p with orthogonal U and diagonal π of the same block-diagonal structure as that of p and $x = U\xi U^T$.

It is shown in [8] that the unique (due to strong convexity of the function ω) optimal solution ξ^* to the latter problem is a diagonal matrix. Thus, when solving (1.30), we may from the very beginning restrict ourselves with diagonal ξ, and with this restriction the problem becomes

$$\min_{\xi\in\mathbf{R}^n}\left\{\sum_i (\xi_i+\sigma)\ln(\xi_i+\sigma) + \pi^T\xi : \xi \geq 0, \sum_i \xi_i = 1\right\}, \qquad (1.31)$$

which is exactly the the same problem as in the case of the simplex setup with $X = \Delta_n$. We see that the only extra work needed in the case of the spectahedron setup, as compared to the simplex one, is in the necessity to find the eigenvalue decomposition of p. The latter task is easy, provided that the diagonal blocks in the matrices in question are of small sizes. Note that this favourable situation does occur in several important applications, e.g., in Shape Design.
Case Sp.B is completely similar to the previous one; the only difference is that the role of (1.30) is now played by the problem

$$\min_{\xi\in\mathbf{R}^n}\left\{\sum_i(\xi_i+\sigma)\ln(\xi_i+\sigma)+\pi^T\xi : \xi\geq 0, \sum_i \xi_i \leq 1\right\},$$

which we have already considered discussing the simplex setup.

Updating prox-centers. The complexity results stated in Theorem 1.2 are independent of how the prox-centers are updated, so that in this respect one, *in principle*, is completely free. It is reasonable, however, to choose as the prox-center at every stage the best (with the smallest value of f) solution obtained up to the current stage.

Accumulating information. The set X_t summarizes, in a sense, all the information on f accumulated so far and to be used in the sequel. Relation (1.24) allows for a tradeoff between the quality (and the volume) of this information and the computational effort required to solve the auxiliary problems (N_{t-1}). With no restrictions on this effort, the most promising policy for updating X_t's would be to set $X_t = \underline{X}_{t-1}$ ("collecting information without compressing it"). With this policy the NERML algorithm *with the ball setup* is basically identical to the *Prox-Level Algorithm* of Lemarechal, Nemirovski and Nesterov [20]; the "restricted memory" version of the latter method (that is, the generic NERML algorithm with ball setup) was proposed by Kiwiel [16].

1.4 Overview of results

In our work we develop Incremental implementation of NERML aimed to solve optimization problems of minimizing composed nondecreasing convex function of several convex functions. This incremental implementation is based on the idea to divide the set of inner functions into subsets and process these subsets incrementally, one at a time. The algorithm is arranged in cycles: to perform one cycle, it goes through all the subsets. Thus, finishing one cycle is equivalent to performing one "regular"

iteration. To the best of our knowledge, till now, incremental algorithms were used only for processing objectives with simple additive structure, while NERML allowed to handle pretty general objective frames.

We also broaden Non-Euclidean Restricted Memory Level Method to solve problems with functional constraints. Our implementation is rest on the reformulation of original task as a set of non-constrained problems of minimizing composed convex function. The resulting algorithm suits well to work in conjunction with Incremental NERML too.

For both Incremental and Constrained algorithms, we prove convergence and estimate efficiency.

In the next two chapters we present our results.

Chapter 2

Incremental implementation of NERML

In real life applications the objective function of problem (1.1) possesses certain specific structure. For instance, in Tomography Image Reconstruction, the objective is the sum of relatively simple functions, and the number of these functions is very large. In Shape Design, the objective function takes the form of the maximum of several other functions. Thus, it makes sense to consider optimization problems of the type

$$\min_x f(x), \quad f(x) = \mathcal{F}(f_1(x), \ldots, f_m(x)), \tag{2.1}$$

where $\mathcal{F}(\cdot)$ is a known in advance nondecreasing Lipschitz continuous convex function and $f_1(x), \ldots, f_m(x)$ are convex and Lipschitz continuous.

Following the basic NERML scheme, to carry out a step of the algorithm, we need to get from the oracle the value and a subgradient of the function $f(\cdot) = \mathcal{F}(f_1(\cdot), \ldots, f_m(\cdot))$ at certain point x, and this answer requires computing the values and subgradients of all "inner" functions f_i at x. *In principle*, there exists another option – at every step, the oracle is requested to provide the value and a subgradient of a *single* "component" f_i of f at a current point. This "incremental" implementation of optimization algorithms goes back to D. Bertsekas [9, 10, 11]), and rationale behind the idea is as follows: in many cases, the computational effort required to provide the first order information on a single component of the objective is nearly m times cheaper than to provide this information for the entire objective. When speaking about a "cheap" optimization method (with $O(n)$ operations per step, modulo the computational expenses of the oracle), this implies that a usual – "full" – iteration is equivalent, in terms of the computational effort, to m "incremental" iterations. At the same time, computational experience demonstrates that progress in accuracy, especially at the initial steps, with m incremental iterations is significantly better than with a single "full" iteration, so that an "incremental

implementation" improves the practical performance of the algorithm. It should be stressed that, to the best of our knowledge, the only structure of f which, for the time being, was considered as appropriate for an incremental implementation, is the additive structure: $f(x) = f_1(x) + ... + f_m(x)$.

In course of our research, we have found that *the NERML method allows for incremental implementation*, and, moreover, *this implementation can handle pretty general structures of f, not only the simplest additive structure*. Besides this, we have found that under mild additional restrictions on $\mathcal{F}$ (specifically, in the case of *polyhedrally representable* $\mathcal{F}$), we are able to solve easily the auxiliary problems arising in the resulting method.

So, in the next subsections we shall provide an essential information about polyhedral sets, the incremental version of NERML (INERML), its implementation and convergence analysis.

2.1 Polyhedral sets and their polyhedral representations

Recall that a set $M \subset \mathbf{R}^n$ is called *polyhedral* if it is the set of all solutions to a system of finitely many linear inequalities:

$$M = \{x \in \mathbf{R}^n : \mathcal{A}x - b \geq 0, \mathcal{A}_{m\times n}, b_{m\times 1}\}. \tag{2.2}$$

It is well-known that the image, under an affine mapping, of a polyhedral set is polyhedral as well. It follows that a set given as

$$M = \left\{x \in \mathbf{R}^n : \exists u \in \mathbf{R}^m : \mathcal{A}\begin{pmatrix} w \\ z \end{pmatrix} - b \geq 0\right\} \tag{2.3}$$

is polyhedral. Representation (2.3) is called a *polyhedral representation* of M.

Remark 2.1 *The "algorithmic advantage" of "advanced" representations* (2.3) *as compared with "straightforward" representations* (2.2) *is that a polyhedral set with a fairly complicated straightforward representation can admit a pretty simple advanced one. For example, the straightforward polyhedral representation of the hyperoctahedron $M = \{x \in \mathbf{R}^n : \sum\limits_i |x_i| \leq 1\}$ requires 2^n linear inequalities, while the advanced representation*

$$M = \left\{x \in \mathbf{R}^n : \exists t \in \mathbf{R}^n : -t_i \leq x_i \leq t_i, \sum_i t_i \leq 1\right\}$$

requires just $2n + 1$ linear inequalities.

Definition 2.1 *A function $\mathcal{F}(y)$ is called polyhedrally representable, if the epigraph* $\mathrm{Epi}(\mathcal{F}) \equiv \{(y,t) : \mathcal{F}(y) \le t\}$ *of $\mathcal{F}$ is polyhedral. A polyhedral representation of* $\mathrm{Epi}(\mathcal{F})$ *is called a polyhedral representation of $\mathcal{F}$.*

Note that a polyhedral representation of a function $\mathcal{F}$ always can be written as

$$\{(y,t) : t \ge \mathcal{F}(y)\} = \{(y,t) : \exists z : Ay + pt + Bz - b \ge 0\}, \tag{2.4}$$

where the matrices A, B and the vectors p, b are the "data" of the representation. In the sequel, we will use exactly polyhedral representations in the form of (2.4).

Calculus of polyhedrally representable functions. Polyhedrally representable functions admit a kind of "calculus"; in fact, all of them can be obtained from a single "generic" example – an *affine function*

$h(y) = a^T y + c$ [1] by applying operations preserving polyhedral representability. The most frequently used operations of the latter type are as follows:

1. Taking *maximum* of a finite set functions:

$$\left\{ \begin{array}{l} \mathrm{Epi}\{f_i\} = \{(y,t) : \exists z_i : A_i y + p_i t + B_i z_i - b_i \ge 0\},\ i = 1,...,m \\ f = \max\limits_i f_i \end{array} \right.$$
$$\Downarrow$$
$$\mathrm{Epi}\{f\} = \{(y,t) : \exists \{z_i\} : A_i y + p_i t + B_i z_i - b_i \ge 0,\ i = 1,...,m\}$$

2. Taking *linear combination with nonnegative coefficients:*

$$\left\{ \begin{array}{l} \mathrm{Epi}\{f_i\} = \{(y,t_i) : \exists z_i : A_i y + p_i t_i + B_i z_i - b_i \ge 0\},\ i = 1,...,m \\ f = \sum\limits_i \alpha_i f_i \end{array} \right.$$
$$\Downarrow$$
$$\mathrm{Epi}\{f\} = \left\{ (y,t) : \exists \{z_i, t_i\} : \begin{array}{l} A_i y + p_i t_i + B_i z_i - b_i \ge 0,\ i = 1,...,m, \\ t - \sum\limits_i \alpha_i t_i \ge 0 \end{array} \right\}$$

3. *Affine substitution of argument:*

$$\mathrm{Epi}\{F\} = \{(x,t) : \exists z : Ax + pt + Bz - b \ge 0\},\ f(y) = F(Qy + q)$$
$$\Downarrow$$
$$\mathrm{Epi}\{f\} = \{(y,t) : \exists z : A(Qy + q) + pt + Bz - b \ge 0\}$$

[1] An affine function is polyhedrally representable by trivial reasons – its epigraph is given by a single linear inequality. Formally speaking,

$$\mathrm{Epi}(h) = \{(y,t) : Ay + pt + b \ge 0\}, \quad A = -a^T, p = 1, b = c\}.$$

4. *Partial minimization:* assuming that $\min\limits_{x:Qx=y}$ below is always achieved, one has

$$\begin{array}{c}\mathrm{Epi}\{F\} = \{(x,t) : \exists z : Ax + pt + Bz - b \geq 0\},\ f(y) = \min\limits_{x:Qx=y} F(x) \\ \Downarrow \\ \mathrm{Epi}\{f\} = \{(y,t) : \exists (x,z) : Ax + pt + Bz - b, Qx - y \geq 0, -Qx + y \geq 0\}\end{array}$$

5. *Superposition:* let $\mathcal{F} : \mathbf{R}^m \to \mathbf{R} \cup \{+\infty\}$ be monotone and polyhedrally representable, and let $f_i : \mathbf{R}^n \to \mathbf{R}$, $i = 1, ..., m$, be polyhedrally representable. Then the superposition

$$f(x) = \begin{cases} \mathcal{F}(f_1(x), ..., f_m(x)), & f_i(x) < \infty, i = 1, ..., m \\ +\infty, & \text{otherwise} \end{cases}$$

is polyhedrally representable:

$$\begin{array}{c}\left\{\begin{array}{l}\mathrm{Epi}\{\mathcal{F}\} = \{(y,t) : \exists z : Ay + pt + Bz - b \geq 0\} \\ \mathrm{Epi}\{f_i\} = \{(x,t_i) : \exists z_i : A_i x + p_i t_i + B_i z_i - b_i \geq 0\}\end{array}\right. \\ \Downarrow \\ \mathrm{Epi}\{f\} = \left\{(x,t) : \exists (y,z,\{z_i,t_i\}) : \begin{array}{r} A_i x + p_i t_i + B_i z_i - b_i \geq 0, \\ i = 1, ..., m \\ Ay + pt + Bz - b \geq 0, \\ t_i \leq y_i,\ i = 1, ..., m \end{array}\right\}\end{array}$$

For more information on polyhedral sets and polyhedrally representable functions, see [32], [33].

2.2 INERML algorithm

Suppose we have to solve the problem:

$$\min_{x \in X} f(x), \quad f(x) = \mathcal{F}(f_1(x), f_2(x), \ldots, f_m(x)) \tag{2.5}$$

where $X \subset \mathbf{R}^n$ is a convex compact set, $f_i(x)$ is convex and Lipschitz continuous for all $i = 1, \ldots, m$ and $\mathcal{F}(\cdot)$ is a Lipschitz continuous polyhedrally representable and monotone function.

Setup for the Incremental NERML (INERML) scheme, similarly to the general NERML scheme (see Section 1.3.2), is given by a strongly convex C^1 function $\omega(\cdot)$ on X and a norm $\|\cdot\|$ on $\mathbf{R}^n$.

The information. We assume that we have access to the First Order oracle which, given on input an index $i \in \{1, ..., m\}$ and a point $x \in X$, returns the value $f_i(x)$ and a subgradient $g_i(x) \equiv f_i'(x)$ of f_i at x. We assume that the oracle satisfies Assumption A (see the beginning of Section 1.3.1), so that

$$\|g_i(x)\|_* \leq L_{\|\cdot\|}(f_i), \tag{2.6}$$

where $L_{\|\cdot\|}(f_i)$ is the Lipschitz constant of f_i.

Let C_i be Lipschitz constant of $\mathcal{F}(u)$ taken with respect to i-th argument:

$$C_i = \sup_{t>0,u} t^{-1}|\mathcal{F}(u + te_i) - \mathcal{F}(u)|,$$

where e_i is i-th standard basic orth in $\mathbf{R}^m$. We set

$$L(f) = \sum_{i=1}^{m} C_i L_{\|\cdot\|}(f_i); \tag{2.7}$$

note that f is Lipschitz continuous with constant $L(f)$ w.r.t. $\|\cdot\|$.

The idea of incremental implementation is inspired by the fact that NERML admits significant freedom in choosing the models F_t of the objective f: the model should be convex Lipschitz continuous (with an independent of t constant) piecewise linear function which is a minorant of f (see (a_t)). In the case of "structured" objective (2.5), it is natural to build such a model in the form

$$F_t(x) = \mathcal{F}(F_{1,t}(x), ..., F_{m,t}(x)), \tag{2.8}$$

where $F_{i,t}(\cdot)$ is a model of $f_i(\cdot)$, i.e., a piecewise linear Lipschitz continuous, with constant $L_{\|\cdot\|}(f_i)$ w.r.t. the norm $\|\cdot\|$, convex function of x which satisfies the relation

$$F_{i,t}(x) \leq f_i(x),\ x \in X. \tag{2.9}$$

Note that since $\mathcal{F}$ is polyhedrally representable (and thus piecewise linear) and monotone, relation (2.8) indeed defines a model of f, provided that $F_{i,t}$ are models of the components f_i of f.

With the just outlined approach, the policy of building models for f reduces to policies of building models of the components f_i. These latter policies can be implemented in the same manner as in the basic NERML. Namely, whenever the First Order oracle reports information $f_i(u), g_i(u)$ on f_i at a point $u \in X$, we get an affine function

$$h_i^u(x) = f_i(u) + \langle g_i(u), x - u\rangle$$

which is Lipschitz continuous, with constant $L_{\|\cdot\|}(f_i)$ w.r.t. $\|\cdot\|$, minorant of $f_i(x)$; taking maximum of (convex combinations of) these minorants obtained so far, or of a part of these minorants, we get a model of f_i. The main observation underlying the incremental implementation of NERML is that *with the outlined approach, there is no necessity to update at every step all the models of f_i; we could update these models one at a time (say, in cyclic order), which allows at every step to ask the First Order oracle on a single one of the m components f_i of f rather than to ask the oracle at every step on all m components.*

The implementation of the outlined idea is as follows.

Execution of INERML as applied to (2.5) is partitioned into subsequent *phases*. At the beginning of phase s ($s = 1, 2, ...$) we have in our disposal

- the best found so far, in terms of the objective, feasible solution, let the corresponding objective value be f^s;
- a valid lower bound $f_s < f^s$ on the optimal value f_* in (2.5);
- a *prox-center* $c_s \in X$ (which can be an arbitrary point of X). We associate with this point the distance function

$$\omega_s(x) = \omega(x) - \langle x, \omega'(c_s)\rangle$$

To initiate the very first phase, we choose somehow the first prox-center $c_1 \in X$, compute $f_i(c_1), g_i(c_1)$, $i = 1, ..., m$, and set

$$\begin{array}{rcl} F^1(x) & = & \mathcal{F}(h_1^{c_1}(x), ..., h_m^{c_1}(x)), \\ f^1 & = & f(c_1)\ [= F^1(c_1)], \\ f_1 & = & \min\limits_{x\in X} F^1(x). \end{array} \tag{2.10}$$

The outlined data define *s-th level*

$$\ell_s = f_s + \lambda(f^s - f_s),$$

where $\lambda \in (0, 1)$ is a parameter of the method.

Phase s is comprised of subsequent steps; to simplify notation, we mark all entities related to a step by index t of the step, skipping the phase index s.

Step t of phase s is as follows. At the beginning of step t, we have in our disposal

- t-th *search point* x_t of the phase,
- t-th *models* $F_{i,t}(x)$ of the components f_i of the objective, which are Lipschitz continuous, with constants $L_{\|\cdot\|}(f_i)$ w.r.t. $\|\cdot\|$, piecewise linear convex functions satisfying the relation

$$\forall(x \in X) : f_i(x) \geq F_{i,t}(x). \qquad (a_{i,t})$$

  These models, according to (2.8), define the current model

$$F_t(x) = \mathcal{F}(F_{1,t}(x), ..., F_{m,t}(x))$$

  of f;
- t-th *localizer* X_t – a set cut off X by a system of finitely many linear inequalities and intersecting the relative interior of X;
- t-th lower bounds $L_{i,t}$ on the quantities $L_{\|\cdot\|}(f_i)$;
- t-th *lower bound* $f_{s,t} \geq f_s$ on f_*

The outlined entities satisfy the relations

$$x_t = \operatorname*{argmin}_{x \in X_t} \omega_s(x) \qquad (b_t)$$
$$x \in X \backslash X_t \Rightarrow f(x) > \ell_s \quad (c_t)$$

To initialize the first step of phase s, we can set, e.g.,

$$\begin{array}{rcl} x_1 & = & c_s, \\ F_{i,1}(x) & = & h_i^{c_1}(x),\ i = 1, ..., m, \\ X_1 & = & X, \\ L_{i,1} & = & \|g_i(c_1)\|_*,\ i = 1, ..., m, \\ f_{s,1} & = & f_s, \end{array}$$

thus ensuring $(a_{\cdot,1}, b_1, c_1)$.

Our actions at step t are as follows:

1. [calling oracle and enriching the models] We choose somehow *t-th working set* I_t, which is a nonempty subset of the index set $I = \{1, ..., m\}$, compute $f_i(x_t)$, $g_i(x_t)$, $i \in I_t$, update the bounds on Lipschitz constants according to

$$L_{i,t+1} = \begin{cases} \max[L_{i,t}, \|g_i(x_t)\|_*], & i \in I_t \\ L_{i,t}, & \text{otherwise} \end{cases},$$

enrich the models of f_i by setting

$$F_{i,t}^+(x) = \begin{cases} \max\left[F_{i,t}(x), h_i^{x_t}(x)\right], & i \in I_t \\ F_{i,t}(x), & \text{otherwise} \end{cases}$$

and enrich accordingly the model of f by setting

$$F_t^+(x) = \mathcal{F}(F_{1,t}^+(x), ..., F_{m,t}^+(x)).$$

Then we pass to Progress Check (see below); as a result, we either terminate phase s and get a new best found so far solution x^{s+1} along with the corresponding value of the objective f^{s+1} and an updated lower bound f_{s+1} on f_* (in this case we pass to phase $s+1$) or proceed with step t of phase s.

2. [updating the lower bound] We solve the auxiliary optimization problem

$$\tilde{f}_t = \min_{x \in X_t} F_t^+(x) \qquad (L_t)$$

and set

$$f_{s,t+1} = \max[\min[\ell_s, \tilde{f}_t], f_{s,t}].$$

Remark 2.2 *Note that* $\min[\ell_s, \tilde{f}_t]$ *(and thus – $f_{s,t+1}$) is a lower bound on f_*; indeed, on $X \backslash X_t$ we have $f(x) \geq \ell_t$ by (c_t), while on X_t we have $f(x) \geq F_t^+(x) \geq \tilde{f}_t$.*

In the case of

$$f_{s,t+1} \geq \ell_s - \theta(\ell_s - f_s) \qquad (2.11)$$

("significant progress in the lower bound") we terminate the phase s and pass to the phase $s+1$, setting

$$f^{s+1} = f^s, \quad f_{s+1} = f_{s,t+1}.$$

Here, as in the basic NERML, $\theta \in (0,1)$ is a parameter of the method.

3. [updating the search point, the localizer and the models] We solve the auxiliary problem

$$x_{t+1} = \operatorname{argmin}\left\{\omega_s(x) : x \in X_{t+1}^+ \equiv X_t \cap \{x : F_t^+(x) \leq \ell_s\}\right\}. \qquad (N_t)$$

Finally, we

- Choose, as X_{t+1}, any set, cut off X by finitely many linear inequalities, which is in-between the sets X^+_{t+1} and

$$X^-_{t+1} = \{x \in X : \langle x - x_{t+1}, \omega'_s(x_{t+1})\rangle \geq 0\},$$

so that

$$X^-_{t+1} \supset X_{t+1} \supset X^+_{t+1}. \tag{2.12}$$

- Update the models $F^+_{i,t}(\cdot)$ into the models $F_{i,t+1}(\cdot)$ in a way which ensure that
  (a) $F_{i,t+1}(\cdot)$ is a convex piecewise linear Lipschitz continuous, with constant $L_{\|\cdot\|}(f_i)$, w.r.t. $\|\cdot\|$, function satisfying $(a_{i,t+1})$ (cf. Remark 1.6);
  (b) One has

$$i \in I_t \Rightarrow F_{i,t+1}(x_t) = F^+_{i,t}(x_t) \qquad [= f_i(x_t)] \tag{2.13}$$

Step t of phase s is completed, and we loop to step $t+1$.

Progress Check. From now on we assume that the policy for handling the working sets I_t satisfies the following requirement:

> (W) *For certain integer $k \leq m$, the union of the working sets associated with k subsequent steps of a phase is the entire index set $I = \{1, ..., m\}$.*

Requirement (W) is satisfied, e.g., by the following two natural policies:

- *trivial policy* $I_t = I$ (here $k = 1$)
- *cyclic policy*, where I_1, I_2,... are singletons, and the corresponding indices are chosen in I in the cyclic order (here $k = m$).

Now we are ready to describe the Progress Check. Informally, the goal of this procedure is to find out whether a "significant progress in the objective" is achieved (cf. the basic NERML). The implementation is as follows. Consider step t of phase s. It may happen that the union of the working sets associated with step t and all preceding steps of the phase is less than I; in this case, the Progress Check reports to the calling algorithm that the phase should *not* be terminated at step t. Now consider the situation when the union of the working sets associated with the steps $1, ..., t$ of the phase equals I. In this case, we can find the shortest segments of steps $\underline{t}, \underline{t}+1, ..., t$ such that the union of the associated working sets equals I; let us call this segment the *major iteration* associated with step t, and let $\mathcal{J}_t = \{\underline{t}, \underline{t}+1, ..., t\}$. The Progress Check works as follows:

1. For every $\tau \in \mathcal{J}_t$, we build a *guess* $f_i^{\tau,t}$ for the quantity $f_i(x_t)$, $i \in I_\tau$, according to
$$f_i^{\tau,t} = f_i(x_\tau) + L_{i,t}\|x_t - x_\tau\| \tag{2.14}$$
(note that we do know $f_i(x_\tau)$ due to $i \in I_\tau$).

2. Since $\bigcup_{\tau\in\mathcal{J}_t} I_\tau = I$, we, for every $i \in I$, get at least one (or perhaps more) guesses for the quantity $f_i(x_t)$. From these guesses, we choose the smallest one, let it be denoted by f_i^t.

3. The guesses f_i^t imply the guess
$$f^t = \mathcal{F}(f_1^t, ..., f_m^t)$$
for the quantity $f(x_t)$. We compare this guess with ℓ_s, namely

   (a) *In the case of* $f^t - \ell_s > \theta(f^s - \ell_s)$, we conclude that no significant progress in the objective is achieved, so that the phase s should not be terminated.

   (b) *In the case of* $f^t - \ell_s > \theta(f^s - \ell_s)$, we call the First Order oracle to compute all the quantities $f_i(x_t)$, $i = 1, ..., m$, and thus get the exact value $f(x_t)$ of the objective at x_t.

   (b.i.) *In the case of* $f(x_t) - \ell_s > \theta(f^s - \ell_s)$, we again conclude that no significant progress in the objective is achieved, so that the phase s should not be terminated.

   (b.ii.) *In the case of* $f(x_t) - \ell_s \le \theta(f^s - \ell_s)$, we conclude that a significant progress in the objective is achieved, set $x^{s+1} = x_t$, $f^{s+1} = f(x_t)$, $f_{s+1} = f_{s,t}$ and terminate phase s.

2.3 Implementation

Now we have to discuss how to solve arising in the described above scheme minimization problems. Namely, we have got two types of problems:

$$\begin{array}{ll} (a) & \min\limits_{x\in X_\tau} F_\tau^+(x), \quad F_\tau^+(x) = \mathcal{F}(T_{1,t}^+(x), ..., T_{m,t}^+(x)); \\ (b) & \min\limits_{x\in X_\tau} \{\omega_s(x) : F_\tau^+(x) \le \ell_s\}. \end{array} \tag{2.15}$$

Problem (2.15.a). 2.15 Recall that X_τ is cut off X by a system of m_t linear inequalities. Further, the function $\mathcal{F}(\cdot)$ is polyhedrally representable and monotone, while the models $F_{i,t}^+$ are polyhedrally representable. Consequently, $F_\tau^+(x)$ is polyhedrally representable (rule on Superposition in Section 2.1):

$$t \ge F_\tau^+(x) \Leftrightarrow \exists u : Ax + pt + Bu - b \ge 0.$$

Thus, problem (2.15.a) is of the form

$$\tilde{f}_\tau = \min_{x\in X,t,u} \{t : Qx + qt + Ru - r \geq 0\}. \tag{2.16}$$

Note that the system of linear constraints in the latter problem is readily given by the system of linear inequalities from the description of X_τ and the system of linear inequalities involved in the polyhedral representation of F_t^+. The latter inequalities, in turn, are readily given by the initial polyhedral representation of $\mathcal{F}$ and the affine functions involved into the description of the models $F_{i,t}^+$. Besides this, the number of linear inequalities in (2.16) is small, provided that both m_t and the total number n_t of "linear pieces" in the description of the models $F_{i,t}^+$, $i = 1, ..., m$ are small *and* $\mathcal{F}$ has a simple polyhedral representation. Note that both m_t and n_t are under our control (m_t can be made as small as 1, and n_t as small as m); as a result, the number of linear inequality constraints in (2.16) can be made as small as $\pi = m + 1 + \mu$, where μ is the number of linear inequalities in the original polyhedral representation of $\mathcal{F}$. Provided (which normally is the case) that π is a moderate number (at most several tens), a computationally efficient way to solve (2.16) is offered by Lagrange duality. Specifically, let

$$\mathcal{L}(x, t, u; \lambda) = t - \lambda^T(Qx + qt + Ru - r)$$

be the Lagrange function of problem (2.16). By the origin of the problem, the set defined by its linear constraints intersects the relative interior of X and the problem is below bounded, so that by Lagrange Duality Theorem we have

$$\tilde{f}_\tau = \max_{\lambda\geq 0} \Phi(\lambda), \quad \Phi(\lambda) = \min_{x\in X,t,u} \mathcal{L}(x, t, u; \lambda), \tag{2.17}$$

or, equivalently,

$$\tilde{f}_\tau = \max \left\{ \underbrace{\min_{x\in X}[-\lambda^T Qx] + r^T\lambda}_{\Psi(\lambda)} : \lambda \geq 0, q^T\lambda = 1, R^T\lambda = 0 \right\}. \tag{2.18}$$

Since we have assumed that X and $\omega(\cdot)$ are simple, so that minimizing functions of the form $\omega(x) + p^T x$ over X (and thus – minimizing linear functions over X) is easy, the objective in the convex problem (2.18) can be easily computed at every point; to this end it suffices to find a minimizer, over X, of the linear function $\lambda^T Qx$ of x. Given this minimizer, we can easily recover the value and a subgradient of Ψ. Finally, the number π of variables in (2.18) was assumed to be moderate, so that the problem can be rapidly solved by efficient black-box-oriented techniques for *low-dimensional* convex optimization, like bundle algorithms.

Problem (2.15.b). Here the situation is completely similar to the one we have just considered. Indeed, the same arguments as above demonstrate that (2.15.b) can be rewritten equivalently as

$$\min_{x\in X,t,u}\left\{\omega(x)+p^Tx : Qx+qt+Ru-r\geq 0\right\}, \tag{2.19}$$

(the data of linear constraints now are different from those in (2.16), but the number π of these constraints is the same as in (2.16)). Further, we should solve our problem only when the optimal value in (L_τ) is $< \ell_s$, and in this case, as it is immediately seen, the set of solutions to the system of linear inequalities in (2.19) intersects the relative interior of X. Thus, we again can apply Lagrange duality to reduce the problem of interest to a low-dimensional problem

$$\tilde{f}_\tau = \max\left\{\underbrace{\min_{x\in X}[\omega(x)+p^Tx-\lambda^TQx]+r^T\lambda}_{\Psi(\lambda)} : \lambda\geq 0, q^T\lambda=1, R^T\lambda=0\right\}. \tag{2.20}$$

Here again it is easy to compute the value and a subgradient of the objective at a given point, which allows to solve the problem by bundle algorithms. Note that since $\omega(x)$ is strongly convex, a high-accuracy, in terms of the objective, solution to (2.20) allows to recover a high-accuracy approximation to the optimal solution of the problem of interest (2.19), which is exactly what we need[2)]

2.4 Convergence analysis

Theorem 2.1 *Let $\epsilon_s = f^s - f_s$ be the upper bound on inaccuracy of s-th approximate solution generated by* INERML. *Then*

(i) *The number of steps N_s at phase s of the method is bounded from above as follows:*

$$N_s \equiv k\rfloor\frac{8\Omega[\omega(\cdot)]k^2L^2(f)}{\theta^2(1-\lambda)^2\alpha\epsilon_s^2}+1\lfloor, \tag{2.21}$$

where $\Omega[\omega(\cdot)]$ is given by (1.26) *and $k\leq m$ is the maximum number of steps in a major iteration. As a result of phase, the gap is reduced at least by the factor*

$$\gamma = \max[\lambda+\theta(1-\lambda),(1-\lambda)+\theta\lambda] < 1,$$

[2)] In the case of (2.16), the objective in the problem is linear, so that it could be problematic to recover a good approximate solution to this problem from a good approximate solution to its Lagrange dual. Note, however, that in INERML we are not interested in optimal solutions to problems (L_t) at all, all we are interested in are the optimal values.

that is,

$$\epsilon_{s+1} \leq \gamma \epsilon_s. \tag{2.22}$$

(ii) *Consequently, for every $\epsilon > 0$, the total number of oracle calls before the first phase s with $\epsilon_s \leq \epsilon$ is started (i.e., before an ϵ-solution to the problem is built) does not exceed*

$$N(\epsilon) \leq C(\theta, \lambda) k^3 \frac{\Omega[\omega(\cdot)] L^2(f)}{\alpha \epsilon^2}, \tag{2.23}$$

with $C(\epsilon, \lambda)$ depending solely on $\epsilon, \lambda \in (0, 1)$.

Proof. Part (i): Assume that phase s did not terminate at step t, so that the search points $x_1, ..., x_{t+1}$ of the phase are well-defined.

1^0. Let us set

$$d_\tau = \|x_\tau - x_{\tau+1}\|.$$

Lemma 2.1 *One has*

$$\sum_{\tau=1}^{t} \frac{\alpha}{2} d_\tau^2 \leq \Omega[\omega(\cdot)]. \tag{2.24}$$

Proof. Let $1 \leq \tau \leq t$. By (b_τ), the point x_τ minimizes $\omega_s(\cdot)$ over X_τ, and by construction $x_{\tau+1} \in X_\tau$, whence $\langle \omega_s'(x_\tau), x_{\tau+1} - x_\tau \rangle \geq 0$. Since ω_s is α-strongly convex w.r.t. $\|\cdot\|$, we have

$$\omega_s(x_{\tau+1}) \geq \omega_s(x_\tau) + \underbrace{\langle \omega_s'(x_\tau), x_{\tau+1} - x_\tau \rangle}_{\geq 0} + \frac{\alpha}{2} \|x_\tau - x_{\tau+1}\|^2 \geq \omega_s(x_\tau) + \frac{\alpha}{2} d_\tau^2.$$

Summing up the resulting inequalities over $\tau = 1, ..., t$, we get

$$\begin{array}{rcl} \sum_{\tau=1}^{t} \frac{\alpha}{2} d_\tau^2 & \leq & \omega_s(x_{t+1}) - \omega_s(x_1) \\ & = & \omega(x_{t+1}) - \omega(x_1) - \langle \omega'(c_s), x_{t+1} - x_1 \rangle \\ & = & \underbrace{\omega(x_{t+1}) - \omega(c_s) - \langle \omega'(c_s), x_{t+1} - c_s \rangle}_{\leq \Omega[\omega(\cdot)]} \\ & & - \underbrace{[\omega(x_1) - \omega(c_s) - \langle \omega'(c_s), x_1 - c_s \rangle]}_{\geq 0} \\ & \leq & \Omega[\omega(\cdot)], \end{array}$$

as claimed in (2.24). ∎

2^0. Assume that $t \geq k$, so that $\bigcup_{i=1}^{k} I_{t-i+1} = I \equiv \{1, ..., m\}$.

Lemma 2.2 *Assuming that phase s does not terminate at step t, one has*

$$\max_{t-k+1 \leq \tau \leq t} d_\tau \geq \nu \equiv \frac{\theta(f^s - \ell_s)}{2kL(f)} = \frac{\theta(1-\lambda)(f^s - f_s)}{2kL(f)}. \tag{2.25}$$

Proof. Assume, on the contrary to what should be proved, that $d_\tau < \nu$ for $\tau \in \mathcal{T} = \{t-k+1, t-k+2, ..., t\}$. For $i \in I$, let τ_i be the last element τ of $\mathcal{T}$ such that $i \in I_\tau$. Let also

$$\widetilde{F}_i(x) = \begin{cases} F_{i,\tau_i+1}(x), & \tau_i < t \\ F^+_{i,t}(x), & \tau_i = t \end{cases}.$$

Then by construction of the models $F_{i,\tau}$ we have

$$\begin{array}{lrcl} (a) & F^+_{i,t}(x) & \equiv & \widetilde{F}_i(x),\ i \in I, \\ (b) & \widetilde{F}_i(x_{\tau_i}) & = & f_i(x_{\tau_i}),\ i \in I, \\ (c) & \widetilde{F}_i(\cdot) \text{ is Lipschitz continuous w.r.t. } \|\cdot\| \text{ with constant } L_i \equiv L_{\|\cdot\|}(f_i) & & \end{array} \tag{2.26}$$

Combining (2.26.*b-c*) with the fact that $\|x_{\tau_i} - x_{t+1}\| \leq d_{t-k+1} + d_{t-k+2} + ... + d_t \leq k\nu$, we arrive at

$$\widetilde{F}_i(x_{t+1}) \geq f_i(x_{\tau_i}) - k\nu L_i,\ i \in I,$$

whence by the monotonicity and Lipschitz continuity of $\mathcal{F}$ it follows that

$$\begin{array}{rcl} \mathcal{F}(\widetilde{F}_1(x_{t+1}), ..., \widetilde{F}_m(x_{t+1})) & \geq & \mathcal{F}(f_1(x_{\tau_1}), ..., f_m(x_{\tau_m})) - \sum_{i=1}^{m} C_i k\nu L_i \\ & = & \mathcal{F}(f_1(x_{\tau_1}), ..., f_m(x_{\tau_m})) - k\nu L(f) \end{array}$$

(see (2.7)). The left hand side in the resulting inequality, by (2.26.*a*), is nothing but $F^+_t(x_{t+1}) = \mathcal{F}(F^+_{1,t}(x_{t+1}), ..., F^+_{m,t}(x_{t+1}))$, and the latter quantity is $\leq \ell_s$ by definition of x_{t+1}. We have arrived at the inequality

$$\mathcal{F}(f_1(x_{\tau_1}), ..., f_m(x_{\tau_m})) \leq \ell_s + k\nu L(f). \tag{2.27}$$

Further, $\|x_t - x_{\tau_i}\| \leq k\nu$, whence (see Rule 1 in Progress Check)

$$f^t_i \leq f_i^{\tau_i, t} \leq f_i(x_{\tau_i}) + L_{i,t}\|x_t - x_{\tau_i}\| \leq f_i(x_{\tau_i}) + k\nu L_i \tag{2.28}$$

(recall that by construction $L_{i,t} \leq L_i$, see (2.6)). Taking into account once again that $\mathcal{F}$ is monotone and Lipschitz continuous, we conclude from (2.28) and (2.27) that

$$\mathcal{F}(f^t_1, ..., f^t_m) \leq \mathcal{F}(f_1(x_{\tau_1}), ..., f_m(x_{\tau_m})) + k\nu L(f) \leq \ell_s + 2k\nu L(f). \tag{2.29}$$

Since $2k\nu L(f) \le \theta(f^s - \ell_s)$, the guess $f^t = \mathcal{F}(f_1^t, ..., f_m^t)$ is $\le \ell_s + \theta(f^s - \ell_s)$, so that Rule 3(b) in the description of Progress Check was invoked at step t.

We have seen that at step t, Rule 3(b) of Progress Check was invoked. Since $\|x_t - x_{\tau_i}\| \le k\nu$, we have $f_i(x_t) \le f_i(x_{\tau_i}) + k\nu L_i$, which, by monotonicity and Lipschitz continuity of $\mathcal{F}$, implies that

$$\mathcal{F}(f_1(x_t), ..., f_m(x_t)) \le \mathcal{F}(f_1(x_{\tau_1}), ..., f_m(x_{\tau_m})) + k\nu L(f),$$

and the latter quantity is $\le \ell_s + 2k\nu L(f)$ by (2.27). Thus, $\mathcal{F}(f_1(x_t), ..., f_m(x_t)) \le \ell_s + 2k\nu L(f) \le \ell_s + \theta(f^s - \ell_s)$, that is, the Progress Check predicts to terminate phase s at step t, which is not the case (recall that we have assumed that the phase is not terminated at step t). The resulting contradiction completes the proof of Lemma. ∎

3^0. Combining (2.24) with (2.25), we conclude that *the number of steps at phase s does not exceed the quantity*

$$N_s \equiv k\rfloor \frac{8\Omega[\omega(\cdot)]k^2L^2(f)}{\alpha\theta^2(1-\lambda)^2(f^s - f_s)^2} + 1\lfloor,$$

as required in (2.21). Besides this, there are exactly two reasons for terminating phase s:

1. "Significant progress in lower bound" (Rule 3): $f^{s+1} = f^s$, $f_{s+1} \ge \ell_s - \theta(\ell_s - f_s)$. In this case, $\epsilon_{s+1} = f^{s+1} - f_{s+1} \le (f^s - \ell_s) + \theta(\ell_s - f_s) = (1 - \lambda + \theta\lambda)\epsilon_s$ (we have taken into account that $\ell_s = f_s + \lambda(f^s - f_s)$);
2. "Significant progress in objective" (Rule 3.b.ii in Progress Check): $f^{s+1} \le \ell_s + \theta(f^s - \ell_s)$, $f_{s+1} \ge f_s$. In this case, $\epsilon_{s+1} = f^{s+1} - f_{s+1} \le \ell_s + \theta(f^s - \ell_s) - f_s = (\lambda + \theta(1-\lambda))(f^s - f_s) = (\lambda + \theta(1-\lambda))\epsilon_s$.

In both cases, $\epsilon_{s+1} \le \gamma\epsilon_s$, as required in (2.22). (i) is proved.

Proof of (ii): Let us first verify that

$$\epsilon_1 \le L(f)\sqrt{\frac{2\Omega[\omega(\cdot)]}{\alpha}}. \tag{2.30}$$

Indeed, by (2.10) $f^1 = F^1(c_1)$, $f_1 = \min\limits_{x\in X} F^1(x)$, where $F^1(\cdot)$ is a Lipschitz continuous, with constant $L(f)$ w.r.t. the norm $\|\cdot\|$, function on X. It follows that $\epsilon_1 \le L(f)D$, where $D = \max\limits_{x\in X} \|x - c_1\|$. At the same time,

$$\forall(x \in X) : \frac{\alpha}{2}\|x - c_1\|^2 \le \omega(x) - \omega(c_1) - \langle \omega'(c_1), x - c_1\rangle \le \Omega[\omega(\cdot)],$$

whence $D \leq \sqrt{\frac{2\Omega[\omega(\cdot)]}{\alpha}}$, and we arrive at (2.30).

In view of (2.30) and the fact that $\epsilon_s \leq \epsilon_1$ for all s, relation (2.21) can be rewritten as

$$N_s \leq C_1 k^3 \frac{\Omega[\omega(\cdot)]L^2(f)}{\alpha \epsilon_s^2}$$

(from now on, C_i depend solely on θ, λ). Now, in the case of $\epsilon \geq \epsilon_1$, the quantity $N(\epsilon)$ in (ii) clearly equals to 0, so that (2.23) is trivially true. Now let $\epsilon < \epsilon_1$. In view of (2.22) there exists the largest $s = s_*$ such that $\epsilon_s > \epsilon$, and we have

$$\begin{aligned}
N(\epsilon) &= \sum_{s=1}^{s_*} N_s \leq C_1 k^3 \Omega[\omega(\cdot)]L^2(f)\alpha^{-1} \sum_{s=1}^{s_*} \epsilon_s^{-2} \\
&\leq C_2 k^3 \Omega[\omega(\cdot)]L^2(f)\alpha^{-1} \sum_{i=0}^{s_*-s} \gamma^{2i} \epsilon_{s_*}^{-2} \\
&\qquad\qquad [\text{since } \epsilon_{s_*-i} \geq \gamma^{-i}\epsilon_{s_*} \text{ by (2.22)}] \\
&\leq C_3 k^3 \Omega[\omega(\cdot)]L^2(f)\alpha^{-1} \epsilon_{s_*}^{-2} \sum_{i=0}^{\infty} \gamma^{2i} \\
&\leq C_4 k^3 \Omega[\omega(\cdot)]L^2(f)\alpha^{-1} \epsilon_{s_*}^{-2} \\
&\leq C_4 k^3 \Omega[\omega(\cdot)]L^2(f)\alpha^{-1} \epsilon^{-2} \\
&\qquad\qquad [\text{since } \epsilon_{s_*} \geq \epsilon]
\end{aligned}$$

and we arrive at (2.23). ∎

Chapter 3

Problems with functional constrains

Another question which emerged in our research was to extend NERML to the case of convex problem *with functional constraints*:

$$\min_x\{f_0(x) : f_j(x) \leq 0,\ j = 1, ..., m, x \in X\}, \tag{3.1}$$

where X is a "simple" convex compact set and $f, f_1, ..., f_m$ are "black-box represented" convex functions which are Lipschitz continuous on X.

In principle, one could rewrite this problem in the form of

$$\min_x\{f_0(x) : x \in \tilde{X}\},$$

where $\tilde{X} := \{x \in X : f_j(x) \leq 0, j = 1, \ldots, m\}$. But in this case the domain of the problem $\tilde{X}$ may become complicated, while in NERML (and in Mirror Descent in general) the simplicity of the domain is crucial for the possibility to solve the auxiliary problems. Thus, (3.1) needed a dedicated investigation.

• *We did extended NERML to problems with functional constraints, by adapting and extending the approach developed in [20] in the context of Bundle methods.* So, in the next subsections we present our algorithm for Constrained NERML and the corresponding convergence analysis.

3.1 Preliminary remarks and notations

By setting $g(x) := \max_j f_j(x)$ we can reduce the situation to the case of a single functional constraint:

$$\min\{f(x) : g(x) \leq 0, x \in X\}. \tag{3.2}$$

Function $g(x)$ is Lipschitz continuous and convex on X as maximum of such functions. So, we shall concentrate on this latter version of the constrained problem.

We assume that our problem is feasible and that the constraint is meaningful, i.e. there exists a point x in X such that function $g(x)$ is positive.

Now, if we could reformulate our problem in an equivalent form of the type

$$\min\{h(x) : x \in X\},$$

where $h(x)$ is Lipschitz continuous and convex on X, then we could solve it by the basic NERML algorithm. And we really can reformulate it so by defining function $h(x)$ as follows:

$$h(x) := \max\{f(x) - f_*, g(x)\}$$

where f_* is the optimal value of the problem (3.2). Indeed, this function is Lipschitz continuous and convex on X as a maximum of such functions, and the optimal solution of the problem

$$\min\{h(x), x \in X\}, \quad \text{where} \quad h(x) = \max\{f(x) - f_*, g(x)\} \tag{3.3}$$

is the same as an optimal solution x^* of (3.2). (To see this, consider subset G of the set X where function $g(\cdot)$ is not positive: $G := \{x \in X : g(x) \leq 0\}$. Then

$$\min\{h(x), x \in X\} = \min\left[\min\{h(x), x \in G\}, \min\{h(x), x \in X \setminus G\}\right].$$

But on G

$$f(x) - f_* \geq 0 \quad \text{and} \quad f(x^*) - f_* = 0,$$

while on $X \setminus G$ function $g(\cdot)$ is positive, that leads to $h(x) > 0$. So

$$\min\{h(x), x \in X\} = 0 \quad \text{and it is achieved at} \quad x^*.$$

)

However this representation posses some difficulty - we do not know the optimal value of the problem apriori. To overcome this, we can replace this value somehow, say, by parameter r. Now we can define a function $\mathbf{h}(\mathbf{x}, \mathbf{r})$ as follows

$$\mathbf{h}(\mathbf{x}, \mathbf{r}) := \max\{\mathbf{f}(\mathbf{x}) - \mathbf{r}, \mathbf{g}(\mathbf{x})\}.$$

We know that the optimal value of (3.3) is 0. So, our goal is to solve

$$\varepsilon(r) = \min\{\mathbf{h}(\mathbf{x}, \mathbf{r}) : \mathbf{x} \in \mathbf{X}\} = \mathbf{0}. \tag{3.4}$$

Note that function $\varepsilon(\cdot)$ is diminishing for $r \leq f^*$, so if $\varepsilon(r) \leq \epsilon$ for some $r \quad (r \leq f^*)$, then $\varepsilon(f^*) \leq \epsilon$ too. In the light of foregoing, *our plan for solving* (3.4) *may be as follows*:

The solution to the equation (3.4) will be derived in subsequent cycles. For each cycle i we underestimate f^* by r^i using an appropriate models $F^i(x), G^i(x)$ of $f(x)$ and $g(x)$ respectively. These models have to be Lipschitz continuous convex and piecewise linear and to underestimate the target and the constraint functions in the way that ensure

$$r^1 \leq r^2 \leq \ldots \leq f^*.$$

Then we shall solve

$$\min\{\mathbf{h}(\mathbf{x}, \mathbf{r^i}) : \mathbf{x} \in \mathbf{X}\}. \tag{3.5}$$

If $\varepsilon(r^i)$ is greater than 0 significantly, we shall deduce that r^i is pretty rough underestimation for f^* and shall update r^i to r^{i+1} using current approximations $F_i(x), G_i(x)$ of $f(x)$ and $g(x)$.

We will realize this plan with the help of NERML Algorithm.

3.2 Constrained NERML: a description

Setup for Constrained NERML is similar to the general NERML scheme (see Section 1.3.2).

The information. We assume that we have access to the First Order oracle which, given on input a point $x \in X$, returns the values $f(x), g(x)$ and subgradients $f'(x), g'(x)$.

Execution of Constrained NERML as applied to (3.4) is partitioned into subsequent *cycles*.

Cycle i**.** At the beginning of cycle $i (i = 1, 2, \ldots)$ we have in our disposal a valid lower bound r^i on the optimal value f_* in (3.2), and the prox-center c^i.

To initialize the very first cycle we can choose somehow point $c \in X$, compute $f(c), g(c), f'(c), g'(c)$ and set

$$\begin{aligned}
&F^1(x) = f(c) + \langle x - c, f'(c)\rangle;\\
&G^1(x) = g(c) + \langle x - c, g'(c)\rangle;\\
&r^1 = \min\{F^1(x) : G^1(x) \leq 0, x \in X\};\\
&H^1(x, r^1) = \max\{F^1(x) - r^1, G^1(x)\}.
\end{aligned}$$

We start to solve problem (3.5) using regular NERML scheme (see 1.3.2). An initializing of the very first phase of the NERML as applied to the particular cycle i can be done as follows:

The very first phase. Set.

$$c^{i,1} = c^i,$$
$$\delta^{i,1} = \max\{f(c^{i,1}) - r^i, g(c^{i,1})\},$$
$$\delta_1^i = \min\{H^{i,1}(x, r^i), x \in X\}.$$

Pase s of cycle i is initialized by

- a valid lower bound δ_s^i on the quantity

$$\min\{\mathbf{h}(\mathbf{x}, \mathbf{r^i}) : \mathbf{x} \in \mathbf{X}\};$$

- the best found so far feasible solution $\delta^{i,s}$ to

$$\min\{\mathbf{h}(\mathbf{x}, \mathbf{r^i}) : \mathbf{x} \in \mathbf{X}\};$$

- a prox center $c^{i,s} \in X$.

These data define s -th level

$$l_s = (1 - \lambda)\delta_s^i + \lambda\delta^{i,s},$$

where λ is a parameter of the method.

Each phase is comprised of subsequent steps (if possible, we shall skip the phase index s and cycle index i in the notations).

Step t**.** At the beginning of step t we have in our disposal

- t -th search point x_t of the phase;
- t -th model $H_t(x, r^i)$ of the function $\mathbf{h}(\mathbf{x}, \mathbf{r^i})$, which is Lipschitz continuous piecewise linear convex function satisfying the relation

$$\forall x \in X \quad \mathbf{h}(\mathbf{x}, \mathbf{r^i}) \geq \mathbf{H_t}(\mathbf{x}, \mathbf{r^i});$$

- t -th best found value $\delta^{s,t}$ of the $\mathbf{h}(\mathbf{x}, \mathbf{r^i})$;
- t -th localizer X_t.

These entities satisfy the relations

$$x_t = \mathrm{argmin}\{\omega_s(x) : x \in X_t\}$$
$$x \in X \setminus X_t \Rightarrow \mathbf{h}(\mathbf{x}, \mathbf{r^i}) > \mathbf{l_s}.$$

To initialize *the first step* of phase s, we can set

$$\begin{array}{l} x_1 = c_s, \\ X_1 = X, \\ F_1(x) = f(x_1) + \langle x - x_1, f'(x_1)\rangle, \\ G_1(x) = g(x_1) + \langle x - x_1, g'(x_1)\rangle, \\ H_1(x, r^i) = \max\{F_1(x) - r^i, G_1(x)\}, \\ \delta^{s,1} = \delta^s, \\ \delta_{s,1} = \min\{H_1(x), x \in X_1\}. \end{array}$$

Our actions at *step* t are as follows:

1. *calling the oracle, updating the upper bound, enriching the model*

   We compute $f(x_t), g(x_t), f'(x_t), g'(x_t)$ and, using these data, $\mathbf{h}(\mathbf{x_t}, \mathbf{r^i}) = \max\{\mathbf{f}(\mathbf{x_t}) - \mathbf{r^i}, \mathbf{g}(\mathbf{x_t})\}$.

   Then we set

   $$\delta^{s,t+1} = \min\{\delta^{s,t}, \mathbf{h}(\mathbf{x_t}, \mathbf{r^i})\}.$$

   If

   $$\delta^{s,t+1} \le \theta\delta^s + (1-\theta)l_s, \tag{3.6}$$

   where $\theta \in (0,1)$ is a parameter of the method. We terminate phase s and pass to phase $s+1$, setting

   $$\delta^{s+1} = \delta^{s,t+1}, \quad \delta_{s+1} = \delta_s.$$

   Otherwise we enrich the model of $\mathbf{h}(\mathbf{x}, \mathbf{r^i})$ by setting

   $$\begin{array}{l} F_t^+(x) = \max\{F_t(x), f(x_t) + \langle x - x_t, f'(x_t)\rangle\}, \\ G_t^+(x) = \max\{G_t(x), g(x_t) + \langle x - x_t, g'(x_t)\rangle\}, \\ H_t^+(x, r^i) = \max\{F_t^+(x) - r^i, G_t^+(x)\} \end{array}$$

2. *updating the lower bound*

   We solve the auxiliary optimization problem

   $$\widetilde{\delta}_{s,t} = \min\{H_t^+(x, r^i) : x \in X_t\}$$

   and set

   $$\delta_{s,t+1} = \max\{\min[l_s, \widetilde{\delta}_{s,t}], \delta_{s,t}\}.$$

If

$$\delta_{s,t+1} < (1-\theta)l_s,$$

we update the search point, the localizer and the model:

3. *updating the search point, the localizer and the model*

   We solve the problem

   $$x_{t+1} = \mathrm{argmin}\{\omega_s(x) : x \in X_{t+1}^+ \equiv X_t \cap \{x : H_t^+(x) \le l_s\}\}$$

   and choose as X_{t+1} any set, cut off X by finitely many linear inequalities, which is in-between the sets X_{t+1}^+ and $X_{t+1}^- = \{x \in X : \langle x - x_{t+1}, \omega_s'(x_{t+1}) \ge 0\}$, so that

   $$X_{t+1}^- \supset X_{t+1} \supset X_{t+1}^+.$$

   Then we update the models $F_t^+(\cdot), G_t^+(\cdot)$ into the models $F_{t+1}(\cdot), G_{t+1}(\cdot)$ (thus defining $H_{t+1}(\cdot)$). Step t of the phase s is completed and we loop to step $t+1$.

   If not, i.e

   $$\delta_{s,t+1} \ge (1-\theta)l_s,$$

   we terminate the phase s and set

   $$\delta^{s+1} = \delta^s, \quad \delta_{s+1} = \delta_{s,t+1}.$$

4. *passing to a new problem*

   We check also whether $\delta_{s+1} \ge \mu\delta^s$, where μ is a parameter of the method. If so, we terminate the cycle i itself, find the model's optimal value

   $$r^{i+1} = \min\{F_t^+(x) : G_t^+(x) \le 0, x \in X\} \tag{3.7}$$

   and *an approximate solution x^{*i} of cycle i* as the best - with the smallest value of $\mathbf{h}(\mathbf{x}, \mathbf{r^i})$ - of the search points generated when running all the phases of cycle i.

   Then we pass to the new cycle, i.e start to solve

   $$\min\{\mathbf{h}(\mathbf{x}, \mathbf{r^{i+1}}) : \mathbf{x} \in \mathbf{X}\}. \tag{3.8}$$

The approximate solution x^* found in course of i cycles is the best, i.e. which gives $\min_{j \le i} \mathbf{h}(\mathbf{x^{*j}}, \mathbf{r^i})$, of points $x^{*1}, \ldots, x^{*i}$.

3.3 Constrained NERML: convergence analysis

Lemma 3.1 *Let $u(x,y)$ be real valued convex and Lipschitz continuous function with respect to its, say, second argument with constant L_2 on compact set Y and continuous with respect to its first argument on compact set X, i.e*

$$\forall y_1, y_2 \in Y \quad |u(\cdot, y_1) - u(\cdot, y_2)| \leq L_2 \|y_1 - y_2\|.$$

Then $v(y) = \min_x u(x,y)$ is Lipschitz continuous with constant L_2.

◀ For all $x \in X$ $v(y_1) = \min_x u(x, y_1) \leq u(x, y_1)$ and $v(y_2) = \min_x u(x, y_2) \leq u(x, y_2)$. Compactness of X implies there exist $x_1, x_2 \in X$ such that $\min_x u(x, y_1) = u(x_1, y_1)$ and $\min_x u(x, y_2) = u(x_2, y_2)$. In particular, $v(y_1) \leq u(x_2, y_1)$ and $v(y_2) \leq u(x_1, y_2)$. So,

$$v(y_1) - v(y_2) \leq u(x_2, y_1) - u(x_2, y_2) \leq L_2 |y_1 - y_2|,$$

and, from the other hand,

$$v(y_1) - v(y_2) \geq u(x_1, y_1) - u(x_1, y_2) \geq -L_2 |y_1 - y_2|.$$

Q.E.D.▶

Theorem 3.1 *Let X be compact "simple" set and $0 < \theta, \lambda < 1; \frac{1}{2} < \mu < 1$ be parameters of the method. Then*

(i) The number of cycles to perform before we are reaching an accuracy ϵ is bounded from above by

$$\frac{\log_2 \frac{4\sqrt{2}LD\mu}{\epsilon}}{\log_2 2\mu}.$$

(ii) The total number of oracle calls to reach the accuracy epsilon does not exceed

$$M(\epsilon) = \tilde{c}(\theta, \lambda, \mu) \frac{\Omega L^2}{\alpha \epsilon^2} \log_2 \frac{4\sqrt{2}LD\mu}{\epsilon},$$

where $\tilde{c}(\theta, \lambda, \mu)$ depends solely and continuously on the parameters of the method.

Proof:
To simplify the notations we will use the only index i to denote the finest model of each cycle with the help of which we "jump" to the new cycle.

1^0. Function $\mathbf{h}(\mathbf{x}, \mathbf{r})$ is Lipschitz continuous with respect to x with constant L as maximum of such functions. Functions F, G are Lipschitz continuous with the

same constant too, so all $H(x,r)$ are Lipschitz continuous with respect to x with constant L.

$\mathbf{h}, \mathbf{H}$ are also Lipschitz continuous with respect to $r (r_1 \le r \le f^*)$ with constant 1 (indeed, $f(x) - r, F(x) - r$ are Lipschitz continuous with constant 1 and $g(x), G(x)$ are constant for all r).

2^0. As it follows from Lemma 3.1, functions

$$\varepsilon(r) = \min_x \mathbf{h}(\mathbf{x}, \mathbf{r}) \quad \text{and} \quad \Sigma(r) = \min_x H(x,r)$$

are Lipschitz continuous with respect to r with constant 1.

3^0. The rules for a choice of $r^1, r^2, \ldots$ are described in Constrained NERML scheme and satisfy

$$r_1 \le r_2 \le \ldots \le f^*.$$

This choice ensures a nested structure of cycles i:

$$\varepsilon(r^1) \ge \varepsilon(r^2) \ge \ldots \ge \varepsilon(f^*);$$
$$\Sigma(r^1) \ge \Sigma(r^2) \ge \ldots \Sigma(f^*).$$

4^0.

$$\Sigma^1(r^1) \le LD. \tag{3.9}$$

Indeed, by the construction,

$$r^1 = \min\{f(c^1) + \langle x - c^1, f'(c^1)\rangle, g(c^1) + \langle x - c^1, g'(c^1)\rangle \le 0\} = f(c^1) + \langle x^{*1} - c^1, f'(c^1)\rangle. \tag{3.10}$$

So,

$$\Sigma^1(r^1) \le \delta_1^1 = \max\{f(c^1) - r^1, g(c^1)\} = \max\{-\langle x^{*1} - c^1, f'(c^1)\rangle, g(c^1)\}.$$

But from Cauchi-Shvartz inequality and the Lipschitz property of function f

$$-\langle x^{*1} - c^1, f'(c^1)\rangle \le L\|x^{*1} - c^1\| \le LD.$$

Now, if $g(c^1) \le 0$ we get (3.9). If not, we know there exists point $\underline{x}$ such that $g(\underline{x}) \le 0$ and $g(c^1) \le g(c^1) - g(\underline{x}) \le L\|c^1 - \underline{x}\| \le LD$. So, also in this case we get (3.9).

5^0. Suppose, we have finished cycle i and rised the lower bound of (3.2) from r^i to r^{i+1}. It means:

- $\Sigma^i(r^i) \geq \mu\delta^i$ by the rule for terminating the cycle;
- $\Sigma^i(r^{i+1}) = 0$ by construction of r^{i+1};
- $0 = \Sigma^i(r^{i+1}) \geq \Sigma^i(r^i) + \Sigma^{i'}(r^i)(r^{i+1} - r^i)$ as it follows from previous argumentation and by the property of convex function;
- $\Sigma^i(r^{i-1}) \geq \Sigma^i(r^i) + \Sigma^{i'}(r^i)(r^{i-1} - r^i)$ also by the property of convex function.

6^0. Let us consider two subsequent cycles. From two last inequolities we get:

$$-\Sigma^{i'}(r^i) \geq \frac{\Sigma^i(r^i)}{r^{i+1} - r^i}$$

and ("switching" index i to $i+1$)

$$-\Sigma^{i+1'}(r^{i+1}) \leq \frac{\Sigma^{i+1}(r^i) - \Sigma^{i+1}(r^{i+1})}{r^{i+1} - r^i}.$$

So,

$$\frac{-\Sigma^{i+1'}(r^{i+1})}{-\Sigma^{i'}(r^i)} \leq \frac{\Sigma^{i+1}(r^i) - \Sigma^{i+1}(r^{i+1})}{r^{i+1} - r^i} \Big/ \frac{\Sigma^i(r^i)}{r^{i+1} - r^i} = \frac{\Sigma^{i+1}(r^i) - \Sigma^{i+1}(r^{i+1})}{\Sigma^i(r^i)}. \tag{3.11}$$

But, as was said above $\Sigma^i(r^i) \geq \mu\delta^i$, and $\delta^i \geq \varepsilon(r^i)$ by definition of δ^i. We know also that all our models underestimate the real functions, which ensures $\varepsilon(r) \geq \Sigma(r)$ for all $r \leq f^*$. So,

$$\Sigma^i(r^i) \geq \mu\varepsilon(r^i) \geq \mu\Sigma^{i+1}(r^i). \tag{3.12}$$

Combining this with (3.11) we get

$$\frac{-\Sigma^{i+1'}(r^{i+1})}{-\Sigma^{i'}(r^i)} \leq \frac{1}{\mu} \cdot \frac{\Sigma^{i+1}(r^i) - \Sigma^{i+1}(r^{i+1})}{\Sigma^{i+1}(r^i)} = \frac{1}{\mu}\left(1 - \frac{\Sigma^{i+1}(r^{i+1})}{\Sigma^{i+1}(r^i)}\right).$$

Denote $\alpha^i = \frac{\Sigma^{i+1}(r^{i+1})}{\Sigma^{i+1}(r^i)}$. Then we have

$$\frac{-\Sigma^{i+1'}(r^{i+1})}{-\Sigma^{i'}(r^i)} \leq \frac{1}{\mu}(1 - \alpha^i). \tag{3.13}$$

7^0. From the other hand, using (3.12)

$$\frac{\Sigma^{i+1}(r^{i+1})}{\Sigma^i(r^i)} \leq \frac{\Sigma^{i+1}(r^{i+1})}{\mu\Sigma^{i+1}(r^i)} = \frac{1}{\mu}\alpha^i. \tag{3.14}$$

8^0. Suppose now we have performed m cycles. Then by (3.14)and (3.9)

$$\Sigma^m(r^m) = \frac{\Sigma^m(r^m)}{\Sigma^{m-1}(r^{m-1})} \cdot \frac{\Sigma^{m-1}(r^{m-1})}{\Sigma^{m-2}(r^{m-2})} \cdot \ldots \cdot \frac{\Sigma^2(r^2)}{\Sigma^1(r^1)} \cdot \Sigma^1(r^1) \leq \left(\frac{1}{\mu}\right)^{m-1} \times \alpha^{m-1} \ldots \alpha^1 \times LD, \quad (3.15)$$

while similarly, by (3.13) and remembering that $\Sigma^{1\prime}(r^1) \leq 1$, we get

$$-\Sigma^{m\prime}(r^m) \leq \left(\frac{1}{\mu}\right)^{m-1} (1-\alpha^{m-1}) \ldots (1-\alpha^1). \quad (3.16)$$

9^0. From the third fact mentioned in paragraph 5 implied to cycle m and (3.16)

$$\Sigma^m(r^m) \leq -\Sigma^{m\prime}(r^m)(r^{m+1}-r^m) \leq \left(\frac{1}{\mu}\right)^{m-1} (1-\alpha^{m-1}) \ldots (1-\alpha^1)(r^{m+1}-r^m). \quad (3.17)$$

But $r^{m+1}-r^m \leq 2LD$ (Indeed, $r^{m+1}-r^m \leq f'-r^1 = (f(x^*)-f(c^1))-\langle x^{1^*}-c^1, f'(c^1)\rangle \leq 2LD$), which implies

$$\Sigma^m(r^m) \leq \left(\frac{1}{\mu}\right)^{m-1} \times (1-\alpha^{m-1}) \ldots (1-\alpha^1) \times 2LD. \quad (3.18)$$

10^0. Combining (3.15),(3.18) we arrive at

$$(\Sigma^m(r^m))^2 \leq \left(\frac{1}{\mu}\right)^{2(m-1)} \times \left(\frac{1}{4}\right)^{m-1} \times 2(LD)^2$$

(we used the fact that $\alpha(1-\alpha) \leq \frac{1}{4}$) getting finally

$$\Sigma^m(r^m) \leq \left(\frac{1}{2\mu}\right)^{m-1} \sqrt{2}LD \quad (3.19)$$

11^0. Suppose, at the end of m cycles

$$\mu\epsilon \leq \Sigma^m(r^m) \leq \left(\frac{1}{2\mu}\right)^{m-1} \sqrt{2}LD.$$

I.e.

$$1-m \geq \log_{2\mu}\frac{\mu\epsilon}{\sqrt{2}LD}.$$

So, if

$$m \geq 2 - \log_{2\mu} \frac{\mu\epsilon}{\sqrt{2}LD} = \frac{\log_2 \frac{4\sqrt{2}LD\mu}{\epsilon}}{\log_2 2\mu},$$

then

$$\Sigma^m(r^m) \leq \mu\epsilon.$$

But by construction $\mu\delta^m \leq \Sigma^m(r^m) \Rightarrow \delta^m \leq \epsilon$, which means we have arrived at desired accuracy ϵ.

12^0. Inside each cycle i our algorithm acts exactly as the regular NERML, and we terminate a cycle for sure when $(1-\mu)\delta^{i,s} < \epsilon$. So the number of steps to end the cycle i is bounded from above by

$$N_{\theta,\lambda}(\epsilon,\mu) = c(\theta,\lambda)\frac{(1-\mu)^2\Omega L^2}{\alpha\epsilon^2},$$

so the total number of steps (= Oracle calls) to solve the problem (3.4) to the accuracy ϵ is bounded from above by

$$M(\epsilon) = N_{\theta,\lambda}(\epsilon,\mu)\frac{\log_2 \frac{4\sqrt{2}LD\mu}{\epsilon}}{\log_2 2\mu} = \tilde{c}(\theta,\lambda,\mu)\frac{\Omega L^2}{\alpha\epsilon^2}\log_2\frac{4\sqrt{2}LD\mu}{\epsilon}.$$

■

3.4 Constrained NERML: incremental version

In the previous section we have described an incremental version of NERML - INERML, where we solved the minimization problem of the Lipschitz continuous polyhedrally representable monotone function. But our constrained version of NERML is built as sequence of cycles, in each of them we solve the problem of the type:

$$\min \mathcal{F}(x), \quad \text{where} \quad \mathcal{F}(x) = \max\{f(x) - r, g(x)\}.$$

As we have already seen in calculus of polyhedrally representable functions, $\max(\cdot,\cdot)$ is polyhedrally representable. $g(x) = \max\{g_1(x),\ldots,g_m(x)\}$, where $g_1(x),\ldots,g_m(x)$ are our functional constraints - so, g is also polyhedrally representable. Function f can have some structure too, and if this structure is polyhedrally representable function, then the whole $\mathcal{F}$ is polyhedrally representable as a composition of such functions (see "Calculus of polyhedrally representable functions" in the previous section). As a result, we can use our INERML algorithm for the constrained case as well. But we have to pay an attention that to carry out a constrained INERML, we have to remember what data corresponds to the objective and what to the constraint, as we have to build their models properly.

Chapter 4

Conclusions

In this work we developed two accelerations of NERML: INERML and Constrained NERML. Within the framework of extremely large-scale convex optimization, incremental implementations proved to be very useful and result in better performance as compared to corresponding "regular" algorithms. Nowadays, the main use of this technique is for medical imaging reconstruction problems. Moreover, incremental technique originated from this field and is known as Ordered Subsets in medical literature. In medical imaging one of the classical approaches for building an objective function for iterative reconstruction is Maximum-Likelihood principle. As a result one gets objective presented as a sum of several million simple convex functions. This historical reasons along with the methods used for reconstruction caused to develop Ordered Subsets algorithms only for functions with simple additive structure.

In our research we show that Incremental techniques can be applied for more general classes of problems. Even if full, no subset, mode is used, INERML can lead to better performance than NERML, as INERML builds models for each "inner" function independently and then uses the structure of "outer" function to build a model for the whole objective. In contrast to this, general NERML is completely blind - as a member of simple, "black-box" oriented, algorithms, it is not capable to utilize an apriori information about the function at all.

Proceeding from this, we suggest to use this techniques in other industrial fields, where suitable type of objective is present. For instance, in Structural Design.

We also suggest to try INERML in the native field for Ordered Subsets methods - in Medical Imaging. The reason is, that INERML, based on NERML, allows to utilize an information about the function gathered so far, and the depth of this memory is in our full control. This feature is very attractive because the use of memory can lead to better performance as compared to memoryless algorithms, while our full control on the memory depth allows to use this method for problems of huge dimensions.

For adjusting INERML for the particular problem, one has to fit several parameters carefully:

- θ, λ, μ
- number of subsets
- memory depth

Techniques for solving auxiliary problems are very important as well - we have to solve this auxiliary problems to high accuracy. Otherwize convergence properties of our algorithm may not be valid.

Constrained version of NERML broads the class of problems which can be processed by NERML. The construction of this method suits well to the construction of INERML, so that they can accomplish each other.

Bibliography

[1] Beck, A., Teboulle, M., *Mirror Descent and nonlinear projection subgradient methods for convex optimization*, Operations Research Letters (to appear).

[2] Bendsoe, M.P., Guedes, J.M., Haber, R.B., Pedersen, P., and Taylor, J.E., *An analytical model to predict optimal material properties in the context of Optimal Structural Design*, Journal of Applied Mechanics **61**, 1994.

[3] Bendsoe, M.P., *Optimization of structural topology shape and material*, Springer-Verlag, 1995.

[4] Ben-Tal, A., Kočvara, M., Nemirovski, A., and Zowe, J., *Free Material Design via Semidefinite Programming. The Multiload Case with Contact Conditions*, SIAM J. of Optimization **9** (1999), 813-832.

[5] Ben-Tal, A., Nemirovski, A., *Structural Design via Semidefinite Programming*, In: R. Saigal, H. Wolkowitcz, L. Vandenberghe, Eds. *Handbook on Semidefinite Programming*, Kluwer Academic Publishers, 2000, 443-468.

[6] Ben-Tal A., Margalit T. and Nemirovski A., *The Ordered Subsets Mirror Descent Optimization Method with Applications to Tomography*, SIAM Journal on Optimization **12** (2001), 79-108.

[7] Ben-Tal A., Nemirovski A., *Lectures on Modern Convex Optimization: Analysis, Algorithms and Engineering Applications*, MPS-SIAM Serries on Optimization, 2001.

[8] Ben-Tal A., Nemirovski A., *Non-Euclidean Restricted Memory Level Method for Large-Scale Convex Optimization*, Springer-Verlag GmbH ISSN **102(3)** (2005), 407-456

[9] Bertsekas, D.P. (1995), *Nonlinear Programming*, Athena Scientific, Belmont, Massachusetts.

[10] Bertsekas, D.P., *Incremental least squares methods and the extended Kalman filter*, SIAM J. on Optimization **6** (1996), 807-822.

[11] Bertsekas, D.P., *A new class of incremental gradient methods for least squares problems*, SIAM J. on Optimization **7** (1997), 913-926.

[12] Defrise M., Kinahan P.E., Townsend D.W., Michel C., Sibomana M., Newport D.F., *Exact and approximate rebinning algorithms for 3D PET data*, IEEE Trans. on Medical Imaging, **16(2)** (1997), 145-158.

[13] Hudson H. M., Larkin R. S., *Accelerated Image Reconstruction using Ordered Subsets of Projection Data*, IEEE Trans. Med. Imag., **13**, No. 4 (1994), 601-609.

[14] Kinahan P.E., Rogers J.P., *Analytic 3D image reconstruction using all detected event*, IEEE Trans. Nucl. Sci., **36** (1988), 964-968.

[15] Kiwiel K., *An aggregate subgradient method for non-smooth convex minimization*, Mathematical Programming **27** (1983), 320-341.

[16] Kiwiel K., *Proximal level bundle method for convex nondifferentable optimization, saddle point problems and variational inequallities*, Mathematical programming Series B **69** (1995), 89-109.

[17] Kiwiel, K.C., Larson, T., and Lindberg, P.O., *The efficiency of ballstep subgradient level methods for convex optimization*, Mathematics of Operations Research **24** (1999), 237-254.

[18] Lemaréchal C., *Nonsmooth optimization and descent methods*, Research Report 78-4, IIASA, Laxenburg, Austria (1978)

[19] Lemaréchal C., Strodiot J.J., Bihain A., *On a bundle algorithm for nonsmooth optimization*, in: O.L. Magasarian, R.R. Meyer, S.M. Robinson Eds., Nonlinear Programmin 4 (Academic Press, NY, 1981), 245-282.

[20] Lemaréchal C., Nemirovski A., Nesterov Yu., *New variants of bundle methods*, Mathematical Programming Series B **69** (1995), 111-148.

[21] Mifflin R., *A modification and an extension of Lemaréchal's algorithm for nonsmooth minimization*, Mathematical Programming Study **17** (1982), 77-90.

[22] *Modern mathematical Methods of Optimization*, edited by Karl-Heinz Elster, Akademie Verlag, 1993.

[23] Nemirovski A. and Yudin D., *Problem Complexity and Method Efficiency in Optimization*, J. Wiley & Sons, 1983.

[24] Nesterov Yu., *Cutting plane algorithms from analytic centers: complexity estimate*, mathematical Programming **65** (1995), 149-176.

[25] Polyak B.T, *A general method for solving extremal problems*, Soviet Math.Doklady **174** (1967), 33-36.

[26] Renegar J., *A Mathematical Wiew of Interior-Point Methods in Convex Optimization*, MPS-SIAM Series on Optimization **3**, SIAM, Philadelphia, PA, 2001.

[27] Ringertz, U., *On finding the optimal distribution of material properties*, Structural Optimization **5**, 1993.

[28] Rosenfeld A., Kak A.C., *Digital picture processing*, Computer Science and Applied Mathematics, **1**, 1982.

[29] Schramm H., Zowe J., *A version of bundle idea for minimizing a non-smooth function: conceptual idea, convergence analysis, numerical results*, SIAM Journal on Optimization **2** (1992), 121-152.

[30] Shepp L.A, Vardi Y., *Maximum-Likelihood reconstruction for emission tomography*, IEEE Trans. Med. Image., **MI-1** (1982), 113-122

[31] Shor N.Z, *Generalized gradient descent with application to block programming*, Kibernetika **3** (1967) (in Russian).

[32] R. Tyrrell Rockafellar, *Convex Analysis*, Princeton, New Jersey, Princeton University Press, 1972.

[33] R. Tyrrell Rockafellar and Roger J-B. Wets, *Variational Analysis*, Springer **317**, 1998.

[34] Roos C., Terlaky T., and Vial J.-P. *Theory and Algorithms for Linear Optimization: An Interior Point Approach*, J. Wiley & Sons, 1997.

[35] Wright S.J., *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, PA, 1997.

[36] Ye Y. *Interior Point Algorithms: Theory and Analysis*, J. Wiley & Sons, 1997.