August 2010 (revised December 2010) **Report LIDS - 2848**

Incremental Gradient, Subgradient, and Proximal Methods for Convex Optimization: A Survey [1]

Dimitri P. Bertsekas [2]

Abstract

We survey incremental methods for minimizing a sum $\sum_{i=1}^{m} f_i(x)$ consisting of a large number of convex component functions f_i. Our methods consist of iterations applied to single components, and have proved very effective in practice. We introduce a unified algorithmic framework for a variety of such methods, some involving gradient and subgradient iterations, which are known, and some involving combinations of subgradient and proximal methods, which are new and offer greater flexibility in exploiting the special structure of f_i. We provide an analysis of the convergence and rate of convergence properties of these methods, including the advantages offered by randomization in the selection of components. We also survey applications in inference/machine learning, signal processing, and large-scale and distributed optimization.

1. INTRODUCTION

We consider optimization problems with a cost function consisting of a large number of component functions, such as

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^{m} f_i(x) \\ &\text{subject to} \quad x \in X, \end{aligned} \tag{1.1}$$

where $f_i : \Re^n \mapsto \Re$, $i = 1, \ldots, m$, are real-valued functions, and X is a closed convex set.† We focus on the case where the number of components m is very large, and there is an incentive to use incremental

[1] This is an extended version of similarly titled papers that appear in Math. Programming Journal, 2011, Vol. 129, pp. 163-195, and the edited volume Optimization for Machine Learning (S. Sra, S. Nowozin, and S. Wright, Eds.), MIT Press, 2012. This version corrects two flaws of the Dec. 2010 original survey: in the statements and proofs of Props. 3.1 and 5.2. Both corrections are described by footnotes preceding the propositions. A supplementary survey, dealing with aggregated incremental gradient, proximal, and augmented Lagrangian methods is: Bertsekas, D. P., 2015. "Incremental Aggregated Proximal and Augmented Lagrangian Algorithms," Lab. for Information and Decision Systems Report LIDS-P-3176, MIT, September 2015.

[2] The author is with the Dept. of Electr. Engineering and Comp. Science, M.I.T., Cambridge, Mass., 02139. His research was supported by the AFOSR under Grant FA9550-10-1-0412. Thanks are due to Huizhen (Janey) Yu for extensive helpful discussions and suggestions. Comments by Angelia Nedić and Ben Recht are also appreciated.

† Throughout the paper, we will operate within the n-dimensional space $\Re^n$ with the standard Euclidean norm, denoted $\|\cdot\|$. All vectors are considered column vectors and a prime denotes transposition, so $x'x = \|x\|^2$. We will be using standard terminology of convex optimization, as given for example in textbooks such as Rockafellar's [Roc70], or the author's recent book [Ber09].

methods that operate on a single component f_i at each iteration, rather than on the entire cost function. If each incremental iteration tends to make reasonable progress in some "average" sense, then depending on the value of m, an incremental method may significantly outperform (by orders of magnitude) its nonincremental counterpart, as experience has shown.

In this paper, we survey the algorithmic properties of incremental methods in a unified framework, based on the author's recent work on incremental proximal methods [Ber10] (an early version appears in the supplementary algorithms chapter of the book [Ber09]). In this section, we first provide an overview of representative applications, and then we discuss three types of incremental methods: gradient, subgradient, and proximal. We unify these methods, into a combined method, which we use as a vehicle for analysis later.

1.1 Some Examples of Additive Cost Problems

Additive cost problems of the form (1.1) arise in a variety of contexts. Let us provide a few examples where the incremental approach may have an advantage over alternatives.

Example 1.1: (Least Squares and Related Inference Problems)

An important context where cost functions of the form $\sum_{i=1}^m f_i(x)$ arise is inference/machine learning, where each term $f_i(x)$ corresponds to error between some data and the output of a parametric model, with x being the vector of parameters. An example is *linear least squares* problems, where f_i has quadratic structure, except for a regularization function. The latter function may be differentiable/quadratic, as in the classical regression problem

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^m (a_i'x - b_i)^2 + \gamma\|x - \bar{x}\|^2 \\ &\text{subject to} \quad x \in \Re^n, \end{aligned}$$

where $\bar{x}$ is given, or nondifferentiable, as in the ℓ_1*-regularization problem*

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^m (a_i'x - b_i)^2 + \gamma \sum_{j=1}^n |x_j| \\ &\text{subject to} \quad x = (x_1, \ldots, x_n) \in \Re^n, \end{aligned}$$

which will be discussed further in Section 5.

A more general class of additive cost problems is *nonlinear least squares*. Here

$$f_i(x) = \big(h_i(x)\big)^2,$$

where $h_i(x)$ represents the difference between the ith of m measurements from a physical system and the output of a parametric model whose parameter vector is x. Problems of nonlinear curve fitting and regression, as well as problems of training neural networks fall in this category, and they are typically nonconvex.

Another possibility is to use a nonquadratic function to penalize the error between some data and the output of the parametric model. For example in place of the squared error $(a_i'x - b_i)^2$, we may use

$$f_i(x) = \ell_i(a_i'x - b_i),$$

where ℓ_i is a convex function. This is a common approach in robust estimation and some support vector machine formulations.

Still another example is *maximum likelihood estimation*, where f_i is of the form

$$f_i(x) = -\log P_Y(y_i; x),$$

and $y_1, \ldots, y_m$ represent values of independent samples of a random vector whose distribution $P_Y(\cdot; x)$ depends on an unknown parameter vector $x \in \Re^n$ that we wish to estimate. Related contexts include "incomplete" data cases, where the expectation-maximization (EM) approach is used.

The following four examples deal with broadly applicable problem structures that give rise to additive cost functions.

Example 1.2: (Dual Optimization in Separable Problems)

Consider the problem

$$\begin{aligned} &\text{maximize} \quad \sum_{i=1}^m c_i(y_i) \\ &\text{subject to} \quad \sum_{i=1}^m g_i(y_i) \geq 0, \quad y_i \in Y_i, \quad i = 1, \ldots, m, \end{aligned}$$

where $c_i : \Re^\ell \mapsto \Re$ and $g_i : \Re^\ell \mapsto \Re^n$ are functions of a vector $y_i \in \Re^\ell$, and Y_i are given sets of $\Re^\ell$. Then by assigning a dual vector/multiplier $x \in \Re^n$ to the n-dimensional constraint function, we obtain the dual problem

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^n f_i(x) \\ &\text{subject to} \quad x \geq 0, \end{aligned}$$

where

$$f_i(x) = \sup_{y_i \in Y_i} \left\{ c_i(y_i) + x' g_i(y_i) \right\},$$

which has the additive form (1.1). Here Y_i is not assumed convex, so integer programming and other discrete optimization problems are included. However, the dual cost function components f_i are always convex, and their values and subgradients can often be conveniently computed, particularly when y_i is a scalar or Y_i is a finite set.

Example 1.3: (Problems with Many Constraints)

Problems of the form

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \quad g_j(x) \leq 0, \quad j = 1, \ldots, r, \quad x \in X, \end{aligned} \tag{1.2}$$

where the number r of constraints is very large often arise in practice, either directly or via reformulation from other problems. They can be handled in a variety of ways. One possibility is to adopt a penalty function approach, and replace problem (1.2) with

$$\begin{aligned} &\text{minimize} \quad f(x) + c \sum_{j=1}^r P\big(g_j(x)\big) \\ &\text{subject to} \quad x \in X, \end{aligned} \tag{1.3}$$

where $P(\cdot)$ is a scalar penalty function satisfying $P(t) = 0$ if $t \leq 0$, and $P(t) > 0$ if $t > 0$, and c is a positive penalty parameter. For example, one may use the quadratic penalty $P(t) = \big(\max\{0, t\}\big)^2$, or the

nondifferentiable penalty $P(t) = \max\{0, t\}$. In the latter case, it can be shown that the optimal solutions of problems (1.2) and (1.3) coincide when c is sufficiently large (see for example [BNO03], Section 7.3, for the case where f is convex). The cost function of the penalized problem (1.3) is of the additive form (1.1).

Set constraints of the form $x \in \cap_{i=1}^m X_i$, where X_i are closed sets, can also be handled by penalties in a way that gives rise to additive cost functions (a simpler but important special case where such constraints arise is the problem of finding a common point within the sets X_i, $i = 1, \ldots, m$; see Section 5.2). In particular, under relatively mild conditions, problem (1.2) with $X = \cap_{i=1}^m X_i$ is equivalent to the unconstrained minimization of

$$f(x) + c \sum_{j=1}^r P\big(g_j(x)\big) + \gamma \sum_{i=1}^m \text{dist}(x; X_i),$$

where $\text{dist}(x; X_i) = \min_{y \in X_i} \|y - x\|$ and γ is a sufficiently large penalty parameter. We discuss this possibility in Section 5.2.

Example 1.4: (Minimization of an Expected Value - Stochastic Programming)

Consider the minimization of an expected value

$$\begin{aligned} &\text{minimize} \quad E\big\{H(x, w)\big\} \\ &\text{subject to} \quad x \in X, \end{aligned} \tag{1.4}$$

where H is a function of x and a random variable w taking a finite but very large number of values w_i, $i = 1, \ldots, m$, with corresponding probabilities π_i. Here the cost function can be written as the sum of the m functions $\pi_i H(x, w_i)$.

An example is *stochastic programming*, a classical model of two-stage optimization under uncertainty, where a vector $x \in X$ is selected at cost $C(x)$, a random event occurs that has m possible outcomes $w_1, \ldots, w_m$, and then another vector y is selected from some set Y with knowledge of the outcome that occurred. Then the optimal decision problem is to specify a vector $y_i \in Y$ for each outcome w_i, and to minimize over x and y_i the expected cost

$$C(x) + \sum_{i=1}^m \pi_i G_i(y_i),$$

where $G_i(y_i)$ is the cost associated with the occurrence of w_i and π_i is the corresponding probability. This is a problem with an additive cost function.

Additive cost function problems also arise from problem (1.4) in a different way, when the expected value $E\big\{H(x, w)\big\}$ is approximated by an m-sample average

$$F(x) = \frac{1}{m} \sum_{i=1}^m H(x, w_i),$$

where w_i are independent samples of the random variable w. The minimum of the sample average $f(x)$ is then taken as an approximation of the minimum of $E\big\{H(x, w)\big\}$.

Example 1.5: (Weber Problem in Location Theory)

A basic problem in location theory is to find a point x in the plane whose sum of weighted distances from a given set of points $y_1, \ldots, y_m$ is minimized. Mathematically, the problem is

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^m w_i \|x - y_i\| \\ &\text{subject to} \quad x \in \Re^n, \end{aligned}$$

where $w_1, \ldots, w_m$ are given positive scalars. This problem descends from the famous Fermat-Torricelli-Viviani problem (see [BMS99] for an account of the history). The algorithmic approaches of the present paper would be of potential interest when the number of points m is large. We refer to Drezner and Hamacher [DrH04] for a survey of recent research, and to Beck and Teboulle [BeT10] for a discussion that is relevant to our context.

The structure of the additive cost function (1.1) often facilitates the use of a distributed computing system that is well-suited for the incremental approach. The following is an illustrative example.

Example 1.6: (Distributed Incremental Optimization – Sensor Networks)

Consider a network of m sensors where data are collected and are used to solve some inference problem involving a parameter vector x. If $f_i(x)$ represents an error penalty for the data collected by the ith sensor, the inference problem is of the form (1.1). While it is possible to collect all the data at a fusion center where the problem will be solved in centralized manner, it may be preferable to adopt a distributed approach in order to save in data communication overhead and/or take advantage of parallelism in computation. In such an approach the current iterate x_k is passed on from one sensor to another, with each sensor i performing an incremental iteration involving just its local component function f_i, and the entire cost function need not be known at any one location. We refer to Blatt, Hero, and Gauchman [BHG08], and Rabbat and Nowak [RaN04], [RaN05] for further discussion.

The approach of computing incrementally the values and subgradients of the components f_i in a distributed manner can be substantially extended to apply to general systems of asynchronous distributed computation, where the components are processed at the nodes of a computing network, and the results are suitably combined, as discussed by Nedić, Bertsekas, and Borkar [NBB01]. The analysis here relies on ideas from distributed asynchronous gradient methods (both deterministic and stochastic), which were developed in the early 80s by the author and his coworkers [Ber83], [TBA86], [BeT89]), and have been experiencing a resurgence recently (see e.g., Nedić and Ozdaglar [NeO09]).

1.2 Incremental Gradient Methods - Differentiable Problems

Let us consider first the case where the components f_i are differentiable (not necessarily convex). Then, we may use incremental gradient methods, which have the form

$$x_{k+1} = P_X\big(x_k - \alpha_k \nabla f_{i_k}(x_k)\big), \tag{1.5}$$

where α_k is a positive stepsize, $P_X(\cdot)$ denotes projection on X, and i_k is the index of the cost component that is iterated on. Such methods have a long history, particularly for the unconstrained case ($X = \Re^n$), starting with the Widrow-Hoff least mean squares (LMS) method [WiH60] for positive semidefinite quadratic component functions (see e.g., [Luo91], [BeT96], Section 3.2.5, [Ber99], Section 1.5.2). They have also been used extensively for the training of neural networks, a case of nonquadratic/nonconvex cost components, under the generic name "backpropagation methods." There are several variants of these methods, which differ in the stepsize selection scheme, and the order in which components are taken up for iteration (it could be deterministic or randomized). They are supported by convergence analyses under various conditions; see Luo [Luo91], Grippo [Gri93], [Gri00], Luo and Tseng [LuT94], Mangasarian and Solodov [MaS94], Bertsekas [Ber97], Solodov [Sol98], Tseng [Tse98].

When comparing the incremental gradient method with its classical nonincremental gradient counterpart [$m = 1$ and all components lumped into a single function $F(x) = \sum_{i=1}^m f_i(x)$], there are two complementary performance issues to consider:

(a) *Progress when far from convergence.* Here the incremental method can be much faster. For an extreme case let $X = \Re^n$ (no constraints), and take m very large and all components f_i identical to each other. Then an incremental iteration requires m times less computation than a classical gradient iteration, but gives exactly the same result, when the stepsize is appropriately scaled to be m times larger. While this is an extreme example, it reflects the essential mechanism by which incremental methods can be far superior: when the components f_i are not too dissimilar, far from the minimum a single component gradient will point to "more or less" the right direction [see also the discussion of [Ber97], and [Ber99] (Example 1.5.5 and Exercise 1.5.5)].

(b) *Progress when close to convergence.* Here the incremental method is generally inferior. As we will discuss shortly, it converges at a sublinear rate because it requires a diminishing stepsize α_k, compared with the typically linear rate achieved with the classical gradient method when a small constant stepsize is used ($\alpha_k \equiv \alpha$). One may use a constant stepsize with the incremental method, and indeed this may be the preferred mode of implementation, but then the method typically oscillates in the neighborhood of a solution, with size of oscillation roughly proportional to α, as examples and theoretical analysis show.

To understand the convergence mechanism of incremental gradient methods, let us consider the case $X = \Re^n$, and assume that the component functions f_i are selected for iteration according to a cyclic order [i.e., $i_k = (k \text{ modulo } m) + 1$], and let us assume that α_k is constant within a cycle (i.e., for all $\ell = 0, 1, \ldots$, $\alpha_{\ell m} = \alpha_{\ell m+1} = \cdots = \alpha_{\ell m+m-1}$). Then, viewing the iteration (1.5) in terms of cycles, we have for every k that marks the beginning of a cycle ($i_k = 1$),

$$x_{k+m} = x_k - \alpha_k \sum_{i=1}^{m} \nabla f_i(x_{k+i-1}) = x_k - \alpha_k \big(\nabla F(x_k) + e_k\big), \tag{1.6}$$

where F is the cost function/sum of components, $F(x) = \sum_{i=1}^m f_i(x)$, and e_k is given by

$$e_k = \sum_{i=1}^{m} \big(\nabla f_i(x_k) - \nabla f_i(x_{k+i-1})\big),$$

and may be viewed as an error in the calculation of the gradient $\nabla f(x_k)$. For Lipschitz continuous gradient functions ∇f_i, the error e_k is proportional to α_k, and this shows two fundamental properties of incremental gradient methods, which hold generally for the other incremental methods of this paper as well:

(a) A constant stepsize ($\alpha_k \equiv \alpha$) typically cannot guarantee convergence, since then the size of the gradient error $\|e_k\|$ is typically bounded away from 0. Instead (in the case of differentiable components f_i) a peculiar form of convergence takes place for constant but sufficiently small α, whereby the iterates within cycles converge but to different points within a sequence of m points (i.e., the sequence of first points in the cycles converges to a different limit than the sequence of second points in the cycles, etc). This is true even in the most favorable case of a linear least squares problem (see Luo [Luo91], or the textbook analysis of [Ber99], Section 1.5.1).

(b) A diminishing stepsize [such as $\alpha_k = O(1/k)$] leads to diminishing error e_k, so (under the appropriate Lipschitz condition) it can result in convergence to a stationary point of f.

A corollary of these properties is that the price for achieving convergence is the slow (sublinear) asymptotic rate of convergence associated with a diminishing stepsize, which compares unfavorably with the

often linear rate of convergence associated with a constant stepsize and the nonincremental gradient method. However, in practical terms this argument does not tell the entire story, since the incremental gradient method often achieves in the early iterations a much faster convergence rate than its nonincremental counterpart. In practice, the incremental method is usually operated with a stepsize that is either constant or is gradually reduced up to a positive value, which is small enough so that the resulting asymptotic oscillation is of no essential concern. An alternative, is to use a constant stepsize throughout, but reduce over time the degree of incrementalism, so that ultimately the method becomes nonincremental and achieves a linear convergence rate (see [Ber97], [Sol98]).

Aside from extensions to nonidifferentiable cost problems, for $X = \Re^n$, there is an important variant of the incremental gradient method that involves extrapolation along the direction of the difference of the preceding two iterates:

$$x_{k+1} = x_k - \alpha_k \nabla f_{i_k}(x_k) + \beta(x_k - x_{k-1}), \tag{1.7}$$

where β is a scalar in $[0, 1)$ and $x_{-1} = x_0$ (see e.g., [MaS94], [Tse98], [Ber96], Section 3.2). This is sometimes called *incremental gradient method with momentum*. The nonincremental version of this method is the *heavy ball* method of Polyak [Pol64], which can be shown to have faster convergence rate than the corresponding gradient method (see [Pol87], Section 3.2.1). A nonincremental method of this type, but with variable and suitably chosen value of β, has been proposed by Nesterov [Nes83], and has received a lot of attention recently because it has optimal iteration complexity properties under certain conditions (see Nesterov [Nes04], [Nes05], Lu, Monteiro, and Yuan [LMY08], Tseng [Tse08], Beck and Teboulle [BeT09], [BeT10]). However, no incremental analogs of this method with favorable complexity properties are currently known.

Another variant of the incremental gradient method for the case $X = \Re^n$ has been proposed by Blatt, Hero, and Gauchman [BHG08], which (after the first m iterates are computed) has the form

$$x_{k+1} = x_k - \alpha \sum_{\ell=0}^{m-1} \nabla f_{i_{k-\ell}}(x_{k-\ell}) \tag{1.8}$$

[for $k < m$, the summation should go up to $\ell = k$, and α should be replaced by a corresponding larger value, such as $\alpha_k = m\alpha/(k+1)$]. This method also computes the gradient incrementally, one component per iteration, but in place of the single component gradient $\nabla f_{i_k}(x_k)$ in Eq. (1.5), it uses an approximation to the total cost gradient $\nabla f(x_k)$, which is an aggregate of the component gradients computed in the past m iterations. A cyclic order of component function selection [$i_k = (k \text{ modulo } m) + 1$] is assumed in [BHG08], and a convergence analysis is given, including a linear convergence rate result for a sufficiently small constant stepsize α and quadratic component functions f_i. It is not clear how iterations (1.5) and (1.8) compare in terms of rate of convergence, although the latter seems likely to make faster progress when close to convergence. Note that iteration (1.8) bears similarity to the incremental gradient iteration with momentum (1.7) where $\beta \approx 1$. In particular, when $\alpha_k \equiv \alpha$, the sequence generated by Eq. (1.7) satisfies

$$x_{k+1} = x_k - \alpha \sum_{\ell=0}^{k} \beta^\ell \, \nabla f_{i_{k-\ell}}(x_{k-\ell}) \tag{1.9}$$

[both iterations (1.8) and (1.9) involve different types of diminishing dependence on past gradient components]. There are no known analogs of iterations (1.7) and (1.8) for nondifferentiable cost problems.

Among alternative incremental methods for differentiable cost problems, let us also mention versions of the Gauss-Newton method for nonlinear least squares problems, based on the extended Kalman filter (Davidon [Dav76], Bertsekas [Ber96], and Moriyama, Yamashita, and Fukushima [MYF03]). They are

mathematically equivalent to the ordinary Gauss-Newton method for linear least squares, which they solve exactly after a single pass through the component functions f_i, but they often perform much faster than the latter in the nonlinear case, particularly when m is large.

Let us finally note that incremental gradient methods are also related to stochastic gradient methods, which aim to minimize an expected value $E\{H(x,w)\}$ (cf. Example 1.2) by using the iteration

$$x_{k+1} = x_k - \alpha_k \nabla H(x_k, w_k),$$

where w_k is a sample of the random variable w. These methods also have a long history (see Polyak and Tsypkin [PoT73], Ljung [Lju77], Kushner and Clark [KuC78], Tsitsiklis, Bertsekas, and Athans [TBA86], Polyak [Pol87], Bertsekas and Tsitsiklis [BeT89], [BeT96], [BeT00], Gaivoronskii [Gai93], Pflug [Pfl96], Kushner and Yin [KuY97], Bottou [Bot05], Meyn [Mey07], Borkar [Bor08], Nemirovski et. al [NJL09], Lee and Wright [LeW10]), and are strongly connected with stochastic approximation algorithms. The main difference between stochastic and deterministic formulations is that the former involve sequential sampling of cost components from an infinite population under some statistical assumptions, while in the latter the set of cost components is predetermined and finite. However, it is possible to view the incremental gradient method (1.5), with a randomized selection of the component function f_i (i.e., with i_k chosen to be any one of the indexes $1, \ldots, m$, with equal probability $1/m$), as a stochastic gradient method (see [BeT96], Example 4.4, [BeT00], Section 5).

The stochastic formulation of incremental methods just discussed highlights an important application context where the component functions f_i are not given a priori, but rather become known sequentially through some observation process. Then it often makes sense to use an incremental method to process the component functions as they become available, and to obtain approximate solutions as early as possible. In fact this may be essential in time-sensitive and possibly time-varying environments, where solutions are needed "on-line." In such cases, one may hope than an adequate estimate of the optimal solution will be obtained, before all the functions f_i are processed for the first time.

1.3 Incremental Subgradient Methods - Nondifferentiable Problems

We now discuss the case where the component functions f_i are convex and nondifferentiable at some points, and consider incremental subgradient methods. These are similar to their gradient counterparts (1.5) except that an arbitrary subgradient $\tilde{\nabla} f_{i_k}(x_k)$ of the cost component f_{i_k} is used in place of the gradient:†

$$x_{k+1} = P_X\big(x_k - \alpha_k \tilde{\nabla} f_{i_k}(x_k)\big). \tag{1.10}$$

Such methods were first proposed in the general form (1.10) in the Soviet Union by Kibardin [Kib80], following the earlier paper by Litvakov [Lit66] (which considered convex/nondifferentiable extensions of linear least squares problems) and other related subsequent proposals.‡ These works remained unnoticed in the

† In this paper, we use $\tilde{\nabla} f(x)$ to denote a subgradient of a convex function f at a vector x, i.e, a vector such that $f(z) \geq f(x) + \tilde{\nabla} f(x)'(z-x)$ for all $x \in \Re^n$. The choice of $\tilde{\nabla} f(x)$ from within the set of all subgradients at x [the subdifferential at x, denoted $\partial f(x)$] will be clear from the context. Note that if f is real-valued, $\partial f(x)$ is nonempty and compact for all x. If f is differentiable at x, $\partial f(x)$ consists of a single element, the gradient $\nabla f(x)$.

‡ Generally, in the 60s and 70s, algorithmic ideas relating to simple gradient methods with and without deterministic and stochastic errors were popular in the Soviet scientific community, partly due to an emphasis on stochastic iterative algorithms, such as pseudogradient and stochastic approximation; the works of Ermoliev, Polyak, and Tsypkin, to name a few of the principal contributors, are representative [Erm69], [PoT73], [Erm76], [Pol78], [Pol87]. By contrast the emphasis in the Western literature at the time was in more complex Newton-like and conjugate direction methods.

Western literature, where incremental methods were reinvented often in different contexts and with different lines of analysis; see Solodov and Zavriev [SoZ98], Bertsekas [Ber99] (Section 6.3.2), Ben-Tal, Margalit, and Nemirovski [BMN01], Nedić and Bertsekas [NeB00], [NeB01], [NeB10], Nedić, Bertsekas, and Borkar [NBB01], Kiwiel [Kiw04], Rabbat and Nowak [RaN04], [RaN05], Gaudioso, Giallombardo, and Miglionico [GGM06], Shalev-Shwartz et. al. [SSS07], Helou and De Pierro [HeD09], Johansson, Rabi, and Johansson [JRJ09], Predd, Kulkarni, and Poor [PKP09], and Ram, Nedić, Veeravalli [RNV09], [RNV09], and Duchi, Hazan, and Singer [DHS10].

Incremental subgradient methods have convergence characteristics that are similar in many ways to their gradient counterparts, the most important similarity being the necessity for a diminishing stepsize α_k for convergence. The lines of analysis, however, tend to be different, since incremental gradient methods rely for convergence on arguments based on decrease of the cost function value, while incremental subgradient methods rely on arguments based on decrease of the iterates' distance to the optimal solution set. The line of analysis of the present paper is of the latter type, similar to earlier works of the author and his collaborators (see [NeB00], [NeB01], [NBB01], and the textbook presentations in [Ber99], [BNO03]).

Note two important ramifications of the lack of differentiability of the component functions f_i:

(1) Convexity of f_i becomes essential, since the notion of subgradient is connected with convexity (subgradient-like algorithms for nondifferentiable/nonconvex problems have been suggested in the literature, but tend to be complicated and have not found much application thus far).

(2) There is more reason to favor the incremental over the nonincremental methods, since (contrary to the differentiable case) nonincremental subgradient methods also require a diminishing stepsize for convergence, and typically achieve a sublinear rate of convergence. Thus the one theoretical advantage of the nonincremental gradient method discussed earlier is not shared by its subgradient counterpart.

Let us finally mention that just as in the differentiable case, there is a substantial literature for stochastic versions of subgradient methods. In fact, as we will discuss in this paper, there is a potentially significant advantage in turning the method into a stochastic one by randomizing the order of selection of the components f_i for iteration.

1.4 Incremental Proximal Methods

We now consider an extension of the incremental approach to proximal algorithms. The simplest one for problem (1.1) is of the form

$$x_{k+1} = \arg\min_{x \in X} \left\{ f_{i_k}(x) + \frac{1}{2\alpha_k} \|x - x_k\|^2 \right\}, \tag{1.11}$$

which relates to the proximal minimization algorithm (Martinet [Mar70], Rockafellar [Roc76]) in the same way that the incremental subgradient method (1.10) relates to the classical nonincremental subgradient method.† Here $\{\alpha_k\}$ is a positive scalar sequence, and we will assume that each $f_i : \Re^n \mapsto \Re$ is a convex function, and X is a nonempty closed convex set. The motivation for this type of method, which was proposed only recently in [Ber10], is that with a favorable structure of the components, the proximal iteration (1.10)

† In this paper we restrict attention to proximal methods with the quadratic regularization term $\|x - x_k\|^2$. Our approach is applicable in principle when a nonquadratic term is used instead in order to match the structure of the given problem. The discussion of such alternative algorithms is beyond our scope.

may be obtained in closed form or be relatively simple, in which case it may be preferable to a gradient or subgradient iteration. In this connection, we note that generally, proximal iterations are considered more stable than gradient iterations; for example in the nonincremental case, they converge essentially for any choice of α_k, while this is not so for gradient methods.

Unfortunately, while some cost function components may be well suited for a proximal iteration, others may not be because the minimization (1.11) is inconvenient, and this leads us to consider combinations of gradient/subgradient and proximal iterations. In fact this has motivated in the past nonincremental combinations of gradient and proximal methods for minimizing the sum of two functions (or more generally, finding a zero of the sum of two nonlinear operators). These methods have a long history, dating to the splitting algorithms of Lions and Mercier [LiM79], Passty [Pas79], and Spingarn [Spi85], and have become popular recently (see Beck and Teboulle [BeT09], [BeT10], and the references they give to specialized algorithms, such as shrinkage/thresholding, cf. Section 5.1). Let us also note that splitting methods are related to alternating direction methods of multipliers (see Gabay and Mercier [GaM76], [Gab83], Bertsekas and Tsitsiklis [BeT89], Eckstein and Bertsekas [EcB92]), which are presently experiencing a revival as viable (nonincremental) methods for minimizing sums of component functions (see the survey by Boyd et. al. [BPC10], which contains extensive references to recent work and applications, and the complexity-oriented work of Goldfarb, Ma, and Scheinberg [GoM09], [GMS10]).

With similar motivation in mind, we adopt in this paper a unified algorithmic framework that includes incremental gradient, subgradient, and proximal methods, and their combinations, and serves to highlight their common structure and behavior. We focus on problems of the form

$$\begin{aligned} &\text{minimize} \quad F(x) \stackrel{\text{def}}{=} \sum_{i=1}^{m} F_i(x) \\ &\text{subject to} \quad x \in X, \end{aligned} \tag{1.12}$$

where for all i,

$$F_i(x) = f_i(x) + h_i(x), \tag{1.13}$$

$f_i : \Re^n \mapsto \Re$ and $h_i : \Re^n \mapsto \Re$ are real-valued convex functions, and X is a nonempty closed convex set.

In Section 2, we consider several incremental algorithms that iterate on the components f_i with a proximal iteration, and on the components h_i with a subgradient iteration. By choosing all the f_i or all the h_i to be identically zero, we obtain as special cases the subgradient and proximal iterations (1.10) and (1.11), respectively. However, our methods offer greater flexibility, and may exploit the special structure of problems where the functions f_i are suitable for a proximal iteration, while the components h_i are not and thus may be preferably treated with a subgradient iteration.

In Section 3, we discuss the convergence and rate of convergence properties of methods that use a cyclic rule for component selection, while in Section 4, we discuss the case of a randomized component selection rule. In summary, the convergence behavior of our incremental methods is similar to the one outlined earlier for the incremental subgradient method (1.10). This includes convergence within a certain error bound for a constant stepsize, exact convergence to an optimal solution for an appropriately diminishing stepsize, and improved convergence rate/iteration complexity when randomization is used to select the cost component for iteration. In Section 5 we illustrate our methods for some example applications.

2. INCREMENTAL SUBGRADIENT-PROXIMAL METHODS

In this section, we consider problem (1.12)-(1.13), and introduce several incremental algorithms that involve

a combination of a proximal and a subgradient iteration. One of our algorithms has the form

$$z_k = \arg\min_{x\in X}\left\{f_{i_k}(x) + \frac{1}{2\alpha_k}\|x - x_k\|^2\right\}, \tag{2.1}$$

$$x_{k+1} = P_X\big(z_k - \alpha_k\tilde{\nabla}h_{i_k}(z_k)\big), \tag{2.2}$$

where $\tilde{\nabla}h_{i_k}(z_k)$ is an arbitrary subgradient of h_{i_k} at z_k. Note that the iteration is well-defined because the minimum in Eq. (2.1) is uniquely attained since f_i is continuous and $\|x - x_k\|^2$ is real-valued, strictly convex, and coercive, while the subdifferential $\partial h_i(z_k)$ is nonempty since h_i is real-valued. Note also that by choosing all the f_i or all the h_i to be identically zero, we obtain as special cases the subgradient and proximal iterations (1.10) and (1.11), respectively.

The iterations (2.1) and (2.2) maintain both sequences $\{z_k\}$ and $\{x_k\}$ within the constraint set X, but it may be convenient to relax this constraint for either the proximal or the subgradient iteration, thereby requiring a potentially simpler computation. This leads to the algorithm

$$z_k = \arg\min_{x\in\Re^n}\left\{f_{i_k}(x) + \frac{1}{2\alpha_k}\|x - x_k\|^2\right\}, \tag{2.3}$$

$$x_{k+1} = P_X\big(z_k - \alpha_k\tilde{\nabla}h_{i_k}(z_k)\big), \tag{2.4}$$

where the restriction $x \in X$ has been omitted from the proximal iteration, and the algorithm

$$z_k = x_k - \alpha_k\tilde{\nabla}h_{i_k}(x_k), \tag{2.5}$$

$$x_{k+1} = \arg\min_{x\in X}\left\{f_{i_k}(x) + \frac{1}{2\alpha_k}\|x - z_k\|^2\right\}, \tag{2.6}$$

where the projection onto X has been omitted from the subgradient iteration. It is also possible to use different stepsize sequences in the proximal and subgradient iterations, but for notational simplicity we will not discuss this type of algorithm.

All of the incremental proximal algorithms given above are new to our knowledge, having first been proposed in the author's recent paper [Ber10] and the on-line chapter of the book [Ber09]. The closest connection to the existing proximal methods is the "proximal gradient" method, which has been analyzed and discussed recently in the context of several machine learning applications by Beck and Teboulle [BeT09], [BeT10] (it can also be interpreted in terms of splitting algorithms [LiM79], [Pas79]). This method is nonincremental, applies to differentiable h_i, and contrary to subgradient and incremental methods, it does not require a diminishing stepsize for convergence to the optimum. In fact, the line of convergence analysis of Beck and Teboulle relies on the differentiability of h_i and the nonincremental character of the proximal gradient method, and is thus different from ours.

Part (a) of the following proposition is a key fact about incremental proximal iterations. It shows that they are closely related to incremental subgradient iterations, with the only difference being that the subgradient is evaluated at the end point of the iteration rather than at the start point. Part (b) of the proposition provides an inequality that is well-known in the theory of proximal methods, and will be useful for our convergence analysis. In the following, we denote by $\mathrm{ri}(S)$ the relative interior of a convex set S, and by $\mathrm{dom}(f)$ the effective domain $\big\{x \mid f(x) < \infty\big\}$ of a function $f : \Re^n \mapsto (-\infty, \infty]$.

Proposition 2.1: Let X be a nonempty closed convex set, and let $f : \Re^n \mapsto (-\infty, \infty]$ be a closed proper convex function such that $\mathrm{ri}(X) \cap \mathrm{ri}\big(\mathrm{dom}(f)\big) \neq \emptyset$. For any $x_k \in \Re^n$ and $\alpha_k > 0$, consider the proximal iteration

$$x_{k+1} = \arg\min_{x \in X} \left\{ f(x) + \frac{1}{2\alpha_k} \|x - x_k\|^2 \right\}. \tag{2.7}$$

(a) The iteration can be written as

$$x_{k+1} = P_X\big(x_k - \alpha_k \tilde{\nabla} f(x_{k+1})\big), \qquad i = 1, \ldots, m, \tag{2.8}$$

where $\tilde{\nabla} f(x_{k+1})$ is some subgradient of f at x_{k+1}.

(b) For all $y \in X$, we have

$$\begin{aligned} \|x_{k+1} - y\|^2 &\leq \|x_k - y\|^2 - 2\alpha_k\big(f(x_{k+1}) - f(y)\big) - \|x_k - x_{k+1}\|^2 \\ &\leq \|x_k - y\|^2 - 2\alpha_k\big(f(x_{k+1}) - f(y)\big). \end{aligned} \tag{2.9}$$

Proof: (a) We use the formula for the subdifferential of the sum of the three functions f, $(1/2\alpha_k)\|x - x_k\|^2$, and the indicator function of X (cf. Prop. 5.4.6 of [Ber09]), together with the condition that 0 should belong to this subdifferential at the optimum x_{k+1}. We obtain that Eq. (2.7) holds if and only if

$$\frac{1}{\alpha_k}(x_k - x_{k+1}) \in \partial f(x_{k+1}) + N_X(x_{k+1}), \tag{2.10}$$

where $N_X(x_{k+1})$ is the normal cone of X at x_{k+1} [the set of vectors y such that $y'(x - x_{k+1}) \leq 0$ for all $x \in X$, and also the subdifferential of the indicator function of X at x_{k+1}; see [Ber09], p. 185]. This is true if and only if

$$x_k - x_{k+1} - \alpha_k \tilde{\nabla} f(x_{k+1}) \in N_X(x_{k+1}),$$

for some $\tilde{\nabla} f(x_{k+1}) \in \partial f(x_{k+1})$, which in turn is true if and only if Eq. (2.8) holds, by the projection theorem.

(b) We have

$$\|x_k - y\|^2 = \|x_k - x_{k+1} + x_{k+1} - y\|^2 = \|x_k - x_{k+1}\|^2 - 2(x_k - x_{k+1})'(y - x_{k+1}) + \|x_{k+1} - y\|^2. \tag{2.11}$$

Also since from Eq. (2.10), $\frac{1}{\alpha_k}(x_k - x_{k+1})$ is a subgradient at x_{k+1} of the sum of f and the indicator function of X, we have (using also the assumption $y \in X$)

$$f(x_{k+1}) + \frac{1}{\alpha_k}(x_k - x_{k+1})'(y - x_{k+1}) \leq f(y).$$

Combining this relation with Eq. (2.11), the result follows. **Q.E.D.**

Based on Prop. 2.1(a), we see that all the preceding iterations can be written in an incremental subgradient format:

(a) Iteration (2.1)-(2.2) can be written as

$$z_k = P_X\big(x_k - \alpha_k \tilde{\nabla} f_{i_k}(z_k)\big), \qquad x_{k+1} = P_X\big(z_k - \alpha_k \tilde{\nabla} h_{i_k}(z_k)\big). \tag{2.12}$$

(b) Iteration (2.3)-(2.4) can be written as

$$z_k = x_k - \alpha_k \tilde{\nabla} f_{i_k}(z_k), \qquad x_{k+1} = P_X\big(z_k - \alpha_k \tilde{\nabla} h_{i_k}(z_k)\big). \tag{2.13}$$

(c) Iteration (2.5)-(2.6) can be written as

$$z_k = x_k - \alpha_k \tilde{\nabla} h_{i_k}(x_k), \qquad x_{k+1} = P_X\big(z_k - \alpha_k \tilde{\nabla} f_{i_k}(x_{k+1})\big). \tag{2.14}$$

Note that in all the preceding updates, the subgradient $\tilde{\nabla} h_{i_k}$ can be *any* vector in the subdifferential of h_{i_k}, while the subgradient $\tilde{\nabla} f_{i_k}$ must be a *specific* vector in the subdifferential of f_{i_k}, specified according to Prop. 2.1(a). Note also that iteration (2.13) can be written as

$$x_{k+1} = P_X\big(x_k - \alpha_k \tilde{\nabla} F_{i_k}(z_k)\big),$$

and resembles the incremental subgradient method for minimizing over X the cost $F(x) = \sum_{i=1}^m F_i(x)$ [cf. Eq. (1.12)], the only difference being that the subgradient of F_{i_k} is taken at z_k rather than x_k.

An important issue which affects the methods' effectiveness is the order in which the components $\{f_i, h_i\}$ are chosen for iteration. In this paper, we consider two possibilities:

(1) A *cyclic order*, whereby $\{f_i, h_i\}$ are taken up in the fixed deterministic order $1, \ldots, m$, so that i_k is equal to (k modulo m) plus 1. A contiguous block of iterations involving $\{f_1, h_1\}, \ldots, \{f_m, h_m\}$ in this order and exactly once is called a *cycle*. We assume that the stepsize α_k is constant within a cycle (for all k with $i_k = 1$ we have $\alpha_k = \alpha_{k+1} \ldots = \alpha_{k+m-1}$).

(2) A *randomized order based on uniform sampling*, whereby at each iteration a component pair $\{f_i, h_i\}$ is chosen randomly by sampling over all component pairs with a uniform distribution, independently of the past history of the algorithm.

It is essential to include all components in a cycle in the cyclic case, and to sample according to the uniform distribution in the randomized case, for otherwise some components will be sampled more often than others, leading to a bias in the convergence process.

Another technique for incremental methods, popular in neural network training practice, is to reshuffle randomly the order of the component functions after each cycle. This alternative order selection scheme leads to convergence, like the preceding two. Moreover, this scheme has the nice property of allocating exactly one computation slot to each component in an m-slot cycle (m incremental iterations). By comparison, choosing components by uniform sampling allocates one computation slot to each component *on the average*, but some components may not get a slot while others may get more than one. A nonzero variance in the number of slots that any fixed component gets within a cycle, may be detrimental to performance, and indicates that reshuffling randomly the order of the component functions after each cycle works better; this is consistent

with experimental observations shared with us by B. Recht (private communication). While it seems difficult to establish this fact analytically, a justification is suggested by the view of the incremental method as a gradient-like method that uses as descent direction the true gradient at the start of the cycle plus an "error" [due to the calculation of the component gradients at points intermediate within a cycle; cf. Eq. (1.6)]. The error has apparently greater variance in the uniform sampling method than in the randomly shuffled order method (in fact the variance of the error would seem relatively larger as m increases, although other factors such as variance of size of component gradients would also play a role). Heuristically, if the variance of the error is larger, the direction of descent deteriorates, suggesting slower convergence. In this paper, we will focus on the easier-to-analyze uniform sampling method, and show by analysis that it is superior to the cyclic order.

For the remainder of the paper, we denote by F^* the optimal value of problem (1.12):

$$F^* = \inf_{x \in X} F(x),$$

and by X^* the set of optimal solutions (which could be empty):

$$X^* = \big\{x^* \mid x^* \in X,\, F(x^*) = F^*\big\}.$$

Also, for a nonempty closed convex set X, we denote by $\mathrm{dist}(\cdot; X)$ the distance function given by

$$\mathrm{dist}(x; X) = \min_{z \in X} \|x - z\|, \qquad x \in \Re^n.$$

In our convergence analysis of Section 4, we will use the following well-known theorem (see Neveu [Nev75], p. 33). We will use a much simpler deterministic version of the theorem in Section 3.

Proposition 2.2: (Supermartingale Convergence Theorem) Let Y_k, Z_k, and W_k, $k = 0, 1, \ldots$, be three sequences of random variables and let $\mathcal{F}_k$, $k = 0, 1, \ldots$, be sets of random variables such that $\mathcal{F}_k \subset \mathcal{F}_{k+1}$ for all k. Suppose that:

(1) The random variables Y_k, Z_k, and W_k are nonnegative, and are functions of the random variables in $\mathcal{F}_k$.

(2) For each k, we have
$$E\big\{Y_{k+1} \mid \mathcal{F}_k\big\} \leq Y_k - Z_k + W_k.$$

(3) There holds, with probability 1, $\sum_{k=0}^{\infty} W_k < \infty$.

Then we have $\sum_{k=0}^{\infty} Z_k < \infty$, and the sequence Y_k converges to a nonnegative random variable Y, with probability 1.

3. CONVERGENCE FOR METHODS WITH CYCLIC ORDER

In this section, we discuss convergence under the cyclic order. We consider a randomized order in the next section. We focus on the sequence $\{x_k\}$ rather than $\{z_k\}$, which need not lie within X in the case of iterations

(2.13) and (2.14) when $X \neq \Re^n$. In summary, the idea is to show that the effect of taking subgradients of f_i or h_i at points near x_k (e.g., at z_k rather than at x_k) is inconsequential, and diminishes as the stepsize α_k becomes smaller, as long as some subgradients relevant to the algorithms are uniformly bounded in norm by some constant. This is similar to the convergence mechanism of incremental gradient methods described in Section 1.2. We use the following assumptions throughout the present section.

Assumption 3.1: [For iterations (2.12) and (2.13)] There is a constant $c \in \Re$ such that for all k

$$\max\left\{\|\tilde{\nabla} f_{i_k}(z_k)\|, \|\tilde{\nabla} h_{i_k}(z_k)\|\right\} \leq c. \tag{3.1}$$

Furthermore, for all k that mark the beginning of a cycle (i.e., all $k > 0$ with $i_k = 1$), we have

$$\max\left\{f_j(x_k) - f_j(z_{k+j-1}),\, h_j(x_k) - h_j(z_{k+j-1})\right\} \leq c\,\|x_k - z_{k+j-1}\|, \qquad \forall\ j = 1, \ldots, m. \tag{3.2}$$

Assumption 3.2: [For iteration (2.14)] There is a constant $c \in \Re$ such that for all k

$$\max\left\{\|\tilde{\nabla} f_{i_k}(x_{k+1})\|, \|\tilde{\nabla} h_{i_k}(x_k)\|\right\} \leq c. \tag{3.3}$$

Furthermore, for all k that mark the beginning of a cycle (i.e., all $k > 0$ with $i_k = 1$), we have

$$\max\left\{f_j(x_k) - f_j(x_{k+j-1}),\, h_j(x_k) - h_j(x_{k+j-1})\right\} \leq c\,\|x_k - x_{k+j-1}\|, \qquad \forall\ j = 1, \ldots, m, \tag{3.4}$$

$$f_j(x_{k+j-1}) - f_j(x_{k+j}) \leq c\,\|x_{k+j-1} - x_{k+j}\|, \qquad \forall\ j = 1, \ldots, m. \tag{3.5}$$

Note that the condition (3.2) is satisfied if for each i and k, there is a subgradient of f_i at x_k and a subgradient of h_i at x_k, whose norms are bounded by c. Conditions that imply the preceding assumptions are:

(a) For algorithm (2.12): f_i and h_i are Lipschitz continuous over the set X.

(b) For algorithms (2.13) and (2.14): f_i and h_i are Lipschitz continuous over the entire space $\Re^n$.

(c) For all algorithms (2.12), (2.13), and (2.14): f_i and h_i are polyhedral [this is a special case of (a) and (b)].

(d) For all algorithms (2.12), (2.13), and (2.14): The sequences $\{x_k\}$ and $\{z_k\}$ are bounded [since then f_i and h_i, being real-valued and convex, are Lipschitz continuous over any bounded set that contains $\{x_k\}$ and $\{z_k\}$].

The following proposition provides a key estimate that reveals the convergence mechanism of our

methods.†

Proposition 3.1: Let $\{x_k\}$ be the sequence generated by any one of the algorithms (2.12)-(2.14), with a cyclic order of component selection. Then for all $y \in X$ and all k that mark the beginning of a cycle (i.e., all k with $i_k = 1$), we have

$$\|x_{k+m} - y\|^2 \leq \|x_k - y\|^2 - 2\alpha_k\big(F(x_k) - F(y)\big) + \alpha_k^2 \beta m^2 c^2, \tag{3.6}$$

where $\beta = \frac{1}{m} + 4$.

Proof: We first prove the result for algorithms (2.12) and (2.13), and then indicate the modifications necessary for algorithm (2.14). Using Prop. 2.1(b), we have for all $y \in X$ and k,

$$\|z_k - y\|^2 \leq \|x_k - y\|^2 - 2\alpha_k\big(f_{i_k}(z_k) - f_{i_k}(y)\big). \tag{3.7}$$

Also, using the nonexpansion property of the projection [i.e., $\big\|P_X(u) - P_X(v)\big\| \leq \|u - v\|$ for all $u, v \in \Re^n$], the definition of subgradient, and Eq. (3.1), we obtain for all $y \in X$ and k,

$$\begin{aligned} \|x_{k+1} - y\|^2 &= \big\|P_X\big(z_k - \alpha_k \tilde{\nabla} h_{i_k}(z_k)\big) - y\big\|^2 \\ &\leq \|z_k - \alpha_k \tilde{\nabla} h_{i_k}(z_k) - y\|^2 \\ &= \|z_k - y\|^2 - 2\alpha_k \tilde{\nabla} h_{i_k}(z_k)'(z_k - y) + \alpha_k^2 \big\|\tilde{\nabla} h_{i_k}(z_k)\big\|^2 \\ &\leq \|z_k - y\|^2 - 2\alpha_k\big(h_{i_k}(z_k) - h_{i_k}(y)\big) + \alpha_k^2 c^2. \end{aligned} \tag{3.8}$$

Combining Eqs. (3.7) and (3.8), and using the definition $F_j = f_j + h_j$, we have

$$\begin{aligned} \|x_{k+1} - y\|^2 &\leq \|x_k - y\|^2 - 2\alpha_k\big(f_{i_k}(z_k) + h_{i_k}(z_k) - f_{i_k}(y) - h_{i_k}(y)\big) + \alpha_k^2 c^2 \\ &= \|x_k - y\|^2 - 2\alpha_k\big(F_{i_k}(z_k) - F_{i_k}(y)\big) + \alpha_k^2 c^2. \end{aligned} \tag{3.9}$$

Let now k mark the beginning of a cycle (i.e., $i_k = 1$). Then at iteration $k + j - 1$, $j = 1, \ldots, m$, the selected components are $\{f_j, h_j\}$, in view of the assumed cyclic order. We may thus replicate the preceding inequality with k replaced by $k + 1, \ldots, k + m - 1$, and add to obtain

$$\|x_{k+m} - y\|^2 \leq \|x_k - y\|^2 - 2\alpha_k \sum_{j=1}^{m} \big(F_j(z_{k+j-1}) - F_j(y)\big) + m\alpha_k^2 c^2,$$

or equivalently, since $F = \sum_{j=1}^m F_j$,

$$\|x_{k+m} - y\|^2 \leq \|x_k - y\|^2 - 2\alpha_k\big(F(x_k) - F(y)\big) + m\alpha_k^2 c^2 + 2\alpha_k \sum_{j=1}^{m} \big(F_j(x_k) - F_j(z_{k+j-1})\big). \tag{3.10}$$

† The original version of this report gave $\beta = \frac{1}{m} + 4$ for the case of algorithms (2.12) and (2.13), and $\beta = \frac{5}{m} + 4$ for the case of algorithm (2.14), because a loose bound was used in the following calculation. The tighter version for algorithm (2.14) given here was prompted by an observation by M. Andersen and P. C. Hansen in Oct. 2013.

The remainder of the proof deals with appropriately bounding the last term above.

From Eq. (3.2), we have for $j = 1, \ldots, m$,

$$F_j(x_k) - F_j(z_{k+j-1}) \leq 2c \, \|x_k - z_{k+j-1}\|. \tag{3.11}$$

We also have

$$\|x_k - z_{k+j-1}\| \leq \|x_k - x_{k+1}\| + \cdots + \|x_{k+j-2} - x_{k+j-1}\| + \|x_{k+j-1} - z_{k+j-1}\|, \tag{3.12}$$

and by the definition of the algorithms (2.12) and (2.13), the nonexpansion property of the projection, and Eq. (3.1), each of the terms in the right-hand side above is bounded by $2\alpha_k c$, except for the last, which is bounded by $\alpha_k c$. Thus Eq. (3.12) yields $\|x_k - z_{k+j-1}\| \leq \alpha_k(2j-1)c$, which together with Eq. (3.11), shows that

$$F_j(x_k) - F_j(z_{k+j-1}) \leq 2\alpha_k c^2(2j-1). \tag{3.13}$$

Combining Eqs. (3.10) and (3.13), we have

$$\|x_{k+m} - y\|^2 \leq \|x_k - y\|^2 - 2\alpha_k\big(F(x_k) - F(y)\big) + m\alpha_k^2 c^2 + 4\alpha_k^2 c^2 \sum_{j=1}^{m}(2j-1),$$

and finally

$$\|x_{k+m} - y\|^2 \leq \|x_k - y\|^2 - 2\alpha_k\big(F(x_k) - F(y)\big) + m\alpha_k^2 c^2 + 4\alpha_k^2 c^2 m^2,$$

which is of the form (3.6) with $\beta = \frac{1}{m} + 4$.

For the algorithm (2.14), a similar argument goes through using Assumption 3.2. In place of Eq. (3.7), using the nonexpansion property of the projection, the definition of subgradient, and Eq. (3.3), we obtain for all $y \in X$ and $k \geq 0$,

$$\|z_k - y\|^2 \leq \|x_k - y\|^2 - 2\alpha_k\big(h_{i_k}(x_k) - h_{i_k}(y)\big) + \alpha_k^2 c^2, \tag{3.14}$$

while in place of Eq. (3.8), using Prop. 2.1(b), we have

$$\|x_{k+1} - y\|^2 \leq \|z_k - y\|^2 - 2\alpha_k\big(f_{i_k}(x_{k+1}) - f_{i_k}(y)\big). \tag{3.15}$$

Combining these equations, in analogy with Eq. (3.9), we obtain

$$\begin{aligned} \|x_{k+1} - y\|^2 &\leq \|x_k - y\|^2 - 2\alpha_k\big(f_{i_k}(x_{k+1}) + h_{i_k}(x_k) - f_{i_k}(y) - h_{i_k}(y)\big) + \alpha_k^2 c^2 \\ &= \|x_k - y\|^2 - 2\alpha_k\big(F_{i_k}(x_k) - F_{i_k}(y)\big) + \alpha_k^2 c^2 + 2\alpha_k\big(f_{i_k}(x_k) - f_{i_k}(x_{k+1})\big). \end{aligned} \tag{3.16}$$

As earlier, we let k mark the beginning of a cycle (i.e., $i_k = 1$). We replicate the preceding inequality with k replaced by $k+1, \ldots, k+m-1$, and add to obtain [in analogy with Eq. (3.10)]

$$\begin{aligned} \|x_{k+m} - y\|^2 \leq \|x_k - y\|^2 &- 2\alpha_k\big(F(x_k) - F(y)\big) + m\alpha_k^2 c^2 \\ &+ 2\alpha_k \sum_{j=1}^{m}\big(F_j(x_k) - F_j(x_{k+j-1})\big) + 2\alpha_k \sum_{j=1}^{m}\big(f_j(x_{k+j-1}) - f_j(x_{k+j})\big). \end{aligned} \tag{3.17}$$

We now bound the two sums in Eq. (3.17), using Assumption 3.2. From Eq. (3.4), we have

$$F_j(x_k) - F_j(x_{k+j-1}) \leq 2c\|x_k - x_{k+j-1}\| \leq 2c\big(\|x_k - x_{k+1}\| + \cdots + \|x_{k+j-2} - x_{k+j-1}\|\big),$$

and since by Eq. (3.3) and the definition of the algorithm, each of the norm terms in the right-hand side above is bounded by $2\alpha_k c$,

$$F_j(x_k) - F_j(x_{k+j-1}) \leq 4\alpha_k c^2(j-1).$$

Also from Eqs. (3.3) and (3.5), and the nonexpansion property of the projection, we have

$$f_j(x_{k+j-1}) - f_j(x_{k+j}) \leq c\,\|x_{k+j-1} - x_{k+j}\| \leq 2\alpha_k c^2.$$

Combining the preceding relations and adding, we obtain

$$\begin{aligned} 2\alpha_k \sum_{j=1}^{m} \big(F_j(x_k) - F_j(x_{k+j-1})\big) + 2\alpha_k \sum_{j=1}^{m} \big(f_j(x_{k+j-1}) - f_j(x_{k+j})\big) &\leq 8\alpha_k^2 c^2 \sum_{j=1}^{m}(j-1) + 4\alpha_k^2 c^2 m \\ &= 4\alpha_k^2 c^2(m^2-m) + 4\alpha_k^2 c^2 m = \left(4 + \frac{1}{m}\right)\alpha_k^2 c^2 m^2, \end{aligned}$$

which together with Eq. (3.17), yields Eq. (3.6). **Q.E.D.**

Among other things, Prop. 3.1 guarantees that with a cyclic order, given the iterate x_k at the start of a cycle and any point $y \in X$ having lower cost than x_k (for example an optimal point), the algorithm yields a point x_{k+m} at the end of the cycle that will be closer to y than x_k, provided the stepsize α_k is less than

$$\frac{2\big(F(x_k) - F(y)\big)}{\beta m^2 c^2}.$$

In particular, for any $\epsilon > 0$ and assuming that there exists an optimal solution x^*, either we are within $\frac{\alpha_k \beta m^2 c^2}{2} + \epsilon$ of the optimal value,

$$F(x_k) \leq F(x^*) + \frac{\alpha_k \beta m^2 c^2}{2} + \epsilon,$$

or else the squared distance to x^* will be strictly decreased by at least $2\alpha_k\epsilon$,

$$\|x_{k+m} - x^*\|^2 < \|x_k - x^*\|^2 - 2\alpha_k\epsilon.$$

Thus, using Prop. 3.1, we can provide various types of convergence results. As an example, for a constant stepsize ($\alpha_k \equiv \alpha$), convergence can be established to a neighborhood of the optimum, which shrinks to 0 as $\alpha \to 0$, as stated in the following proposition.

Proposition 3.2: Let $\{x_k\}$ be the sequence generated by any one of the algorithms (2.12)-(2.14), with a cyclic order of component selection, and let the stepsize α_k be fixed at some positive constant α.

(a) If $F^* = -\infty$, then

$$\liminf_{k\to\infty} F(x_k) = F^*.$$

(b) If $F^* > -\infty$, then

$$\liminf_{k\to\infty} F(x_k) \leq F^* + \frac{\alpha\beta m^2 c^2}{2},$$

where c and β are the constants of Prop. 3.1.

Proof: We prove (a) and (b) simultaneously. If the result does not hold, there must exist an $\epsilon > 0$ such that

$$\liminf_{k\to\infty} F(x_{km}) - \frac{\alpha\beta m^2 c^2}{2} - 2\epsilon > F^*.$$

Let $\hat{y} \in X$ be such that

$$\liminf_{k\to\infty} F(x_{km}) - \frac{\alpha\beta m^2 c^2}{2} - 2\epsilon \geq F(\hat{y}),$$

and let k_0 be large enough so that for all $k \geq k_0$, we have

$$F(x_{km}) \geq \liminf_{k\to\infty} F(x_{km}) - \epsilon.$$

By combining the preceding two relations, we obtain for all $k \geq k_0$,

$$F(x_{km}) - F(\hat{y}) \geq \frac{\alpha\beta m^2 c^2}{2} + \epsilon.$$

Using Prop. 3.1 for the case where $y = \hat{y}$ together with the above relation, we obtain for all $k \geq k_0$,

$$\|x_{(k+1)m} - \hat{y}\|^2 \leq \|x_{km} - \hat{y}\|^2 - 2\alpha\big(F(x_{km}) - F(\hat{y})\big) + \beta\alpha^2 m^2 c^2 \leq \|x_{km} - \hat{y}\|^2 - 2\alpha\epsilon.$$

This relation implies that for all $k \geq k_0$,

$$\|x_{(k+1)m} - \hat{y}\|^2 \leq \|x_{(k-1)m} - \hat{y}\|^2 - 4\alpha\epsilon \leq \cdots \leq \|x_{k_0} - \hat{y}\|^2 - 2(k+1-k_0)\alpha\epsilon,$$

which cannot hold for k sufficiently large – a contradiction. **Q.E.D.**

The next proposition gives an estimate of the number of iterations needed to guarantee a given level of optimality up to the threshold tolerance $\alpha\beta m^2 c^2/2$ of the preceding proposition.

Proposition 3.3: Assume that X^* is nonempty. Let $\{x_k\}$ be a sequence generated as in Prop. 3.2. Then for $\epsilon > 0$, we have

$$\min_{0\leq k\leq N} F(x_k) \leq F^* + \frac{\alpha\beta m^2 c^2 + \epsilon}{2}, \tag{3.18}$$

where N is given by

$$N = m \left\lfloor \frac{\mathrm{dist}(x_0; X^*)^2}{\alpha\epsilon} \right\rfloor. \tag{3.19}$$

Proof: Assume, to arrive at a contradiction, that Eq. (3.18) does not hold, so that for all k with $0 \leq km \leq N$, we have

$$F(x_{km}) > F^* + \frac{\alpha\beta m^2 c^2 + \epsilon}{2}.$$

By using this relation in Prop. 3.1 with α_k replaced by α and y equal to the vector of X^* that is at minimum distance from x_{km}, we obtain for all k with $0 \leq km \leq N$,

$$\begin{aligned}
\mathrm{dist}(x_{(k+1)m}; X^*)^2 &\leq \mathrm{dist}(x_{km}; X^*)^2 - 2\alpha\big(F(x_{km}) - F^*\big) + \alpha^2\beta m^2 c^2 \\
&\leq \mathrm{dist}(x_{km}; X^*)^2 - (\alpha^2\beta m^2 c^2 + \alpha\epsilon) + \alpha^2\beta m^2 c^2 \\
&= \mathrm{dist}(x_{km}; X^*)^2 - \alpha\epsilon.
\end{aligned}$$

Adding the above inequalities for $k = 0, \ldots, \frac{N}{m}$, we obtain

$$\text{dist}(x_{N+m}; X^*)^2 \leq \text{dist}(x_0; X^*)^2 - \left(\frac{N}{m} + 1\right) \alpha\epsilon,$$

so that

$$\left(\frac{N}{m} + 1\right) \alpha\epsilon \leq \text{dist}(x_0; X^*)^2,$$

which contradicts the definition of N. **Q.E.D.**

According to Prop. 3.3, to achieve a cost function value within $O(\epsilon)$ of the optimal, the term $\alpha\beta m^2 c^2$ must also be of order $O(\epsilon)$, so α must be of order $O(\epsilon/m^2c^2)$, and from Eq. (3.19), the number of necessary iterations N is $O(m^3c^2/\epsilon^2)$, and the number of necessary cycles is $O\big((mc)^2/\epsilon^2\big)$. This is the same type of estimate as for the nonincremental subgradient method [i.e., $O(1/\epsilon^2)$, counting a cycle as one iteration of the nonincremental method, and viewing mc as a Lipschitz constant for the entire cost function F], and does not reveal any advantage for the incremental methods given here. However, in the next section, we demonstrate a much more favorable iteration complexity estimate for the incremental methods that use a randomized order of component selection.

Exact Convergence for a Diminishing Stepsize

We can also obtain an exact convergence result for the case where the stepsize α_k diminishes to zero. The idea is that with a constant stepsize α we can get to within an $O(\alpha)$-neighborhood of the optimum, as shown above, so with a diminishing stepsize α_k, we should be able to reach an arbitrarily small neighborhood of the optimum. However, for this to happen, α_k should not be reduced too fast, and should satisfy $\sum_{k=0}^{\infty} \alpha_k = \infty$ (so that the method can "travel" infinitely far if necessary).

Proposition 3.4: Let $\{x_k\}$ be the sequence generated by any one of the algorithms (2.12)-(2.14), with a cyclic order of component selection, and let the stepsize α_k satisfy

$$\lim_{k\to\infty} \alpha_k = 0, \qquad \sum_{k=0}^{\infty} \alpha_k = \infty.$$

Then,

$$\liminf_{k\to\infty} F(x_k) = F^*.$$

Furthermore, if X^* is nonempty and

$$\sum_{k=0}^{\infty} \alpha_k^2 < \infty,$$

then $\{x_k\}$ converges to some $x^* \in X^*$.

Proof: For the first part, it will be sufficient to show that $\liminf_{k\to\infty} F(x_{km}) = F^*$. Assume, to arrive at a contradiction, that there exists an $\epsilon > 0$ such that

$$\liminf_{k\to\infty} F(x_{km}) - 2\epsilon > F^*.$$

Then there exists a point $\hat{y} \in X$ such that

$$\liminf_{k\to\infty} F(x_{km}) - 2\epsilon > F(\hat{y}).$$

Let k_0 be large enough so that for all $k \geq k_0$, we have

$$F(x_{km}) \geq \liminf_{k\to\infty} F(x_{km}) - \epsilon.$$

By combining the preceding two relations, we obtain for all $k \geq k_0$,

$$F(x_{km}) - F(\hat{y}) > \epsilon.$$

By setting $y = \hat{y}$ in Prop. 3.1, and by using the above relation, we have for all $k \geq k_0$,

$$\|x_{(k+1)m} - \hat{y}\|^2 \leq \|x_{km} - \hat{y}\|^2 - 2\alpha_{km}\epsilon + \beta\alpha_{km}^2 m^2 c^2 = \|x_{km} - \hat{y}\|^2 - \alpha_{km}\left(2\epsilon - \beta\alpha_{km}m^2c^2\right).$$

Since $\alpha_k \to 0$, without loss of generality, we may assume that k_0 is large enough so that

$$2\epsilon - \beta\alpha_k m^2 c^2 \geq \epsilon, \qquad \forall\ k \geq k_0.$$

Therefore for all $k \geq k_0$, we have

$$\|x_{(k+1)m} - \hat{y}\|^2 \leq \|x_{km} - \hat{y}\|^2 - \alpha_{km}\epsilon \leq \cdots \leq \|x_{k_0 m} - \hat{y}\|^2 - \epsilon \sum_{\ell=k_0}^{k} \alpha_{\ell m},$$

which cannot hold for k sufficiently large. Hence $\liminf_{k\to\infty} F(x_{km}) = F^*$.

To prove the second part of the proposition, note that from Prop. 3.1, for every $x^* \in X^*$ and $k \geq 0$ we have

$$\|x_{(k+1)m} - x^*\|^2 \leq \|x_{km} - x^*\|^2 - 2\alpha_{km}\big(F(x_{km}) - F(x^*)\big) + \alpha_{km}^2\beta m^2 c^2. \tag{3.20}$$

The Supermartingale Convergence Theorem (Prop. 2.2)† and the hypothesis $\sum_{k=0}^{\infty}\alpha_k^2 < \infty$, imply that $\big\{\|x_{km} - x^*\|\big\}$ converges for every $x^* \in X^*$. Since then $\{x_{km}\}$ is bounded, it has a limit point $\bar{x} \in X$ that satisfies

$$F(\bar{x}) = \liminf_{k\to\infty} F(x_{km}) = F^*.$$

This implies that $\bar{x} \in X^*$, so it follows that $\big\{\|x_{km} - \bar{x}\|\big\}$ converges, and that the entire sequence $\{x_{km}\}$ converges to $\bar{x}$ (since $\bar{x}$ is a limit point of $\{x_{km}\}$).

Finally, to show that the entire sequence $\{x_k\}$ also converges to $\bar{x}$, note that from Eqs. (3.1) and (3.3), and the form of the iterations (2.12)-(2.14), we have $\|x_{k+1} - x_k\| \leq 2\alpha_k c \to 0$. Since $\{x_{km}\}$ converges to $\bar{x}$, it follows that $\{x_k\}$ also converges to $\bar{x}$. **Q.E.D.**

† Actually we use here a deterministic version/special case of the theorem, where Y_k, Z_k, and W_k are nonnegative scalar sequences satisfying $Y_{k+1} \leq Y_k - Z_k + W_k$ with $\sum_{k=0}^{\infty} W_k < \infty$. Then the sequence Y_k must converge. This version is given with proof in many sources, including [BeT96] (Lemma 3.4), and [BeT00] (Lemma 1).

4. CONVERGENCE FOR METHODS WITH RANDOMIZED ORDER

In this section, we discuss convergence for the randomized component selection order and a constant stepsize α. The randomized versions of iterations (2.12), (2.13), and (2.14), are

$$z_k = P_X\big(x_k - \alpha\tilde{\nabla} f_{\omega_k}(z_k)\big), \qquad x_{k+1} = P_X\big(z_k - \alpha\tilde{\nabla} h_{\omega_k}(z_k)\big), \tag{4.1}$$

$$z_k = x_k - \alpha\tilde{\nabla} f_{\omega_k}(z_k), \qquad x_{k+1} = P_X\big(z_k - \alpha\tilde{\nabla} h_{\omega_k}(z_k)\big), \tag{4.2}$$

$$z_k = P_X\big(x_k - \alpha\tilde{\nabla} h_{\omega_k}(z_k)\big), \qquad x_{k+1} = z_k - \alpha\tilde{\nabla} f_{\omega_k}(x_{k+1}), \tag{4.3}$$

respectively, where $\{\omega_k\}$ is a sequence of random variables, taking values from the index set $\{1,\ldots,m\}$.

We assume the following throughout the present section.

Assumption 4.1: [For iterations (4.1) and (4.2)]

(a) $\{\omega_k\}$ is a sequence of random variables, each uniformly distributed over $\{1,\ldots,m\}$, and such that for each k, ω_k is independent of the past history $\{x_k, z_{k-1}, x_{k-1}, \ldots, z_0, x_0\}$.

(b) There is a constant $c \in \Re$ such that for all k, we have with probability 1

$$\max\big\{\|\tilde{\nabla} f_i(z_k^i)\|,\, \|\tilde{\nabla} h_i(z_k^i)\|\big\} \le c, \qquad \forall\ i = 1,\ldots,m, \tag{4.4}$$

$$\max\big\{f_i(x_k) - f_i(z_k^i),\, h_i(x_k) - h_i(z_k^i)\big\} \le c\|x_k - z_k^i\|, \qquad \forall\ i = 1,\ldots,m, \tag{4.5}$$

where z_k^i is the result of the proximal iteration, starting at x_k if ω_k would be i, i.e.,

$$z_k^i = \arg\min_{x\in X}\left\{f_i(x) + \frac{1}{2\alpha_k}\|x - x_k\|^2\right\},$$

in the case of iteration (4.1), and

$$z_k^i = \arg\min_{x\in \Re^n}\left\{f_i(x) + \frac{1}{2\alpha_k}\|x - x_k\|^2\right\},$$

in the case of iteration (4.2).

Assumption 4.2: [For iteration (4.3)]

(a) $\{\omega_k\}$ is a sequence of random variables, each uniformly distributed over $\{1,\ldots,m\}$, and such that for each k, ω_k is independent of the past history $\{x_k, z_{k-1}, x_{k-1}, \ldots, z_0, x_0\}$.

(b) There is a constant $c \in \Re$ such that for all k, we have with probability 1

$$\max\big\{\|\tilde{\nabla} f_i(x^i_{k+1})\|,\, \|\tilde{\nabla} h_i(x_k)\|\big\} \leq c, \qquad \forall\ i = 1,\ldots,m, \tag{4.6}$$

$$f_i(x_k) - f_i(x^i_{k+1}) \leq c\|x_k - x^i_{k+1}\|, \qquad \forall\ i = 1,\ldots,m, \tag{4.7}$$

where x^i_{k+1} is the result of the iteration, starting at x_k if ω_k would be i, i.e.,

$$x^i_{k+1} = P_X\big(z^i_k - \alpha_k \tilde{\nabla} f_i(x^i_{k+1})\big),$$

with

$$z^i_k = x_k - \alpha_k \tilde{\nabla} h_i(x_k).$$

Note that condition (4.5) is satisfied if there exist subgradients of f_i and h_i at x_k with norms less or equal to c. Thus the conditions (4.4) and (4.5) are similar, the main difference being that the first applies to "slopes" of f_i and h_i at z^i_k while the second applies to the "slopes" of f_i and h_i at x_k. As in the case of Assumption 3.1, these conditions are guaranteed by Lipschitz continuity assumptions on f_i and h_i. The convergence analysis of the randomized algorithms of this section is somewhat more complicated than the one of the cyclic order counterparts, and relies on the Supermartingale Convergence Theorem. The following proposition deals with the case of a constant stepsize, and parallels Prop. 3.2 for the cyclic order case.

Proposition 4.1: Let $\{x_k\}$ be the sequence generated by one of the randomized incremental methods (4.1)-(4.3), and let the stepsize α_k be fixed at some positive constant α.

(a) If $F^* = -\infty$, then with probability 1

$$\inf_{k\geq 0} F(x_k) = F^*.$$

(b) If $F^* > -\infty$, then with probability 1

$$\inf_{k\geq 0} F(x_k) \leq F^* + \frac{\alpha\beta m c^2}{2},$$

where $\beta = 5$.

Proof: Consider first algorithms (4.1) and (4.2). By adapting the proof argument of Prop. 3.1 with F_{i_k}

replaced by F_{ω_k} [cf. Eq. (3.9)], we have

$$\|x_{k+1}-y\|^2 \le \|x_k-y\|^2 - 2\alpha\big(F_{\omega_k}(z_k)-F_{\omega_k}(y)\big)+\alpha^2c^2, \quad \forall\, y\in X, \quad k\ge 0.$$

By taking the conditional expectation with respect to $\mathcal{F}_k=\{x_k,z_{k-1},\ldots,z_0,x_0\}$, and using the fact that ω_k takes the values $i=1,\ldots,m$ with equal probability $1/m$, we obtain for all $y\in X$ and k,

$$\begin{aligned} E\big\{\|x_{k+1}-y\|^2 \mid \mathcal{F}_k\big\} &\le \|x_k-y\|^2 - 2\alpha E\big\{F_{\omega_k}(z_k)-F_{\omega_k}(y)\mid \mathcal{F}_k\big\}+\alpha^2c^2 \\ &= \|x_k-y\|^2 - \frac{2\alpha}{m}\sum_{i=1}^m\big(F_i(z_k^i)-F_i(y)\big)+\alpha^2c^2 \\ &= \|x_k-y\|^2 - \frac{2\alpha}{m}\big(F(x_k)-F(y)\big)+\frac{2\alpha}{m}\sum_{i=1}^m\big(F_i(x_k)-F_i(z_k^i)\big)+\alpha^2c^2. \end{aligned} \tag{4.8}$$

By using Eqs. (4.4) and (4.5),

$$\sum_{i=1}^m\big(F_i(x_k)-F_i(z_k^i)\big)\le 2c\sum_{i=1}^m\|x_k-z_k^i\| = 2c\alpha\sum_{i=1}^m\|\tilde\nabla f_i(z_k^i)\|\le 2m\alpha c^2.$$

By combining the preceding two relations, we obtain

$$\begin{aligned} E\big\{\|x_{k+1}-y\|^2 \mid \mathcal{F}_k\big\} &\le \|x_k-y\|^2 - \frac{2\alpha}{m}\big(F(x_k)-F(y)\big)+4\alpha^2c^2+\alpha^2c^2 \\ &= \|x_k-y\|^2 - \frac{2\alpha}{m}\big(F(x_k)-F(y)\big)+\beta\alpha^2c^2, \end{aligned} \tag{4.9}$$

where $\beta=5$.

The preceding equation holds also for algorithm (4.3). To see this note that Eq. (3.16) yields for all $y\in X$

$$\|x_{k+1}-y\|^2 \le \|x_k-y\|^2 - 2\alpha\big(F_{\omega_k}(x_k)-F_{\omega_k}(y)\big)+\alpha^2c^2+2\alpha\big(f_{\omega_k}(x_k)-f_{\omega_k}(x_{k+1})\big), \tag{4.10}$$

and similar to Eq. (4.8), we obtain

$$E\big\{\|x_{k+1}-y\|^2 \mid \mathcal{F}_k\big\} \le \|x_k-y\|^2 - \frac{2\alpha}{m}\big(F(x_k)-F(y)\big)+\frac{2\alpha}{m}\sum_{i=1}^m\big(f_i(x_k)-f_i(x_{k+1}^i)\big)+\alpha^2c^2. \tag{4.11}$$

From Eq. (4.7), we have

$$f_i(x_k)-f_i(x_{k+1}^i)\le c\|x_k-x_{k+1}^i\|,$$

and from Eq. (4.6) and the nonexpansion property of the projection,

$$\|x_k-x_{k+1}^i\| \le \big\|x_k-z_k^i+\alpha\tilde\nabla f_i(x_{k+1}^i)\big\| = \big\|x_k-x_k+\alpha\tilde\nabla h_i(x_k)+\alpha\tilde\nabla f_i(x_{k+1}^i)\big\|\le 2\alpha c.$$

Combining the preceding inequalities, we obtain Eq. (4.9) with $\beta=5$.

Let us fix a positive scalar γ, consider the level set L_γ defined by

$$L_\gamma = \begin{cases} \Big\{x\in X \mid F(x)<-\gamma+1+\frac{\alpha\beta mc^2}{2}\Big\} & \text{if } F^*=-\infty, \\ \Big\{x\in X \mid F(x)<F^*+\frac{2}{\gamma}+\frac{\alpha\beta mc^2}{2}\Big\} & \text{if } F^*>-\infty, \end{cases}$$

and let $y_\gamma \in X$ be such that

$$F(y_\gamma) = \begin{cases} -\gamma & \text{if } F^* = -\infty, \\ F^* + \frac{1}{\gamma} & \text{if } F^* > -\infty. \end{cases}$$

Note that $y_\gamma \in L_\gamma$ by construction. Define a new process $\{\hat{x}_k\}$ that is identical to $\{x_k\}$, except that once x_k enters the level set L_γ, the process terminates with $\hat{x}_k = y_\gamma$. We will now argue that for any fixed γ, $\{\hat{x}_k\}$ (and hence also $\{x_k\}$) will eventually enter L_γ, which will prove both parts (a) and (b).

Using Eq. (4.9) with $y = y_\gamma$, we have

$$E\big\{\|\hat{x}_{k+1} - y_\gamma\|^2 \mid \mathcal{F}_k\big\} \leq \|\hat{x}_k - y_\gamma\|^2 - \frac{2\alpha}{m}\big(F(\hat{x}_k) - F(y_\gamma)\big) + \beta\alpha^2 c^2,$$

from which

$$E\big\{\|\hat{x}_{k+1} - y_\gamma\|^2 \mid \mathcal{F}_k\big\} \leq \|\hat{x}_k - y_\gamma\|^2 - v_k, \tag{4.12}$$

where

$$v_k = \begin{cases} \frac{2\alpha}{m}\big(F(\hat{x}_k) - F(y_\gamma)\big) - \beta\alpha^2 c^2 & \text{if } \hat{x}_k \notin L_\gamma, \\ 0 & \text{if } \hat{x}_k = y_\gamma. \end{cases}$$

The idea of the subsequent argument is to show that as long as $\hat{x}_k \notin L_\gamma$, the scalar v_k (which is a measure of progress) is strictly positive and bounded away from 0.

(a) Let $F^* = -\infty$. Then if $\hat{x}_k \notin L_\gamma$, we have

$$\begin{aligned} v_k &= \frac{2\alpha}{m}\big(F(\hat{x}_k) - F(y_\gamma)\big) - \beta\alpha^2 c^2 \\ &\geq \frac{2\alpha}{m}\left(-\gamma + 1 + \frac{\alpha\beta m c^2}{2} + \gamma\right) - \beta\alpha^2 c^2 \\ &= \frac{2\alpha}{m}. \end{aligned}$$

Since $v_k = 0$ for $\hat{x}_k \in L_\gamma$, we have $v_k \geq 0$ for all k, and by Eq. (4.12) and the Supermartingale Convergence Theorem (cf. Prop. 2.2), $\sum_{k=0}^{\infty} v_k < \infty$ implying that $\hat{x}_k \in L_\gamma$ for sufficiently large k, with probability 1. Therefore, in the original process we have

$$\inf_{k\geq 0} F(x_k) \leq -\gamma + 1 + \frac{\alpha\beta m c^2}{2}$$

with probability 1. Letting $\gamma \to \infty$, we obtain $\inf_{k\geq 0} F(x_k) = -\infty$ with probability 1.

(b) Let $F^* > -\infty$. Then if $\hat{x}_k \notin L_\gamma$, we have

$$\begin{aligned} v_k &= \frac{2\alpha}{m}\big(F(\hat{x}_k) - F(y_\gamma)\big) - \beta\alpha^2 c^2 \\ &\geq \frac{2\alpha}{m}\left(F^* + \frac{2}{\gamma} + \frac{\alpha\beta m c^2}{2} - F^* - \frac{1}{\gamma}\right) - \beta\alpha^2 c^2 \\ &= \frac{2\alpha}{m\gamma}. \end{aligned}$$

Hence, $v_k \geq 0$ for all k, and by the Supermartingale Convergence Theorem, we have $\sum_{k=0}^{\infty} v_k < \infty$ implying that $\hat{x}_k \in L_\gamma$ for sufficiently large k, so that in the original process,

$$\inf_{k\geq 0} F(x_k) \leq F^* + \frac{2}{\gamma} + \frac{\alpha\beta m c^2}{2}$$

with probability 1. Letting $\gamma \to \infty$, we obtain $\inf_{k\geq 0} F(x_k) \leq F^* + \alpha\beta mc^2/2$. **Q.E.D.**

By comparing Prop. 4.1(b) with Prop. 3.2(b), we see that when $F^* > -\infty$ and the stepsize α is constant, the randomized methods (4.1), (4.2), and (4.3), have a better error bound (by a factor m) than their nonrandomized counterparts. In fact an example given in p. 514 of [BNO03] for the incremental subgradient method can be adapted to show that the bound of Prop. 3.2(b) is tight in the sense that for a bad problem/cyclic order we have $\liminf_{k\to\infty} F(x_k) - F^* = O(\alpha m^2 c^2)$. By contrast the randomized method will get to within $O(\alpha mc^2)$ with probability 1 for any problem, according to Prop. 4.1(b). Thus with the randomized algorithm we do not run the risk of choosing by accident a bad cyclic order. A related result is provided by the following proposition, which should be compared with Prop. 3.3 for the nonrandomized methods.

Proposition 4.2: Assume that X^* is nonempty. Let $\{x_k\}$ be a sequence generated as in Prop. 4.1. Then for any positive scalar ϵ, we have with probability 1

$$\min_{0\leq k\leq N} F(x_k) \leq F^* + \frac{\alpha\beta mc^2 + \epsilon}{2}, \tag{4.13}$$

where N is a random variable with

$$E\{N\} \leq m\,\frac{\mathrm{dist}(x_0; X^*)^2}{\alpha\epsilon}. \tag{4.14}$$

Proof: Let $\hat{y}$ be some fixed vector in X^*. Define a new process $\{\hat{x}_k\}$ which is identical to $\{x_k\}$ except that once x_k enters the level set

$$L = \left\{ x \in X \;\middle|\; F(x) < F^* + \frac{\alpha\beta mc^2 + \epsilon}{2} \right\},$$

the process $\{\hat{x}_k\}$ terminates at $\hat{y}$. Similar to the proof of Prop. 4.1 [cf. Eq. (4.9) with y being the closest point of $\hat{x}_k$ in X^*], for the process $\{\hat{x}_k\}$ we obtain for all k,

$$\begin{aligned} E\{\mathrm{dist}(\hat{x}_{k+1}; X^*)^2 \mid \mathcal{F}_k\} &\leq E\{\|\hat{x}_{k+1} - y\|^2 \mid \mathcal{F}_k\} \\ &\leq \mathrm{dist}(\hat{x}_k; X^*)^2 - \frac{2\alpha}{m}\big(F(\hat{x}_k) - F^*\big) + \beta\alpha^2 c^2 \\ &= \mathrm{dist}(\hat{x}_k; X^*)^2 - v_k, \end{aligned} \tag{4.15}$$

where $\mathcal{F}_k = \{x_k, z_{k-1}, \ldots, z_0, x_0\}$ and

$$v_k = \begin{cases} \frac{2\alpha}{m}\big(F(\hat{x}_k) - F^*\big) - \beta\alpha^2 c^2 & \text{if } \hat{x}_k \notin L, \\ 0 & \text{otherwise.} \end{cases}$$

In the case where $\hat{x}_k \notin L$, we have

$$v_k \geq \frac{2\alpha}{m}\left(F^* + \frac{\alpha\beta mc^2 + \epsilon}{2} - F^*\right) - \beta\alpha^2 c^2 \;= \frac{\alpha\epsilon}{m}. \tag{4.16}$$

By the Supermartingale Convergence Theorem (cf. Prop. 2.2), from Eq. (4.15) we have

$$\sum_{k=0}^{\infty} v_k < \infty$$

with probability 1, so that $v_k = 0$ for all $k \geq N$, where N is a random variable. Hence $\hat{x}_N \in L$ with probability 1, implying that in the original process we have

$$\min_{0 \leq k \leq N} F(x_k) \leq F^* + \frac{\alpha\beta mc^2 + \epsilon}{2}$$

with probability 1. Furthermore, by taking the total expectation in Eq. (4.15), we obtain for all k,

$$E\big\{\mathrm{dist}(\hat{x}_{k+1}; X^*)^2\big\} \leq E\big\{\mathrm{dist}(\hat{x}_k; X^*)^2\big\} - E\{v_k\} \leq \mathrm{dist}(\hat{x}_0; X^*)^2 - E\left\{\sum_{j=0}^{k} v_j\right\},$$

where in the last inequality we use the facts $\hat{x}_0 = x_0$ and $E\big\{\mathrm{dist}(\hat{x}_0; X^*)^2\big\} = \mathrm{dist}(\hat{x}_0; X^*)^2$. Therefore, letting $k \to \infty$, and using the definition of v_k and Eq. (4.16),

$$\mathrm{dist}(\hat{x}_0; X^*)^2 \geq E\left\{\sum_{k=0}^{\infty} v_k\right\} = E\left\{\sum_{k=0}^{N-1} v_k\right\} \geq E\left\{\frac{N\alpha\epsilon}{m}\right\} = \frac{\alpha\epsilon}{m} E\big\{N\big\}.$$

Q.E.D.

Like Prop. 4.1, a comparison of Props. 3.3 and 4.2 again suggests an advantage for the randomized methods: compared to their deterministic counterparts, they achieve a much smaller error tolerance (a factor of m), in the same *expected* number of iterations. Note, however, that the preceding assessment is based on upper bound estimates, which may not be sharp on a given problem [although the bound of Prop. 3.2(b) is tight with a worst-case problem selection as mentioned earlier; see [BNO03], p. 514]. Moreover, the comparison based on worst-case values versus expected values may not be strictly valid. In particular, while Prop. 3.3 provides an upper bound estimate on N, Prop. 4.2 provides an upper bound estimate on $E\{N\}$, which is not quite the same.

Finally for the case of a diminishing stepsize, let us give the following proposition, which parallels Prop. 3.4 for the cyclic order.

Proposition 4.3: Let $\{x_k\}$ be the sequence generated by one of the randomized incremental methods (4.1)-(4.3), and let the stepsize α_k satisfy

$$\lim_{k \to \infty} \alpha_k = 0, \qquad \sum_{k=0}^{\infty} \alpha_k = \infty.$$

Then, with probability 1,

$$\liminf_{k \to \infty} F(x_k) = F^*.$$

Furthermore, if X^* is nonempty and

$$\sum_{k=0}^{\infty} \alpha_k^2 < \infty,$$

then $\{x_k\}$ converges to some $x^* \in X^*$ with probability 1.

Proof: The proof of the first part is nearly identical to the corresponding part of Prop. 3.4. To prove the second part, similar to the proof of Prop. 4.1, we obtain for all k and all $x^* \in X^*$,

$$E\big\{\|x_{k+1} - x^*\|^2 \mid \mathcal{F}_k\big\} \leq \|x_k - x^*\|^2 - \frac{2\alpha_k}{m}\big(F(x_k) - F^*\big) + \beta\alpha_k^2 c^2 \tag{4.17}$$

[cf. Eq. (4.9) with α and y replaced with α_k and x^*, respectively], where $\mathcal{F}_k = \{x_k, z_{k-1}, \ldots, z_0, x_0\}$. By the Supermartingale Convergence Theorem (Prop. 2.2), for each $x^* \in X^*$, there is a set Ω_{x^*} of sample paths of probability 1 such that for each sample path in Ω_{x^*}

$$\sum_{k=0}^{\infty} \frac{2\alpha_k}{m}\big(F(x_k) - F^*\big) < \infty, \tag{4.18}$$

and the sequence $\{\|x_k - x^*\|\}$ converges.

Let $\{v_i\}$ be a countable subset of the relative interior $\mathrm{ri}(X^*)$ that is dense in X^* [such a set exists since $\mathrm{ri}(X^*)$ is a relatively open subset of the affine hull of X^*; an example of such a set is the intersection of X^* with the set of vectors of the form $x^* + \sum_{i=1}^p r_i\xi_i$, where $\xi_1, \ldots, \xi_p$ are basis vectors for the affine hull of X^* and r_i are rational numbers]. Let also Ω_{v_i} be the set of sample paths defined earlier that corresponds to v_i. The intersection

$$\bar{\Omega} = \cap_{i=1}^{\infty}\Omega_{v_i}$$

has probability 1, since its complement $\bar{\Omega}^c$ is equal to $\cup_{i=1}^{\infty}\Omega_{v_i}^c$ and

$$\mathrm{Prob}\,(\cup_{i=1}^{\infty}\Omega_{v_i}^c) \leq \sum_{i=1}^{\infty}\mathrm{Prob}\,(\Omega_{v_i}^c) = 0.$$

For each sample path in $\bar{\Omega}$, all the sequences $\{\|x_k - v_i\|\}$ converge so that $\{x_k\}$ is bounded, while by the first part of the proposition [or Eq. (4.18)] $\liminf_{k\to\infty} F(x_k) = F^*$. Therefore, $\{x_k\}$ has a limit point $\bar{x}$ in X^*. Since $\{v_i\}$ is dense in X^*, for every $\epsilon > 0$ there exists $v_{i(\epsilon)}$ such that $\|\bar{x} - v_{i(\epsilon)}\| < \epsilon$. Since the sequence $\{\|x_k - v_{i(\epsilon)}\|\}$ converges and $\bar{x}$ is a limit point of $\{x_k\}$, we have $\lim_{k\to\infty}\|x_k - v_{i(\epsilon)}\| < \epsilon$, so that

$$\limsup_{k\to\infty}\|x_k - \bar{x}\| \leq \lim_{k\to\infty}\|x_k - v_{i(\epsilon)}\| + \|v_{i(\epsilon)} - \bar{x}\| < 2\epsilon.$$

By taking $\epsilon \to 0$, it follows that $x_k \to \bar{x}$. **Q.E.D.**

5. SOME APPLICATIONS

In this section we illustrate our methods in the context of two types of practical applications, and discuss relations with known algorithms.

5.1 Regularized Least Squares

Let us consider least squares problems, involving minimization of a sum of quadratic component functions $f_i(x)$ that correspond to errors between data and the output of a model that is parameterized by a vector

x. Often a convex regularization function $R(x)$ is added to the least squares objective, to induce desirable properties of the solution. This gives rise to problems of the form

$$\begin{aligned} &\text{minimize} \quad R(x) + \frac{1}{2}\sum_{i=1}^{m}(c_i'x - d_i)^2 \\ &\text{subject to} \quad x \in \Re^n, \end{aligned} \tag{5.1}$$

where c_i and d_i are given vectors and scalars, respectively, and γ is a positive scalar. When R is differentiable (e.g., quadratic), and either m is very large or the data (c_i, d_i) become available sequentially over time, it makes sense to consider incremental gradient methods, which have a long history of applications over the last 50 years, starting with the Widrow-Hoff least mean squares (LMS) method [WiH60].

The classical type of regularization involves a quadratic function R (as in classical regression and the LMS method), but nondifferentiable regularization functions have become increasingly important recently. On the other hand, to apply our incremental methods, a quadratic R is not essential. What is important is that R has a simple form that facilitates the use of proximal algorithms, such as for example a separable form, so that the proximal iteration on R is simplified through decomposition. As an example, consider the ℓ_1-*regularization problem*, where

$$R(x) = \gamma\|x\|_1 = \gamma\sum_{j=1}^{n}|x^j|, \tag{5.2}$$

γ is a positive scalar and x^j is the jth coordinate of x. Then the proximal iteration

$$z_k = \arg\min_{x\in\Re^n}\left\{\gamma\,\|x\|_1 + \frac{1}{2\alpha_k}\|x - x_k\|^2\right\}$$

decomposes into the n one-dimensional minimizations

$$z_k^j = \arg\min_{x^j\in\Re}\left\{\gamma\,|x^j| + \frac{1}{2\alpha_k}|x^j - x_k^j|^2\right\}, \qquad j = 1,\ldots,n,$$

and can be done in closed form

$$z_k^j = \begin{cases} x_k^j - \gamma\alpha_k & \text{if } \gamma\alpha_k \le x_k^j, \\ 0 & \text{if } -\gamma\alpha_k < x_k^j < \gamma\alpha_k, \\ x_k^j + \gamma\alpha_k & \text{if } x_k^j \le -\gamma\alpha_k, \end{cases} \qquad j = 1,\ldots,n. \tag{5.3}$$

We refer to Figueiredo, Nowak, and Wright [FNW07], Wright, Nowak, and Figueiredo [WNF08], Beck and Teboulle [BeT10], and the references given there, for a discussion of a broad variety of applications in estimation and signal processing problems, where nondifferentiable regularization functions play an important role.

We now note that the incremental algorithms of this paper are well-suited for solution of ℓ_1-regularization problems of the form (5.1)-(5.2). For example, the kth incremental iteration may consist of selecting a data pair (c_{i_k}, d_{i_k}) and performing a proximal iteration of the form (5.3) to obtain z_k, followed by a gradient iteration on the component $\frac{1}{2}(c_{i_k}'x - d_{i_k})^2$, starting at z_k:

$$x_{k+1} = z_k - \alpha_k c_{i_k}(c_{i_k}'z_k - d_{i_k}).$$

This algorithm is the special case of the algorithms (2.12)-(2.14) (here $X = \Re^n$, and all three algorithms coincide), with $f_i(x)$ being $\gamma\|x\|_1$ (we use m copies of this function) and $h_i(x) = \frac{1}{2}(c_i'x - d_i)^2$. It can be

viewed as an incremental version of a popular class of algorithms in signal processing, known as iterative shrinkage/thresholding (see Chambolle et. al. [CDL98], Figueiredo and Nowak [FiN03], Daubechies, Defrise, and Mol [DDM04], Combettes and Wajs [CoW05], Bioucas-Dias and Figueiredo [BiF07], Elad, Matalon, and Zibulevsky [EMZ07], Beck and Teboulle [BeT09], [BeT10]). Our methods bear the same relation to this class of algorithms as the LMS method bears to gradient algorithms for the classical linear least squares problem with quadratic regularization function.

Finally, let us note that as an alternative, the proximal iteration (5.3) could be replaced by a proximal iteration on $\gamma\,|x^j|$ for some selected index j, with all indexes selected cyclically in incremental iterations. Randomized selection of the data pair (c_{i_k}, d_{i_k}) would also be interesting, particularly in contexts where the data has a natural stochastic interpretation.

5.2 Iterated Projection Algorithms

A feasibility problem that arises in many contexts involves finding a point with certain properties within a set intersection $\cap_{i=1}^m X_i$, where each X_i is a closed convex set. For the case where m is large and each of the sets X_i has a simple form, incremental methods that make successive projections on the component sets X_i have a long history (see e.g., Gubin, Polyak, and Raik [GPR67], and recent papers such as Bauschke [Bau01], Bauschke, Combettes, and Kruk [BCL06], and Cegielski and Suchocka [CeS08], and their bibliographies). We may consider the following generalized version of the classical feasibility problem,

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \quad x \in \cap_{i=1}^m X_i, \end{aligned} \tag{5.4}$$

where $f : \Re^n \mapsto \Re$ is a convex cost function, and the method

$$x_{k+1} = P_{X_{i_k}}\big(x_k - \alpha_k \tilde{\nabla} f(x_k)\big), \tag{5.5}$$

where the index i_k is chosen from $\{1, \ldots, m\}$ according to a randomized rule. Incremental algorithms for problem (5.4), which bear some relation with ours have been recently proposed by Nedić [Ned10]. Actually, the algorithm of [Ned10] involves an additional projection on a special set X_0 at each iteration, but for simplicity we will take $X_0 = \Re^n$. The incremental approach is particularly well-suited for problems of the form (5.4) where the sets X_i are not known in advance, but are revealed as the algorithm progresses.

While the problem (5.4) does not involve a sum of component functions, it may be converted into one that does by using an exact penalty function. In particular, consider the problem

$$\begin{aligned} &\text{minimize} \quad f(x) + \gamma \sum_{i=1}^m \text{dist}(x; X_i) \\ &\text{subject to} \quad x \in \Re^n, \end{aligned} \tag{5.6}$$

where γ is a positive penalty parameter. Then for f Lipschitz continuous and γ sufficiently large, problems (5.4) and (5.6) are equivalent. We show this for the case where $m = 1$ and then we generalize.

Proposition 5.1: Let $f : Y \mapsto \Re$ be a function defined on a subset Y of $\Re^n$, and let X be a nonempty closed subset of Y. Assume that f is Lipschitz continuous over Y with constant L, i.e.,

$$\big|f(x) - f(y)\big| \leq L\|x - y\|, \qquad \forall\ x, y \in Y,$$

and let γ be a scalar with $\gamma > L$. Then the set of minima of f over X coincides with the set of minima of

$$f(x) + \gamma\, \mathrm{dist}(x; X)$$

over Y.

Proof: Denote $F(x) = f(x) + \gamma\, \mathrm{dist}(x; X)$. For a vector $x \in Y$, let $\hat{x}$ denote a vector of X that is at minimum distance from X. If $\gamma > L$, we have

$$F(x) = f(x) + \gamma\|x - \hat{x}\| = f(\hat{x}) + \big(f(x) - f(\hat{x})\big) + \gamma\|x - \hat{x}\| \geq f(\hat{x}) + (\gamma - L)\|x - \hat{x}\| \geq F(\hat{x}), \qquad \forall\ x \in Y,$$

with strict inequality if $x \neq \hat{x}$; here the first inequality follows using the Lipschitz property of f to write

$$f(x) - f(\hat{x}) \geq -L\|x - \hat{x}\|,$$

while the second inequality follows from the fact $f(\hat{x}) = F(\hat{x})$. In words, the value of $F(x)$ is strictly reduced when we project an $x \in Y$ with $x \notin X$ onto X. Hence the minima of F over Y can only lie within X, while $F = f$ within X. Thus all minima of F over Y must lie in X and also minimize f over X (since $F = f$ on X). Conversely, all minima of f over X are also minima of F over X (since $F = f$ on X), and by the preceding inequality, they are also minima of F over Y. **Q.E.D.**

We now provide a generalization for $m > 1$.†

Proposition 5.2: Let $f : Y \mapsto \Re$ be a function defined on a subset Y of $\Re^n$, and let X_i, $i = 1, \ldots, m$, be closed subsets of Y with nonempty intersection. Assume that f is Lipschitz continuous over Y with constant L, and that for some scalar $\beta > 0$, we have

$$\mathrm{dist}(x; X_1 \cap \cdots \cap X_m) \leq \beta \sum_{i=1}^{m} \mathrm{dist}(x; X_i), \qquad \forall\ x \in Y. \tag{5.7}$$

Let γ be a scalar with $\gamma > \beta L$. Then the set of minima of f over $\cap_{i=1}^m X_i$ coincides with the set of minima of

$$f(x) + \gamma \sum_{i=1}^{m} \mathrm{dist}(x; X_i)$$

over Y.

† In the original version of this report the assumption on existence of the scalar β in the proposition below was neglected, due to a faulty application of Prop. 5.1 in its proof. This was noted in a paper by Kundu, Bach, and Bhattacharrya in Oct. 2017. If the sets X_i are polyhedral this assumption is not necessary; this is Hoffman's lemma.

Proof: The proof is similar to the proof of Prop. 5.1, using Eq. (5.7) to modify the main inequality. Denote $F(x) = f(x) + \gamma \sum_{i=1}^m \operatorname{dist}(x; X_i)$ and $X = X_1 \cap \cdots \cap X_m$. For a vector $x \in Y$, let $\hat{x}_i$ denote a vector of X_i that is at minimum distance from x, and let $\hat{x}$ denote a vector of X that is at minimum distance from x. If $\gamma > \beta L$, we have

$$F(x) = f(x)+\gamma \sum_{i=1}^m \|x-\hat{x}_i\| \geq f(\hat{x})+\big(f(x)-f(\hat{x})\big)+\frac{\gamma}{\beta}\|x-\hat{x}\| \geq f(\hat{x})+\left(\frac{\gamma}{\beta}-L\right)\|x-\hat{x}\| \geq F(\hat{x}), \qquad \forall\, x \in Y,$$

with strict inequality if $x \neq \hat{x}$. The proof now proceeds as in the proof of Prop. 5.1. **Q.E.D.**

Regarding algorithmic solution, from Prop. 5.2, it follows that we may consider in place of the original problem (5.4) the additive cost problem (5.6) for which our algorithms apply. In particular, let us consider the algorithms (2.12)-(2.14), with $X = \Re^n$, which involve a proximal iteration on one of the functions $\gamma \operatorname{dist}(x; X_i)$ followed by a subgradient iteration on f. A key fact here is that the proximal iteration

$$z_k = \arg\min_{x \in \Re^n} \left\{ \gamma \operatorname{dist}(x; X_{i_k}) + \frac{1}{2\alpha_k}\|x - x_k\|^2 \right\} \tag{5.8}$$

involves a projection on X_{i_k} of x_k, followed by an interpolation. This is shown in the following proposition.

Proposition 5.3: Let z_k be the vector produced by the proximal iteration (5.8). If $x_k \in X_{i_k}$ then $z_k = x_k$, while if $x_k \notin X_{i_k}$,

$$z_k = \begin{cases} (1-\beta_k)x_k + \beta_k P_{X_{i_k}}(x_k) & \text{if } \beta_k < 1, \\ P_{X_{i_k}}(x_k) & \text{if } \beta_k \geq 1, \end{cases} \tag{5.9}$$

where

$$\beta_k = \frac{\alpha_k \gamma}{\operatorname{dist}(x_k; X_{i_k})}.$$

Proof: The case $x_k \in X_{i_k}$ is evident, so assume that $x_k \notin X_{i_k}$. From the nature of the cost function in Eq. (5.8) we see that z_k is a vector that lies in the line segment between x_k and $P_{X_{i_k}}(x_k)$. Hence there are two possibilities: either

$$z_k = P_{X_{i_k}}(x_k), \tag{5.10}$$

or $z_k \notin X_{i_k}$ in which case by setting to 0 the gradient at z_k of the cost function in Eq. (5.8) yields

$$\gamma \frac{z_k - P_{X_{i_k}}(z_k)}{\left\|z_k - P_{X_{i_k}}(z_k)\right\|} = \frac{1}{\alpha_k}(x_k - z_k).$$

This equation implies that x_k, z_k, and $P_{X_{i_k}}(z_k)$ lie on the same line, so that $P_{X_{i_k}}(z_k) = P_{X_{i_k}}(x_k)$ and

$$z_k = x_k - \frac{\alpha_k \gamma}{\operatorname{dist}(x_k; X_{i_k})}\big(x_k - P_{X_{i_k}}(x_k)\big) = (1-\beta_k)x_k + \beta_k P_{X_{i_k}}(x_k). \tag{5.11}$$

By calculating and comparing the value of the cost function in Eq. (5.8) for each of the possibilities (5.10) and (5.11), we can verify that (5.11) gives a lower cost if and only if $\beta_k < 1$. **Q.E.D.**

Let us now consider the problem

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^{m} \big(f_i(x) + h_i(x)\big) \\ &\text{subject to} \quad x \in \cap_{i=1}^{m} X_i. \end{aligned}$$

Based on the preceding analysis, we can convert this problem to the unconstrained minimization problem

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^{m} \big(f_i(x) + h_i(x) + \gamma \text{dist}(x; X_i)\big) \\ &\text{subject to} \quad x \in \Re^n, \end{aligned}$$

where γ is sufficiently large. The algorithm (2.14), applied to this problem, yields the iteration

$$y_k = x_k - \alpha_k \tilde{\nabla} h_{i_k}(x_k), \qquad z_k = y_k - \alpha_k \tilde{\nabla} f_{i_k}(z_k), \qquad x_{k+1} = \begin{cases} (1-\beta_k) z_k + \beta_k P_{X_{i_k}}(z_k) & \text{if } \beta_k < 1, \\ P_{X_{i_k}}(z_k) & \text{if } \beta_k \geq 1, \end{cases}$$

where

$$\beta_k = \frac{\alpha_k \gamma}{\text{dist}(z_k; X_{i_k})},$$

[cf. Eq. (5.9)]. The index i_k may be chosen either randomly or according to a cyclic rule.

Let us finally note another problem where our incremental methods apply:

$$\begin{aligned} &\text{minimize} \quad f(x) + c \sum_{j=1}^{r} \max\big\{0, g_j(x)\big\} \\ &\text{subject to} \quad x \in \cap_{i=1}^{m} X_i. \end{aligned}$$

This type of problem is obtained by replacing convex inequality constraints of the form $g_j(x) \leq 0$ with the nondifferentiable penalty terms $c \max\big\{0, g_j(x)\big\}$, where $c > 0$ is a penalty parameter. Then a possible incremental method at each iteration, would either do a subgradient or proximal iteration on f, or select one of the violated constraints (if any) and perform a subgradient iteration on the corresponding function g_j, or select one of the sets X_i and do an interpolated projection on it. Related methods may also be obtained when f is replaced by a cost function of the form

$$\sum_{i=1}^{m} \big(f_i(x) + h_i(x)\big),$$

and the components f_i are dealt with a proximal iteration while the components h_i are dealt with a subgradient iteration.

6. CONCLUSIONS

We have surveyed incremental algorithms, which can deal with many of the challenges posed by large data sets in machine learning applications, as well as with the additive structure of many interesting problems,

including those arising in the context of duality. We have used a unified analytical framework that includes incremental proximal algorithms and their combinations with the more established incremental gradient and subgradient methods. This allows the flexibility to separate the cost function into the parts that are conveniently handled by proximal iterations (e.g., in essentially closed form), and the remaining parts to be handled by subgradient iterations. We have outlined the convergence properties of these methods, and we have shown that our algorithms apply to some important problems that have been the focus of recent research.

Much work remains to be done to apply and evaluate our methods within the broad context of potential applications. Let us mention some possibilities that may extend the range of applications of our approach, and are interesting subjects for further investigation: alternative proximal and projected subgradient iterations, involving nonquadratic proximal terms and/or subgradient projections, alternative stepsize rules, distributed asynchronous implementations along the lines of [NBB01], polyhedral approximation (bundle) variants of the proximal iterations in the spirit of [BeY09], and variants for methods with errors in the calculation of the subgradients along the lines of [NeB10].

7. REFERENCES

[BCL03] Bauschke, H. H., Combettes, P. L., and Luke, D. R., 2003. "Hybrid Projection-Reflection Method for Phase Retrieval," Journal of the Optical Society of America, Vol. 20, pp. 1025-1034.

[BCK06] Bauschke, H. H., Combettes, P. L., and Kruk, S. G., 2006. "Extrapolation Algorithm for Affine-Convex Feasibility Problems," Numer. Algorithms, Vol. 41, pp. 239-274.

[BHG08] Blatt, D., Hero, A. O., Gauchman, H., 2008. "A Convergent Incremental Gradient Method with a Constant Step Size," SIAM J. Optimization, Vol. 18, pp. 29-51.

[BMN01] Ben-Tal, A., Margalit, T., and Nemirovski, A., 2001. "The Ordered Subsets Mirror Descent Optimization Method and its Use for the Positron Emission Tomography Reconstruction," in Inherently Parallel Algorithms in Feasibility and Optimization and their Applications (D. Butnariu, Y. Censor, and S. Reich, eds.), Elsevier, Amsterdam, Netherlands.

[BMS99] Boltyanski, V., Martini, H., and Soltan, V., 1999. Geometric Methods and Optimization Problems, Kluwer, Boston.

[BNO03] Bertsekas, D. P., Nedić, A., and Ozdaglar, A. E., 2003. Convex Analysis and Optimization, Athena Scientific, Belmont, MA.

[BPC10] Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J., 2010. "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," working paper on line, Stanford, Univ.

[Bau01] Bauschke, H. H., 2001. "Projection Algorithms: Results and Open Problems," in Inherently Parallel Algorithms in Feasibility and Optimization and their Applications (D. Butnariu, Y. Censor, and S. Reich, eds.), Elsevier, Amsterdam, Netherlands.

[BeT89] Bertsekas, D. P., and Tsitsiklis, J. N., 1989. Parallel and Distributed Computation: Numerical Methods, Prentice-Hall, Englewood Cliffs, N. J.

[BeT96] Bertsekas, D. P., and Tsitsiklis, J. N., 1996. Neuro-Dynamic Programming, Athena Scientific, Belmont, MA.

[BeT00] Bertsekas, D. P., and Tsitsiklis, J. N., 2000. "Gradient Convergence in Gradient Methods," SIAM J. Opti-

mization, Vol. 10, pp. 627-642.

[BeT09] Beck, A., and Teboulle, M., 2009. "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," SIAM J. on Imaging Sciences, Vol. 2, pp. 183-202.

[BeT10] Beck, A., and Teboulle, M., 2010. "Gradient-Based Algorithms with Applications to Signal-Recovery Problems," in Convex Optimization in Signal Processing and Communications (Y. Eldar and D. Palomar, eds.), Cambridge University Press, pp. 42-88.

[BeY09] Bertsekas, D. P., and Yu, H., 2009. "A Unifying Polyhedral Approximation Framework for Convex Optimization," Lab. for Information and Decision Systems Report LIDS-P-2820, MIT; to appear in SIAM J. on Optimization.

[Ber83] Bertsekas, D. P., 1983. "Distributed Asynchronous Computation of Fixed Points," Mathematical Programming, Vol. 27, pp. 107-120.

[Ber96] Bertsekas, D. P., 1996. "Incremental Least Squares Methods and the Extended Kalman Filter," SIAM J. on Optimization, Vol. 6, pp. 807-822.

[Ber97] Bertsekas, D. P., 1997. "A Hybrid Incremental Gradient Method for Least Squares," SIAM J. on Optimization, Vol. 7, pp. 913-926.

[Ber99] Bertsekas, D. P., 1999. Nonlinear Programming, 2nd edition, Athena Scientific, Belmont, MA.

[Ber09] Bertsekas, D. P., 2009. Convex Optimization Theory, Athena Scientific, Belmont, MA; also, this book's on-line supplementary chapter on algorithms.

[Ber10] Bertsekas, D. P., 2010. "Incremental Proximal Methods for Large Scale Convex Optimization," Lab. for Info. and Decision Systems Report LIDS-P-2847, MIT, Cambridge, MA; to appear in Math. Programming J.

[BiF07] Bioucas-Dias, J., and Figueiredo, M. A. T., 2007. "A New TwIST: Two-Step Iterative Shrinkage/Thresholding Algorithms for Image Restoration," IEEE Trans. Image Processing, Vol. 16, pp. 2992-3004.

[Bor08] Borkar, V. S., 2008. Stochastic Approximation: A Dynamical Systems Viewpoint, Cambridge Univ. Press.

[Bot05] Bottou, L., 2005. "SGD: Stochastic Gradient Descent," http://leon.bottou.org/projects/sgd.

[CDL98] Chambolle, A., DeVore, R. A., Lee, N. Y., and Lucier, B. J., 1998. "Nonlinear Wavelet Image Processing: Variational Problems, Compression, and Noise Removal Through Wavelet Shrinkage," IEEE Trans. Image Processing, Vol. 7, pp. 319-335.

[CeS08] Cegielski, A., and Suchocka, A., 2008. "Relaxed Alternating Projection Methods," SIAM J. Optimization, Vol. 19, pp. 1093-1106.

[CoW05] Combettes, P. L., and Wajs, V. R., 2005. "Signal Recovery by Proximal Forward-Backward Splitting," Multiscale Modeling and Simulation, Vol. 4, pp. 1168-1200.

[DDM04] Daubechies, I., Defrise, M., and Mol, C. D., 2004. "An Iterative Thresholding Algorithm for Linear Inverse Problems with a Sparsity Constraint," Comm. Pure Appl. Math., Vol. 57, pp. 1413-1457.

[DrH04] Drezner, Z., and Hamacher, H. W., 2004. Facility Location: Applications and Theory, Springer, N. Y.

[DHS10] Duchi, J., Hazan, E., and Singer, Y., 2010. "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," UC Berkeley EECS Technical Report 2010-24, to appear in J. of Machine Learning Research.

[Dav76] Davidon, W. C., 1976. "New Least Squares Algorithms," J. Optimization Theory and Applications, Vol. 18,

pp. 187-197.

[EMZ07] Elad, M., Matalon, B., and Zibulevsky, M., 2007. "Coordinate and Subspace Optimization Methods for Linear Least Squares with Non-Quadratic Regularization," J. on Applied and Computational Harmonic Analysis, Vol. 23, pp. 346-367.

[EcB92] Eckstein, J., and Bertsekas, D. P., 1992. "On the Douglas-Rachford Splitting Method and the Proximal Point Algorithm for Maximal Monotone Operators," Math. Programming, Vol. 55, pp. 293-318.

[Erm69] Ermoliev, Yu. M., "On the Stochastic Quasi-Gradient Method and Stochastic Quasi-Feyer Sequences," Kibernetika, No. 2, 1969, pp. 73–83.

[Erm76] Ermoliev, Yu. M., Stochastic Programming Methods, Nauka, Moscow, 1976.

[FiN03] Figueiredo, M. A. T., and Nowak, R. D., 2003. "An EM Algorithm for Wavelet-Based Image Restoration," IEEE Trans. Image Processing, Vol. 12, pp. 906-916.

[FNW07] Figueiredo, M. A. T., Nowak, R. D., and Wright, S. J., 2007. "Gradient Projection for Sparse Reconstruction: Application to Compressed Sensing and Other Inverse Problems," IEEE J. Sel. Topics in Signal Processing, Vol. 1, pp. 586-597.

[GGM06] Gaudioso, M., Giallombardo, G., and Miglionico, G., 2006. "An Incremental Method for Solving Convex Finite Min-Max Problems," Math. of Operations Research, Vol. 31, pp. 173-187.

[GMS10] Goldfarb, D., Ma, S., and Scheinberg, K., 2010. "Fast Alternating Linearization Methods for Minimizing the Sum of Two Convex Functions", Columbia Univ. report, on line.

[GPR67] Gubin, L. G., Polyak, B. T., and Raik, E. V., 1967. "The Method of Projection for Finding the Common Point in Convex Sets," U.S.S.R. Comput. Math. Phys., Vol. 7, pp. 124 (English Translation).

[GaM76] Gabay, D., and Mercier, B., 1979. "A Dual Algorithm for the Solution of Nonlinear Variational Problems via Finite-Element Approximations," Comp. Math. Appl., Vol. 2, pp. 17-40.

[Gab83] Gabay, D., 1983. "Applications of the Method of Multipliers to Variational Inequalities," in M. Fortin and R. Glowinski, eds., Augmented Lagrangian Methods: Applications to the Solution of Boundary-Value Problems, North-Holland, Amsterdam.

[GoM09] Goldfarb, D., and Ma, S., 2009. "Fast Multiple Splitting Algorithms for Convex Optimization," Columbia Univ. report, on line.

[Gri94] Grippo, L., 1994. "A Class of Unconstrained Minimization Methods for Neural Network Training," Optim. Methods and Software, Vol. 4, pp. 135-150.

[Gri00] Grippo, L., 2000. "Convergent On-Line Algorithms for Supervised Learning in Neural Networks," IEEE Trans. Neural Networks, Vol. 11, pp. 1284-1299.

[HeD09] Helou, E. S., and De Pierro, A. R., 2009. "Incremental Subgradients for Constrained Convex Optimization: A Unified Framework and New Methods," SIAM J. on Optimization, Vol. 20, pp. 1547-1572.

[JRJ09] Johansson, B., Rabi, M., and Johansson, M., 2009. "A Randomized Incremental Subgradient Method for Distributed Optimization in Networked Systems," SIAM J. on Optimization, Vol. 20, pp. 1157-1170.

[Kib80] Kibardin, V. M., 1980. "Decomposition into Functions in the Minimization Problem," Automation and Remote Control, Vol. 40, pp. 1311-1323.

[Kiw04] Kiwiel, K. C., 2004. "Convergence of Approximate and Incremental Subgradient Methods for Convex Optimization," SIAM J. on Optimization, Vol. 14, pp. 807-840.

[KuC78] Kushner, H. J., and Clark, D. S., 1978. Stochastic Approximation Methods for Constrained and Unconstrained Systems, Springer-Verlag, N. Y.

[KuY97] Kushner, H. J., and Yin, G., 1997. Stochastic Approximation Methods, Springer-Verlag, N. Y.

[LMY08] Lu, Z., Monteiro, R. D. C., and Yuan, M., 2008. "Convex Optimization Methods for Dimension Reduction and Coefficient Estimation in Multivariate Linear Regression," Report, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta; appeared on line in Math. Programming J., 2010.

[LeW10] Lee, S., and Wright, S. J., 2010. "Sparse Nonlinear Support Vector Machines via Stochastic Approximation," Univ. of Wisconsin Report, submitted.

[LiM79] Lions, P. L., and Mercier, B., 1979. "Splitting Algorithms for the Sum of Two Nonlinear Operators," SIAM J. on Numerical Analysis, Vol. 16, pp. 964-979.

[Lit66] Litvakov, B. M., 1966. "On an Iteration Method in the Problem of Approximating a Function from a Finite Number of Observations," Avtom. Telemech., No. 4, pp. 104-113.

[Lju77] Ljung, L., 1977. "Analysis of Recursive Stochastic Algorithms," IEEE Trans. on Automatic Control, Vol. 22, pp. 551-575.

[LuT94] Luo, Z. Q., and Tseng, P., 1994. "Analysis of an Approximate Gradient Projection Method with Applications to the Backpropagation Algorithm," Optimization Methods and Software, Vol. 4, pp. 85-101.

[Luo91] Luo, Z. Q., 1991. "On the Convergence of the LMS Algorithm with Adaptive Learning Rate for Linear Feedforward Networks," Neural Computation, Vol. 3, pp. 226-245.

[MYF03] Moriyama, H., Yamashita N., and Fukushima, M., 2003. "The Incremental Gauss-Newton Algorithm with Adaptive Stepsize Rule," Computational Optimization and Applications, Vol. 26, pp. 107-141.

[MaS94] Mangasarian, O. L., and Solodov, M. V., 1994. "Serial and Parallel Backpropagation Convergence Via Nonmonotone Perturbed Minimization," Opt. Methods and Software, Vol. 4, pp. 103-116.

[Mar70] Martinet, B., 1970. "Regularisation d' Inéquations Variationelles par Approximations Successives," Revue Fran. d'Automatique et Infomatique Rech. Opérationelle, Vol. 4, pp. 154-159.

[Mey07] Meyn, S., 2007. Control Techniques for Complex Networks, Cambridge University Press, N. Y.

[NBB01] Nedić, A., Bertsekas, D. P., and Borkar, V., 2001. "Distributed Asynchronous Incremental Subgradient Methods," in Inherently Parallel Algorithms in Feasibility and Optimization and their Applications (D. Butnariu, Y. Censor, and S. Reich, eds.), Elsevier, Amsterdam, Netherlands.

[NJL09] Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A., 2009. "Robust Stochastic Approximation Approach to Stochastic Programming," SIAM Journal on Optimization, Vol. 19, pp. 1574-1609.

[NeB00] Nedić, A., and Bertsekas, D. P., 2000. "Convergence Rate of the Incremental Subgradient Algorithm," in Stochastic Optimization: Algorithms and Applications, Eds., S. Uryasev and P. M. Pardalos, Kluwer Academic Publishers, pp. 263-304.

[NeB01] Nedić, A., and Bertsekas, D. P., 2001. "Incremental Subgradient Methods for Nondifferentiable Optimization," SIAM J. on Optimization, Vol. 12, 2001, pp. 109-138.

[NeB10] Nedić, A., and Bertsekas, D. P., 2010. "The Effect of Deterministic Noise in Subgradient Methods," Math. Programming, Ser. A, Vol. 125, pp. 75-99.

[NeO09] Nedić, A., and Ozdaglar, A., 2009. "Distributed Subgradient Methods for Multi-Agent Optimization," IEEE Trans. on Aut. Control, Vol. 54, pp. 48-61.

[Ned10] Nedić, A., 2010. "Random Projection Algorithms for Convex Minimization Problems," Univ. of Illinois Report; appear in Math. Programming Journal.

[Nes83] Nesterov, Y., 1983. "A Method for Unconstrained Convex Minimization Problem with the Rate of Convergence $O(1/k^2)$," Doklady AN SSSR 269, pp. 543-547; translated as Soviet Math. Dokl.

[Nes04] Nesterov, Y., 2004. Introductory Lectures on Convex Optimization, Kluwer Academic Publisher, Dordrecht, The Netherlands.

[Nes05] Nesterov, Y., 2005. "Smooth Minimization of Nonsmooth Functions," Math. Programming, Vol. 103 pp. 127-152.

[Nev75] Neveu, J., 1975. Discrete Parameter Martingales, North-Holland, Amsterdam, The Netherlands.

[PKP09] Predd, J. B., Kulkarni, S. R., and Poor, H. V., 2009. "A Collaborative Training Algorithm for Distributed Learning," IEEE Transactions on Information Theory, Vol. 55, pp. 1856-1871.

[Pas79] Passty, G. B., 1979. "Ergodic Convergence to a Zero of the Sum of Monotone Operators in Hilbert Space," J. Math. Anal. Appl., Vol. 72, pp. 383-390.

[Pfl96] Pflug, G., 1996. Optimization of Stochastic Models. The Interface Between Simulation and Optimization, Kluwer, Boston.

[PoT73] Polyak, B. T., and Tsypkin, Y. Z., 1973. "Pseudogradient Adaptation and Training Algorithms," Automation and Remote Control, Vol. 12, pp. 83-94.

[Pol64] Poljak, B. T., 1964. "Some Methods of Speeding up the Convergence of Iteration Methods," Z. VyČisl. Mat. i Mat. Fiz., Vol. 4, pp. 1-17.

[Pol87] Polyak, B. T., 1987. Introduction to Optimization, Optimization Software Inc., N. Y.

[Pol78] Polyak, B. T., 1978. "Nonlinear Programming Methods in the Presence of Noise," Math. Programming, Vol. 14, pp. 87–97.

[Pol87] Polyak, B. T., 1987. Introduction to Optimization, Optimization Software Inc., N. Y.

[RNV09] Ram, S. S., Nedić, A., and Veeravalli, V. V., 2009. "Incremental Stochastic Subgradient Algorithms for Convex Optimization," SIAM Journal on Optimization, Vol. 20, pp. 691-717.

[RNV10] Ram, S. S., Nedić, A., and Veeravalli, V. V., 2010. "Distributed Stochastic Subgradient Projection Algorithms for Convex Optimization," Journal of Optimization Theory and Applications, Vol. 147, pp. 516-545.

[RaN04] Rabbat, M. G., and Nowak, R. D., 2004. "Distributed Optimization in Sensor Networks," in Proc. Inf. Processing Sensor Networks, Berkeley, CA, pp. 20-27.

[RaN05] Rabbat M. G., and Nowak R. D., 2005. "Quantized Incremental Algorithms for Distributed Optimization," IEEE Journal on Select Areas in Communications, Vol. 23, pp. 798-808.

[Roc70] Rockafellar, R. T., 1970. Convex Analysis, Princeton University Press, Princeton, NJ.

[Roc76] Rockafellar, R. T., 1976. "Monotone Operators and the Proximal Point Algorithm," SIAM Journal on Control and Optimization, Vol. 14, pp. 877-898.

[SSS07] Shalev-Shwartz, S., Singer, Y., Srebro, N., and Cotter, A., 2007. "Pegasos: Primal Estimated Subgradient Solver for SVM," in ICML 07, New York, N. Y., pp. 807-814.

[SoZ98] Solodov, M. V., and Zavriev, S. K., 1998. "Error Stability Properties of Generalized Gradient-Type Algorithms," J. Opt. Theory and Appl., Vol. 98, pp. 663-680.

[Sol98] Solodov, M. V., 1998. "Incremental Gradient Algorithms with Stepsizes Bounded Away from Zero," Comput. Opt. Appl., Vol. 11, pp. 28-35.

[Spi85] Spingarn, J. E., 1985. "Applications of the Method of Partial Inverses to Convex Programming: Decomposition," Math. Programming, Vol. 32, pp. 199-223.

[TBA86] Tsitsiklis, J. N., Bertsekas, D. P., and Athans, M., 1986. "Distributed Asynchronous Deterministic and Stochastic Gradient Optimization Algorithms," IEEE Trans. Automatic Control, Vol. AC-31, pp. 803-812.

[Tse98] Tseng, P., 1998. "An Incremental Gradient(-Projection) Method with Momentum Term and Adaptive Stepsize Rule," SIAM J. on Optimization, Vol. 8, pp. 506-531.

[Tse08] Tseng, P., 2008. "On Accelerated Proximal Gradient Methods for Convex-Concave Optimization," Report, Math. Dept., Univ. of Washington.

[VoU07] Vonesch, C., and Unser, M., 2007. "Fast Iterative Thresholding Algorithm for Wavelet-Regularized Deconvolution," in Proc. SPIE Optics and Photonics 2007 Conference on Mathematical Methods: Wavelet XII, Vol. 6701, San Diego, CA, pp. 1-5.

[WNF08] Wright, S. J., Nowak, R. D., and Figueiredo, M. A. T., 2008. "Sparse Reconstruction by Separable Approximation," in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008), pp. 3373-3376.

[WiH60] Widrow, B., and Hoff, M. E., 1960. "Adaptive Switching Circuits," Institute of Radio Engineers, Western Electronic Show and Convention, Convention Record, Part 4, pp. 96-104.