June 2015 (revised September 2015) **Report LIDS - 3176**

Incremental Aggregated Proximal and Augmented Lagrangian Algorithms

Dimitri P. Bertsekas†

Abstract

We consider minimization of the sum of a large number of convex functions, and we propose an incremental aggregated version of the proximal algorithm, which bears similarity to the incremental aggregated gradient and subgradient methods that have received a lot of recent attention. Under cost function differentiability and strong convexity assumptions, we show linear convergence for a sufficiently small constant stepsize. This result also applies to distributed asynchronous variants of the method, involving bounded interprocessor communication delays.

We then consider dual versions of incremental proximal algorithms, which are incremental augmented Lagrangian methods for separable equality-constrained optimization problems. Contrary to the standard augmented Lagrangian method, these methods admit decomposition in the minimization of the augmented Lagrangian, and update the multipliers far more frequently. Our incremental aggregated augmented Lagrangian methods bear similarity to several known decomposition algorithms, most of which, however, are not incremental in nature: the augmented Lagrangian decomposition algorithm of Stephanopoulos and Westerberg [StW75], and the related methods of Tadjewski [Tad89] and Ruszczynski [Rus95], and the alternating direction method of multipliers (ADMM) and more recent variations. We compare these methods in terms of their properties, and highlight their potential advantages and limitations.

We also address the solution of separable inequality-constrained optimization problems through the use of nonquadratic augmented Lagrangians such as the exponential, and we dually consider a corresponding incremental aggregated version of the proximal algorithm that uses nonquadratic regularization, such as an entropy function. We finally propose a closely related linearly convergent method for minimization of large differentiable sums subject to an orthant constraint, which may be viewed as an incremental aggregated version of the mirror descent method.

1. INCREMENTAL GRADIENT, SUBGRADIENT, AND PROXIMAL METHODS

We consider optimization problems with a cost function that consists of additive components:

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

† Dimitri Bertsekas is with the Dept. of Electr. Engineering and Comp. Science, and the Laboratory for Information and Decision Systems, M.I.T., Cambridge, Mass., 02139.

where $f_i : \Re^n \mapsto \Re$, $i = 1, \ldots, m$, are convex 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 methods that operate on a single component f_i at each iteration, rather than on the entire cost function F. Problems of this type arise often in various practical contexts and have received a lot of attention recently.

Suitable algorithms include the *incremental subgradient method* (abbreviated IS), where a cost component f_{i_k} is selected at iteration k, and an arbitrary subgradient $\tilde\nabla f_{i_k}(x_k)$ of f_{i_k} is used in place of a full subgradient of F at x_k:†

$$x_{k+1} = P_X\big(x_k - \alpha_k \tilde\nabla f_{i_k}(x_k)\big), \tag{1.2}$$

where α_k is a positive stepsize, and $P_X(\cdot)$ denotes projection on X. It is important here that all components are taken up for iteration with equal long-term frequency, using either a cyclic or a random selection scheme. Methods of this type and their properties have been studied for a long time, and the relevant literature, beginning in the 60's, is too voluminous to list here. The author's survey [Ber10] discusses the history of this algorithm, its convergence properties, and its connections with stochastic approximation methods. Generally, a diminishing stepsize α_k is needed for convergence, even when the components f_i are differentiable. Moreover the convergence rate properties are generally better when the index i_k is selected by randomization over the set $\{1, \ldots, m\}$ than by a deterministic cyclic rule, as first shown by Nedić and Bertsekas [NeB01]; see also [BNO03].

Another method, introduced by the author in [Ber10] and further studied in [Ber11], [Ber12], is the *incremental proximal method* (abbreviated IP),

$$x_{k+1} \in \arg\min_{x \in X} \left\{ f_{i_k}(x) + \frac{1}{2\alpha_k} \|x - x_k\|^2 \right\}. \tag{1.3}$$

This method relates to the proximal algorithm (Martinet [Mar70], Rockafellar [Roc76a]) in the same way that the IS method (1.2) relates to the classical nonincremental subgradient method. Similar to the IS method, it is important that all components are taken up for iteration with equal long-term frequency. The theoretical convergence properties of the IS and IP algorithms are similar, but it is generally believed that IP is more robust, a property inherited from its nonincremental counterpart.

It turns out that the structures of the IS and IP methods (1.2) and (1.3) are quite similar. An important fact in this regard is that the IP method (1.3) can be equivalently written as

$$x_{k+1} = P_X\big(x_k - \alpha_k \tilde\nabla f_{i_k}(x_{k+1})\big), \tag{1.4}$$

† 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 $x'x = \|x\|^2$. The scalar coordinates of an optimization vector such as x are denoted by superscripts, $x = (x^1, \ldots, x^n)$, while sequences of iterates are indexed by subscripts. We use $\tilde\nabla f(x)$ to denote a subgradient of a convex function f at a vector $x \in \Re^n$, i.e, a vector such that $f(z) \geq f(x) + \tilde\nabla f(x)'(z - x)$ for all $z \in \Re^n$. The choice of $\tilde\nabla f(x)$ from within the set of all subgradients at x will be clear from the context. If f is differentiable at x, $\tilde\nabla f(x)$ is the gradient $\nabla f(x)$.

where $\tilde{\nabla} f_{i_k}(x_{k+1})$ is a *special* subgradient of f_{i_k} at the new point x_{k+1} (see Bertsekas [Ber10], Prop. 2.1, [Ber11], Prop. 1, or [Ber15], Prop. 6.4.1). This special subgradient is determined from the optimality conditions for the proximal maximization (1.3). For example if $X = \Re^n$, we have

$$\tilde{\nabla} f_{i_k}(x_{k+1}) = \frac{x_k - x_{k+1}}{\alpha_k},$$

which is consistent with Eq. (1.4). Thus determining the special subgradient $\tilde{\nabla} f_{i_k}(x_{k+1})$ may be a difficult problem, and in most cases it is preferable to implement the iteration in the proximal form (1.3) rather than the projected form (1.4). However, the equivalent form of the IP iteration (1.4), when compared with the IS iteration (1.2), suggests the close connection between the IS and IP iterations. In fact this connection is the basis for a combination of the two methods to provide flexibility for the case where some of the cost components f_i are well suited for the proximal minimization of Eq. (1.3), while others are not; see [Ber10], [Ber11], [Ber12].

Incremental Aggregated Gradient and Subgradient Methods

Incremental aggregated methods aim to provide a better approximation of a subgradient of the entire cost function F, while preserving the economies accrued from computing a single component subgradient at each iteration. In particular, the aggregated subgradient method (abbreviated IAS), has the form

$$x_{k+1} = P_X\left(x_k - \alpha_k \sum_{i=1}^{m} \tilde{\nabla} f_i(x_{\ell_i})\right), \tag{1.5}$$

where $\tilde{\nabla} f_i(x_{\ell_i})$ is a "delayed" subgradient of f_i at some earlier iterate x_{ℓ_i}. We assume that the indexes ℓ_i satisfy

$$k - b \le \ell_i \le k, \qquad \forall\ i, k, \tag{1.6}$$

where b is a fixed nonnegative integer. Thus the algorithm uses outdated subgradients from previous iterations for the components f_i, $i \neq i_k$, and need not compute a subgradient of these components at iteration k.

The IAS method was first proposed, to our knowledge, by Nedić, Bertsekas, and Borkar [NBB01]. It was motivated primarily by distributed asynchronous solution of dual separable problems, similar to the ones to be discussed in Section 2 (in a distributed asynchronous context, it is natural to assume that subgradients are used with some delays). A convergence result was shown in [NBB01] assuming that the stepsize sequence $\{a_k\}$ is diminishing, and satisfies the standard conditions

$$\sum_{k=0}^{\infty} \alpha_k = \infty, \qquad \sum_{k=0}^{\infty} \alpha_k^2 < \infty. \tag{1.7}$$

This result covers the case of iteration (1.5) for the case $X = \Re^n$; the more general case where $X \neq \Re^n$ admits a similar analysis. We note that distributed algorithms that involve bounded delays in the iterates have a long history, and are common in various distributed asynchronous computation contexts, including gradient-like and coordinate descent methods; see [BeT89], Sections 7.5-7.8.

Note a limitation of this iteration over the IS iteration: one has to store the past subgradients $\tilde{\nabla} f_i(x_{\ell_i})$, $i \neq i_k$. Moreover, whatever effect the use of previously computed subgradients has, it will not be fully manifested until a subgradient of each component has been computed; this is significant when the number of components m is large. We note also that there are other approaches for approximating a full subgradient of the cost function, which aim at computational economies, such as ϵ-subgradient methods (see Nedić and Bertsekas [NeB10] and the references quoted there), and surrogate subgradient methods (see Bragin et. al [BLY15] and the references quoted there).

The IAS method (1.5) contains as a special case the incremental aggregated gradient method (abbreviated IAG) for the case where the components f_i are differentiable:

$$x_{k+1} = x_k - \alpha_k \sum_{i=1}^{m} \nabla f_i(x_{\ell_i}), \tag{1.8}$$

where $\ell_i \in [k-b, k]$ for all i and k. This method has attracted considerable attention thanks to a particularly interesting convergence result. For the favorable case where the component gradients ∇f_i are Lipschitz continuous and F is strongly convex, it has been shown that the IAG method is linearly convergent to the solution with a sufficiently small but constant stepsize $\alpha_k \equiv \alpha$. This result was first given by Blatt, Hero, and Gauchman [BHG08], for the case where the cost components f_i are quadratic and the delayed indexes ℓ_i satisfy certain restrictions that are consistent with a cyclic selection of components for iteration (see also [AFB06]). The linear convergence result has been subsequently extended for nonquadratic problems and for various forms of the method by several other authors, including Schmidt, Le Roux, and Bach [SLB13], Mairal [Mai13], [Mai14], and Defazio, Caetano, and Domke [DCD14]. Several schemes have been proposed to address the limitation of having to store the past subgradients $\tilde{\nabla} f_i(x_{\ell_i})$, $i \neq i_k$. Moreover, several experimental studies have confirmed the theoretical convergence rate advantage of the IAG method over the corresponding incremental gradient method under the preceding favorable conditions. The use of arbitrary indexes $\ell_i \in [k-b, k]$ in the IAG method was introduced in the paper by Gurbuzbalaban, Ozdaglar, and Parillo [GOP15], who gave a simple linear convergence analysis.

Incremental Aggregated Proximal Algorithm

In this paper, we consider an incremental aggregated proximal algorithm (abbreviated IAP), which has the form

$$x_{k+1} \in \arg\min_{x \in X} \left\{ f_{i_k}(x) + \sum_{i \neq i_k} \tilde{\nabla} f_i(x_{\ell_i})'(x - x_k) + \frac{1}{2\alpha_k} \|x - x_k\|^2 \right\}, \tag{1.9}$$

where $\tilde{\nabla} f_i(x_{\ell_i})$ is a "delayed" subgradient of f_i at some earlier iterate x_{ℓ_i}. We assume that the indexes ℓ_i

satisfy the boundedness condition $\ell_i \in [k-b, k]$, cf. Eq. (1.6). Intuitively, the idea is that the term

$$\sum_{i \neq i_k} \tilde{\nabla} f_i(x_{\ell_i})'(x - x_k)$$

in the proximal minimization (1.9) is a linear approximation to the term

$$\sum_{i \neq i_k} f_i(x)$$

[minus the constant $\sum_{i \neq i_k} f_i(x_k)$], which would be used in the standard proximal algorithm

$$x_{k+1} \in \arg\min_{x \in X} \left\{ F(x) + \frac{1}{2\alpha_k} \|x - x_k\|^2 \right\}. \tag{1.10}$$

It is straightforward to verify the following equivalent form of the IAP iteration (1.9):

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

where

$$z_k = x_k - \alpha_k \sum_{i \neq i_k} \tilde{\nabla} f_i(x_{\ell_i}). \tag{1.12}$$

In this form the algorithm is executed as a two-step process: first use x_k and preceding subgradients to compute z_k via Eq. (1.12), and then execute an IP iteration starting from z_k. Note a limitation of this iteration over the IP iteration, which is shared with other incremental aggregated methods: to keep updating the vector z_k, one has to store the past subgradients $\tilde{\nabla} f_i(x_{\ell_i})$, $i \neq i_k$.

Similar to the IP iteration (1.4), the IAP iteration (1.9) and its equivalent form (1.11)-(1.12) can be written as

$$x_{k+1} = P_X \Big(z_k - \alpha_k \tilde{\nabla} f_{i_k}(x_{k+1}) \Big), \tag{1.13}$$

so when executing the iteration, we typically can obtain the subgradient $\tilde{\nabla} f_{i_k}(x_{k+1})$, which can be used in subsequent IAP iterations. For example, in the unconstrained case where $X = \Re^n$, from Eq. (1.13), we see that

$$\tilde{\nabla} f_{i_k}(x_{k+1}) = \frac{z_k - x_{k+1}}{\alpha_k}.$$

It is possible to prove various convergence results for the IAP iteration (1.9), or its equivalent forms (1.11)-(1.12) and (1.12)-(1.13), for the case where the stepsize α_k is diminishing and satisfies the standard conditions (1.7). These results are in line with similar results for the IP method, given in [Ber10], [Ber11], and for the IAS method (1.5), given in [NBB01]. Since the difference between the IAP and IAS methods is the use of $\tilde{\nabla} f_{i_k}(x_{k+1})$ in IAP in place of $\tilde{\nabla} f_{i_k}(x_{\ell_{i_k}})$ in IAS, intuitively, for a diminishing stepsize, the asymptotic performance of the two methods should be similar, and indeed the convergence proofs for the two methods are fairly similar, under comparable assumptions. We will thus not go into this convergence analysis.

Incremental Aggregated Proximal Algorithm for Unconstrained Problems

In the unconstrained case where $X = \Re^n$ and the component functions f_i are differentiable, the IAP iteration (1.13) can be written as

$$x_{k+1} = x_k - \alpha_k \left(\nabla f_{i_k}(x_{k+1}) + \sum_{i \neq i_k} \nabla f_i(x_{\ell_i}) \right). \tag{1.14}$$

In this case, one may expect similar convergence behavior for the IAP and IAG methods, under favorable conditions which allow the use of a constant stepsize $\alpha_k \equiv \alpha$. In particular, we prove the following for the IAP method.

Proposition 1.1: Assume that $X = \Re^n$ and that the functions f_i are convex and differentiable, and satisfy

$$\big\|\nabla f_i(x) - \nabla f_i(z)\big\| \leq L_i \|x - z\|, \qquad \forall\ x, z \in \Re^n,$$

for some constants L_i. Assume further that the function $F = \sum_{i=1}^m f_i$ is strongly convex with unique minimum denoted x^*. Then there exists $\overline{\alpha} > 0$ such that for all $\alpha \in (0, \overline{\alpha}]$, the sequence $\{x_k\}$ generated by the IAP iteration (1.14) with constant stepsize $\alpha_k \equiv \alpha$ converges to x^* linearly, in the sense that $\|x_k - x^*\| \leq \gamma \rho^k$ for some scalars $\gamma > 0$ and $\rho \in (0, 1)$, and all k.

The proof, given in Section 3 relies on the similarity of the iterations (1.14) and (1.8) [the use of the term $\nabla f_{i_k}(x_{k+1})$ in place of the term $\nabla f_{i_k}(x_{\ell_{i_k}})$]. A key idea is to view the IAP iteration (1.14) as a gradient method with errors in the calculation of the gradient, i.e.,

$$x_{k+1} = x_k - \alpha_k \big(\nabla F(x_k) + e_k \big), \tag{1.15}$$

where $\nabla F(x_k) = \sum_{i=1}^m \nabla f_i(x_k)$, and

$$e_k = \nabla f_{i_k}(x_{k+1}) - \nabla f_{i_k}(x_k) + \sum_{i \neq i_k} \big(\nabla f_i(x_{\ell_i}) - \nabla f_i(x_k) \big), \tag{1.16}$$

and then to appropriately bound the size of the errors e_k. This is similar to known lines of convergence proofs for gradient and subgradient methods with errors. The proof of Section 3 applies also to a diagonally scaled version of IAP, where a separate but constant stepsize is used for each coordinate.

We note that the line of proof of Prop. 1.1 does not readily extend to the constrained case when $X \neq \Re^n$, nor is it clear whether and under what conditions linear convergence can be proved. In Section 4, however, we will consider an incremental aggregated proximal algorithm that uses a nonquadratic regularization term and seems to cope better with the case of nonnegativity constraints, i.e., $X = \{x \mid x \geq 0\}$.

We finally return to the similarity of the IAP method (1.9) with the IAS method (1.5), and note that the two methods admit similar distributed asynchronous implementations, which was described in the paper [NBB01]. In this context, we have a central processor that executes the proximal iteration (1.9) for some selected component f_{i_k}, while other processors compute subgradients for other components f_i at points x_{ℓ_i}, which are supplied by the central processor. These subgradients involve a "delay" that may be unpredictable, hence the asynchronous character of the computation.

Local Versions of Proximal Algorithms

While the analysis of this paper requires that f_i and X are convex, there is a straightforward way to extend our incremental proximal methods to nonconvex problems involving twice differentiable functions, which we will describe briefly. The idea is to use a local version of the proximal algorithm, proposed in the author's paper [Ber79] and based on a local version of the Fenchel duality framework given in [Ber78]. The algorithm applies to the problem

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \quad g(x) = 0, \end{aligned} \tag{1.17}$$

where $f : \Re^n \mapsto \Re$ and $g : \Re^n \mapsto \Re^r$ are twice continuously differentiable functions, such that f is "locally convex" over the set $\big\{x \mid g(x) = 0\big\}$ (this is defined in terms of assumptions that relate to second order sufficiency conditions of nonlinear programming; see [Ber78], [Ber79]). The local proximal algorithm has the form

$$x_{k+1} \in \arg \min_{g(x)=0} \left\{ f(x) + \frac{1}{2\alpha_k} \|x - x_k\|^2 \right\}, \tag{1.18}$$

where α_k is sufficiently small to ensure that the function minimized in Eq. (1.18) is convex over $\Re^n$ [not just locally over the set $\big\{x \mid g(x) = 0\big\}$]. A Newton-like version of this algorithm was also given in [Ber79].

There is an incremental version of the local proximal iteration (1.18) for problems involving sums of functions. In particular, consider the problem

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^m f_i(x) \\ &\text{subject to} \quad g(x) = 0, \end{aligned} \tag{1.19}$$

where $f_i : \Re^n \mapsto \Re$ and $g : \Re^n \mapsto \Re^r$ are twice continuously differentiable functions, such that each f_i is "locally convex" over the set $\big\{x \mid g(x) = 0\big\}$, for all i. This incremental local proximal iteration is

$$x_{k+1} \in \arg \min_{g(x)=0} \left\{ f_{i_k}(x) + \frac{1}{2\alpha_k} \|x - x_k\|^2 \right\}, \tag{1.20}$$

where i_k is the index of the cost component that is iterated on. One may also consider an aggregated form of this incremental iteration. The convergence properties of these algorithms are an interesting subject for investigation, which lies, however, outside the scope of the present paper.

There is also another way to combine local proximal and incremental ideas for the case of the (nonconvex) separable problem in the vector $x = (x^1, \ldots, x^m)$,

$$\begin{aligned} &\text{minimize} \quad f(x) \overset{\text{def}}{=} \sum_{i=1}^{m} f_i(x^i) \\ &\text{subject to} \ \ g(x) \overset{\text{def}}{=} \sum_{i=1}^{m} g_i(x^i) = 0, \end{aligned} \tag{1.21}$$

where $f_i : \Re^{n_i} \mapsto \Re$ and $g_i : \Re^{n_i} \mapsto \Re^r$ are twice continuously differentiable functions, and are such that the problem admits a solution-Lagrange multiplier pair (x^*, λ^*) satisfying standard second order sufficiency conditions. In this approach, also developed in [Ber78], [Ber79], the problem (1.21) is converted to the equivalent problem

$$\begin{aligned} &\text{minimize} \quad \phi_\gamma(z) \overset{\text{def}}{=} \min_{g(x)=0} \left\{ f(x) + \frac{1}{2\gamma}\|x - z\|^2 \right\} \\ &\text{subject to} \ \ z \in \Re^{n_1 + \cdots + n_m}, \end{aligned} \tag{1.22}$$

where γ is sufficiently small so that for fixed z, $f(x) + \frac{1}{2\gamma}\|x - z\|^2$ is convex in x locally, for all x in a suitably small neighborhood of x^*, i.e., γ should be such that $\frac{1}{\gamma}I + \nabla^2 f(x^*)$ is positive definite. Since the minimization problem (1.22), which defines $\phi_\gamma(z)$, is separable of the form

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^{m} \left(f_i(x^i) + \frac{1}{2\gamma}\|x^i - z^i\|^2 \right) \\ &\text{subject to} \ \ x \in \Re^{n_1 + \cdots + n_m}, \quad \sum_{i=1}^{m} g_i(x^i) = 0, \end{aligned} \tag{1.23}$$

and locally convex in x, for fixed z and suitably small values of γ, it can be solved using the augmented Lagrangian-based methods of the next section. Denoting $x(z, \gamma)$ the optimal solution of this problem for given z and γ, it is shown in [Ber79] (Prop. 2.1) (see also [Ber78], Prop. 2) that ϕ_γ is differentiable and

$$\nabla \phi_\gamma(z) = \frac{1}{\gamma}\big(z - x(z, \gamma)\big).$$

Thus the gradient algorithm

$$z_{k+1} = z_k - \gamma \nabla \phi_\gamma(z_k), \tag{1.24}$$

can be written as $z_{k+1} = x(z_k, \gamma)$ or equivalently, using Eqs. (1.22) and (1.23), in the (local) proximal form

$$x_{k+1} \in \arg \min_{\sum_{i=1}^m g_i(x^i)=0} \left\{ \sum_{i=1}^{m} \left(f_i(x^i) + \frac{1}{2\gamma}\|x^i - x_k^i\|^2 \right) \right\}. \tag{1.25}$$

Note that the above minimization is amenable to decomposition, including solution using the incremental aggregated augmented Lagrangian and ADMM methods of the next section, assuming γ is sufficiently small to induce the required amount of convexification to make problem (1.25) convex (locally within a neighborhood of x^*).

The convergence properties of this algorithm are developed in [Ber79], based on a local theory of conjugate functions and Fenchel duality developed in [Ber78]. We refer to these papers for a discussion of the local aspects of the minimization (1.25), as well as for the implementation of the Newton iteration

$$z_{k+1} = z_k - \big(\nabla^2 \phi_\gamma(z_k)\big)^{-1} \nabla \phi_\gamma(z_k), \tag{1.26}$$

in analogy with the gradient method (1.24). A further analysis is again outside the scope of the present paper, and is an interesting subject for investigation.

2. INCREMENTAL AUGMENTED LAGRANGIAN METHODS

A second objective of this paper is to consider the application of the IP and IAP methods in a dual setting, where they take the form of incremental augmented Lagrangian algorithms for the separable constrained optimization problem

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^m h_i(y^i) \\ &\text{subject to} \quad y^i \in Y_i, \quad i = 1, \ldots, m, \qquad \sum_{i=1}^m (A_i y^i - b_i) = 0, \end{aligned} \tag{2.1}$$

as shown in [Ber15], Section 6.4.3. Here $h_i : \Re^{n_i} \mapsto \Re$ are convex functions (n_i is a positive integer, which may depend on i), Y_i are nonempty closed convex subsets of $\Re^{n_i}$, A_i are given $r \times n_i$ matrices, and $b_i \in \Re^r$ are given vectors. The optimization vector is $y = (y^1, \ldots, y^m)$, and our objective is to consider algorithms that allow decomposition in the minimization of the augmented Lagrangian, so that m separate augmented Lagrangian minimizations are performed, each with respect to a single component y^i. Note that the problem (2.1) is unaffected by redefinition of the scalars b_i, as long as $\sum_{i=1}^m b_i$ is not changed. It may be beneficial to adjust the scalars b_i so that the residuals $A_i y^i - b_i$ are small near the optimal, and this may in fact be attempted in the course of some algorithms as a form of heuristic.

Following a standard analysis, the dual function for problem (2.1) is given by

$$Q(\lambda) = \inf_{y^i \in Y_i,\, i=1,\ldots,m} \left\{ \sum_{i=1}^m \big(h_i(y^i) + \lambda'(A_i y^i - b_i) \big) \right\}, \tag{2.2}$$

where $\lambda \in \Re^r$ is the dual vector. By decomposing the minimization over the components y^i, Q can be expressed in the additive form

$$Q(\lambda) = \sum_{i=1}^m q_i(\lambda),$$

where q_i is the concave function

$$q_i(\lambda) = \inf_{y^i \in Y_i} \big\{ h_i(y^i) + \lambda'(A_i y^i - b_i) \big\}, \qquad i = 1, \ldots, m. \tag{2.3}$$

Assuming that the dual function components q_i are real-valued (which is true for example if Y_i is compact), the dual function $Q(\lambda)$ can be minimized with the classical subgradient method.† This method takes the form

$$\lambda_{k+1} = \lambda_k + \alpha_k \sum_{i=1}^{m} \tilde{\nabla} q_i(\lambda_k), \tag{2.4}$$

where $\alpha_k > 0$ is the stepsize and the subgradients $\tilde{\nabla} q_i(\lambda_k)$ are obtained as

$$\tilde{\nabla} q_i(\lambda_k) = A_i y^i_{k+1} - b_i, \qquad i = 1, \ldots, m,$$

with all components y^i updated according to

$$y^i_{k+1} \in \arg \min_{y^i \in Y_i} \big\{ h_i(y^i) + \lambda_k'(A_i y^i - b_i) \big\}, \qquad i = 1, \ldots, m.$$

The additive form of the dual function Q makes it suitable for application of incremental methods, including the IAS method described in Section 1, which in fact was proposed in [NBB01] with the separable problem (2.1) in mind. In the case where the components q_i are differentiable [which is true if the infimum in the definition (2.3) is attained uniquely for all λ], one may also use the IAG method with a constant but sufficiently small stepsize. This is an incremental aggregated version of a classical dual gradient method proposed in the 60s and often attributed to Everett [Eve63]. It takes the form

$$\lambda_{k+1} = \lambda_k + \alpha \left(\nabla q_{i_k}(\lambda_k) + \sum_{i \neq i_k} \nabla q_i(\lambda_{\ell_i}) \right); \tag{2.5}$$

cf. Eq. (1.8). The gradient of the dual function component q_i is given by

$$\nabla q_i(\lambda) = A_i y^i(\lambda) - b_i,$$

where $y^i(\lambda)$ is the minimizer over Y_i of

$$f_i(y^i) + \lambda' A_i y^i,$$

which is assumed to be unique for differentiability of q_i. By streamlining the computations using the preceding relations, we see that the iteration has the following form.

† In the case where q_i is not real-valued, the dual function can be maximized over the set $\cap_{i=1}^m \Lambda_i$, where $\Lambda_i = \big\{ \lambda \mid q_i(\lambda > -\infty \big\}$. This can be done by using incremental constraint projection methods involving projection or proximal maximization over a single set Λ_i at a time. Methods of this type have been proposed in [Ber11], [Ned11], [WaB13], [WaB15], but their discussion is beyond the scope of the present paper.

Incremental Aggregated Dual Gradient Iteration (IADG)

Select a component index i_k, and update the single component y^{i_k} according to

$$y^{i_k}_{k+1} \in \arg \min_{y^{i_k} \in Y_{i_k}} \big\{ h_{i_k}(y^{i_k}) + \lambda_k' A_{i_k} y^{i_k} \big\}, \tag{2.6}$$

while keeping the others unchanged, $y^i_{k+1} = y^i_k$ for all $i \neq i_k$. Then update λ according to

$$\lambda_{k+1} = \lambda_k + \alpha \left(A_{i_k} y^{i_k}_{k+1} + \sum_{i \neq i_k} A_i y^i_{\ell_i} - b \right). \tag{2.7}$$

The convergence properties of the method are governed by the known results for the IAG method, which were noted in Section 1. In particular, we obtain linear convergence with a constant sufficiently small stepsize α, assuming Lipschitz continuity of ∇q_i and strong convexity of Q, and that the long-term frequency of updating y^i is the same for all i. Note, however, that this linear convergence result cannot be used when the primal problem (2.1) has additional convex inequality constraints, because then the corresponding dual problem involves nonnegativity constraints.

Augmented Lagrangian-Based Algorithms for Separable Problems

The nonincremental and incremental subgradient and gradient methods just described are convenient for the purposes of decomposition, but their convergence properties tend to be fragile. On the other hand, the more stable augmented Lagrangian methods have a major drawback: when a quadratic penalty term is added to the Lagrangian function, the resulting augmented Lagrangian

$$\sum_{i=1}^{m} \big(h_i(y^i) + \lambda'(A_i y^i - b_i) \big) + \frac{\alpha_k}{2} \left\| \sum_{i=1}^{m} (A_i y^i - b_i) \right\|^2$$

is not separable any more, and is not amenable to minimization by decomposition. This is a well-known limitation of the augmented Lagrangian approach that has been addressed by a number of authors with various algorithmic proposals, which we will now survey.

The first proposal of this type was the paper by Stephanopoulos and Westerberg [StW75], which was based on enforced decomposition: minimizing the augmented Lagrangian separately with respect to each component vector y^i, while holding the other components fixed at some estimated values. Minimization over the components y^i is followed by a multiplier update (using the standard augmented Lagrangian formula).

The decomposition method of [StW75] attracted considerable attention and motivated further research, including the similarly structured methods by Tadjewski [Tad89] and by Ruszczynski [Rus95], which include convergence analyses and give references to earlier works. Our incremental aggregated proximal algorithm bears similarity with the methods of [StW75], [Tad89], and [Rus95]. We note, however, that the methods of [StW75] and [Tad89] were motivated by nonconvex separable problems for which there is a duality gap, while our analysis requires a convex programming structure, where there is no duality gap. The method of [Rus95] is applied to convex separable problems, including linear programming.

Another method for convex separable problems that uses augmented Lagrangian minimizations is given by Deng, Lai, Peng, and Yin [DLP14], who give several related references, including the paper by Chen and Teboulle [ChT94]. The method is based on the use of primal proximal terms in the augmented Lagrangian (in addition to the quadratic penalty term). This is in the spirit of Rockafellar's proximal method of multipliers [Roc76b], and involves two separate penalty parameters, which for convergence should satisfy certain restrictions. The papers by Hong and Luo [HoL13], and Robinson and Tappenden [RoT15] also propose algorithms that use primal proximal terms and two penalty parameters, but differ from the algorithm of [DLP14] in that they update the primal variables in Gauss-Seidel rather than Jacobi fashion, while requiring additional assumptions (see also Dang and Lan [DaL15] for a related algorithm). Gauss-Seidel updating is somewhat similar to the incremental mode of iteration of this paper, and based on the results of experiments in [WHM13] and [RoT15], it appears to be beneficial.

A different possibility to deal with nonconvex separable problems is based on the convexification provided by the local proximal algorithm that was discussed at the end of the preceding section. Its application to nonconvex separable problems is described in [Ber79]; see also Tanikawa and Mukai [TaM85], who proposed a method that aims at improved efficiency relative to the approach of [Ber79]. A discussion of additional proposals of decomposition methods that use augmented Lagrangians is given in the recent paper by Hamdi and Mishra [HaM11].

Still another approach that has been used to exploit the structure of the separable problem (2.1) is the alternating direction method of multipliers (ADMM), a popular method for convex programming, first proposed by Glowinskii and Morocco [GlM75], and Gabay and Mercier [GaM76], and further developed by Gabay [Gab79], [Gab83]. This method applies to the problem

$$\begin{aligned} &\text{minimize} \quad f_1(x) + f_2(z) \\ &\text{subject to} \quad x \in \Re^n,\ z \in \Re^m, \quad Ax = z, \end{aligned} \tag{2.8}$$

where $f_1 : \Re^n \mapsto (-\infty, \infty]$ and $f_2 : \Re^m \mapsto (-\infty, \infty]$ are closed convex functions, and A is a given $m \times n$ matrix. The method is better suited than the augmented Lagrangian method for exploiting special structures, including separability, and is capable of decoupling the vectors x and z in the augmented Lagrangian

$$f_1(x) + f_2(z) + \lambda'(Ax - z) + \frac{\alpha}{2}\|Ax - z\|^2.$$

For a discussion of the properties and the many applications of the method, we refer to its extensive literature, including the books [BeT89], Section 3.4.4, [Ber15], Section 5.4, and [BPC11], which give many references. The form of the ADMM for separable problems to overcome the coupling of variables in the augmented Lagrangian minimization was first derived in Bertsekas and Tsitsiklis [BeT89], Section 3.4, pp. 249-254 (see also [Ber15], Section 5.4.2). We will describe the form of this specialized ADMM later in this section.

We will now consider the incremental proximal methods IP [cf. Eq. (1.3)] and IAP [cf. Eq. (1.9)] for maximizing the dual function $\sum_{i=1}^{m} q_i(\lambda)$. Taking into account the concavity of the components q_i, the IP method takes the form

$$\lambda_{k+1} \in \arg\max_{\lambda\in\Re^r} \left\{ q_{i_k}(\lambda) - \frac{1}{2\alpha_k}\|\lambda - \lambda_k\|^2 \right\}, \tag{2.9}$$

where i_k is the index of the component chosen for iteration and α_k is a positive parameter. This method was given in [Ber15], Section 6.4.3, where it was shown that it can be implemented through the use of decoupled augmented Lagrangian minimizations, each involving a single component vector y^i. The IAP method takes the form

$$\lambda_{k+1} \in \arg\max_{\lambda\in\Re^r} \left\{ q_{i_k}(\lambda) + \sum_{i\neq i_k} \tilde\nabla q_i(\lambda_{\ell_i})'(\lambda - \lambda_k) - \frac{1}{2\alpha_k}\|\lambda - \lambda_k\|^2 \right\}, \tag{2.10}$$

and has not been considered earlier within the dual separable constrained optimization context of this section. The convergence results noted in Section 1 apply to this method. In particular, by Prop. 1.1, the IAP method (2.10) is convergent with a sufficiently small constant stepsize, assuming that each q_i is differentiable with Lipschitz continuous gradient and Q is strongly concave. Of course, the differentiability of q_i is a restrictive assumption, and it amounts to attainment of the minimum at a unique point $y^i \in Y_i$ in the definition (2.3) of $q_i(\lambda)$ for all $\lambda \in \Re^r$.

We will now describe how the incremental proximal methods IP and IAP can be implemented in terms of augmented Lagrangian minimizations, which decompose with respect to components y^i and have an incremental character. To this end, we will review the well-known Fenchel duality relation between proximal and augmented Lagrangian iterations, given first by Rockafellar [Roc73], [Roc76b], and subsequently in many sources, including the author's monograph and textbook accounts [Ber82], Chapter 5, and [Ber15], Section 5.2.1.

Duality Between Proximal and Augmented Lagrangian Iterations

Given a proper convex function $P : \Re^r \mapsto (-\infty, \infty]$, let $Q : \Re^r \mapsto [-\infty, \infty)$ be the closed proper concave function defined by†

$$Q(\lambda) = \inf_{u\in\Re^r} \big\{ P(u) + \lambda' u \big\}. \tag{2.11}$$

† Here and later, for concave functions Q, we use terminology used for convex functions as applied to $-Q$.

This is a conjugacy relation, since $Q(\lambda) = -P^\star(-\lambda)$, where $P^\star$ is the conjugate convex function of P. Moreover, if P is closed, it can be recovered from Q using the conjugacy theorem,

$$P(u) = P^{\star\star}(u) = \sup_{\lambda\in\Re^r} \big\{\lambda' u + Q(-\lambda)\big\}, \tag{2.12}$$

where $P^{\star\star}$ is the conjugate convex function of $P^\star$ (see, e.g., [Ber09], Prop. 1.6.1).

A key fact, assuming that P is closed, is that the proximal iteration

$$\lambda_{k+1} \in \arg\max_{\lambda\in\Re^r} \left\{ Q(\lambda) - \frac{1}{2\alpha_k}\|\lambda - \lambda_k\|^2 \right\}, \tag{2.13}$$

can be equivalently implemented in two steps as

$$u_{k+1} \in \arg\min_{u\in\Re^r} \left\{ P(u) + \lambda_k' u + \frac{\alpha_k}{2}\|u\|^2 \right\}, \tag{2.14}$$

followed by

$$\lambda_{k+1} = \lambda_k + \alpha_k u_{k+1}; \tag{2.15}$$

see, e.g., [Ber15], Section 5.2.1. Moreover, u_{k+1} is a subgradient of Q at λ_{k+1}:

$$u_{k+1} = \tilde\nabla Q(\lambda_{k+1}). \tag{2.16}$$

These relations are shown by straightforward application of the Fenchel duality theorem to the maximization of Eq. (2.13), which involves the sum of the concave functions Q and $-(1/2\alpha_k)\|\lambda - \lambda_k\|^2$. The closedness of P is used both to ensure that the duality relation (2.12) holds, and to guarantee that the minimum in Eq. (2.14) is attained. Note that Eq. (2.14) has the form of an augmented Lagrangian minimization relating to the (somewhat contrived) problem of minimizing P subject to the equality constraint $u = 0$.

Augmented Lagrangian Method

We will now translate the duality between the proximal and augmented Lagrangian iterations just described to the constrained optimization context, setting the stage for using this duality in an incremental context. Consider a generic convex programming problem of the form

$$\begin{aligned} &\text{minimize} \quad H(y) \\ &\text{subject to} \quad y \in Y, \qquad Ay - b = 0, \end{aligned} \tag{2.17}$$

where $H : \Re^n \mapsto \Re$ is a convex function, Y is a convex set, A is an $r \times n$ matrix, and $b \in \Re^r$. Consider also the corresponding primal and dual functions

$$P(u) = \inf_{y\in Y,\, Ay-b=u} H(y), \qquad Q(\lambda) = \inf_{y\in Y} \big\{ H(y) + \lambda'(Ay - b)\big\},$$

which are convex and concave, respectively. We assume that P is closed and proper, and that the optimal value of the problem is finite, so that Q is also closed proper and concave, and there is no duality gap (see [Ber09], Section 4.2).

There is a well-known relation between the primal and dual functions. In particular, Q has the equivalent form

$$Q(\lambda) = \inf_{u\in\Re^r} \inf_{y\in Y,\, Ay-b=u} \big\{H(y) + \lambda'(Ay-b)\big\} = \inf_{u\in\Re^r} \big\{P(u) + \lambda' u\big\},$$

so P and Q satisfy the conjugacy relation (2.11). Based on the preceding discussion [cf. (2.11)-(2.16)], it follows that the proximal iteration (2.13) can be equivalently written as the two-step process (2.14)-(2.15)

$$u_{k+1} \in \arg\min_{u\in\Re^r} \left\{P(u) + \lambda_k' u + \frac{\alpha_k}{2}\|u\|^2\right\}, \tag{2.18}$$

followed by

$$\lambda_{k+1} = \lambda_k + \alpha_k u_{k+1}. \tag{2.19}$$

Moreover, from Eqs. (2.15) and (2.16), we have

$$u_{k+1} = \frac{\lambda_{k+1} - \lambda_k}{\alpha_k} = \tilde\nabla Q(\lambda_{k+1}). \tag{2.20}$$

We will now write the iteration (2.18)-(2.19) in terms of the augmented Lagrangian, and obtain the classical (first order) augmented Lagrangian method. Using the definition of the primal function P, we see that the minimization in Eq. (2.18) can be written as

$$\begin{aligned}
&\inf_{u\in\Re^r} \left\{P(u) + {\lambda_k}' u + \frac{\alpha_k}{2}\|u\|^2\right\} \\
&\quad= \inf_{u\in\Re^r} \left\{\inf_{y\in Y,\, Ay-b=u} \big\{H(y)\big\} + {\lambda_k}' u + \frac{\alpha_k}{2}\|u\|^2\right\} \\
&\quad= \inf_{u\in\Re^r} \inf_{y\in Y,\, Ay-b=u} \left\{H(y) + \lambda_k'(Ay-b) + \frac{\alpha_k}{2}\|Ay-b\|^2\right\} \\
&\quad= \inf_{y\in Y} \left\{H(y) + \lambda_k'(Ay-b) + \frac{\alpha_k}{2}\|Ay-b\|^2\right\} \\
&\quad= \inf_{y\in Y} L_{\alpha_k}(y, \lambda_k),
\end{aligned}$$

where for any $\alpha > 0$, L_α is the augmented Lagrangian function

$$L_\alpha(y,\lambda) = H(y) + \lambda'(Ay-b) + \frac{\alpha}{2}\|Ay-b\|^2, \qquad y \in \Re^n,\ \lambda \in \Re^r. \tag{2.21}$$

From the preceding calculation it also follows that for any $y_{k+1} \in Y$ that minimizes the augmented Lagrangian over Y:

$$y_{k+1} \in \arg\min_{y\in Y} L_{\alpha_k}(y, \lambda_k), \tag{2.22}$$

we have $u_{k+1} = Ay_{k+1} - b$, and the iteration (2.19) can be equivalently written as the multiplier iteration

$$\lambda_{k+1} = \lambda_k + \alpha_k(Ay_{k+1} - b). \tag{2.23}$$

This is precisely the (first order) augmented Lagrangian method. It is equivalent to the proximal iteration

$$\lambda_{k+1} \in \arg \max_{\lambda \in \Re^r} \left\{ Q(\lambda) - \frac{1}{2\alpha_k} \|\lambda - \lambda_k\|^2 \right\},$$

[cf. Eq. (2.13)]. In view of Eqs. (2.20) and (2.23), it can also be written in the gradient-like form

$$\lambda_{k+1} = \lambda_k + \alpha_k \tilde{\nabla} Q(\lambda_{k+1}), \tag{2.24}$$

where $\tilde{\nabla} Q(\lambda_{k+1})$, the special subgradient of Q at λ_{k+1}, is given by

$$\tilde{\nabla} Q(\lambda_{k+1}) = A y_{k+1} - b. \tag{2.25}$$

Note that the minimizing y_{k+1} in Eq. (2.22) need not exist or be unique. Its existence must be assumed in some way, e.g., by assuming that H has compact level sets. As an example, it can be verified that for the two-dimensional/single constraint problem of minimizing $H(y) = e^{y^1}$, subject to $y^1 + y^2 = 0$, $y^1 \in \Re$, $y^2 \geq 0$, the dual optimal solution is $\lambda^* = 0$, but there is no primal optimal solution. For this problem, the augmented Lagrangian algorithm will generate sequences $\{\lambda_k\}$ and $\{y_k\}$ such that $\lambda_k \to 0$ and $y_k \to -\infty$.

Incremental Augmented Lagrangian Methods

The duality between the proximal and augmented Lagrangian minimizations outlined above is generic, and holds in other related contexts, based on a similar use of the Fenchel duality theorem. In the context of the separable problem (2.1), it holds in an incremental form where $Q(\lambda)$ is replaced by

$$q_{i_k}(\lambda),$$

as in the IP iteration (2.9), or is replaced by

$$q_{i_k}(\lambda) + \sum_{i \neq i_k} \tilde{\nabla} q_i(\lambda_{\ell_i})'(\lambda - \lambda_k),$$

as in the IAP iteration (2.10). We refer to these two methods as the *incremental augmented Lagrangian method* (abbreviated IAL), and the *incremental aggregated augmented Lagrangian method* (abbreviated IAAL).

Based on the discussion of the algorithm (2.22)-(2.24), the IAL method,

$$\lambda_{k+1} \in \arg \max_{\lambda \in \Re^r} \left\{ q_{i_k}(\lambda) - \frac{1}{2\alpha_k} \|\lambda - \lambda_k\|^2 \right\},$$

can be implemented as follows, as already noted in [Ber15], Section 6.4.3.

Incremental Augmented Lagrangian Iteration (IAL)

Select a component index i_k, and update the single component y^{i_k} according to

$$y^{i_k}_{k+1} \in \arg \min_{y^{i_k} \in Y_{i_k}} \left\{ h_{i_k}(y^{i_k}) + \lambda'_k(A_{i_k} y^{i_k} - b_{i_k}) + \frac{\alpha_k}{2} \|A_{i_k} y^{i_k} - b_{i_k}\|^2 \right\}, \tag{2.26}$$

while keeping the others unchanged, $y^i_{k+1} = y^i_k$ for all $i \neq i_k$. Then update λ according to

$$\lambda_{k+1} = \lambda_k + \alpha_k (A_{i_k} y^{i_k}_{k+1} - b_{i_k}). \tag{2.27}$$

As in the IP method, all component indexes should be selected for iteration in Eq. (2.26) with equal long-term frequency. Note that the augmented Lagrangian minimization is decoupled with respect to the components y^i, thus overcoming the major limitation of the augmented Lagrangian approach for separable problems.

To derive the IAAL method, we use the equivalent form (1.11)-(1.12) of the IAP algorithm. We see then that the method has similar form to the IAL method, except that λ_k is first translated by a multiple of the sum of the delayed subgradients. In particular, the IAAL iteration takes the form

$$\lambda_{k+1} \in \arg \max_{\lambda \in \Re^r} \left\{ q_{i_k}(\lambda) - \frac{1}{2\alpha_k} \|\lambda - \nu_k\|^2 \right\},$$

where

$$\nu_k = \lambda_k + \alpha_k \sum_{i \neq i_k} \tilde{\nabla} q_i(\lambda_{\ell_i}). \tag{2.28}$$

Applying the relations (2.22)-(2.24), it follows that we can write the IAAL iteration in two steps: Select a component index i_k, and update the single component y^{i_k} according to

$$y^{i_k}_{k+1} \in \arg \min_{y^{i_k} \in Y_{i_k}} \left\{ h_{i_k}(y^{i_k}) + \nu'_k(A_{i_k} y^{i_k} - b_{i_k}) + \frac{\alpha_k}{2} \|A_{i_k} y^{i_k} - b_{i_k}\|^2 \right\}, \tag{2.29}$$

while keeping the others unchanged, $y^i_{k+1} = y^i_k$ for all $i \neq i_k$. Then update λ according to

$$\lambda_{k+1} = \nu_k + \alpha_k (A_{i_k} y^{i_k}_{k+1} - b_{i_k}). \tag{2.30}$$

Note that the subgradients $\tilde{\nabla} q_i(\lambda_{\ell_i})$, needed for the computation of ν_k in Eq. (2.28), are generated by

$$\tilde{\nabla} q_i(\lambda_{\ell_i}) = A_i y^i_{\ell_i} - b_i, \qquad \forall\ i \neq i_k,$$

[cf. Eq. (2.25)]. Thus by streamlining the preceding relations, we see that the IAAL updates are written as

$$y_{k+1}^{i_k} \in \arg \min_{y^{i_k} \in Y_{i_k}} \left\{ h_{i_k}(y^{i_k}) + \lambda_k'(A_{i_k} y^{i_k} - b_{i_k}) + \frac{\alpha_k}{2} \left\| A_{i_k} y^{i_k} - b_{i_k} + \sum_{i \neq i_k} (A_i y_{\ell_i}^i - b_i) \right\|^2 \right\},$$

$$\lambda_{k+1} = \lambda_k + \alpha_k \left(A_{i_k} y_{k+1}^{i_k} - b_{i_k} + \sum_{i \neq i_k} (A_i y_{\ell_i}^i - b_i) \right).$$

If we denote $b = \sum_{i=1}^m b_i$, and neglect the constant term $-\lambda_k' b_{i_k}$ from the augmented Lagrangian, we can write the iteration in a way that it depends on the scalars b_i only through their sum b.

Incremental Aggregated Augmented Lagrangian (IAAL) Iteration

Select a component index i_k, and update the single component y^{i_k} according to

$$y_{k+1}^{i_k} \in \arg \min_{y^{i_k} \in Y_{i_k}} \left\{ h_{i_k}(y^{i_k}) + \lambda_k' A_{i_k} y^{i_k} + \frac{\alpha_k}{2} \left\| A_{i_k} y^{i_k} + \sum_{i \neq i_k} A_i y_{\ell_i}^i - b \right\|^2 \right\}, \tag{2.31}$$

while keeping the others unchanged, $y_{k+1}^i = y_k^i$ for all $i \neq i_k$. Then update λ according to

$$\lambda_{k+1} = \lambda_k + \alpha_k \left(A_{i_k} y_{k+1}^{i_k} + \sum_{i \neq i_k} A_i y_{\ell_i}^i - b \right). \tag{2.32}$$

By comparing the IAL method (2.26)-(2.27) with the IAAL method (2.31)-(2.32), we see that they require comparable computations per iteration. While the IAL method requires a diminishing stepsize α_k for convergence, the IAAL method can converge with a constant stepsize, assuming that the dual function components have Lipschitz continuous gradients, and the dual function is strongly concave (cf. Prop. 1.1). Intuitively, if it can use a constant stepsize, the IAAL method should be asymptotically more effective than the IAL method. Of course, if Q is not strongly convex (as for example in the important case where Q is polyhedral, which arises in integer programming), our analysis guarantees the convergence of the IAAL method only if the stepsize α_k is diminishing. In this case it is unclear which of the IAL and IAAL methods is more effective on a given problem.

Both the IAL and IAAL algorithms require an initial multiplier λ_0. Regarding the delayed indexes ℓ_i in the IAAL algorithm, if the iteration is executed at a single processor, it is most appropriate to choose ℓ_i to be the iteration index at which the component y^i was last changed prior to the current index k, so $\ell_i \leq k$ (if a component y^i has not yet been updated prior to k, we take $\ell_i = 0$ and let y_0^i be some initial choice for y^i). In this case, the formal statement of the IAAL method is again given by Eqs. (2.29)-(2.30), with ℓ_i replaced by k for all $i \neq i_k$. However, a different value of ℓ_i may apply if the iteration is executed in a distributed asynchronous computing environment, as in the corresponding IAS method of [NBB01].

Note that the multiplier λ_k is updated each time a component y^i is updated, which suggests that the stepsize α_k should be chosen carefully, possibly through some experimentation. Moreover, the strong convexity assumption of Q is essential for the convergence of the method with a constant stepsize. Indeed a three-dimensional example by Chen, He, Ye, and Yuan [CHH14] can be used to show that the IAAL algorithm need not converge for any value of constant stepsize if the strong convexity assumption is violated.† An alternative possibility is to perform a batch of component updates y^i of the form (2.29) between multiplier updates of the form (2.30). For example, one may restructure the IAAL iteration so that it consists of a full cycle of updates of $y^1, \ldots, y^m$, sequentially according to Eq. (2.31), to obtain y^i_{k+1}, $i = 1, \ldots, m$, and only then to update λ according to

$$\lambda_{k+1} = \lambda_k + \alpha_k \left(\sum_{i=1}^{m} A_i y^i_{k+1} - b \right).$$

Note that this sequential update of $y^1, \ldots, y^m$ according to Eq. (2.31) amounts to a cycle of coordinate descent iterations for minimizing the augmented Lagrangian. Therefore, this variant of the IAAL iteration may be viewed as an implementation of the augmented Lagrangian method with approximate minimization of the augmented Lagrangian using coordinate descent. An algorithm of this type may be interesting and has been suggested in the past (see Bertsekas and Tsitsiklis [BeT89], Example 4.4, and Eckstein [Eck12]). Its linear convergence has been shown under certain assumptions by Hong and Luo [HoL13]. The algorithm is worthy of further investigation, particularly in view of favorable computational results given by Wang, Hong, Ma, and Luo [WHM13]. Let us also note that the work by Hong, Chang, Wang, Razaviyayn, Ma, and Luo [HCW14] derives an algorithm for the separable problem (2.1) that is quite similar to the IAAL algorithm, using different assumptions and line of development. The paper [HCW14] proves convergence but not a linear convergence rate result.

Comparison with ADMM

We will now compare the IAAL iteration with the ADMM. We note that there is a well-known connection of the ADMM and augmented Lagrangian methods, which was clarified long ago through a series of papers. In particular, Lions and Mercier [LiM79] proposed a splitting algorithm for finding a zero of the sum of two maximal monotone operators, known as the Douglas-Ratchford algorithm. It turns out that this algorithm

† While the paper [CHH14] is entitled “The Direct Extension of ADMM for Multi-Block Convex Minimization Problems ...,” it considers an algorithm that is not a special case of ADMM, so a convergence counterexample is possible. A correct specialization of ADMM for separable problems (dating from 1989 but unknown to the authors of [CHH14]) will be given shortly, and is convergent under the same broadly applicable conditions as ADMM.

contains as a special case the ADMM, as shown in [Gab83]. The paper by Eckstein and Bertsekas [EcB92] showed that the general form of the proximal algorithm for finding a zero of maximal monotone operator, proposed by Rockafellar [Roc76a], [Roc76b], contains as a special case the Douglas-Ratchford algorithm and hence also the ADMM. Thus the ADMM and the augmented Lagrangian method have a common ancestry: they are both special cases of the general form of the proximal algorithm for finding a zero of a maximal monotone operator. The common underlying structure of the two methods is reflected in similar formulas, but ADMM has the advantage of flexibility to allow decomposition, at the expense of a typically slower practical convergence rate.

A convenient decomposition-based form of ADMM for the separable problem (2.1) was derived (together with the corresponding coordinate descent version of the augmented Lagrangian method) in [BeT89], Section 3.4 and Example 4.4 (see also [Ber15], Section 5.4.2). Wang, Hong, Ma, and Luo [WHM13], apparently unaware of this form of ADMM, give related algorithms (referred to as Algorithms 2 and 3 in their paper), which, however, involve updating m multiplier vectors in place of the single multiplier update of the following algorithm. At iteration k, and given λ_k, the ADMM algorithm of [BeT89] generates λ_{k+1} as follows.

ADMM Iteration for Separable Problems

Perform a separate augmented Lagrangian minimization over y^i, for each $i = 1, \ldots, m$,

$$y^i_{k+1} \in \arg\min_{y^i \in Y_i} \left\{ h_i(y^i) + \lambda_k' A_i y^i + \frac{\alpha}{2} \left\| A_i y^i - A_i y^i_k + \frac{1}{m} \left(\sum_{j=1}^m A_j y^j_k - b \right) \right\|^2 \right\}, \qquad i = 1, \ldots, m, \tag{2.33}$$

and then update λ_k according to

$$\lambda_{k+1} = \lambda_k + \frac{\alpha}{m} \left(\sum_{i=1}^m A_i y^i_{k+1} - b \right). \tag{2.34}$$

Note that contrary to the augmented Lagrangian method, where the best strategy for adjusting α is usually clear, see e.g., [Ber82], there is no clear way to adjust the parameter α to improve performance in ADMM. As a result for efficiency α is often determined by trial and error. A closely related but more refined form of ADMM, also derived in [BeT89a], Section 3.4, Example 4.4, aims to improve the parameter selection by exploiting the structure of the matrices A_i. It uses a coordinate-dependent parameter $\frac{\alpha}{m_j}$ in iteration (2.34), in place of α/m, where m_j is the number of submatrices A_i that have nonzero jth row. In this version, the multiplier update essentially involves diagonal scaling. The iteration maintains additional vectors $z^i_k \in \Re^r$, $i = 1, \ldots, m$, which represent estimates of $A_i y^i$ at the optimum, and has the following form, where A_{ji} denotes the jth row of the matrix A_i.

Diagonally Scaled ADMM Iteration for Separable Problems

Perform a separate augmented Lagrangian minimization over y^i, for each $i = 1, \dots, m$,

$$y^i_{k+1} \in \arg \min_{y^i \in Y_i} \left\{ h_i(y^i) + \lambda_k' A_i y^i + \frac{\alpha}{2} \left\| A_i y^i - z^i_k \right\|^2 \right\}, \qquad i = 1, \dots, m, \tag{2.35}$$

and then update λ_k and z_k according to

$$\lambda^j_{k+1} = \lambda^j_k + \frac{\alpha}{m_j} \left(\sum_{i=1}^m A_{ji} y^i_{k+1} - b_j \right), \qquad j = 1, \dots, r, \tag{2.36}$$

$$z^i_{k+1} = A_i y^i_{k+1} + \frac{\lambda_k - \lambda_{k+1}}{\alpha}, \qquad i = 1, \dots, m. \tag{2.37}$$

Note that the preceding two ADMM iterations coincide when there is no nonzero row in any of the matrices A_i, i.e., $m_j = m$ for all j. In comparing the IAAL iteration (2.31)-(2.32), and the ADMM iterations (2.33)-(2.34) and (2.35)-(2.37), we note that they involve fairly similar operations. In particular, the ADMM mutiplier update (2.34) approximates an average (over a full cycle of m components) of the IAAL multiplier updates (2.32), and is executed m times less frequently; this is reminiscent of the difference between the proximal and incremental proximal iterations. The different multiplier update frequencies of IAAL and ADMM suggests that assuming IAAL converges, its stepsize α_k should be chosen much smaller than the stepsize α in ADMM, say

$$\alpha_k \in \left[\frac{\alpha}{m}, \frac{\alpha}{m^2} \right],$$

as a crude approximation, for comparable performance. There are also two other major differences:

(a) The ADMM iterations have guaranteed convergence for any constant stepsize α, and under weaker conditions (differentiability of q_i and strong convexity of Q are not required). On the other hand the IAAL method requires a diminishing stepsize in general, or (under Lipschitz continuity of ∇q_i and strong convexity of Q) a constant stepsize that is not arbitrary, but must be sufficiently small.

(b) In the IAAL method a *single* component y^i is updated at each iteration, while in the ADMM *all* components y^i are updated. For some problems, this may work in favor of IAAL, particularly for large m, a case that generally seems to favor incremental methods.

Thus for the separable problems of this section, one may roughly view the IAAL method as an incremental variant of ADMM, where the advantage of incrementalism may be offset by less solid convergence properties.

A computational comparison of the two methods will be helpful in clarifying their relative merits.

The diagonally scaled ADMM iteration (2.35)-(2.37) suggests also a similar diagonal scaling for the IAAL iteration. The simplest way to accomplish this is to use the IAAL method (2.31)-(2.32) after scaling the constraints, i.e., after multiplying the r constraint equations with different scaling factors, which in turn will introduce diagonal scaling for the dual variables. Proposition 1.1 will still apply under this form of scaling, assuming Lipschitz continuity of ∇q_i and strong convexity of Q.

Comparison with the Methods of Tadjewski [Tad89] and Ruszczynski [Rus95]

The methods of [Tad89] and [Rus95] are motivated by the earlier algorithm of [StW75], and apply to the separable constrained optimization problem of this section. They are similar to each other, but use different assumptions. The method of [Tad89] requires differentiability and second order sufficiency assumptions, but applies to nonconvex separable problems that may have a duality gap, while the method of [Rus95] applies to separable problems with convex, possibly nondifferentiable cost function. These methods are also similar to our IAAL method (2.29)-(2.30), but they use different approximations of the quadratic penalty terms. In particular, instead of the vectors $y^i_{\ell_i}$ that appear in Eqs. (2.29) and (2.30), they use other terms that are iteratively adjusted, with the aim to improve the approximation of the quadratic penalty terms of the standard augmented Lagrangian. Both papers [Tad89] and [Rus95] provide a convergence analysis, involving suitable choices of various parameters, although the convergence results obtained are not as strong as the ones for ADMM. A major difference of the methods of [Tad89] and [Rus95] from our IAAL method is that, like the ADMM, they update all the components y^i simultaneously at each iteration, so they are not incremental in character.

3. PROOF OF PROPOSITION 1.1

Similar to other convergence proofs of incremental gradient methods, including the one of [GOP15] for the IAG method, the proof of Prop. 1.1 is based on viewing the IAP iteration with constant stepsize $\alpha_k \equiv \alpha$,

$$x_{k+1} = x_k - \alpha \left(\nabla f_{i_k}(x_{k+1}) + \sum_{i \neq i_k} \nabla f_i(x_{\ell_i}) \right), \tag{3.1}$$

as a gradient method with errors in the calculation of the gradient [cf. Eqs. (1.15), (1.16)]. To deal with the delays in the iterates, we use the following lemma, due to Feyzmahdavian, Aytekin, and Johansson [FAJ14]:

Lemma 3.1: Let $\{\beta_k\}$ be a nonnegative sequence satisfying

$$\beta_{k+1} \leq p\beta_k + q \max_{\max\{0,k-d\}\leq \ell \leq k} \beta_\ell, \qquad \forall\ k = 0, 1, \ldots,$$

for some positive integer d and nonnegative scalars p and q such that $p + q < 1$. Then we have

$$\beta_k \leq \rho^k \beta_0, \qquad \forall\ k = 0, 1, \ldots,$$

where $\rho = (p + q)^{\frac{1}{1+d}}$.

In the following proof we take the stepsize α as small as is needed for the various calculations to be valid. Also for convenience in expressing various formulas involving delays, we consider the algorithm for large enough iteration indexes, so that all the delayed iteration indexes in the following calculations are larger than 0 (for this it will be sufficient to consider the algorithm as starting at an iteration $k \geq 2b$). Note that the Lipschitz condition on ∇f_i implies a Lipschitz condition and a bound on ∇F. In particular, denoting

$$L = \sum_{i=1}^m L_i,$$

we have for all $x, z \in \Re^n$,

$$\big\|\nabla F(x) - \nabla F(z)\big\| = \left\|\sum_{i=1}^m \nabla f_i(x) - \sum_{i=1}^m \nabla f_i(z)\right\| \leq \sum_{i=1}^m \big\|\nabla f_i(x) - \nabla f_i(z)\big\| \leq \sum_{i=1}^m L_i\|x - z\| = L\|x - z\|. \quad (3.2)$$

As a special case, for $z = x^*$, where x^* is the unique minimum of F, we have

$$\big\|\nabla F(x_\ell)\big\| = \big\|\nabla F(x_\ell) - \nabla F(x^*)\big\| \leq L\|x_\ell - x^*\|, \qquad \forall\ \ell \geq 0. \quad (3.3)$$

We break down the proof of Prop. 1.1 in steps, first writing the iteration (3.1) as a gradient iteration with errors, then carrying along the errors in a standard line of linear convergence analysis of gradient methods without errors, then bounding the errors, and finally using Lemma 3.1:

(a) We write the iteration (3.1) as a gradient method with errors

$$x_{k+1} = x_k - \alpha\big(\nabla F(x_k) + e_k\big), \quad (3.4)$$

where the error term e_k is given by

$$e_k = \nabla f_{i_k}(x_{k+1}) - \nabla f_{i_k}(x_k) + \sum_{i \neq i_k} \big(\nabla f_i(x_{\ell_i}) - \nabla f_i(x_k)\big). \quad (3.5)$$

(b) We relate the gradient error e_k to the distance $\|x_k - x^*\|$ by verifying the relation

$$\|x_{k+1} - x^*\|^2 = \|x_k - x^*\|^2 - 2\alpha\nabla F(x_k)'(x_k - x^*) + \alpha^2\big\|\nabla F(x_k)\big\|^2 + E_k, \tag{3.6}$$

where

$$E_k = \alpha^2\|e_k\|^2 - 2\alpha\big(x_k - \alpha\nabla F(x_k) - x^*\big)'e_k. \tag{3.7}$$

This is done by subtracting x^* from both sides of Eq. (3.4), norm-squaring both sides, and carrying out the straightforward calculation.

(c) We use Eq. (3.7) to bound $|E_k|$ according to

$$|E_k| \le \alpha^2\|e_k\|^2 + 2\alpha\|e_k\|\,\big\|x_k - x^*\big\|, \tag{3.8}$$

for all sufficiently small α. In particular, from Eq. (3.7), we have

$$|E_k| \le \alpha^2\|e_k\|^2 + 2\alpha\|e_k\|\,\big\|x_k - x^* - \alpha\nabla F(x_k)\big\|,$$

and Eq. (3.8) is obtained from the preceding relation by using the inequality

$$\big\|x_k - x^* - \alpha\nabla F(x_k)\big\| \le \|x_k - x^*\|.$$

which holds for α sufficiently small; this is a consequence of the fact that under the gradient Lipschitz assumption, a gradient iteration (with no error) reduces the distance to x^* for $\alpha \in (0, 1/L]$ (see e.g., [Ber15], Prop. 6.1.6).

(d) We use the strong convexity assumption

$$\big(\nabla F(x) - \nabla F(y)\big)'(x - y) \ge \sigma\|x - y\|^2, \qquad \forall\, x, y \in \Re^n, \tag{3.9}$$

where σ is the coefficient of strong convexity and the Lipschitz condition (3.2), to invoke the relation

$$\nabla F(x_k)'(x_k - x^*) \ge \frac{\sigma L}{\sigma + L}\|x_k - x^*\|^2 + \frac{1}{\sigma + L}\big\|\nabla F(x_k)\big\|^2; \tag{3.10}$$

see e.g., [Nes14], Th. 2.1.22, or [Ber15], Prop. 6.1.9(b). This will be used to bound the term $\nabla F(x_k)'(x_k - x^*)$ of Eq. (3.6).

(e) We show that for $\alpha \le \frac{2}{\sigma + L}$, we have

$$\|x_{k+1} - x^*\|^2 \le \left(1 - 2\alpha\frac{\sigma L}{\sigma + L}\right)\|x_k - x^*\|^2 + |E_k|. \tag{3.11}$$

In particular, using the relations (3.6) and (3.10), we have

$$\begin{aligned}\|x_{k+1} - x^*\|^2 &\le \|x_k - x^*\|^2 - 2\alpha\left(\frac{\sigma L}{\sigma + L}\|x_k - x^*\|^2 + \frac{1}{\sigma + L}\big\|\nabla F(x_k)\big\|^2\right) + \alpha^2\big\|\nabla F(x_k)\big\|^2 + |E_k| \\ &\le \left(1 - 2\alpha\frac{\sigma L}{\sigma + L}\right)\|x_k - x^*\|^2 + \alpha\left(\alpha - \frac{2}{\sigma + L}\right)\big\|\nabla F(x_k)\big\|^2 + |E_k|,\end{aligned}$$

from which Eq. (3.11) follows.

(f) We prove that the error e_k is proportional to the stepsize α, and to the maximum distance of the iterates from x^* over the past $2b$ iterates:

$$\|e_k\| \leq O(\alpha) \max_{k-2b\leq \ell \leq k} \|x_\ell - x^*\|. \tag{3.12}$$

This is straightforward, using the Lipschitz assumption on ∇f_i and the bound (3.3) on ∇F.

In particular, from Eq. (3.5), we have

$$\begin{aligned} \|e_k\| &\leq \big\|\nabla f_{i_k}(x_{k+1}) - \nabla f_{i_k}(x_k)\big\| + \sum_{i\neq i_k} \big\|\nabla f_i(x_{\ell_i}) - \nabla f_i(x_k)\big\| \\ &\leq L_{i_k}\|x_{k+1} - x_k\| + \sum_{i\neq i_k} L_i \|x_k - x_{\ell_i}\| \\ &\leq L_{i_k}\|x_{k+1} - x_k\| + \sum_{i\neq i_k} L_i \big(\|x_k - x_{k-1}\| + \cdots + \|x_{\ell_i+1} - x_{\ell_i}\|\big). \end{aligned} \tag{3.13}$$

Moreover from Eqs. (3.3) and (3.4),

$$\|x_{\ell+1} - x_\ell\| = \alpha\big\|\nabla F(x_\ell)\big\| + \alpha\|e_\ell\| \leq \alpha L\|x_\ell - x^*\| + \alpha\|e_\ell\|, \qquad \forall\ \ell \geq 0. \tag{3.14}$$

Using this relation for ℓ in the range $[k-b, k]$ in Eq. (3.13), we obtain

$$(1-\alpha L_{i_k})\|e_k\| \leq O(\alpha)\left(\sum_{\ell=k-b}^{k} \|x_\ell - x^*\| + \sum_{\ell=k-b}^{k-1} \|e_\ell\|\right),$$

where for $p \geq 1$, we generically use $O(\alpha^p)$ to denote any function of α such that for some scalar $\gamma > 0$, we have $\big|O(\alpha^p)\big| \leq \gamma\alpha^p$ for all α in some bounded open interval containing the origin. Thus,

$$\|e_k\| \leq O(\alpha)\left(\sum_{\ell=k-b}^{k} \|x_\ell - x^*\| + \sum_{\ell=k-b}^{k-1} \|e_\ell\|\right). \tag{3.15}$$

From Eq. (3.5), we also have

$$\begin{aligned} \|e_\ell\| &\leq L_{i_\ell}\|x_{\ell+1} - x_\ell\| + \sum_{i\neq i_\ell} L_i\|x_\ell - x_{\ell_i}\| \\ &\leq L\left(\|x_{\ell+1} - x^*\| + \|x_\ell - x^*\| + \sum_{i\neq i_\ell} L_i\big(\|x_\ell - x^*\| + \|x_{\ell_i} - x^*\|\big)\right). \end{aligned} \tag{3.16}$$

Since for ℓ in the range $[k-b, k-1]$, ℓ_i lies in the range $[k-2b, k-1]$, it follows that

$$\|e_\ell\| \leq c \max_{k-2b\leq \ell \leq k} \|x_\ell - x^*\|, \qquad \forall\ \ell \in [k-b, k-1],$$

where c is some constant that is independent of k and ℓ. Combining this with Eq. (3.15), we obtain Eq. (3.12).

(g) We use Eqs. (3.8), (3.11), and (3.12) to obtain

$$\|x_{k+1} - x^*\|^2 \leq \left(1 - 2\alpha \frac{\sigma L}{\sigma + L}\right) \|x_k - x^*\|^2 + O(\alpha^2) \max_{k-2b \leq \ell \leq k} \|x_\ell - x^*\|^2. \tag{3.17}$$

In particular, the two terms bounding $|E_k|$ in Eq. (3.8) are $\alpha^2 \|e_k\|^2$ and $\alpha \|e_k\| \, \big\|x_k - x^*\big\|$, which in view of Eq. (3.12) are bounded by terms that are $O(\alpha^4)$ and $O(\alpha^2)$ times $\max_{k-2b \leq \ell \leq k} \|x_\ell - x^*\|^2$, respectively.

(h) We use Eq. (3.17) and Lemma 3.1, with $d = 2b$, $\beta_k = \|x_k - x^*\|^2$, $p = 1 - 2\alpha \frac{\sigma L}{\sigma + L}$, and $q = O(\alpha^2)$, so that $p + q < 1$ for sufficiently small α. This shows that $\sqrt{\beta_k} = \|x_k - x^*\|$ converges linearly to 0, and completes the proof. **Q.E.D.**

Convergence Rate Comparison for Small Stepsizes

Note that Eq. (3.17) provides a more refined rate of convergence estimate. While this estimate is not very precise, because of the second order term on the right in Eq. (3.17), it shows that the ratio

$$\frac{\sigma L}{\sigma + L} = \frac{L}{1 + L/\sigma}$$

where $L = \sum_{i=1}^m L_i$ and σ is the coefficient of strong convexity, plays an important role, and in particular the convergence rate is improved when the “condition number” L/σ is small. The role of the ratio L/σ in determining the convergence rate of gradient methods (without error) is well-known; see e.g., [Nes04], [Ber15].

Convergence rate estimates like the one of Eq. (3.17) can also be similarly derived for IAG, and for the standard nonincremental gradient method [for which the error term $|E_k|$ in Eq. (3.11) is equal to 0]. These estimates, to first order [i.e., after neglecting the second order term in the right-hand side of Eq. (3.17)], are identical for IAP, IAG, and for the standard nonincremental gradient method. This suggests that for very small values of α, IAP and IAG perform comparably, while the nonincremental gradient method performs much worse because it requires m times as much overhead per iteration to calculate the full gradient of the cost function.

4. NONQUADRATIC INCREMENTAL PROXIMAL AND AUGMENTED LAGRANGIAN METHODS

The augmented Lagrangian methods of Section 2 apply to linear equality constrained problems for which the multiplier vector λ is unconstrained. This allows the application of the linear convergence result of Prop. 1.1. We will now consider convex inequality constraints, whose multipliers must be nonnegative. As a result the dual problem involves an orthant constraint, and the linear convergence result of Prop. 1.1 does not apply.

Unfortunately, when there is an orthant constraint [i.e., $X = \{x \mid x \geq 0\}$ instead of $X = \Re^n$ in Eq. (1.9)], the proof of Prop. 1.1 breaks down because the critical inequality (3.3) fails. In fact, to our knowledge, a linear convergence rate result for the IAG method (1.8) applied with an orthant constraint is not currently available. Moreover, the convergence of the augmented Lagrangian-like methods discussed in Section 2 has been analyzed only for the equality-constrained case. In this section we will try to address this difficulty by using a different (nonquadratic) proximal approach.

In particular, we will introduce incremental augmented Lagrangian methods for convex inequality constraints, where the quadratic penalty in the augmented Lagrangian is replaced by a suitable nonquadratic penalty. One of our objectives is to develop linearly convergent methods that can exploit separability, similar to the ones of Section 2. A second objective is to develop corresponding dual linearly convergent incremental aggregated gradient and proximal methods for differentiable minimization subject to nonnegativity constraints.

Nonquadratic Augmented Lagrangian Methods for Inequality Constraints

Consider the convex programming problem

$$\begin{aligned} &\text{minimize} \quad H(y) \\ &\text{subject to} \ \ y \in Y, \qquad G_j(y) \leq 0, \quad j = 1, \ldots, r, \end{aligned} \tag{4.1}$$

where $H : \Re^n \mapsto (-\infty, \infty)$ and $G_j : \Re^n \mapsto (-\infty, \infty)$ are convex functiona, and Y is a convex set. The corresponding dual problem is

$$\begin{aligned} &\text{maximize} \quad Q(\mu) \\ &\text{subject to} \ \ \mu \geq 0, \end{aligned} \tag{4.2}$$

where $Q : \Re^r \mapsto [-\infty, \infty)$ is the concave function of the multiplier vector $\mu = (\mu^1, \ldots, \mu^r)$, given by

$$Q(\mu) = \inf_{y \in Y} \left\{ H(y) + \sum_{j=1}^{r} \mu^j G_j(y) \right\}, \qquad \mu \in \Re^r. \tag{4.3}$$

We will apply an augmented Lagrangian method, first proposed by Kort and Bertsekas [KoB72], and further developed in a number of subsequent works, including the monograph [Ber82] (Chapter 5). The method makes use of a nonquadratic penalty function $\psi : \Re \mapsto \Re$ with the following properties:

(i) ψ is twice differentiable and $\nabla^2 \psi(t) > 0$ for all $t \in \Re$,

(ii) $\psi(0) = 0$, $\nabla \psi(0) = 1$,

(iii) $\lim_{t \to -\infty} \psi(t) > -\infty$,

(iv) $\lim_{t \to -\infty} \nabla \psi(t) = 0$ and $\lim_{t \to \infty} \nabla \psi(t) = \infty$.

The most common and interesting special case is the exponential

$$\psi(s) = \exp(s) - 1, \qquad s \in \Re. \tag{4.4}$$

The corresponding *exponential augmented Lagrangian method* and its dual, a proximal algorithm known as the *entropy minimization algorithm*, has been analyzed first in [KoB72] and [Ber82], and then by Tseng and Bertsekas [TsB93]. Related classes of methods, which also contain the exponential and entropy methods as special cases, were proposed and analyzed later by Iusem, Svaiter, and Teboulle [IST94]; see also the survey by Iusem [Ius99], which contains followup work and many references.

The augmented Lagrangian algorithm corresponding to ψ and problem (4.2) maintains multipliers $\mu_k^j > 0$, $j = 1, \ldots, r$, for the inequality constraints, and consists of finding

$$y_{k+1} \in \arg\min_{y \in Y} \left\{ H(y) + \sum_{j=1}^{r} \frac{\mu_k^j}{\alpha_k^j} \psi\big(\alpha_k^j G_j(y)\big) \right\}, \tag{4.5}$$

where $\alpha_k^j > 0$, $j = 1, \ldots, r$, are penalty parameters, followed by the multiplier iteration

$$\mu_{k+1}^j = \mu_k^j \nabla\psi\big(a_k^j G_j(y_{k+1})\big), \qquad j = 1, \ldots, r. \tag{4.6}$$

Alternatively and equivalently, based on the Fenchel duality theorem, one may show that the multiplier iteration can be written in the proximal form

$$\mu_{k+1} \in \arg\max_{\mu \in \Re^r} \left\{ Q(\mu) - \sum_{j=1}^{r} \frac{\mu_k^j}{\alpha_k^j} \psi^\star \left(\frac{\mu^j}{\mu_k^j} \right) \right\}, \tag{4.7}$$

where Q is the dual function given by Eq. (4.3), and $\psi^\star$ is the convex conjugate of ψ.

To see the equivalence of the expressions (4.6) and (4.7), let us write

$$u_{k+1}^j = G_j(y_{k+1}), \qquad j = 1, \ldots, r,$$

and note that the augmented Lagrangian minimization (4.5) yields

$$u_{k+1} \in \arg \min_{u=(u^1,\ldots,u^r) \in \Re^r} \left\{ P(u) + \sum_{j=1}^{r} \frac{\mu_k^j}{\alpha_k^j} \psi(\alpha_k^j u^j) \right\}, \tag{4.8}$$

where P is the primal function

$$P(u) = \inf_{y \in Y,\ G_j(y) \le u^j,\, j=1,\ldots,r} H(y).$$

Then the minimization in Eq. (4.8) is the Fenchel dual to the maximization (4.7). By applying the Fenchel duality theorem, we have that the maximizing vector in Eq. (4.7) is equal to the gradient

$$\nabla \left(\sum_{j=1}^{r} \frac{\mu_k^j}{\alpha_k^j} \psi(\alpha_k^j u^j) \right) \Bigg|_{u=u_{k+1}},$$

so it is given by the formula (4.6).

Note that while the dual problem is to maximize $Q(\mu)$ subject to $\mu \geq 0$, the proximal maximization (4.7) is unconstrained. The reason is that the conjugate $\psi^\star$ takes the value ∞ outside the nonnegative orthant, and has the character of a barrier function within the nonnegative orthant. As an example, for the exponential function (4.4) the conjugate is the entropy function

$$\psi^\star(t) = \begin{cases} t\big(\ln(t) - 1\big) + 1 & \text{if } t > 0, \\ 1 & \text{if } t = 0, \\ \infty & \text{if } t < 0. \end{cases} \tag{4.9}$$

An important advantage of the nonquadratic augmented Lagrangian method versus its quadratic counterpart, is that it leads to twice differentiable augmented Lagrangians. This advantage also carries over to the incremental augmented Lagrangian methods to be presented next.

Nonquadratic Incremental Augmented Lagrangian Methods for Inequality Constraints

Consider now the separable constrained optimization problem

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^m h_i(y^i) \\ &\text{subject to} \quad y^i \in Y_i, \quad i = 1, \ldots, m, \qquad \sum_{i=1}^m g_{ji}(y^i) \leq 0, \end{aligned} \tag{4.10}$$

where h_i and g_{ji} are convex real-valued functions, and Y_i are convex sets. Similar to the development of Section 2, the corresponding incremental aggregated augmented Lagrangian method, which parallels IAAL, maintains a vector $\mu_k > 0$ and operates as follows.

Incremental Aggregated Augmented Lagrangian Iteration for Inequalities (IAALI)

Select a component index i_k, and update the single component y^{i_k} according to

$$y_{k+1}^{i_k} \in \arg \min_{y^{i_k} \in Y_{i_k}} \left\{ h_{i_k}(y^{i_k}) + \sum_{j=1}^r \frac{\mu_k^j}{\alpha_k^j} \psi \left(\alpha_k^j \left(g_{ji_k}(y^{i_k}) + \sum_{i \neq i_k} g_{ji}(y_{\ell_i}^i) \right) \right) \right\}, \tag{4.11}$$

while keeping the others unchanged, $y_{k+1}^i = y_k^i$ for all $i \neq i_k$. Then update μ according to

$$\mu_{k+1}^j = \mu_k^j \nabla \psi \left(a_k^j \left(g_{ji_k}(y_{k+1}^{i_k}) + \sum_{i \neq i_k}^m g_{ji}(y_{\ell_i}^i) \right) \right), \quad j = 1, \ldots, r. \tag{4.12}$$

Note that the minimization (4.11) is of low dimension, but involves the nonquadratic penalty function ψ. Thus even when the component y^{i_k} is one-dimensional, this minimization will likely require some form of iterative line search. Note also that the update formula (4.12) can equivalently be written as

$$\mu_{k+1} \in \arg\max_{\mu\in\Re^r} \left\{ q_{i_k}(\mu) + \sum_{i\neq i_k} \nabla q_i(\mu_{\ell_i})'(\mu-\mu_k) - \sum_{j=1}^{r} \frac{\mu_k^j}{\alpha_k^j}\psi^\star\left(\frac{\mu^j}{\mu_k^j}\right)\right\}, \tag{4.13}$$

where q_i are the dual function components, given by

$$q_i(\mu) = \inf_{y^i\in Y^i}\left\{ h_i(y^i) + \sum_{j=1}^{r} \mu^j g_{ij}(y^i)\right\}, \qquad \mu\in\Re^r, \qquad i=1,\ldots,m.$$

The form (4.13) of the method can be viewed as an incremental aggregated proximal method for maximizing $Q(\mu)=\sum_{i=1}^m q_i(\mu)$ over $\mu\geq 0$, where

$$q_i(\mu) = \inf_{y^i\in Y_i}\left\{ h_i(y^i) + \sum_{j=1}^{r} \mu^j g_{ji}(y^i)\right\}, \qquad i=1,\ldots,m; \tag{4.14}$$

cf. Eq. (2.3). The convergence properties of the IAALI and the corresponding incremental aggregated proximal method (4.13) for solving the dual problem

$$\begin{aligned} &\text{maximize} \quad \sum_{i=1}^{m} q_i(\mu) \\ &\text{subject to} \quad \mu\geq 0, \end{aligned}$$

are interesting research subjects, as we will now discuss.

Nonquadratic Incremental Aggregated Proximal Algorithm for Nonnegativity Constraints

Consider the minimization problem

$$\begin{aligned} &\text{minimize} \quad F(x) \overset{\text{def}}{=} \sum_{i=1}^{m} f_i(x) \\ &\text{subject to} \quad x\geq 0, \end{aligned} \tag{4.15}$$

where $f_i:\Re^n\mapsto\Re$, $i=1,\ldots,m$, are convex real-valued functions. When translated to this minimization context, the algorithm (4.13) maintains a vector $x_k>0$ that is updated as follows.

Nonquadratic Incremental Aggregated Proximal Iteration for $X=\{x\mid x\geq 0\}$

Select a component index i_k, and obtain x_{k+1} as

$$x_{k+1} \in \arg\min_{x\in\Re^n} \left\{ f_{i_k}(x) + \sum_{i\neq i_k} \nabla f_i(x_{\ell_i})'(x-x_k) + \sum_{j=1}^{n} \frac{x_k^j}{\alpha_k^j}\psi^\star\left(\frac{x^j}{x_k^j}\right)\right\}. \tag{4.16}$$

The analysis of the convergence properties of this algorithm is beyond the scope of this paper, and will be the subject of a separate publication. In particular, it is interesting to investigate the linear convergence of the method (4.16) when the parameters a_k^j are constant (but sufficiently small), under the appropriate Lipschitz continuity and strong convexity assumptions, similar to Prop. 1.1. Note that by differentiating the cost function in the minimization of Eq. (4.16), we obtain the optimality condition, which can be written as

$$\nabla f_{i_k}(x_{k+1}) + \sum_{i \neq i_k} \nabla f_i(x_{\ell_i}) + \begin{pmatrix} \frac{1}{\alpha_k^1} \nabla \psi^\star \left(\frac{x_{k+1}^1}{x_k^1} \right) \\ \vdots \\ \frac{1}{\alpha_k^m} \nabla \psi^\star \left(\frac{x_{k+1}^m}{x_k^m} \right) \end{pmatrix} = 0. \tag{4.17}$$

This expression may be used in the line of proof of Section 3 in place of the corresponding formula (1.14) for the unconstrained IAP algorithm (1.14), which can be written in the form

$$\nabla f_{i_k}(x_{k+1}) + \sum_{i \neq i_k} \nabla f_i(x_{\ell_i}) + \frac{x_{k+1} - x_k}{\alpha} = 0. \tag{4.18}$$

When ψ (and hence also $\psi^\star$) is quadratic and $\alpha_k^j \equiv \alpha$, the two preceding formulas coincide. However, contrary to iteration (4.18), the iteration (4.17) preserves the strict positivity of the iterates ($x_k > 0$ for all k), and addresses the orthant-constrained problem (4.15).

Entropy-Based Incremental Aggregated Proximal Algorithm for Nonnegativity Constraints

For an illustration of the algorithm (4.16), consider the special case where ψ is the exponential function and $\psi^\star$ is the entropy function, so that

$$\psi(s) = \exp(s) - 1, \qquad \psi^\star(t) = \begin{cases} t\big(\ln(t) - 1\big) + 1 & \text{if } t > 0, \\ 1 & \text{if } t = 0, \\ \infty & \text{if } t < 0, \end{cases} \qquad \nabla \psi^\star(t) = \begin{cases} \ln(t) & \text{if } t > 0, \\ \text{does not exist} & \text{if } t \leq 0, \end{cases}$$

[cf. Eqs. (4.4) and (4.9)]. Then by using a constant stepsize α^j for each coordinate, Eq. (4.17) takes the form

$$\ln\left(\frac{x_{k+1}^j}{x_k^j} \right) = -\alpha^j \left(\frac{\partial f_{i_k}(x_{k+1})}{\partial x^j} + \sum_{i \neq i_k} \frac{\partial f_i(x_{\ell_i})}{\partial x^j} \right), \qquad j = 1, \ldots, n, \tag{4.19}$$

where i_k is the component index selected for iteration k. We can write this iteration as

$$\ln\left(\frac{x_{k+1}^j}{x_k^j} \right) = -\alpha^j \left(\frac{\partial F(x_k)}{\partial x^j} + e_k^j \right), \tag{4.20}$$

where $e_k = (e_k^1, \ldots, e_k^n)$ is the error vector

$$e_k = \nabla f_{i_k}(x_{k+1}) - \nabla f_{i_k}(x_k) + \sum_{i \neq i_k} \big(\nabla f_i(x_{\ell_i}) - \nabla f_i(x_k) \big), \tag{4.21}$$

that played an important role in the proof of Prop. 1.1 [cf. Eq. (3.5)].

We will use the line of analysis of Section 3 to speculate about the linear convergence of iteration (4.19) and its equivalent form (4.20)-(4.21). Assume that the minimum x^* satisfies the strict complementary slackness condition

$$\frac{\partial F(x^*)}{\partial x^j} > 0, \qquad \forall\ j \in J^0, \tag{4.22}$$

where $J^0 = \big\{j \mid (x^j)^* = 0\big\}$, and speculate on the behavior of $\{x_k\}$ in a small neighborhood around x^*.

Consider first the iterates x_k^j, $j \in J^0$, in a small neighborhood around x^*. We note that the errors e_k^j of Eq. (4.21) are near 0 and by Eq. (4.22), are negligible relative to the gradient components $\frac{\partial F(x_k)}{\partial x^j}$, for all $j \in J^0$. In view of the form of iteration (4.20) and the condition (4.22), the logarithms $\ln(x_{k+1}^j/x_k^j)$, $j \in J^0$, are negative, and hence the ratios x_{k+1}^j/x_k^j, $j \in J^0$, are within $[0, 1)$, so the sequences $\{x_k^j\}$, $j \in J^0$, are linearly decreasing towards 0.

Consider next the iterates x_k^j, $j \notin J^0$, in a small neighborhood around x^*. They are close to the corresponding positive numbers $(x^j)^*$, $j \notin J^0$, and they are iterated according to

$$\ln(x_{k+1}^j) = \ln(x_k^j) - \alpha^j \left(\frac{\partial F(x_k)}{\partial x^j} + e_k^j\right), \qquad j \notin J^0, \tag{4.23}$$

[cf. Eq. (4.20)]. This looks like an incremental aggregated gradient iteration in the logarithms $\ln(x^j)$, $j \notin J^0$. Indeed by making the transformation of variables $z^j = \ln(x^j)$, $j = 1, \ldots, n$, for $x^j > 0$, and introducing the function

$$H(z^1, \ldots, z^n) = F\big(\exp(z^1), \ldots, \exp(z^n)\big),$$

and its gradient, which is related to the gradient of F through the relation

$$\frac{\partial H(z)}{\partial z^j} = \exp(z^j)\frac{\partial F\big(\exp(z^1), \ldots, \exp(z^n)\big)}{\partial x^j} = x^j \frac{\partial F(x)}{\partial x^j}, \qquad j = 1, \ldots, n,$$

we see that the iteration (4.23) can be written as

$$z_{k+1}^j = z_k^j - \frac{\alpha^j}{x_k^j}\left(\frac{\partial H(z_k)}{\partial z^j}\right) + \alpha^j e_k^j, \qquad j \notin J^0,$$

where $x_k^j = \exp(z_k^j)$. Thus, neglecting the effect of the coordinates x^j, $j \in J^0$, that are fast diminishing to 0, the iteration behaves like the IAP method restricted to the space of the coordinate logarithms $z^j = \ln(x^j)$, $j \notin J^0$, with coordinate-dependent stepsizes $\frac{\alpha^j}{x_k^j}$ that are close to the positive constants $\frac{\alpha^j}{(x^j)^*}$, $j \in J^0$, for x_k near x^*.

By combining the preceding argument with the proof of Prop. 1.1, we can show that the method converges to x^* locally, i.e., when started sufficiently close to x^*, assuming the strict complementarity condition (4.22), and the appropriate stepsize, Lipschitz continuity, and strong convexity conditions. The proof is long and will be deferred to a future publication. Moreover, for $j \notin J^0$, $\big\{\ln(x_k^j)\big\}$ converges to

$\ln\big((x^j)^*\big)$ linearly, while for $j \in J^0$, $\{x_k^j\}$ also converges to $(x^j)^*$ linearly. However, a more sophisticated argument is needed to show global and linear convergence of $\{x_k\}$ to x^*, by combining the line of proof of Prop. 1.1 with the existing convergence proofs of the entropy minimization algorithm and its dual, the exponential method of multipliers.

Entropy-Based Incremental Aggregated Gradient Algorithm for Nonnegativity Constraints

Finally let us note the analog of the IAG method for nonnegativity constraints. In analogy with Eq. (4.19) it has the form

$$\ln\left(\frac{x_{k+1}^j}{x_k^j}\right) = -\alpha^j \sum_{i=1}^m \frac{\partial f_i(x_{\ell_i})}{\partial x^j}, \qquad j = 1, \ldots, n,$$

or equivalently

$$x_{k+1}^j = x_k^j \exp\left(-\alpha^j \sum_{i=1}^m \frac{\partial f_i(x_{\ell_i})}{\partial x^j}\right), \qquad j = 1, \ldots, n, \tag{4.24}$$

[the difference from Eq. (4.19) is the use of $\frac{\partial f_{i_k}(x_{\ell_{i_k}})}{\partial x^j}$ in place of $\frac{\partial f_{i_k}(x_{k+1})}{\partial x^j}$]. This iteration should be compared with the IAS method (1.5), for the case where the functions f_i are differentiable, and the stepise α_k is a constant α:

$$x_{k+1} = \left[x_k - \alpha \sum_{i=1}^m \nabla f_i(x_{\ell_i})\right]^+, \tag{4.25}$$

where $[\cdot]^+$ denotes projection onto the nonnegative orthant. We may view the method (4.25) as the constrained version of the IAG method (1.8) with constant stepsize for which, however, no linear convergence proof is presently available.†

The iteration (4.24) may also be viewed as an incremental version of the mirror descent method; see Beck and Teboulle [BeT03], the surveys by Juditsky and Nemirovski [JuN11a], [JunN11b], and the references quoted there, and the author's presentation in [Ber15], Section 6.6. Using similar arguments to the case of iteration (4.19), we can show that the iteration (4.24) converges linearly to x^*, when started sufficiently close to x^*, assuming the strict complementarity condition (4.22), and the appropriate constant stepsize, and other conditions. Note that the iteration (4.24) may be implemented more conveniently than the proximal iteration (4.16), as it does not require a proximal minimization. However, the iteration (4.24) is not suitable as the basis for the development of an incremental augmented Lagrangian method, such as IAALI [cf. Eqs. (4.11)-(4.12)].

† A local linear convergence result for the constrained IAG method (4.25) is possible, assuming the strict complementarity condition (4.22). In particular, it can be shown that there is a sphere centered at x^* such that if x_0 belongs to that sphere, then the sequence generated by iteration (4.25) stays within that sphere and converges linearly to x^*. The idea of the proof is that after the first iteration, all the iterates satisfy $x_k^j = 0$ for all indices $j \in J^0$, so the method essentially reduces to the IAG method in the space of variables x^j, $j \notin J^0$.

A final comment relates to the choice of the stepsizes α^j in iteration (4.24). For the coordinates that are bounded away from 0 (i.e., for $j \notin J^0$) we have asymptotically $\sum_{i=1}^m \frac{\partial f_i(x_{\ell_i})}{\partial x^j} \approx 0$, so from a Taylor expansion of the exponential in Eq. (4.24), we obtain

$$x_{k+1}^j = x_k^j \left(1 + \left(-\alpha^j \sum_{i=1}^m \frac{\partial f_i(x_{\ell_i})}{\partial x^j} \right) + \frac{1}{2} \left(-\alpha^j \sum_{i=1}^m \frac{\partial f_i(x_{\ell_i})}{\partial x^j} \right)^2 + \cdots \right).$$

By discarding the second and higher order terms for $j \notin J^0$, we see that approximately,

$$x_{k+1}^j \approx x_k^j - \alpha^j x_k^j \sum_{i=1}^m \frac{\partial f_i(x_{\ell_i})}{\partial x^j}, \qquad j \notin J^0.$$

This suggests scaling the stepsizes α^j for $j \notin J^0$, so that α^j is inversely proportional to the optimal value $(x^j)^*$. On the other hand, for $j \in J^0$, it makes sense to choose α^j large (subject to a positive lower bound) in order to accelerate the convergence of x_k^j to $(x^j)^* = 0$. Thus a reasonable heuristic is to set

$$\alpha^j = \frac{\alpha}{\max\{\bar{x}^j,\, \delta\}}, \qquad j = 1, \ldots, n,$$

where $\bar{x}^j$ is an estimate for the optimal coordinate value $(x^j)^*$, α is some positive scalar, which corresponds to the stepsize of the constrained IAG iteration (4.25), and δ is a small positive constant. One may also consider updating the values α^j in the course of the algorithm, as better estimates $\bar{x}^j$ are obtained.

5. CONCLUDING REMARKS

In this paper we have proposed IAP, an incremental aggregated proximal method, and we have shown that under favorable assumptions, it attains a linear convergence rate, using a constant (but sufficiently small) stepsize. The application of this method in a dual context, to separable constrained optimization problems, yields the IAAL method, an incremental augmented Lagrangian method that preserves and exploits the separable structure. The principal difference of our method relative to the several alternative augmented Lagrangian-based proposals, is its incremental character and its high update frequency of the multiplier λ_k; the alternative methods, except Algorithm 1 of [WHM13] and the one of [RoT15], but including the proper version of ADMM for separable problems, update all the primal variables y^i, $i = 1, \ldots, m$, simultaneously rather than sequentially, so they are not incremental in nature. Moreover, the alternative methods update the multipliers m times less frequently than IAAL. A systematic computational comparison of our methods with the nonincremental alternatives will be helpful in clarifying what advantages our incremental approach may hold.

There are several analytical issues relating to the IAAL method, which require further investigation. For example a more refined convergence rate analysis may point the way to adaptive stepsize adjustment schemes, and/or forms of scaling based on second derivatives of the cost function and the matrices A_i. There are analyses of this type for ADMM; see the paper by Giselsson and Boyd [GiB15], and the references cited there. Another possibility is to use a momentum term in the updating formula for the multiplier λ. A third possibility is to control the degree of incrementalism by "batching" multiple augmented Lagrangian iterations involving multiple components.

We have also proposed linearly converging extensions of IAAL for problems with convex inequality constraints. These are based on a nonquadratic augmented Lagrangian approach such as the exponential, and its dual version, which is an incremental aggregated entropy algorithm (4.19). The fuller investigation of this method, as well as the method (4.24), which is the exponential analog of the IAG method for nonnegativity constraints, are important research subjects.

6. REFERENCES

[AFB06] Ahn, S., Fessler, J., Blatt, D., and Hero, A. O., 2006. "Convergent Incremental Optimization Transfer Algorithms: Application to Tomography," IEEE Transactions on Medical Imaging, Vol. 25, pp. 283-296.

[BLY15] Bragin, M. A., Luh, P. B., Yan, J. H., Yu, N., and Stern, G. A., 2015. "Convergence of the Surrogate Lagrangian Relaxation Method," J. of Optimization Theory and Applications, Vol. 164, pp. 173-201.

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

[BPC11] Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J., 2011. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers, Now Publishers Inc, Boston, MA.

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

[BeT03] Beck, A., and Teboulle, M., 2003. "Mirror Descent and Nonlinear Projected Subgradient Methods for Convex Optimization," Operations Research Letters, Vol. 31, pp. 167-175.

[Ber78] Bertsekas, D. P., 1978. "Local Convex Conjugacy and Fenchel Duality," Preprints of 7th Triennial World Congress of IFAC, Helsinki, Finland, Vol. 2, pp. 1079-1084.

[Ber79] Bertsekas, D. P., 1979. "Convexification Procedures and Decomposition Methods for Nonconvex Optimization Problems," J. of Optimization Theory and Applications, Vol. 29, pp. 169-197.

[Ber82] Bertsekas, D. P., 1982. Constrained Optimization and Lagrange Multiplier Methods, Academic Press, NY; republished in 1996 by Athena Scientific, Belmont, MA. On line at http://web.mit.edu/dimitrib/www/lagrmult.html.

[Ber09] Bertsekas, D. P., 2009. Convex Optimization Theory, Athena Scientific, Belmont, MA.

[Ber10] Bertsekas, D. P., 2010. "Incremental Gradient, Subgradient, and Proximal Methods for Convex Optimization: A Survey," Lab. for Information and Decision Systems Report LIDS-P-2848, MIT; arXiv:1507.01030.

[Ber11] Bertsekas, D. P., 2011. "Incremental Proximal Methods for Large Scale Convex Optimization," Math. Programming, Vol. 129, pp. 163-195.

[Ber15] Bertsekas, D. P., 2015. Convex Optimization Algorithms, Athena Scientific, Belmont, MA.

[CHH14] Chen, C., He, B., Ye, Y., and Yuan, X., 2014. “The Direct Extension of ADMM for Multi-Block Convex Minimization Problems is not Necessarily Convergent,” Mathematical Programming, published on line.

[ChT94] Chen, G., and Teboulle, M., 1994. “A Proximal-Based Decomposition Method for Convex Minimization Problems,” Mathematical Programming, Vol. 64, pp. 81-101.

[DLP14] Deng, W., Lai, M. J., Peng, Z., and Yin, W., 2014. “Parallel Multi-Block ADMM with O (1/k) Convergence,” arXiv preprint arXiv:1312.3040v2.

[DaL15] Dang, C., and Lan, G., (2015). “Randomized First-order Methods for Saddle Point Optimization,” arXiv preprint arXiv:1409.8625v3.

[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.

[Eck12] Eckstein, J., 2012. “Augmented Lagrangian and Alternating Direction Methods for Convex Optimization: A Tutorial and Some Illustrative Computational Results,” RUTCOR Research Report RRR 32-2012, Rutgers, Univ.

[Eve63] Everett, H., 1963. “Generalized Lagrange Multiplier Method for Solving Problems of Optimal Allocation of Resources,” Operations Research, Vol. 11, pp. 399-417.

[FAJ14] Feyzmahdavian, H. R., Aytekin, A., and Johansson, M., 2014. “A Delayed Proximal Gradient Method with Linear Convergence Rate,” in Prop. of 2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP), pp. 1-6.

[GOP15] Gurbuzbalaban, M., Ozdaglar, A., and Parrilo, P., 2015. “On the Convergence Rate of Incremental Aggregated Gradient Algorithms,” arXiv preprint arXiv:1506.02081.

[GaM76] Gabay, D., and Mercier, B., 1976. “A Dual Algorithm for the Solution of Nonlinear Variational Problems via Finite-Element Approximations,” Comp. Math. Appl., Vol. 2, pp. 17-40.

[Gab79] Gabay, D., 1979. Methodes Numeriques pour l’Optimization Non Lineaire, These de Doctorat d’Etat et Sciences Mathematiques, Uni. Pierre at Marie Curie (Paris VI).

[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.

[GiB15] Giselsson, P., and Boyd, S., 2015. “Metric Selection in Douglas-Rachford Splitting and ADMM,” arXiv preprint arXiv:1410.8479v4.

[GlM75] Glowinski, R. and Marrocco, A., 1975. “Sur l’ Approximation par Elements Finis d’ Ordre un et la Resolution par Penalisation-Dualite d’une Classe de Problemes de Dirichlet Non Lineaires” Revue Francaise d’Automatique Informatique Recherche Operationnelle, Analyse Numerique, R-2, pp. 41-76.

[HCW14] Hong, M., Chang, T.-H., Wang, X., Razaviyayn, M., Ma, S., and Luo, Z.-Q., 2013. “A Block Successive Upper Bound Minimization Method of Multipliers for Linearly Constrained Convex Optimization,? arXiv preprint arXiv:1401.7079v1.

[HaM11] Hamdi, A., and Mishra, S. K., 2011. “Decomposition Methods Based on Augmented Lagrangians: A Survey,” in Topics in Nonconvex Optimization, Springer, N. Y., pp. 175-203.

[HoL13] Hong, M., and Luo, Z. Q., 2013. "On the Linear Convergence of the Alternating Direction Method of Multipliers," arXiv preprint arXiv:1208.3922v3.

[IST94] Iusem, A. N., Svaiter, B. F., and Teboulle, M., 1994. "Entropy-Like Proximal Methods in Convex Programming," Math. of Operations Research, Vol. 19, pp. 790-814.

[Ius99] Iusem, A. N., 1999. "Augmented Lagrangian Methods and Proximal Point Methods for Convex Minimization," Investigacion Operativa, Vol. 8, pp. 11-49.

[JuN11a] Juditsky, A., and Nemirovski, A., 2011. "First Order Methods for Nonsmooth Convex Large-Scale Optimization, I: General Purpose Methods," in Optimization for Machine Learning, by Sra, S., Nowozin, S., and Wright, S. J. (eds.), MIT Press, Cambridge, MA, pp. 121-148.

[JuN11b] Juditsky, A., and Nemirovski, A., 2011. "First Order Methods for Nonsmooth Convex Large-Scale Optimization, II: Utilizing Problem's Structure," in Optimization for Machine Learning, by Sra, S., Nowozin, S., and Wright, S. J. (eds.), MIT Press, Cambridge, MA, pp. 149-183.

[KoB72] Kort, B. W., and Bertsekas, D. P., 1972. "A New Penalty Function Method for Constrained Minimization," Proc. 1972 IEEE Confer. Decision Control, New Orleans, LA, pp. 162-166.

[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.

[Mai13] Mairal, J., 2013. "Optimization with First-Order Surrogate Functions," arXiv preprint arXiv:1305.3120.

[Mai14] Mairal, J., 2014. "Incremental Majorization-Minimization Optimization with Application to Large-Scale Machine Learning," arXiv preprint arXiv:1402.4419.

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

[NBB01] Nedić, A., Bertsekas, D. P., and Borkar, V., 2001. "Distributed Asynchronous Incremental Subgradient Methods," Proc. of 2000 Haifa Workshop "Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications," by D. Butnariu, Y. Censor, and S. Reich, Eds., Elsevier, Amsterdam.

[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.

[Ned11] Nedić, A., 2011. "Random Algorithms for Convex Minimization Problems," Math. Programming, Ser. B, Vol. 129, pp. 225-253.

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

[RoT15] Robinson, D. P., and Tappenden, R. E., 2015. "A Flexible ADMM Algorithm for Big Data Applications," arXiv preprint arXiv:1502.04391.

[Roc73] Rockafellar, R. T., 1973. "A Dual Approach to Solving Nonlinear Programming Problems by Unconstrained Optimization," Math. Programming, pp. 354-373.

[Roc76a] Rockafellar, R. T., 1976. "Monotone Operators and the Proximal Point Algorithm," SIAM J. on Control and Optimization, Vol. 14, pp. 877-898.

[Roc76b] Rockafellar, R. T., 1976. "Augmented Lagrangians and Applications of the Proximal Point Algorithm in Convex Programming," Math. of Operations Research, Vol. 1, pp. 97-116.

[Rus95] Ruszczynski, A., 1995. "On Convergence of an Augmented Lagrangian Decomposition Method for Sparse Convex Optimization," Math. of Operations Research, Vol. 20, pp. 634-656.

[SLB13] Schmidt, M., Le Roux, N., and Bach, F., 2013. "Minimizing Finite Sums with the Stochastic Average Gradient," arXiv preprint arXiv:1309.2388.

[StW75] Stephanopoulos, G., and Westerberg, A. W., 1975. "The Use of Hestenes' Method of Multipliers to Resolve Dual Gaps in Engineering System Optimization," J. Optimization Theory and Applications, Vol. 15, pp. 285-309.

[TaM85] Tanikawa, A., and Mukai, M., 1985. "A New Technique for Nonconvex Primal-Dual Decomposition of a Large-Scale Separable Optimization Problem," IEEE Trans. Autom. Control, Vol. AC-30, pp. 133-143

[Tad89] Tatjewski, P., 1989. "New Dual-Type Decomposition Algorithm for Nonconvex Separable Optimization Problems," Automatica, Vol. 25, pp. 233-242.

[TsB93] Tseng, P., and Bertsekas, D. P., 1993. "On the Convergence of the Exponential Multiplier Method for Convex Programming," Math. Programming, Vol. 60, pp. 1-19.

[WHM13] Wang, X., Hong, M., Ma, S., Luo, Z. Q., 2013. "Solving Multiple-Block Separable Convex Minimization Problems Using Two-Block Alternating Direction Method of Multipliers," arXiv preprint arXiv:1308.5294.

[WaB13] Wang, M., and Bertsekas, D. P., 2013. "Incremental Constraint Projection-Proximal Methods for Nonsmooth Convex Optimization," Lab. for Information and Decision Systems Report LIDS-P-2907, MIT, to appear in SIAM J. on Optimization.

[WaB15] Wang, M., and Bertsekas, D. P., 2015. "Incremental Constraint Projection Methods for Variational Inequalities," Mathematical Programming, Vol. 150, pp. 321-363.