Proximal Algorithm Meets a Conjugate descent

Matthieu Kowalski

Proximal Algorithm Meets a Conjugate descent

Matthieu Kowalski
Laboratoire des Signaux et Systèmes
UMR 8506 CNRS - SUPELEC - Univ Paris-Sud 11

July 26, 2010

Abstract

An extansion of the non linear conjugate gradient algorithm is proposed for some non smooth problems. We extend some results of descent algorithm in the smooth case for convex non smooth functions. We then construct a conjugate descente algorithm based on the proximity operator to obtain a descent direction. Analysis of convergence of this algorithm is provided.

keywords: non smooth optimization, conjugate descent algorithm, proximal algorithm

1 Introduction

A common and convenient formulation to deal with an inverse problem is to model it as a variational problem, giving rise to a convex optimization problem. In this article, we focus on the following formulation:

$$\underset{x\in\mathbb{R}^N}{\text{minimize}}\, F(x) = f_1(x) + f_2(x) \ , \tag{1}$$

assuming that

Assumption 1.
- *f_1 is a proper convex lower semi-continuous function, $L-$Lipshitz differentiable, with $L > 0$,*
- *f_2 is a non-smooth proper convex lower semi-continuous function,*
- *F is coercive.*

A wide range of inverse problems belongs to this category. In the past decades, several algorithms have been proposed to deal with this general framework, intensively used in the signal processing community, as stressed in Combettes *et al.* [8]. An outstanding illustration concerns regularized or constrained least squares. For about 15 years, the convex non-smooth $\ell_2 - \ell_1$ case, known as Basis Pursuit (Denoising) [7] in signal processing or as Lasso [25] in machine learning and statistics, has been widely studied both in a theoretical and practical point of view. This specific problem highlights interesting properties, in particular the sparsity principle which finds a typical application with the compressive sensing [10],[6].

Within the general framework given by (1) and Assumption 1,[1] we aim to generalize a classical algorithm used in smooth optimization: the non-linear conjugate gradient algorithm. To solve Problem (1), we propose to take advantage of the forward-backward proximal approach to find a good descent direction and to construct a practical conjugate descent algorithm. To our knowledge, such a method has not been proposed in this context, although a generalization of the steepest residual methods was proposed in the past for non-smooth problem [27].

The paper is organized as follows. Section 2 recalls definitions and results on convex analysis. In Section 3, we give a brief state of the art concerning the methods that deal with Problem (1), and describe more precisely the two algorithms which inspired ours: the forward-backward proximal algorithm [8] and the non-linear conjugate gradient method [23]. We then extend some results known in the smooth case for (conjugate) gradient descent to the non-smooth case in Section 4. Hence, we derive and analyze the resulting algorithm in Section 5.

2 Reminder on convex analysis

This section is devoted to important definitions, properties and theorems issued from convex analysis, which will be intensively used in the rest of the paper. First, we focus on directional derivatives and subgradients which are important concepts to deal with non differentiable functionals. In this context, we define the notion of a descent direction and give some important properties used to state some results of convergence in the following sections. Finally, the foundations concerning proximity operators are recalled together with an important theorem of convex optimization.

Definition 1 (Directional derivative)**.** *Let F be a lower semi-continuous convex function on $\mathbb{R}^N$. Then, for all $x \in \mathbb{R}^N$, for all $d \in \mathbb{R}^N$, the directional derivative exists and is defined by*

$$F'(x;d) = \lim_{\lambda \downarrow 0} \frac{F(x+\lambda d) - F(x)}{\lambda} \ .$$

In addition, we also give the definition of the subdifferential which is a significant notion of convex analysis.

Definition 2 (Subdifferential)**.** *Let F be a lower semi-continuous convex function on $\mathbb{R}^N$. The subdifferential of F at x is the set defined by*

$$\partial F(x) = \left\{ g \in \mathbb{R}^N \ , F(y) - F(x) \geq \langle g, y-x \rangle \text{ for all } y \in \mathbb{R}^N \right\} \ ,$$

[1] In here and what follows, the denomination Problem (1) refers to this combination.

or equivalently

$$\partial F(x) = \left\{ g \in \mathbb{R}^N \ , \langle g, d \rangle \leq F'(x; d) \ \textit{for all} \ d \in \mathbb{R}^N \right\} .$$

An element of the subdifferential is called a subgradient. A consequence of this definition is that

$$\sup_{g \in \partial F(x)} \langle g, d \rangle = F'(x; d) \ ,$$

and we will denote

$$g_s(x; d) = \operatorname*{arg\,sup}_{g \in \partial F(x)} \langle g, d \rangle \ . \tag{2}$$

As we are interested by descent methods for optimization, we recall the definition of a descent direction.

Definition 3 (Descent direction)**.** *Let $F : \mathbb{R}^N \to \mathbb{R}$ be a convex function. d is a descent direction for F at x if and only if there exists $\alpha > 0$ such that $F(x + \alpha d) < F(x)$.*

More precisely, we have the following proposition usefull for convex optimization.

Proposition 1. *Let $F : \mathbb{R}^N \to \mathbb{R}$ be a convex function. d is a descent direction for F at x if and only if, for all $g \in \partial F(x)$, $\langle d, g \rangle < 0$.*

In the following, in order to proove some convergence results, we will also need the following propositions, that specify some kind of continuity properties of the subgradient.

Proposition 2. *Let $F : \mathbb{R}^N \to \mathbb{R}$ be a convex function and $\partial F(x)$ its subdifferential at x. The function $x \mapsto \partial F(x)$ has a closed graph. i.e, let $\{x_k\}$ be a sequence of $\mathbb{R}^N$ such that $\lim_{k\to\infty} x_k = \bar{x}$, and $g_k \in \partial F(x_k)$ a sequence such that $\lim_{k\to\infty} g_k = \bar{g}$. Then*

$$\bar{g} \in \partial F(\bar{x}) \ .$$

However, as stressed in [5], we *do not have*:

$$x_k \to \bar{x}, \bar{g} \in \partial F(\bar{x}) \Rightarrow \exists g_k \in \partial F(x_k) \to \bar{g} \ .$$

Because of this lack of continuity, the steepest descent method for non-smooth convex functions does not necessarily converge (see [5] for a counter example). Nevertheless, we can proove the following proposition.

Proposition 3. *Let $F : \mathbb{R}^N \to \mathbb{R}$ be a convex function. Let $d \in \mathbb{R}^N$ be a descent direction at $x \in \mathbb{R}^N$. Let $\{\alpha_k\}$ be a sequence of $\mathbb{R}_+$ such that $\lim_{k\to\infty} \alpha_k = 0$. Then*

$$\lim_{k\to\infty} \langle g_s(x + \alpha_k d; d), d \rangle = \langle g_s(x; d), d \rangle \ ,$$

where g_s is defined in Eq. (2).

Proof. With proposition 2 states that $\lim_{k\to\infty} \langle g_s(x+\alpha_k d; d), d\rangle \leq \langle g_s(x; d), d\rangle$. We proove here that $\lim_{k\to\infty} \langle g_s(x + \alpha_k d; d), d\rangle \geq \langle g_s(x; d), d\rangle$.

$$
\begin{aligned}
\lim_{k\to\infty} \langle g_s(x + \alpha_k d; d), d\rangle &= \lim_{k\to\infty} \lim_{\lambda\downarrow 0} \frac{F(x + \alpha_k d + \lambda d) - F(x + \alpha_k d)}{\lambda} \\
&\geq \lim_{k\to\infty} \lim_{\lambda\downarrow 0} \frac{F(x + \alpha_k d + \lambda d) - F(x)}{\lambda} \quad \text{because } d \text{ is a descent direction} \\
&\geq \lim_{k\to\infty} \lim_{\mu_k\downarrow \alpha_k} \frac{F(x + \mu_k d) - F(x)}{\mu_k - \alpha_k} \\
&\geq \lim_{k\to\infty} \lim_{\mu_k\downarrow \alpha_k} \frac{F(x + \mu_k d) - F(x)}{\mu_k} \\
&\geq \langle g_s(x; d), d\rangle
\end{aligned}
$$

where the second inequality comes from that d is a descent direction for F at x. ∎

As this work is based on the forward-backward algorithm, we will also deal with the proximity operator introduced by Moreau [16], which is intensively used in convex optimisation algorithms.

Definition 4 (Proximity operator)**.** *Let $\varphi : \mathbb{R}^N \to \mathbb{R}$ be a lower semi-continuous convex function. The proximity operator associated with φ denoted by $\text{prox}_\varphi : \mathbb{R}^N \to \mathbb{R}^N$ is given by*

$$
\text{prox}_\varphi(y) = \frac{1}{2} \underset{x\in\mathbb{R}^N}{\arg\min} \|y - x\|_2^2 + \varphi(x) \ . \tag{3}
$$

Furthermore, proximity operators are firmly non expansive, hence continuous (See [8] for more details concerning proximity operators).

To conclude this section, we state an important theorem of convex optimization [22], usefull to prove convergence of optimization algorithm in a finite dimensional setting.

Theorem 1. *Let $F : \mathbb{R}^N \to \mathbb{R}$ be a convex function, which admits a set of minimizer X^*. Let $\{x_k\}$ be a sequence satisfying $\lim_{k\to\infty} F(x_k) = F(x^*)$, with $x^* \in X^*$. Then all convergent subsequences of $\{x_k\}$ converge to a point of X^*.*

Before going further into the proximal-conjugate algorithm, we present a brief state of the art of the main existing algorithms in convex optimization. A particular attention will be paid on the two algorithms which inspire the present paper.

3 State of the art

We first expose the non-linear conjugate gradient algorithm for smooth functions, and then the Iterative Shrinkage/Thresholding Algorithm (ISTA). We conclude by a short review of popular algorithms used for convex non-smooth optimization.

3.1 Non-linear conjugate gradient (NLCG)

The conjugate gradient algorithm was first introduced to minimize quadratic functions [14], and was extended to minimize general smooth functions (non necessarily convex). This extension is usually called the non-linear conjugate gradient algorithm. There exists an extensive literature about the (non-linear) conjugate gradient. One can refer to the popular paper of Shewchuck [24] available on line, but also to the book [23] of Pytlak dedicated to conjugate gradient algorithms or to the recent survey [13].

The non-linear conjugate gradient algorithm has the following form:

Algorithm 1 (NLCG)**.** *Repeat until convergence:*

1. $p_k = -\nabla F(x_k)$
2. $d_k = p_k + \beta_k d_{k-1}$
3. *choose a step length* $\alpha_k > 0$
4. $x_{k+1} = x_k + \alpha_k d_k$

where β_k is the conjugate gradient update parameter that relies in $\mathbb{R}$. Various choices can be made for β_k. Some of the most popular are

$$\beta_k^{HS} = \frac{\langle \nabla F(x_{k+1}), \nabla F(x_{k+1}) - \nabla F(x_k) \rangle}{\langle d_k, \nabla F(x_{k+1}) - \nabla F(x_k) \rangle} \ , \tag{4}$$

$$\beta_k^{FR} = \frac{\|\nabla F(x_{k+1})\|^2}{\|\nabla F(x_k)\|^2} \ , \tag{5}$$

$$\beta_k^{PRP} = \frac{\langle \nabla F(x_{k+1}), \nabla F(x_{k+1}) - \nabla F(x_k) \rangle}{\|\nabla F(x_k)\|^2} \ . \tag{6}$$

β_k^{HS} was proposed in the original paper of Hestenes and Stiefel [14]; β_k^{FR}, introduced by Fletcher and Reeves [12], is useful for some results as the Al-Baali theorem []; β_k^{PRP}, by Polak and Ribière [20] and Polyak [21], is known to have good practical behavior. One can refer to [13] for a more exhaustive presentation of the possible choices for β_k.

3.2 Forward-backward proximal algorithm

A simple algorithm used to deal with functionals as (1) is ISTA, also known as Thresholded Landweber [9] or forward-backward proximal algorithm [8]. Let us recall that f_1 must be $L-$Lipshitz differentiable.

Algorithm 2 (ISTA)**.** *epeat until convergence:*

1. $x_{k+1} = \mathrm{prox}_{\mu f_2}\left(x_k - \mu \nabla f_1(x)\right)$

where $0 < \mu < 2/L$.

One can equivalently write the previous algorithm as a descent algorithm.

Algorithm 3 (ISTA as a descent algorithm)**.** *nitialization: Choose* $x^{(0)} \in \mathbb{R}^N$ *(for example* $\mathbf{0}$*).*
Repeat until convergence:

1. $p_k = \mathrm{prox}_{\mu\ f_2}(x_k - \mu \nabla f_1(x))$
2. $s_k = p_k - x_k$
3. $x_{k+1} = x_k + s_k$

with a constant step size equals to one.

Moreover, we are sure that s_k is a descent direction. Indeed, since f_1 is L-Lipshitz differentiable,

$$0 \leq f_1(x) - f_1(y) - \langle \nabla f_1(y), x - y \rangle \leq L/2 \|x - y\|^2 \ . \tag{7}$$

Let us introduce now the surrogate:

$$F^{sur}(x, y) = f_1(y) + \langle \nabla f_1(y), x - y \rangle + L/2 \|x - y\|^2 + f_2(x) \ . \tag{8}$$

Hence, for all $x, y \in \mathbb{R}^N$

$$F(x) = F^{sur}(x, x) \leq F^{sur}(x, y) \ . \tag{9}$$

Let us denote by x_{k+1} the minimizer of $F^{sur}(., x_k)$. Then, one can prove that

$$x_{k+1} = \arg\min_x F^{sur}(x, x_k) = \mathrm{prox}_{\frac{1}{L} f_2}(x_k - \nabla f_1(x_k)/L) \ . \tag{10}$$

Such a choice assures to decrease the value of the functional:

$$\begin{aligned} f_1(x_{k+1}) + f_2(x_{k+1}) &= F^{sur}(x_{k+1}, x_{k+1}) \\ &\leq F^{sur}(x_{k+1}, x_k) \\ &\leq F^{sur}(x_k, x_k) \\ &\leq f_1(x_k) + f_2(x_k) \ . \end{aligned}$$

Consequently, $s_k = x_{k+1} - x_k$ is a descent direction for F at x_k.

It is well known that ISTA converges to a minimizer of F [8], [9]. We can state the following corollary of this convergence results.

Corollary 1. *Let* F *be the function as defined in* (1)*. Let* $\{x_k\}$ *be generated by a descent algorithm, and let* $p_k = \mathrm{prox}_{\mu f_2}(x_k - \frac{1}{L}\nabla f_1(x_k))$*, with* $0 < \mu < 2/L$*. If* $\lim_{k\to\infty} x_k - p_k = 0$*, then all convergent subsequences of* $\{x_k\}$ *converge to a minimizer of* F*.*

Proof. $F(x_k)$ is a decreasing sequence bounded from bellow. As F is continuous and stand in a finite dimensional space, one can extract a convergent subsequence of $\{x_k\}$, with $\tilde{x}$ being its limit. As the prox operator is continuous, let $\{\tilde{p}_k\}$ being the corresponding subsequence of $\{p_k\}$ obtained from $\{x_k\}$.

Then, for $\varepsilon/2 > 0$, there exists $K > 0$ such that for all $k > K$, we have $\|\tilde{p}_k - \tilde{x}_k\| < \varepsilon/2$ and $\|\tilde{x}_k - \tilde{x}\| < \varepsilon/2$. Hence, for all $k > K$, $\|\tilde{p}_k - \tilde{x}\| \leq \|\tilde{p}_k - \tilde{x}_k\| + \|\tilde{x}_k - \tilde{x}\| < \varepsilon$. Thus, $\tilde{x}$ is proven to be a fixed point of $\mathrm{prox}_{\frac{1}{L} f_2}(. - \frac{1}{L}\nabla f_1(.))$. Moreover, one can state that $\tilde{x}$ is a minimizer of F, using Propostion 3.1 from [8].

Finally, applying Theorem 1 leads to Corollary 1. ∎

3.3 Others algorithms

As already mentioned in the introduction, a various range of algorithms were developed during the past years. In particular, one can cite algorithms inspired by the significant works of Nesterov [18, 17], such as the Fast Iterative Shrinkage/Thresholding Algorithm (FISTA) of Beck and Teboulle [3]. The main advantages of this algorithm is the speed on convergence, in $\mathcal{O}(\frac{1}{k^2})$, where k is the number of iterations, which must be compared to the speed of ISTA in $\mathcal{O}(\frac{1}{k})$. This theoretical results are often verified in practice: ISTA is much slower than FISTA to reach a good estimation of the sought minimizer. Tn [26], Paul Tseng gives a good overview, with generalizations and extensions of such accelerated first order algorithm. Other accelerated algorithms were proposed, such as SPARSA by Wright *et al.* [28] or the alternating direction methods via the augmented Lagrangian [19].

4 A general conjugate descent algorithm

In this section, we generalize some theoretical results known for gradient descent in the smooth case, to a general descent algorithm which can be used to minimize a convex functional. We first present a general conjugate descent algorithm, not studied yet as far as we know in the non smooth case, and discuss the choice of the step length thanks to an extension of the Wolfe conditions known for the smooth case (see for example [4, 23]). We then study the convergence of the algorithm for different choices of the step length. For this, we extend the notion of "uniformly gradient related" descent proposed by Bertsekas [4] and generalize the Al-Baali theorem [1], which assures that the conjugation provides a descent direction under some conditions for the choice of the conjugate parameter.

4.1 A general (conjugate) descent algorithm for non-smooth functions

We extend the non linear conjugate gradient Algorithm 1 by presenting the following general conjugate descent algorithm.

Algorithm 4.
Initialization: choose $x^{(0)} \in \mathbb{R}^N$ *(for example* $\mathbf{0}$*).*
Repeat until convergence:

1. *find* s_k*, a descent direction at* x_k *for* F
2. *choose* β_k*, the conjugate parameter*
3. $d_k = s_k + \beta_k d_{(k-1)}$
4. *find a step length* $\alpha_k > 0$
5. $x_{k+1} = x_k + \alpha_k d_k$

This algorithm obviously reduces to a classical general descent algorithm as Algorithm 3 with an adaptive step length if $\beta_k = 0$.

Ideally, one would find the optimal step size α_k. However, in the general case, one does not have access to a closed form of this optimal step size. Then, a line search, based on the Wolfe conditions, must be performed.

4.2 (Modified) Wolfe conditions

Wolfe conditions are usually defined for smooth functions, in order to perform a line search with a proper step size. These conditions were extended to convex, non necessarily differentiable, functions in [29]. At each iteration k, let x_k be updated as in step 5 of Algorithm 4. One can perform a line search to choose the step size α_k in order to verify the Wolfe conditions:

$$F(x_k + \alpha_k d_k) - F(x_k) \leq c_1 \alpha_k \langle g_s(x_k; d_k) d_k \rangle \tag{11}$$

$$\langle g_s(x_k + \alpha_k d_k; d_k), d_k \rangle \geq c_2 \langle g_s(x_k; d_k), d_k \rangle \ , \tag{12}$$

with $0 < c_1 < c_2 < 1$, and g_s the element of the subgradient defined as in (2).

As in the smooth case, one can extend these conditions to obtain the strong Wolfe conditions by replacing (12) by

$$|\langle g_s(x_k + \alpha_k d_k; d_k), d_k \rangle| \leq -c_2 \langle g_s(x_k; d_k), d_k \rangle \ . \tag{13}$$

One can prefer the Mifflin-Wolfe conditions proposed by Mifflin in [15] for non smooth problems (although the latter is not referred as "Wolfe conditions" by the author):

$$F(x_k + \alpha_k d_k) - F(x_k) \leq -c_1 \alpha_k \|d_k\|^2 \tag{14}$$

$$\langle g_s(x_k + \alpha_k d_k; d_k), d_k \rangle \geq -c_2 \|d_k\|^2 \ , \tag{15}$$

with $0 < c_1 < c_2 < 1$.

Mifflin proposed a procedure which converges in a *finite number of iterations* to a solution α satisfying the Mifflin-Wolfe conditions. The procedure is the following:

Algorithm 5 (Line search)**.**
Initialization: Choose $\alpha > 0$*. Set* $\alpha_L = 0, \alpha_N = +\infty$*.*
Repeat until α *verifies* (14) *and* (15)

1. *If* α *verifies* (14) *set* $\alpha_L = \alpha$

   Else $\alpha_N = \alpha$

2. *If* $\alpha_N = +\infty$ *set* $\alpha = 2\alpha$

   Else $\alpha = \frac{\alpha_L + \alpha_N}{2}$

Now that we have defined rules to choose the step length, we pay attention to the convergence properties of Algorithm 4.

4.3 Convergence results

We first provide general results about the descent method for convex non-smooth functional, which generalize the ones obtained in the smooth case. We begin by stating the following theorem.

Theorem 2. *Let $F : \mathbb{R}^N \to \mathbb{R}$ be a convex function. Assume that $\{x_k\}$ is a sequence generated by algorithm 4, and that d_k is a descent direction for all k such that $F(x_k + \alpha_k d_k) < F(x_k)$.*

If α_k is a constant step size or satisfies the Mifflin-Wolfe, then

$$\lim_{k\to\infty} \|d_k\| = 0 \ .$$

If α_k is the optimal step size or satisfies the Wolfe conditions, then

$$\lim_{k\to\infty} \langle g_s(x_k, d_k), d_k\rangle = 0 \ ,$$

where g_s is a subgradient as defined in (2)*.*

Proof. We provide here the proof for the Mifflin-Wolfe conditions. The proof for others are straightforward. Since d_k is a descent direction, the sequence of $F(x_k)$ is decreasing, and as it is bounded from bellow, converges to some F^*. Then $\sum_{k=0}^{\infty} F(x_k) - F(x_{k+1}) < +\infty$.

From the first Mifflin-Wolfe condition, we can state that

$$\lim_{k\to\infty} \alpha_k \|d_k\|^2 = 0 \ .$$

Suppose that $\lim\limits_{k\to 0} \alpha_k = 0$ and $\|d_k\|$ doest not tend to 0. Then, during Algorithm 5, we can find α such that:

$$F(x_k + \alpha d_k) - F(x_k) > -c_1 \alpha \|d_k\|^2 \ .$$

Thus,

$$F(x_k + \alpha d_k) - F(x_k + \alpha_k d_k) > -c_1(\alpha - \alpha_k)\|d_k\|^2 \ ,$$

and because F is weakly upper semi-smooth,

$$\lim \inf_{\alpha\downarrow\alpha_k} \langle g_s(x_k+\alpha d_k; d_k), d_k\rangle \geq \lim \sup_{\alpha\downarrow\alpha_k} \frac{F(x_k + \alpha d_k) - F(x_k + \alpha_k d_k)}{\alpha - \alpha_k} \geq -c_1\|d_k\| \ .$$

Thanks to proposition 3, we also have that

$$\lim_{\alpha_k\downarrow 0} \langle g_s(x_k + \alpha_k d_k; d_k), d_k\rangle = \langle g_s(x_k; d_k), d_k\rangle \ .$$

Therefore, there exists a number $K \in \mathbb{N}^*$, such that for all $k \geq K$:

$$\langle g_s(x_k; d_k), d_k\rangle \geq -c_1\|d_k\|^2 \ ,$$

i.e.

$$c_1 \geq \frac{|\langle g_s(x_k; d_k), d_k \rangle|}{\|d_k\|^2} \ .$$

Then, as $c_1 < 1$ for $k > K$, we have

$$|\langle g_s(x_k; d_k), d_k \rangle| \leq \|d_k\|^2 \ .$$

From the second Mifflin-Wolfe condition, we obtain that for all $k > K$:

$$\begin{aligned} \langle g_s(x_k + 1; d_k) - g_s(x_k; d_k), d_k \rangle &\geq \langle g_s(x_{k+1}; d_k), d_k \rangle - \langle g_s(x_k; d_k), d_k \rangle \\ &\geq -c_2 \|d_k\| - \langle g_s(x_k; d_k), d_k \rangle \\ &\geq (1 - c_2)\|d_k\|^2 \ , \end{aligned}$$

with $c_2 < 1$, contradicting that $\lim\limits_{k\to\infty} \langle g(x_k + \alpha_k d_k; d_k), d_k \rangle = \langle g_s(x_k; d_k), d_k \rangle$. Then $\lim\limits_{k\to 0} \|d_k\| = 0$. ∎

Remark 1. *Usually, such results are obtained in the smooth case assuming that the gradient is Lipshtiz continuous (see for example [23]). Even if such an hypothesis simplifies the proof, we have seen in the previous proof that it is not at all necessary.*

Such a theorem is not sufficient to ensure convergence of the descent algorithm. Indeed, one needs stronger hypothesis when $\langle g_s(x_k, d_k), d_k \rangle \to 0$. For that, we adapt the definition of the *uniformly gradient related descent* of [4] to the non differentiable convex case.

Definition 5. *Let $F : \mathbb{R}^N \to \mathbb{R}$ be a convex function, and $\partial F(x)$ its subdifferential at x. Let $\{x_k\}$ be a sequence generated by a descent method, with $x_{k+1} = x_k + \alpha_k d_k$. The sequence $\{d_k\}$ is uniformly subgradient related to $\{x_k\}$ if for every convergent subsequence $\{x_k\}_K$ for which*

$$\lim_{k\to\infty, k\in K} 0 \notin \partial F(x_k) \ ,$$

there holds

$$0 < \liminf_{k\to\infty, k\in K} |F'(x_k; d_k)| \ , \qquad \limsup_{k\to\infty, k\in K} |d_k| < \infty \, .$$

Thanks to this definition, if d_k is uniformly subgradient related to x_k, then with the Wolfe conditions, one can conclude that $g_s(x_k; d_k) \to 0$, i.e. the descent algorithm converges to a minimizer of the functional. Furthermore, we will see that if s_k is properly chosen, one can assure the convergence results under the Mifflin-Wolfe conditions.

4.4 A uniformly subgradient related conjugation

In the case of an optimal choice, then we are sure to obtain a descent direction at each iteration:

Lemma 1. *Let $F : \mathbb{R}^N \to \mathbb{R}$ be a convex function. Let $\alpha_k^* = \arg\min_{\alpha>0} F(x_k + \alpha d_k)$, where d_k is a descent direction for F at x_k. If s_{k+1} is a descent direction for F at $x_{k+1} = x_k + \alpha_k^* d_k$, then for all $\beta_k > 0$, $d_{k+1} = s_{k+1} + \beta_k d_k$ is also descent direction for F at x_{k+1}.*

Proof. For all $g(x_{k+1}) \in \partial F(x_{k+1})$, by definition of α_k^*, $\langle d_k, g(x_{k+1})\rangle = 0$. Hence, for all $g(x_{k+1}) \in \partial F(x_{k+1})$,

$$\begin{aligned}\langle d_{k+1}, g(x_{k+1})\rangle &= \langle s_{k+1} + \beta_k d_k, \partial F(x_{k+1})\rangle \\ &= \langle s_{k+1}, g(x_{k+1})\rangle < 0 \ ,\end{aligned}$$

because s_{k+1} is a descent direction. ∎

However, as we do not usually have access to the optimal step, it would be interesting to know when the conjugacy parameter β_k assures to obtain an descent direction. Inspired by Al-Baali theorem [1], we provide the following theorem.

Theorem 3. *Let $F : \mathbb{R}^N \to \mathbb{R}$ be a convex function. Let $\{x_k\}$ a sequence generated by the conjugate descent algorithm 4, where for all k, the step size α_k was chosen under the strong Wolfe conditions (11), (13). Let $d_k = s_k + \beta_k d_{k-1}$, such that s_k is uniformly subgradient related. If $\beta_k \leq \frac{\langle g_s(x_{k+1};d_{k+1}), s_{k+1}\rangle}{\langle g_s(x_k;d_k), s_k\rangle}$, then d_k is a uniformly gradient related descent direction.*

Proof. By induction, distinguish two cases. 1) If $\langle g_s(x_{k+1}, d_{k+1}), d_k\rangle \leq 0$, then conclusion follows immediately.

2) If $\langle g_s(x_{k+1}, d_{k+1}), d_k\rangle > 0$, then

$$|\langle g_s(x_{k+1}; d_{k+1}), d_k\rangle| \leq |\langle g_s(x_{k+1}; d_k), d_k\rangle| \ ,$$

and, with the strong Wolfe conditions ?

$$|\langle g_s(x_{k+1}; d_{k+1}), d_k\rangle| \leq -c_2 \langle g_s(x_k; d_k), d_k\rangle \ .$$

We have

$$\frac{\langle g_s(x_{k+1}; d_{k+1}), d_{k+1}\rangle}{|\langle g_s(x_{k+1}; d_{k+1}), s_{k+1}\rangle|} = \frac{\langle g_s(x_{k+1}; d_{k+1}), s_{k+1}\rangle}{|\langle g_s(x_{k+1}; d_{k+1}), s_{k+1}\rangle|} + \beta_{k+1} \frac{\langle g_s(x_{k+1}; d_{k+1}), d_k\rangle}{|\langle g_s(x_{k+1}; d_{k+1}), s_{k+1}\rangle|} \ .$$

Consequently

$$\begin{aligned}\frac{\langle g_s(x_{k+1}; d_{k+1}), d_{k+1}\rangle}{|\langle g_s(x_{k+1}; d_{k+1}), s_{k+1}\rangle|} &\leq -1 - c_2 \beta_{k+1} \frac{\langle g_s(x_k; d_k), d_k\rangle}{|\langle g_s(x_{k+1}; d_{k+1}), s_{k+1}\rangle|} \\ &\leq -1 - c_2 \frac{\langle g_s(x_k; d_k), d_k\rangle}{|\langle g_s(x_k; d_k), s_k\rangle|} \ .\end{aligned}$$

By definition of $g_s(x_k, d_k)$, we have that $-1 \leq \frac{\langle g_s(x_k;d_k),d_k\rangle}{|\langle g_s(x_k;d_k),s_k\rangle|}$, and finally,

$$\frac{\langle g_s(x_{k+1};d_{k+1}),d_{k+1}\rangle}{|\langle g_s(x_{k+1};d_{k+1}),s_{k+1}\rangle|} \leq -1 + c_2 < 0 \ .$$

∎

Note that in the smooth case, the bound on β_k reduces to the conjugate parameter proposed by Fletcher and Reeves, in which case Theorem 3 corresponds to Al-Baali's results.

5 Proximal conjugate algorithm

This section is dedicated to the proposed proximal conjugate algorithm, which gives a practical choice to choose an appropriate descent direction, thanks to the proximity operator. We begin with a study of the algorithm and show that it is an authentic conjugate gradient algorithm when f_2 is a quadratic function. We also analyze its asymptotic speed of convergence.

5.1 The algorithm

The idea is to construct a conjugate direction, based on the descent $p_k - x_k$. This gives the following algorithm:

Algorithm 6 (Proximal Conjugate Algorithm)**.** *Repeat until convergence:*

1. $p_k = prox_{f_2/L}\left(x_k - \frac{1}{L}\nabla f_1(x_k)\right)$
2. $s_k = p_k - x_k$
3. *Choose the conjugate parameter* β_k
4. $d_k = s_k + \beta_k d^{(k-1)}$
5. *Choose the step length* α_k
6. $x_{k+1} = x_k + \alpha_k d_k$

First, we prove that the descent direction s_k provided by the proximal operation is uniformly subgradient related.

Proposition 4. *Let F be a convex function, defined as in Eq.* (1) *under Assumption 1, $\{x_k\}$ be a sequence generated by a descent method, $p_k = \mathrm{prox}_{\frac{1}{L}f_2}\left(x_k - \frac{1}{L}\nabla f_1(x_k)\right)$ and $s_k = p_k - x_k$. Then the sequence $\{s_k\}$ is uniformly subgradient related.*

Proof. Let $\tilde{x}_k$ a convergent subsequence of $\tilde{x}$ such that $\lim_{k\to\infty} \tilde{x}_k = \tilde{x}$, $\tilde{p}_k = \mathrm{prox}_{f_2/L}\left(\tilde{x}_k - \frac{1}{L}\nabla f_1(\tilde{x}_k)\right)$, and $\lim_{k\to\infty} \tilde{p}_k = \tilde{p}$. We also denote $\tilde{s}_k = \tilde{p}_k - \tilde{x}_k$ and $\lim_{k\to\infty} \tilde{s}_k = \tilde{s}$. Assume that $\tilde{x}$ is not a critical point.

We first prove that, if x is a critical point of $F^{sur}(., x_k)$, then

$$F^{sur}(x+h, x_k) - F^{sur}(x, x_k) \geq L\|\mathbf{h}\|_2^2 \ .$$

For that, we compute $\partial_x F^{sur}(x, a)$:

$$\partial_x F^{sur}(x, a) = \nabla f_1(x) + L(x-a) + \partial f_2(x) \ ,$$

and define:

$$g_s^{sur}(x, a; d) = \underset{g \in \partial_x F^{sur}(x,a)}{\arg\sup} \ \langle g, d \rangle \ .$$

As a consequence $g_s^{sur}(x, x; d) = g_s(x; d)$. One can check that

$$\begin{aligned} F^{sur}(x+h, x_k) - F^{sur}(x, x_k) = \langle \partial F^{sur}(x, x_k), h\rangle + L/2\|h\|_2^2 \\ + \{f_2(x+h) - f_2(x) - \langle \partial f_2(x), h\rangle\} \ . \end{aligned}$$

Since x is a critical point of $F^{sur}(., x_k)$, for all h, we have $\langle \partial F^{sur}(x, x_k), h\rangle = 0$, then

$$F^{sur}(x+h, x_k) - F^{sur}(x, x_k) = L/2\|h\|_2^2 + \{f_2(x+h) - f_2(x) - \langle \partial f_2(x), h\rangle\} \ .$$

By definition of the subgradient, an element v belongs to $\partial f_2(x)$ if and only if for all y, $f_2(x) + \langle v, y-x\rangle \leq f_2(y)$. In particular, when $y = x+h$, for all h and for all $v \in \partial f_2(x)$, we have that

$$f_2(x) + \langle v, h\rangle \leq f_2(y) \text{ i.e. } 0 \leq f_2(x+h) - f_2(x) - \langle \partial f_2(x), h\rangle \ ,$$

and

$$F^{sur}(x+h, x_k) - F^{sur}(x, x_k) \geq L/2\|h\|_2^2 \ .$$

Now, we apply the previous inequality to $x = p_k$, which is a critical point of $F^{sur}(., x_k)$ as seen in Section 3.2, and to $h = -s_k$. This gives

$$\begin{aligned} -L/2\|s_k\| &\geq F^{sur}(p_k, x_k) - F^{sur}(p_k - s_k, x_k) \\ &\geq F^{sur}(p_k, x_k) - F^{sur}(x_k, x_k) \\ &\geq \langle g_s^{sur}(x_k, x_k; s_k), s_k\rangle \\ &\geq \langle g_s(x_k; s_k), s_k\rangle \ , \end{aligned}$$

where the third inequality comes from the definition of the subgradient $g_s^{sur}(x_k, x_k; s_k)$, for the descent direction $s_k = p_k - x_k$. Taking the limit, we have then

$$L/2\|\tilde{s}\|^2 \leq \liminf |\langle g_s(\tilde{x}, \tilde{s}), \tilde{s}\rangle| \ ,$$

as $\tilde{s} \neq 0$ (otherwise, $\tilde{x}$ is a critical point), which concludes the proof. ∎

Then, if α_k is chosen with the Wolfe conditions, the proximal conjugate algorithm converges (assuming that d_k is a descent direction for all k). Furthermore, if α_k is chosen with the Mifflin-Wolfe conditions, we also have the convergence of the algorithm.

Theorem 4. *Let F be a convex function, defined as in Eq.* (1) *under Assumption 1. Let $\{x_k\}$ be a sequence generated by Algorithm 6. Assume that for all k, α_k is chosen thanks to the Mifflin-Wolfe conditions, d_k is a descent direction, and β_k is bounded. Then all convergent subsequences of $\{x_k\}$ converge to a minimizer of F.*

Proof. Immediate using Theorem 2 and Corollary 1. ∎

5.2 Remarks on the step size

Variants of ISTA estimate at each iteration the Lipshitz-parameter L in order to assure convergence of the Algorithm. Such a variant is restated in Algorithm 7. One can refer for example to [3] for more details.

Algorithm 7 (ISTA with Line search)**.** *Let $\eta > 1$.*
Repeat until convergence:

1. *Find the smallest integer i_k such that with $\mu_k = \frac{1}{\eta^{i_k} L_{k-1}}$ and with*

$$x_{k+1} = \operatorname{prox}_{\mu_k f_2}\left(x_k - \mu_k \nabla f_1(x)\right) \ ,$$

we have $F(x_{k+1}) \leq \bar{F}^{sur}(x_{k+1}, x_k)$, where $\bar{F}^{sur}$ is defined as in Eq. (8) *replacing L by $\eta^{i_k} L_{k-1}$.*

Then, in frameworks like SPARSA [28], the authors propose to use μ_k as a step parameter, and propose strategies as the Bazilei-Borwein choice to set it up. The following lemma establishes a necessary and sufficient condition which states that when μ_k is equivalent to the step-size parameter α_k in Algorithm 6 when the conjugate parameter β_k is set to zero.

Lemma 2. *Let F be a convex function defined as in Eq.* (1) *under Assumption 1, $p_k = \operatorname{prox}_{\frac{1}{L} f_2}\left(x_k - \frac{1}{L}\nabla f_1(x)\right)$, $x_{k+1} = x_k + \alpha_k(p_k - x_k)$. We also have $x_{k+1} = \operatorname{prox}_{\frac{\alpha_k}{L} f_2}\left(x_k - \frac{\alpha_k}{L}\nabla f_1(x)\right)$ if and only if $\partial f_2(p_k) \cap \partial f_2(x_{k+1}) \neq \emptyset$.*

Proof. By definition of the proximity operator, $x_k - \frac{1}{L}\nabla f_1(x_k) - p_k \in \frac{1}{L}\partial f_2(p_k)$.
Let us denote by $p_k^\alpha = \operatorname{prox}_{\frac{\alpha_k}{L} f_2}\left(x_k - \frac{\alpha_k}{L}\nabla f_1(x)\right)$. Then

$$\begin{aligned}
p_k^\alpha = x_k + \alpha_k(p_k - x_k) &\Leftrightarrow x_k - \frac{\alpha_k}{L}\nabla f_1(x_k) - x_k - \alpha_k(p_k - x_k) \in \frac{\alpha_k}{L}\partial f_2(p_k^\alpha) \\
&\Leftrightarrow 0 \in -\frac{\alpha_k}{L}\nabla f_1(x_k) + \frac{\alpha_k}{L}\left(\nabla f_1(x_k) + \partial f_2(p_k)\right) - \frac{\alpha_k}{L}\partial f_2(p_k^\alpha) \\
&\Leftrightarrow 0 \in \partial f_2(p_k) - \partial f_2(p_k^\alpha) \\
&\Leftrightarrow \partial f_2(p_k) \cap \partial f_2(x_{k+1}) \neq \emptyset
\end{aligned}$$

∎

However, the necessary and sufficient condition given in the previous Lemma is hard to check, and can never occur for certain choices of function f_2 (for example, if f_2 is differentiable).

5.3 The quadratic case

A natural question concerns the behavior of this proximal-conjugate descent algorithm when f_2 is quadratic, i.e.

$$f_2(x) = \frac{1}{2}\langle x, Qx\rangle + \langle c, x\rangle \ ,$$

with Q a symmetric definite positive linear application, and $c \in \mathbb{R}^N$. We have then

$$\begin{aligned}
\hat{x} = \mathrm{prox}_{\mu f_2}(y) &= \arg\min_x \frac{1}{2}\|y-x\|^2 + \mu f_2(x) \\
\iff 0 &= \hat{x} - y + \mu Q x + \mu c \\
\iff \hat{x} &= (I + Q\mu)^{-1}(y - \mu c)
\end{aligned}$$

Hence, the descent direction s_k given in the proximal conjugate algorithm is

$$\begin{aligned}
s_k &= \mathrm{prox}_{\mu f_2}(x_k - \mu \nabla f_1(x_k)) - x_k \\
&= (I + \mu Q)^{-1}(x_k - \mu \nabla f_1(x_k) - \mu c) - x_k \\
&= (I + \mu Q)^{-1}(-\mu \nabla f_1(x_k) - \mu c - \mu Q x_k) \\
&= -(\frac{1}{\mu} I + Q)^{-1}(\nabla f_1(x_k) + \nabla f_2(x_k))
\end{aligned}$$

The proximal conjugate descent is then the classical conjugate gradient algorithm preconditioned by $\frac{1}{\mu} I + Q$.

5.4 Speed of convergence

Intuitively, the conjugate algorithm has asymptotically the same behavior as ISTA. Then, one can expect that the speed of convergence will be $O(1/k)$, for k large enough. This is stated with the following theorem.

Theorem 5. *Let F be a convex function satisfying Assumption 1 and $x*$ a minimizer of F. Let $\{x_k\}$ the sequence generated by the proximal conjugate Algorithm 6. Then, there exist $K > 0$ such that for all $k > K$, $F(x_k) - F(x^*) \leq \frac{L\|x^* - x_k\|^2}{2(k-K+1)}$.*

Proof. The proof is based on the one given by Tseng in [26] for the speed of convergence of ISTA.

Let

$$\ell_F(x; y) = f(y) + \langle \nabla f(y), x - y\rangle + \lambda P(x) \ .$$

We can recall the "three points property": if $z_+ = \arg\min_x \psi(x) + \frac{1}{2}\|x - z\|2$, then

$$\psi(x) + \frac{1}{2}\|x - z\| \geq \psi(z_+) + \frac{1}{2}\|z_+ - z\|^2 + \frac{1}{2}\|x - z_+\|^2$$

Moreover, with the following inequality

$$F(x) \geq \ell_F(x;y) \geq F(x) - \frac{L}{2}\|x-y\|^2 \ ,$$

$$F(p_k) \leq F(x) + \frac{L}{2}\|x-x_k\|^2 - \frac{L}{2}\|x-p_k\|^2$$

$$\sum_{n=K}^{k} F(p_n) - F(x) \leq \frac{L}{2}\sum_{n=K} k(\|x-x_n\|^2 - \|x-p_n\|^2)$$

Since the sequence of $F(p_k)$ is decreasing, we have

$$\begin{aligned}
(k-K+1)(F(p_k)-F(x)) &\leq \frac{L}{2}\sum_{n=K}^{k}(\|x-x_n\|^2 - \|x-p_n\|^2) \\
&\leq \frac{L}{2}\sum_{n=K}^{k}(\|x-x_n\|^2 - \|x-x_{n+1}\|^2 - \|x_{n+1}-p_n\|^2) \\
&\leq \frac{L}{2}\|x-x_k\|^2 - \frac{L}{2}\|x-x_{k+1}\| - \frac{L}{2}\sum_{n=K}^{k}\|x_{n+1}-p_n\| \\
&\leq \frac{L}{2}\|x-x_k\|^2 - \frac{L}{2}\sum_{n=K}^{k}\|x_{n+1}-p_n\|
\end{aligned}$$

For all ε_1, there exists a number K_1 for which all $k \geq K_1$ $|F(x_k) - F(p_k)| < \varepsilon_1$. Moreover, for all ε_2, there exists a number K_2 such that for all $k \geq K_2$ $\|x_{k+1} - p_k\| < \varepsilon_2$. The choices $\varepsilon_1 = \frac{L}{2}\varepsilon_2$ and $K = max(K_1, K_2)$, ensure that

$$\begin{aligned}
F(x_k) - F(x^*) &\leq \frac{L\|x^*-x_k\|^2}{2(k-K+1)} - \frac{L}{2}\varepsilon_2 + \varepsilon_1 \\
F(x_k) - F(x^*) &\leq \frac{L\|x^*-x_k\|^2}{2(k-K+1)} \ .
\end{aligned}$$

∎

5.5 An approximate proximal conjugate descent algorithm

In Algorithm 6, one must be able to compute exactly the proximity operator of function f_2. However, in many cases, one do not have access to a close form solution, but can only approximate it thanks to iterative algorithms. In that case a natural question arises: how does behave the proposed algorithm when we cannot have a close form formula for the proximity operator?

The study made in section 4 shows that one needs to obtain a descent direction s_k to construct the conjugate direction d_k. Remember that the proximity operator has exactly the form of the general optimization problem given by

Eq. (1). Then, any iterative algorithm able to deal with this kind of problem can estimate the solution of the proximity operator, within an inner loop of the main prox-conj algorithm.

Using such a procedure may be computationnaly costly. Nevertheless, with a few iterations of the inner loop, the functional decreases. Since we only need a descent direction, as defined in Definition 3, we are looking for an algorithm where step 1. in Algorithm 6 is replaced by:

$$1.\text{ Find } \check{p}_k \text{ such that } F^{sur}(\check{p}_k, x_k) < F^{sur}(x_k, x_k)$$

Indeed, in that case we have

$$F(\check{p}_k) = F^{sur}(\check{p}_k, \check{p}_k) \leq F^{sur}(\check{p}_k, x_k) \leq F^{sur}(x_k, x_k) = F(x_k) \ ,$$

regarding the definition of the surrogate F^{sur} given by Eq. (8) and the inequality (9). Then at Step 2. of the prox-conj algorithm, $s_k = \check{p}_k - x_k$ is guaranteed to be a descent direction. But, this descent direction may not be uniformly subgradient related anymore and there is no more guaranty to converge to a minimizer of the functional. Nevertheless for a certain class of function f_2, we can establish a strategie which ensure the convergence. From now, we assume the following.

Assumption 2. *There exists a linear operator* $\Phi : \mathbb{R}^N \to \mathbb{R}^M$ *and a function* $\tilde{f} : \mathbb{R}^M \to \mathbb{R}^M$ *such that* $f_2 : \mathbb{R}^N \to \mathbb{R}$ *can be written as*

$$\mu f_2(x) = \tilde{f}(\Phi x) \ .$$

Denoting by $\tilde{f}^*$*, the Fenchel conjugate of* $\tilde{f}$*, we suppose that the proximity operator of* $\tilde{f}^*$ *is given by a closed form.*

Again, we do not have access to a close formula for $\operatorname{prox}_{\mu f_2}$. However, using the Fenchel dual formulation we can rewrite this minimization problem such that

$$\min_u \frac{1}{2}\|y - u\|_2^2 + \tilde{f}(\Phi u) = \min_v \|y - \Phi^* v\| + \langle \phi^* v, y \rangle + \tilde{f}^*(v) \ .$$

Moreover, thanks to the KKT conditions, the following relationship between the primal variable u and the dual variable v holds:

$$u^\star = \Phi^* v^\star - y \ .$$

Hence, one can use any known algorithm to obtain an approximation of the proximal solution at step 1 of Algorithm 6. Such a strategy is already used in practice (see for example [11, 2]). However, this inner loop is usually run in order to obtain a estimate close to the true minimizer, and may can be a computational burden. In the light of the remark above, we propose to stop the inner loop as soon as a point allowing to decrease the original functional is obtained. This strategy is summarized in the following algorithm using ISTA in the inner loop[2].

[2]We choose ISTA here in order to keep the Algorithm simple. One can choose any other algorithm as FISTA, if the stopping criterion remains the same.

Algorithm 8 (Approximate Proximal Conjugate Algorithm)**.** *Repeat until convergence:*

1. $y_k = x_k - \frac{1}{L}\nabla f_1(x_k)$
2. *Computation of* p_k *such that* $F^{sur}(p_k, x_k) \le F^{sur}(x_k, p_k)$*, with* $p_k = x_k$ *only if* $F^{sur}(x_k, x_k) = \min_p F^{sur}(p, x_k)$*.*
   Repeat while $F^{sur}(x_k, x_k) < F^{sur}(p_k, x_k)$
   (a) $v_{\ell+1} = \operatorname{prox}_{\tilde{f}}(v_\ell - \Phi^*(\Phi y_k - v_\ell))$
   (b) $v_k = v_{\ell+1}$
   (c) $p_k = G^* v_k - y$
3. $s_k = x_k - p_k$
4. *Choose the conjugate parameter* β_k
5. $d_k = -s_k + \beta_k d^{(k-1)}$
6. *Choose the step length* α_k
7. $x_{k+1} = x_k + \alpha_k d_k$

When β_k is set to zero at each iteration, the step size α_k is set to one and the inner loop is run until "convergence", in which case the algorithm is reduced to the one proposed for the Total Variation regularized inverse problems in [11]. Here, we propose a simple criterion to stop the inner loop, and the convergence is given by the following theorem.

Theorem 6. *Let* $\{x_k\}$ *be a sequence generated by Algorithm 8. Assume that for all k,* d_k *is a descent direction and* β_k *is bounded. Then, if* α_k *is chosen thanks to the Mifflin-Wolfe conditions, or is a constant step size,* $\{x_k\}$ *converges to a minimizer of* F*.*

Proof. We first show that, in a finite number of iterations, we can find $p_k = \Phi^* v_k - y$, such that $F^{sur}(p_k, x_k) < F^{sur}(x_k, x_k)$, if x_k is not a minimizer of $F^{sur}(., x_k)$. Assume the opposite: $\forall \ell \; F^{sur}(p_\ell, x_k) \ge F^{sur}(x_k, x_k)$. Then v_ℓ converges to a fixed point of $\operatorname{prox}_{\tilde{f}}(. - \Phi^*(\Phi y_k - .))$, and by definition of the Fenchel duality, p_ℓ converges to $\arg\min_p \frac{1}{2}\|y_k - p\|^2 + \lambda f_2(p)$. Hence $\lim_{\ell\to\infty} F^{sur}(p_\ell, x_k) = F^{sur}(x_k, x_k)$, contradicting that x_k is not a minimizer of $F^{sur}(., x_k)$.

Secondly, using the same arguments than in Theorem 2, we have $\lim_{k\to 0} \|d_k\| = 0$, and then $\lim_{k\to 0} \|s_k\| = 0$. Let $\tilde{x}$ be an accumulation point of $\{x_k\}$, which is also an accumulation point of $\{p_k\}$. We have

$$\begin{aligned}\lim_{k\to\infty} F^{sur}(p_k, x_k) &= F^{sur}(\tilde{x}, \tilde{x}) \\ &= \min_p F^{sur}(p, \tilde{x}) \\ &= \min_x F(x) \quad \text{by definition of } F^{sur} \;.\end{aligned}$$

Then, applying Theorem 1, Algorithm 8 converges. ∎

References

[1] M. Al-Baali. Descent property and global convergence of the fletcher-reeves method with inexact line search. *IMA Journal of Numerical Analysis*, 5:121–124, 1985.

[2] A. Beck and M. Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring. *IEEE Transactions on Image Processing*, 18(11):2419–2434, 2009.

[3] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[4] Dimitri P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, 1982.

[5] Frédéric Bonnans, J. Charles Gilbert, Claude Lemaréchal, and Claudia A. Sagastiábal. *Numerical Optimization*. Springer, 2003.

[6] E. J. Candès and T. Tao. Near optimal signal recovery from random projections : universal encoding strategies ? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006.

[7] S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

[8] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling and Simulation*, 4(4):1168–1200, November 2005.

[9] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. Pure Appl. Math.*, 57(11):1413 – 1457, Aug 2004.

[10] David L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[11] Jalal Fadili and Gabriel Peyré. Total variation projection with first order schemes. Technical report, 2009.

[12] R. Fletcher and C. Reeves. Function minimization by conjugate gradients. *Comput. Journal*, 7:149–154, 1964.

[13] William G. Hager and Hongchao Zhang. A survey of nonlinear conjugate gradient methods. *Pacific Journal of Optimization - Special Issue on Conjugate Gradient and Quasi-Newton Methods for Nonlinear Optimization*, 2(1):35 – 58, Jan 2006.

[14] Magnus R. Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49:409–436, 1952.

[15] R. Mifflin. An algorithm for constrained optimization with semismooth functions. *Math. Oper. Res.*, 2:191 – 207, 1977.

[16] J.-J. Moreau. Proximité et dualité dans un espace hilbertien. *Bull. Soc. Math. France*, 93:273–299, 1965.

[17] Y.E. Nesterov. Gradient methods for minimizing composite objective function. Technical report, 2007. CORE discussion paper – Université Catholique de Louvain.

[18] Yurii E. Nesterov. method for solving the convex programming problem with convergence rate $o(1/k^2)$. *Dokl. Akad. Nauk SSSR*, 269(3):543–547, 1983.

[19] M. Ng, P. Weiss, and X.-M. Yuan. Solving constrained total-variation image restoration and reconstruction problems via alternating direction methods. Technical report, 2009.

[20] E. Polak and Ribière. Note sur la convergence de directions conjuguées. *Revue Fran caise d'Informatique et de Recherche Opérationelle*, 3(16):35–43, 1969.

[21] B.T. Polyak. The conjugate gradient method in extreme problems. *USSR Comp. Math. Math. Phys.*, 9:94–112, 1969.

[22] B.T. Polyak. *Introduction to Optimization*. Translation Series in Mathematics and Engineering, Optimization Software, 1987.

[23] Radoslaw Pytlak. *Conjugate Gradient Algorithms in Nonconvex Optimization*. Springer, 2009.

[24] Jonathan Richard Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. 1994.

[25] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Serie B*, 58(1):267–288, 1996.

[26] Paul Tseng. Approximation accuracy, gradient methods, and error bound for structured convex optimization. Technical report, 2009.

[27] Philip Wolfe. A method of conjugate subgradients for minimizing nondifferentiable functions. *Mathematical Programming Studies*, 3:145–173, 1975.

[28] S. Wright, R. Nowak, and M. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7):2479–2493, 2009.

[29] Jin Yu, S.V.N. Vishwanathan, Simon Gunter, and Schraudolph Nicol N. A quasi-newton approach to nonsmooth convex optimization problems in machine learning. *Journal of Machine Learning Research*, 11:1 – 57, 2010.