Un petit tutoriel sur les méthodes primales-duales proximales pour l'optimisation convexe

Audrey REPETTI, Émilie CHOUZENOUX, Jean-Christophe PESQUET

Laboratoire d'Informatique Gaspard Monge
5, Boulevard Descartes, Champs sur Marne, 77454 MARNE LA VALLEE Cedex 2, France
audrey.repetti@univ-paris-est.fr, emilie.chouzenoux@univ-paris-est.fr,
jean-christophe.pesquet@univ-paris-est.fr

Résumé – Les développements récents en imagerie et en analyse de données se sont accompagnés d'un besoin accru en méthodes rapides pour la résolution de problèmes d'optimisation convexe de très grande dimension. Ces dernières années, un engouement particulièrement vif a été suscité par les approches de type primales-duales, se démarquant par leur simplicité et leur flexibilité. Cet article tutoriel se propose de donner au lecteur une vue d'ensemble de ces méthodes, de leurs conditions d'applications, de convergence, et de leur potentiel d'accélération via des stratégies stochastiques par blocs.

Abstract – Recent developments in imaging and data analysis techniques came along with an increasing need for fast convex optimization methods for solving large scale problems. In the last years, primal-dual approaches have raised a strong enthusiasm, because of their simplicity of use and their flexibility. This tutorial paper provides an overview of these algorithms, their applicability conditions, their convergence, and their potential acceleration using stochastic block coordinate strategies.

1 Introduction

De nombreux champs d'application, tels que le traitement d'images, la vision par ordinateur et l'apprentissage, requièrent la conception de stratégies efficaces pour la résolution de problèmes d'optimisation convexe formulés de la manière suivante :

$$\text{trouver } \widehat{\mathbf{x}} \in \underset{\mathbf{x}\in\boldsymbol{\mathcal{H}}}{\operatorname{Argmin}}\; \mathsf{f}(\mathbf{x}) + \mathsf{h}(\mathbf{x}) + \mathsf{g}(\mathbf{L}\mathbf{x}), \tag{1}$$

où $\boldsymbol{\mathcal{H}}$ et $\boldsymbol{\mathcal{G}}$ sont des espaces hilbertiens réels, f, g, et h sont des fonctions convexes définies respectivement sur $\boldsymbol{\mathcal{H}}$, $\boldsymbol{\mathcal{G}}$ et $\boldsymbol{\mathcal{H}}$, h est de plus supposée de gradient Lipschitz, et $\mathbf{L}$ est un opérateur linéaire borné de $\boldsymbol{\mathcal{H}}$ vers $\boldsymbol{\mathcal{G}}$. On autorisera les fonctions f et g à prendre leurs valeurs dans $]-\infty,+\infty]$, de façon à pouvoir modéliser des contraintes sur la solution recherchée. Parmi les nombreuses approches disponibles pour résoudre (1), les stratégies de type primales-duales se détachent, notamment dans le cadre de résolution de problèmes de grande taille, de par leur aptitude à éclater le critère initial en une somme de termes plus simples qui peuvent être traités individuellement, de façon parallèle, sans nécessiter l'inversion des opérateurs linéaires pouvant apparaître dans le problème. Les termes différentiables peuvent être traités par des étapes de gradient et, si des fonctions non lisses sont impliquées, par exemple lorsque des contraintes ou des mesures de parcimonie interviennent, leur opérateur proximal peut être employé. Cela conduit aux algorithmes proximaux primaux-duaux dont cet article effectue un tour d'horizon.

Tout d'abord, notons que les algorithmes proximaux primaux-duaux existants peuvent être décomposés principalement en trois classes :

- des méthodes basées sur l'algorithme explicite-implicite [16, 31, 9, 29] : ces méthodes alternent à chaque itération une étape de gradient et une étape proximale.
- des méthodes basées sur l'algorithme explicite-implicite-explicite [7, 14, 5, 11] : en comparaison aux approches précédentes, ces méthodes ajoutent une étape de gradient supplémentaire à chaque itération. Ce sont cependant les premières méthodes primales-duales qui ont été proposées dans la littérature pour résoudre (1) en exploitant le caratère lisse de h.
- des méthodes basées sur des projections [2] : il s'agit de la classe la plus récente, dont le principal avantage, par comparaison aux autres méthodes, est de ne pas avoir besoin de connaître la norme de l'opérateur linéaire $\mathbf{L}$.

Ce tutoriel est centré sur la description de la première classe de méthodes, peut être plus simple à appréhender, et sur laquelle des développements récents ont été réalisés. Après avoir défini nos notations en section 2, nous présenterons dans la section 3, des algorithmes proximaux primaux-duaux parallèles où l'ensemble des variables primales et duales est mis à jour à chaque itération. Nous montrerons ensuite dans la section 4 comment réduire le coût de calcul et de mémoire de ces méthodes, en adoptant des stratégies d'alternance aléatoire par bloc.

2 Problèmes primaux-duaux

2.1 Problème de minimisation

Dans toute la suite de l'article, on supposera sans toujours le rappeler explicitement que toutes les fonctions intervenant sont convexes, semi-continues inférieurement et de domaine non vide.

L'objectif est de trouver une solution du problème (1). Ce problème faisant intervenir la composition d'une fonction non lisse avec un opérateur linéaire peut être résolu en traitant à la fois (1) et sa forme duale correspondante [25] :

$$\text{trouver } \widehat{\mathbf{v}} \in \underset{\mathbf{v}\in\boldsymbol{\mathcal{G}}}{\operatorname{Argmin}}\, (\mathsf{f}^* \,\square\, \mathsf{h}^*)(-\mathbf{L}^*\mathbf{v}) + \mathsf{g}^*(\mathbf{v}). \tag{2}$$

Dans le problème dual ci-dessus, $\mathsf{h}^* \colon \mathcal{H} \to]-\infty, +\infty]$ désigne la conjuguée (de Legendre-Fenchel) de la fonction h qui est définie par $(\forall \mathbf{x} \in \mathcal{H})\ \mathsf{h}^*(\mathbf{x}) = \sup_{\mathbf{y}\in\mathcal{H}} \langle \mathbf{y} \mid \mathbf{x}\rangle - \mathsf{h}(\mathbf{y})$. De plus, $\square$ désigne l'infimale convolution définie par $\mathsf{f}^* \,\square\, \mathsf{h}^* \colon \mathcal{H} \to [-\infty, +\infty] \colon \mathbf{x} \mapsto \inf_{\mathbf{y}\in\mathcal{H}} (\mathsf{f}^*(\mathbf{y}) + \mathsf{h}^*(\mathbf{x}-\mathbf{y}))$. L'élément neutre de l'infimale convolution est la fonction indicatrice de l'ensemble $\{\mathbf{0}\}$, notée $\iota_{\{\mathbf{0}\}}$, où $\mathbf{0}$ représente le vecteur nul de $\mathcal{H}$: $\mathsf{f}^* \,\square\, \iota_{\{\mathbf{0}\}} = \mathsf{f}^*$.

Remarquons que les problèmes (1)-(2) peuvent être reformulés de façon à obtenir un problème de recherche de point selle du Lagrangien [3, Chap. 19] :

$$(\mathbf{x}, \mathbf{v}) \quad \mapsto \quad \mathsf{h}(\mathbf{x}) \;+\; \mathsf{f}(\mathbf{x}) \;-\; \mathsf{g}^*(\mathbf{v}) \;+\; \langle \mathbf{L}\mathbf{x}, \mathbf{v}\rangle. \tag{3}$$

Ainsi, en notant $\partial \mathsf{f}$ et $\partial \mathsf{g}$ les sous-différentiels de f et g, si le couple $(\widehat{\mathbf{x}}, \widehat{\mathbf{v}})$ vérifie les conditions de Karush-Kuhn-Tucker

$$-\mathbf{L}^*\widehat{\mathbf{v}} - \nabla \mathsf{h}(\widehat{\mathbf{x}}) \in \partial \mathsf{f}(\widehat{\mathbf{x}}) \quad \text{et} \quad \mathbf{L}\widehat{\mathbf{x}} \in \partial \mathsf{g}^*(\widehat{\mathbf{v}}), \tag{4}$$

alors $\widehat{\mathbf{x}}$ est une solution du problème primal (1), $\widehat{\mathbf{v}}$ est une solution du problème dual (2), et $(\widehat{\mathbf{x}}, \widehat{\mathbf{v}})$ est une solution du problème (3). L'objectif des algorithmes primaux-duaux est de rechercher un tel couple de vecteurs.

2.2 Formulation par bloc

Afin de pouvoir définir des algorithmes primaux-duaux permettant de décomposer efficacement les variables par blocs, nous allons préciser nos notations et hypothèses.

Soient $(\mathcal{H}_j)_{1\leqslant j\leqslant p}$ et $(\mathcal{G}_n)_{1\leqslant n\leqslant q}$ des espaces hilbertiens réels séparables. Choisissons $\boldsymbol{\mathcal{H}} = \mathcal{H}_1 \oplus \ldots \oplus \mathcal{H}_p$ et $\boldsymbol{\mathcal{G}} = \mathcal{G}_1 \oplus \ldots \oplus \mathcal{G}_q$ où $\oplus$ désigne la somme hilbertienne. Un élément de $\boldsymbol{\mathcal{H}}$ (resp. $\boldsymbol{\mathcal{G}}$) s'écrit ainsi $\mathbf{x} = (\mathsf{x}^{(j)})_{1\leqslant j\leqslant p}$ (resp. $\mathbf{v} = (\mathsf{v}^{(n)})_{1\leqslant n\leqslant q}$) où $(\forall j \in \{1, \ldots, p\})\ \mathsf{x}^{(j)} \in \mathcal{H}_j$ (resp. $(\forall n \in \{1, \ldots, q\})$ $\mathsf{v}^{(n)} \in \mathcal{G}_n$).

Nous nous intéressons au problème (1) où, pour tout $(\mathbf{x}, \mathbf{v}) \in \boldsymbol{\mathcal{H}} \oplus \boldsymbol{\mathcal{G}}$,

$$\begin{aligned} &\mathsf{f}(\mathbf{x}) = \textstyle\sum_{j=1}^p \mathsf{f}_j(\mathsf{x}^{(j)}), && \mathsf{h}(\mathbf{x}) = \textstyle\sum_{j=1}^p \mathsf{h}_j(\mathsf{x}^{(j)}),\\ &\mathbf{L}\mathbf{x} = \Big(\textstyle\sum_{j=1}^p \mathsf{L}_{n,j}\mathsf{x}^{(j)}\Big)_{1\leqslant n\leqslant q}, && \mathsf{g}(\mathbf{v}) = \textstyle\sum_{n=1}^q \mathsf{f}_n(\mathsf{x}^{(n)}). \end{aligned}$$

D'autre part, dans la suite de cet article, nous supposerons que (1) admet une solution et que les conditions de qualification permettant d'appliquer le théorème de dualité de Fenchel-Rockafellar sont satisfaites (c.f. [3, Chap. 15] et [25]). Ce théorème nous assure de l'existence d'une solution au problème dual (2) et de l'absence de saut de dualité, *i.e.*

$$\mathsf{f}(\widehat{\mathbf{x}}) + \mathsf{h}(\widehat{\mathbf{x}}) + \mathsf{g}(\mathbf{L}\widehat{\mathbf{x}}) = -\Big((\mathsf{f}^* \,\square\, \mathsf{h}^*)(-\mathbf{L}^*\widehat{\mathbf{v}}) + \mathsf{g}^*(\widehat{\mathbf{v}})\Big). \tag{5}$$

2.3 Notation supplémentaires

Dans la suite, nous noterons $\mathcal{S}^+(\boldsymbol{\mathcal{H}})$ l'ensemble des opérateurs linéaires auto-adjoints bornés fortement positifs de $\boldsymbol{\mathcal{H}}$ dans $\boldsymbol{\mathcal{H}}$ (un élément de $\mathcal{S}^+(\boldsymbol{\mathcal{H}})$ est donc une matrice définie-positive en dimension finie). Soit $\mathbf{U} \in \mathcal{S}^+(\boldsymbol{\mathcal{H}})$, nous définissons la norme pondérée par $\mathbf{U}$ comme $\|\cdot\|_{\mathbf{U}} = \sqrt{\langle \cdot \mid \mathbf{U}\cdot\rangle}$. L'opérateur proximal (relatif à la métrique induite par $\mathbf{U}$) de la fonction semi-continue inférieurement, convexe et de domaine non vide $\mathsf{f} \colon \boldsymbol{\mathcal{H}} \to]-\infty, +\infty]$ en un point $\mathbf{x} \in \boldsymbol{\mathcal{H}}$, noté $\operatorname{prox}_{\mathbf{U},\mathsf{f}}(\mathbf{x})$, a été défini dans [27] comme étant l'unique minimiseur de la fonction $\mathsf{f} + \frac{1}{2}\|\cdot - \mathbf{x}\|_{\mathbf{U}}^2$. Cet opérateur possède de nombreuses propriétés intéressantes [3] et, en particulier, peut être défini pour des fonctions non lisses.

3 Algorithmes primaux duaux

3.1 Méthodes de lagrangien augmenté

Comme indiqué dans [25], l'algorithme ADMM (*Alternating Direction Method of Multipliers*) [6, 21, 22] peut être vu comme un algorithme primal-dual. Cette méthode permet de résoudre les problèmes (1)-(2), lorsque $\mathbf{L}^*\mathbf{L}$ est un isomorphisme. Remarquons que cet algorithme est équivalent à l'*algorithme de Douglas-Rachford* [17, 13] lorsque ce dernier est appliqué au problème dual (2).

Bien que cet algorithme soit beaucoup utilisé en traitement du signal [23, 24, 19, 20, 1, 30], en pratique, il peut être difficile à mettre en œuvre. En effet, lorsque l'opérateur $\mathbf{L}$ est une matrice de grande taille et/ou n'a pas une structure simple, le calcul de chaque itérée peut se révéler coûteux. D'autre part, la différentiabilité de la fonction h n'est pas exploitée de façon explicite dans cette méthode.

3.2 Algorithme explicite-implicite

Une autre méthode permettant de résoudre à la fois le problème primal (1) et le problème dual (2) est dérivée des travaux de [16, 31]. Ses itérées sont données dans l'algorithme 1. Remarquons que, contrairement à l'algorithme ADMM, ici la matrice $\mathbf{L}$ n'a pas besoin de satisfaire de condition particulière. D'autre part, $\boldsymbol{a}_k$, $\boldsymbol{b}_k$ et $\boldsymbol{c}_k$ représentent des termes d'erreurs pouvant apparaitre respectivement lors des calculs des opérateurs proximaux et du gradient à l'itération k. Observons également que, lorsque g et $\mathbf{L}$ sont identiquement nulles, on retrouve la forme de base de l'algorithme explicite-implicite.

Dans l'algorithme 1, lorsque $p = 1$, $q = 1$ et pour tout $k \in \mathbb{N}$, $\varepsilon_k^{(1)} = \varepsilon_k^{(2)} = 1$ on obtient l'algorithme décrit dans [16,

Algorithme 1 Algorithme primal-dual

Initialisation : Soit $(\boldsymbol{x}_0, \boldsymbol{v}_0)$ à valeurs dans $\boldsymbol{\mathcal{H}} \oplus \boldsymbol{\mathcal{G}}$. Pour tout $j \in \{1, \dots, p\}$, soit $\mathsf{W}_j \in \mathcal{S}^+(\mathcal{H}_j)$, et, pour tout $n \in \{1, \dots, q\}$, soit $\mathsf{U}_n \in \mathcal{S}^+(\mathcal{G}_n)$.

Itérations :

Pour $k = 0, 1, \dots$

$$
\left\lfloor
\begin{array}{l}
\text{pour } j = 1, \dots, p \\
\left\lfloor
\begin{array}{l}
y_k^{(j)} = \varepsilon_k^{(j)} \Big(\operatorname{prox}_{\mathsf{W}_j^{-1}, \mathsf{f}_j} \big(x_k^{(j)} - \mathsf{W}_j (\nabla \mathsf{h}_j (x_k^{(j)}) + c_k^{(j)} \\
\qquad\qquad + \sum_{n=1}^{q} \mathsf{L}_{n,j}^* v_k^{(n)}) \big) + a_k^{(j)} \Big), \\
x_{k+1}^{(j)} = x_k^{(j)} + \lambda_k \varepsilon_k^{(j)} (y_k^{(j)} - x_k^{(j)}),
\end{array}
\right. \\
\text{pour } n = 1, \dots, q \\
\left\lfloor
\begin{array}{l}
u_k^{(n)} = \varepsilon_k^{(p+n)} \Big(\operatorname{prox}_{\mathsf{U}_n^{-1}, \mathsf{g}_n^*} \big(v_k^{(n)} \\
\qquad\qquad + \mathsf{U}_n \sum_{j=1}^{p} \mathsf{L}_{n,j} (2 y_k^{(j)} - x_k^{(j)}) \big) + b_k^{(n)} \Big), \\
v_{k+1}^{(n)} = v_k^{(n)} + \lambda_k \varepsilon_k^{(p+n)} (u_k^{(n)} - v_k^{(n)}).
\end{array}
\right.
\end{array}
\right.
$$

Algorithme 2 Algorithme primal-dual (forme symétrique)

Initialisation : Soit $(\boldsymbol{x}_0, \boldsymbol{v}_0)$ à valeurs dans $\boldsymbol{\mathcal{H}} \oplus \boldsymbol{\mathcal{G}}$. Pour tout $j \in \{1, \dots, p\}$, soit $\mathsf{W}_j \in \mathcal{S}^+(\mathcal{H}_j)$, et, pour tout $n \in \{1, \dots, q\}$, soit $\mathsf{U}_n \in \mathcal{S}^+(\mathcal{G}_n)$.

Itérations :

Pour $k = 0, 1, \dots$

$$
\left\lfloor
\begin{array}{l}
\text{pour } n = 1, \dots, q \\
\left\lfloor
\begin{array}{l}
u_k^{(n)} = \varepsilon_k^{(p+n)} \Big(\operatorname{prox}_{\mathsf{U}_n^{-1}, \mathsf{g}_n^*} \big(v_k^{(n)} \\
\qquad\qquad + \mathsf{U}_n \sum_{j=1}^{p} \mathsf{L}_{n,j} x_k^{(j)} \big) + b_k^{(n)} \Big), \\
v_{k+1}^{(n)} = v_k^{(n)} + \lambda_k \varepsilon_k^{(p+n)} (u_k^{(n)} - v_k^{(n)}),
\end{array}
\right. \\
\text{pour } j = 1, \dots, p \\
\left\lfloor
\begin{array}{l}
y_k^{(j)} = \varepsilon_k^{(j)} \Big(\operatorname{prox}_{\mathsf{W}_j^{-1}, \mathsf{f}_j} \big(x_k^{(j)} - \mathsf{W}_j (\nabla \mathsf{h}_j (x_k^{(j)}) + c_k^{(j)} \\
\qquad\qquad + \sum_{n=1}^{q} \mathsf{L}_{n,j}^* (2 u_k^{(n)} - v_k^{(n)})) \big) + a_k^{(j)} \Big), \\
x_{k+1}^{(j)} = x_k^{(j)} + \lambda_k \varepsilon_k^{(j)} (y_k^{(j)} - x_k^{(j)}).
\end{array}
\right.
\end{array}
\right.
$$

31]. La suite $(\boldsymbol{x}_k, \boldsymbol{v}_k)_{k \in \mathbb{N}}$ générée par l'algorithme 1 converge alors faiblement vers un couple de solutions $(\widehat{\mathbf{x}}, \widehat{\mathbf{v}})$ des problèmes (1)-(2), sous certaines conditions techniques. Par exemple, si l'on suppose que $\mathsf{W}_1 = \tau \mathrm{Id}$, et $\mathsf{U}_1 = \sigma \mathrm{Id}$, avec $(\tau, \sigma) \in]0, +\infty[^2$, il suffit que

(i) $\tau^{-1} - \sigma \|\mathbf{L}\|^2 > \beta/2$, où β est la constante de Lipschitz de h,

(ii) pour tout $k \in \mathbb{N}$, $\lambda_k \in]0, \overline{\lambda}[$, où $\overline{\lambda} = 2 - \beta(\tau^{-1} - \sigma\|\mathbf{L}\|^2)^{-1}/2 \in [1, 2[$,

(iii) $\sum_{k \in \mathbb{N}} \lambda_k (\overline{\lambda} - \lambda_k) = +\infty$,

(iv) $\sum_{k \in \mathbb{N}} \lambda_k \|\boldsymbol{a}_k\| < +\infty$, $\sum_{k \in \mathbb{N}} \lambda_k \|\boldsymbol{b}_k\| < +\infty$, et $\sum_{k \in \mathbb{N}} \lambda_k \|\boldsymbol{c}_k\| < +\infty$.

Des résultats de convergence complémentaires ont récemment été obtenus dans [26].

Notons que le problème primal (1) et le problème dual (2) sont symétriques. Ainsi, une alternative à l'algorithme 1, bénéficiant des mêmes propriétés de convergence que ce dernier, est obtenue en intervertissant les variables primales et duales. Cette méthode est décrite dans l'algorithme 2.

On peut remarquer que l'algorithme présenté dans [9, 29] est un cas particulier de l'algorithme 1. En effet, lorsque $\mathsf{h} \equiv 0$, et les termes d'erreurs $(\boldsymbol{a}_k)_{k \in \mathbb{N}}$, $(\boldsymbol{b}_k)_{k \in \mathbb{N}}$ et $(\boldsymbol{c}_k)_{k \in \mathbb{N}}$ sont tous nuls, l'algorithme 1 se réduit à l'algorithme 3. Quand $p = 1$, $q = 1$, $\varepsilon_k^{(1)} \equiv \varepsilon_k^{(2)} \equiv 1$, les conditions de convergence de cet algorithme sont moins restrictives que celles vues précédemment [16].

Par ailleurs, en utilisant le théorème de décomposition de Moreau [3], en effectuant le changement de variable $(\forall k \in \mathbb{N})$ $\boldsymbol{p}_k = \boldsymbol{v}_k / \sigma$ puis en réarrangeant les mises à jours des variables, on obtient alors l'algorithme présenté dans [8, 18].

Mentionnons enfin qu'une seconde classe d'algorithmes primaux-duaux explicites-implicites a été proposée dans la littérature [10, 12].

Algorithme 3 Algorithme primal-dual simplifié

Initialisation : Soit $(\boldsymbol{x}_0, \boldsymbol{v}_0)$ à valeurs dans $\boldsymbol{\mathcal{H}} \oplus \boldsymbol{\mathcal{G}}$. Pour tout $j \in \{1, \dots, p\}$, soit $\mathsf{W}_j \in \mathcal{S}^+(\mathcal{H}_j)$, et, pour tout $n \in \{1, \dots, q\}$, soit $\mathsf{U}_n \in \mathcal{S}^+(\mathcal{G}_n)$.

Itérations :

Pour $k = 0, 1, \dots$

$$
\left\lfloor
\begin{array}{l}
\text{pour } j = 1, \dots, p \\
\left\lfloor
\begin{array}{l}
y_k^{(j)} = \varepsilon_k^{(j)} \operatorname{prox}_{\mathsf{W}_j^{-1}, \mathsf{f}_j} \big(x_k^{(j)} - \mathsf{W}_j \sum_{n=1}^{q} \mathsf{L}_{n,j}^* v_k^{(n)} \big), \\
x_{k+1}^{(j)} = x_k^{(j)} + \lambda_k \varepsilon_k^{(j)} (y_k^{(j)} - x_k^{(j)}),
\end{array}
\right. \\
\text{pour } n = 1, \dots, q \\
\left\lfloor
\begin{array}{l}
u_k^{(n)} = \varepsilon_k^{(p+n)} \operatorname{prox}_{\mathsf{U}_n^{-1}, \mathsf{g}_n^*} \big(v_k^{(n)} \\
\qquad\qquad + \mathsf{U}_n \sum_{j=1}^{p} \mathsf{L}_{n,j} (2 y_k^{(j)} - x_k^{(j)}) \big), \\
v_{k+1}^{(n)} = v_k^{(n)} + \lambda_k \varepsilon_k^{(p+n)} (u_k^{(n)} - v_k^{(n)}).
\end{array}
\right.
\end{array}
\right.
$$

4 Algorithmes primaux-duaux alternés par blocs

Dans les domaines d'applications où le nombre de données peut être très important, il est utile de pouvoir réduire le nombre d'opérations qui sont traitées à chaque itération ainsi que la mémoire occupée. Une stratégie pour atteindre cet objectif consiste à diviser les données globales en plusieurs blocs, et de n'en traiter qu'un nombre réduit à chaque itération.

Récemment les résultats de convergence des algorithmes présentés dans la section 3 ont été généralisés au cas où l'on utilise une stratégie alternée dans laquelle un ensemble de blocs est activé de manière aléatoire flexible, à chaque itération. Plus précisément, on se place dans le cas où $p \geqslant 1$, $q \geqslant 1$ et $\boldsymbol{\varepsilon}_k$ est un vecteur de variables aléatoire booléennes choisies aléatoirement à l'itération k de telle sorte que chacune de ses composantes soit égale à 1 avec une probabilité non nulle.

On peut alors montrer que les résultats de convergence de la section précédente se généralisent au cas stochastique, sous certaines conditions techniques [15, 28]. Plus précisément la convergence presque sûre vers un couple de solutions du problème primal-dual est assurée pourvu que les vecteurs aléatoires $(\varepsilon_k)_{k\in\mathbb{N}}$ soient indépendants et identiquement distribués, et que les erreurs maintenant autorisées à être aléatoires vérifient des hypothèses de sommabilité.

5 Conclusion

Les méthodes proximales primales-duales présentées offrent des solutions pertinentes aux problèmes d'optimisation de grande dimension. Leurs extensions stochastiques récentes ouvrent un nouvel univers de possibilités. On peut, en particulier, en déduire des versions distribuées de ces algorithmes [28, 4]. Le fait de pouvoir considérer des erreurs aléatoires dans ces méthodes permet aussi d'envisager de nouvelles méthodes d'approximations stochastiques dont les applications sont très larges, notamment dans le domaine de l'apprentissage. En pratique, un utilisateur pourra orienter son choix vers une méthode plutôt qu'une autre, en fonction de la structure du problème à traiter, de l'adéquation entre l'algorithme et l'architecture de calculs, ainsi que des restrictions imposées par les conditions de convergence.

Références

[1] M.V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo. An augmented lagrangian approach to the constrained optimization formulation of imaging inverse problems. 20(3) :681–695, Mar. 2011.

[2] A. Alotaibi, P. L. Combettes, and N. Shahzad. Solving coupled composite monotone inclusions by successive Fejér approximations of their Kuhn-Tucker set. *SIAM J. Optim.*, 24(4) :2076–2095, Dec. 2014.

[3] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, New York, 2011.

[4] P. Bianchi, W. Hachem, and F. Iutzeler. A stochastic coordinate descent primal-dual algorithm and applications to large-scale composite optimization. Technical report, 2014. http ://arxiv.org/abs/1407.0898.

[5] R. I. Boţ and C. Hendrich. Convergence analysis for a primal-dual monotone + skew splitting algorithm with applications to total variation minimization. *J. Math. Imaging Vis.*, 49(3) :551–568, Jul. 2014.

[6] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Machine Learn.*, 8(1) :1–122, 2011.

[7] L. M. Briceños Arias and P. L. Combettes. A monotone + skew splitting model for composite monotone inclusions in duality. *SIAM J. Optim.*, 21(4) :1230–1250, Oct. 2011.

[8] M. Burger, A. Sawatzky, and G. Steidl. First order algorithms in variational image processing. In S. Osher R. Glowinski and W. Yin, editors, *To appear in Operator Splittings and Alternating Direction Methods*. 2014. http ://www.mathematik.uni-kl.de/fileadmin/image/steidl/publications/algs_book_rev_final.pdf.

[9] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imag. Vision*, 40(1) :120–145, 2010.

[10] P. Chen, J. Huang, and X. Zhang. A primal-dual fixed point algorithm for convex separable minimization with applications to image restoration. *Inverse Problems*, 29(2) :025011, 2013.

[11] P. L. Combettes. Systems of structured monotone inclusions : duality, algorithms, and applications. 23(4) :2420–2447, Dec. 2013.

[12] P. L Combettes, L. Condat, J.-C. Pesquet, and B. C. Vũ. A forward-backward view of some primal-dual optimization methods in image recovery. In *Proc. IEEE Int. Conf. Image Process. (ICIP)*, pages 4141–4145, Paris, France, 27-30 Oct. 2014.

[13] P. L. Combettes and J.-C. Pesquet. A Douglas-Rachford splitting approach to nonsmooth convex variational signal recovery. *IEEE J. Selected Topics Signal Process.*, 1(4) :564–574, Dec. 2007.

[14] P. L. Combettes and J.-C. Pesquet. Primal-dual splitting algorithm for solving inclusions with mixtures of composite, Lipschitzian, and parallel-sum type monotone operators. *Set-Valued Var. Anal.*, pages 1–24, 2011.

[15] P. L. Combettes and J.-C. Pesquet. Stochastic quasi-Fejér block-coordinate fixed point iterations with random sweeping. *To appear in SIAM J. Optim.*, 2015. http ://www.optimization-online.org/DB_HTML/2014/04/4333.html.

[16] L. Condat. A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms. *J. Optim. Theory Appl.*, 158(2) :460–479, Aug. 2013.

[17] J. Eckstein and D. P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Program.*, 55(1-3) :293–318, 1992.

[18] E. Esser, X. Zhang, and T. Chan. A general framework for a class of fist order primal-dual algorithms for convex optimization in imaging science. *SIAM J. Imaging Sci.*, 3(4) :1015–1046, 2010.

[19] M. A. T. Figueiredo and J. M. Bioucas-Dias. Restoration of Poissonian images using alternating direction optimization. 19(12) :3133–3145, Dec. 2010.

[20] M. A. T. Figueiredo and R. D. Nowak. Deconvolution of Poissonian images using variable splitting and augmented Lagrangian optimization. In *Proc. IEEE Workshop Stat. Sign. Proc.*, pages x+4, Cardiff, United Kingdom, 31 Aug. – 3 Sept. 2009.

[21] M. Fortin and R. Glowinski, editors. *Augmented Lagrangian Methods : Applications to the Numerical Solution of Boundary-Value Problems*. Elsevier Science Ltd, Amsterdam : North-Holland, 1983.

[22] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite elements approximations. *Comput. Math. Appl.*, 2 :17–40, 1976.

[23] J.-F. Giovannelli and A. Coulais. Positive deconvolution for superimposed extended source and point sources. *Astron. Astrophys.*, 439 :401–412, 2005.

[24] T. Goldstein and S. Osher. The split Bregman method for ℓ_1-regularized problems. 2 :323–343, 2009.

[25] N. Komodakis and J.-C. Pesquet. Playing with duality : An overview of recent primal-dual approaches for solving large-scale optimization problems. *To appear in IEEE Signal Process. Mag.*, 2015. http ://www.optimization-online.org/DB_HTML/2014/06/4398.html.

[26] J. Liang, J. Fadili, and G. Peyré. Convergence rates with inexact nonexpansive operators. Technical report, 2014. http ://arxiv.org/abs/1404.4837.

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

[28] J.-C. Pesquet and A. Repetti. A class of randomized primal-dual algorithms for distributed optimization. *To appear in J. Nonlinear Convex Anal.*, 2015. http ://arxiv.org/abs/1406.6404.

[29] T. Pock, A. Chambolle, D. Cremers, and H. Bischof. A convex relaxation approach for computing minimal partitions. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn. (CVPR)*, pages 810–817, 20-25 June 2009.

[30] Q. Tran-Dinh, A. Kyrillidis, and V. Cevher. Composite self-concordant minimization. *To appear in J. Mach. Learn. Res.*, 2015. http ://arxiv.org/pdf/1308.2867v1.pdf.

[31] B. C. Vũ. A splitting algorithm for dual monotone inclusions involving cocoercive operators. *Adv. Comput. Math.*, 38(3) :667–681, 2013.