A Multicore Convex Optimization Algorithm with Applications to Video Restoration

Feriel Abboud, Emilie Chouzenoux, Jean-Christophe Pesquet, Hugues Talbot

A MULTICORE CONVEX OPTIMIZATION ALGORITHM WITH APPLICATIONS TO VIDEO RESTORATION

Feriel Abboud$^{(1)}$*, Emilie Chouzenoux*$^{(2,3)}$*, Jean-Christophe Pesquet*$^{(2)}$ *and Hugues Talbot*$^{(3)}$

$^{(1)}$ WITBE France
$^{(2)}$ CVN, INRIA Saclay, CentraleSupélec, Univ. Paris Saclay, France
$^{(3)}$ LIGM, UMR CNRS 8049, ESIEE, Univ. Paris Est Marne-la-Vallée, France

ABSTRACT

In this paper, we present a new distributed algorithm for minimizing a sum of non-necessarily differentiable convex functions composed with arbitrary linear operators. The overall cost function is assumed strongly convex. Each involved function is associated with a node of a hypergraph having the ability to communicate with neighboring nodes sharing the same hyperedge. Our algorithm relies on a primal-dual splitting strategy with established convergence guarantees. We show how it can be efficiently implemented to take full advantage of a multicore architecture. The good numerical performance of the proposed approach is illustrated in a problem of video sequence denoising, where a significant speedup is achieved.

***Index Terms*—** convex optimization, distributed algorithms, proximal methods, video processing, parallel programming.

1. INTRODUCTION

Numerous tasks in image processing, such as video restoration, can be formulated as nonsmooth optimization problems over large datasets. In this context, it is necessary to propose parallel/distributed methods to compute efficiently the solutions to the corresponding high-dimensional optimization problems. In this work, we focus on the case when the objective function is a sum of several convex non-necessarily smooth functions [1]. In the general case, a closed form expression of the solution does not exist, and developing iterative strategies becomes necessary.

Primal-dual splitting methods are used prominently when dealing with convex optimization problems where large-size linear operators are involved [2, 3, 4, 5]. This class of algorithms is well suited for large-scale problems encountered in image processing [6, 7]. Primal-dual techniques are based on several well-known strategies such as the Forward-Backward iteration [8, 9], the Forward-Backward-Forward iteration [10, 11], the Douglas-Rachford algorithm [12, 13], or the Alternating Direction Method of Multipliers [14, 15, 16, 17]. Recently, primal-dual algorithms have been combined with block-coordinate approaches [18, 19]. These algorithms can achieve a fast convergence speed with reduced memory requirements. Both stochastic [20, 21] and deterministic [22, 23] versions of these have been used in image processing and machine learning applications. In the latter context, algorithms based on a dual Forward-Backward approach are often refered to as dual ascent methods.

The aforementioned algorithms were originally proposed with a centralized implementation, which may be suboptimal or even unsuitable when dealing with massive datasets. Various asynchronous or distributed extensions have recently been proposed [14, 24, 25, 26], where each term is handled independently by a processing unit and the convergence toward an aggregate solution is ensured via a suitable communication strategy between those processing units.

In this paper, we propose a new proximal algorithm applicable to multicore architectures for minimizing a sum of convex functions involving linear operators when the global cost function is strongly convex. The proposed algorithm extends the algorithm that was recently proposed in [22, 27] to a distributed asynchronous scenario. In this algorithm, each involved function is locally related to a node of a connected hypergraph, where communications are allowed only between neighboring nodes. Our approach takes advantage of the sparse structure of the involved linear operators, which limits the local memory cost required for each node. Our proposal benefits from the classical key advantages of primal-dual splitting strategies, in particular their ability to handle a finite sum of convex functions without having to invert any linear operator, and its convergence is guaranteed.

The remainder of this paper is organized as follows: in Section 2 we recall some fundamental background and state the problem. In Section 3, our asynchronous block dual forward-backward optimization algorithm is presented. In particular, we describe our dimension reduction strategy. Section 4 shows the good performance of the proposed algorithm in the context of video denoising. Finally, some conclusions are drawn in Section 5.

2. PROBLEM FORMULATION

2.1. Notation

Let ψ be a proper lower-semicontinuous convex function from $\mathbb{R}^N$ to $]-\infty,+\infty]$ and let $B \in \mathbb{R}^{N\times N}$ be a symmetric positive definite matrix. The proximity operator of ψ at $\widetilde{x} \in \mathbb{R}^N$ relative to the metric induced by B, denoted by $\operatorname{prox}_{B,\psi}(\widetilde{x})$, is defined as the unique solution to the following minimization problem [28, 1]:

$$\underset{x\in\mathbb{R}^N}{\text{minimize}} \;\; \psi(x) + \frac{1}{2}\|x-\widetilde{x}\|_B^2, \tag{1}$$

where $\|\cdot\|_B$ denotes the usual Euclidean norm of $\mathbb{R}^N$, weighted by B.

2.2. Minimization problem

This paper aims at solving the following general form of strongly convex optimization problem:

$$\text{Find} \;\; \widetilde{x} = \underset{x\in\mathbb{R}^N}{\operatorname{argmin}} \;\; G(x) + \frac{1}{2}\|x-\widetilde{x}\|^2, \tag{2}$$

where G is a proper lower-semicontinuous convex functions from $\mathbb{R}^N$ to $]-\infty,+\infty]$ and $\widetilde{x}$ is a given point of $\mathbb{R}^N$. Note that this optimization task is also equivalent to computing $\operatorname{prox}_{I,G}(\widetilde{x})$. In accordance with many variational models used in image processing [29, 30], we will assume that function G can be split as follows:

$$(\forall x \in \mathbb{R}^N) \quad G(x) = \sum_{j=1}^{J} g_j(A_j x), \tag{3}$$

where, for every $j \in \{1,\dots,J\}$, $g_j : \mathbb{R}^{M_j} \to]-\infty,+\infty]$ is a proper lower-semicontinuous convex possibly nonsmooth function and A_j is a real-valued matrix of dimension $M_j \times N$. In addition, we will need the technical assumption that $\bigcap_{j=1}^{J} \operatorname{dom}(g_j \circ A_j)$ is nonempty.

A number of primal-dual algorithms [8, 12, 13, 14] can be applied to Problem (2) by making use of its dual formulation.

3. ASYNCHRONOUS DUAL FORWARD-BACKWARD SCHEME

3.1. Consensus formulation

Efficient distributed schemes can be obtained by resorting to a global consensus technique [31, 14, 24, 2] and rewriting the problem under the following form:

$$\text{Find} \;\; \widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}=(x^j)_{1\leqslant j\leqslant J}\in\Lambda}{\operatorname{argmin}} \;\; \sum_{j=1}^{J} g_j(A_j x^j) + \frac{1}{2}\sum_{j=1}^{J}\omega_j\|x^j-\widetilde{x}\|^2, \tag{4}$$

Algorithm 1 Proposed distributed algorithm

Initialization:
$\mathbb{T}_t^* \equiv$ index set of nodes using block $t \in \{1,\dots,T\}$,
$\{\omega_{j,t} \mid 1 \leqslant j \leqslant J, t \in \mathbb{T}_j\} \subset]0,1]$ such that $(\forall t \in \{1,\dots,T\}) \sum_{j\in\mathbb{T}_t^*} \omega_{j,t} = 1$,
$y_0^j \in \mathbb{R}^{M_j}$, $(\forall t \in \mathbb{T}_j)$ $[x_0^j]_t = [\widetilde{x}]_t - \omega_{j,t}^{-1}\mathcal{A}_{j,t}^\top y_0^j$,
$\epsilon \in]0,1]$ and $(\forall \ell \in \{1,\dots,L\})$ $\vartheta_\ell = \min_{j\in\mathbb{V}_\ell, t\in\mathbb{T}_j} \omega_{j,t}$.
Main loop:
for $n = 0,1,\dots$ **do**
  $\gamma_n \in [\epsilon, 2-\epsilon]$
  $j_n \in \{1,\dots,J+L\}$
  if $j_n \leqslant J$ **then**
    Local optimization:
    $\widetilde{y}_n^{j_n} = y_n^{j_n} + \gamma_n B_{j_n}^{-1} \sum_{t\in\mathbb{T}_{j_n}} \mathcal{A}_{j_n,t}[x_n^{j_n}]_t$
    $y_{n+1}^{j_n} = \widetilde{y}_n^{j_n} - \gamma_n B_{j_n}^{-1} \operatorname{prox}_{\gamma_n B_{j_n}^{-1}, g_{j_n}}\left(\gamma_n^{-1} B_{j_n}\widetilde{y}_n^{j_n}\right)$
    $y_{n+1}^j = y_n^j, \quad j \in \{1,\dots,J\}\setminus\{j_n\}$
    for $t \in \mathbb{T}_{j_n}$ **do**
      $[x_{n+1}^{j_n}]_t = [x_n^{j_n}]_t - \omega_{j_n,t}^{-1}\mathcal{A}_{j_n,t}^\top(y_{n+1}^{j_n} - y_n^{j_n})$
    end for
    $([x_{n+1}^j]_t)_{t\in\mathbb{T}_j} = ([x_n^j]_t)_{t\in\mathbb{T}_j}, \quad j \in \{1,\dots,J\}\setminus\{j_n\}$
  else
    Projection:
    $\ell_n = j_n - J$
    $y_{n+1}^j = y_n^j, \quad j \in \{1,\dots,J\}$
    for $j \in \mathbb{V}_{\ell_n}$ **do**
      for $t \in \mathbb{T}_j$ **do**
        $[x_{n+1}^j]_t = [x_n^j]_t +$
        $\gamma_n\vartheta_{\ell_n}\omega_{j,t}^{-1}\left(\operatorname{mean}\left(([x_n^{j'}]_t)_{j'\in\mathbb{V}_{\ell_n}\cap\mathbb{T}_t^*}\right) - [x_n^j]_t\right)$
      end for
    end for
    $([x_{n+1}^j]_t)_{t\in\mathbb{T}_j} = ([x_n^j]_t)_{t\in\mathbb{T}_j}, \quad j \notin \mathbb{V}_{\ell_n}$.
  end if
end for

where $(\omega_j)_{1\leqslant j\leqslant J} \in]0,1]^J$ are such that $\sum_{j=1}^{J}\omega_j = 1$. Essentially, the global variable x has been replaced by a collection of vectors $(x^j)_{1\leqslant j\leqslant J}$, each one being associated with one of the terms in (3). In this reformulation, Λ is a vector subspace of $\mathbb{R}^{NJ}$ defined so as to introduce suitable coupling constraints on the latter variables.

The second ingredient of our distributed approach is that it relies on a hypergraph formulation. Each node $j \in \{1,\dots,J\}$ is associated with the function g_j, which is considered local and processes its own private data. Moreover, each node j is allowed to communicate with nodes that belong to the same group $\mathbb{V}_\ell$ with $\ell \in \{1,\dots,L\}$. The sets $(\mathbb{V}_\ell)_{1\leqslant\ell\leqslant L}$ can then be viewed as the hyperedges of the hypergraph (the standard graph topology is recovered when every $\mathbb{V}_\ell$ has only 2 elements). In this context, the constraint space Λ is decomposed in a collection of constraint spaces $(\Lambda_\ell)_{1\leqslant \ell\leqslant L}$ of lower dimension in such a way that, for every $\boldsymbol{x} = [(x^1)^\top, \ldots, (x^J)^\top]^\top \in \mathbb{R}^{NJ}$,

$$\boldsymbol{x} \in \Lambda \quad \Leftrightarrow \quad (\forall \ell \in \{1,\ldots,L\}) \quad (x^j)_{j\in\mathbb{V}_\ell} \in \Lambda_\ell. \tag{5}$$

3.2. Proposed distributed strategy

When the operators $(A_j)_{1\leqslant j\leqslant J}$ have no specific structure, the DBFB algorithm from [22, 27] leads to a distributed algorithm that we proposed as preliminary work in [32]. This method however requires each node of the hypergraph to handle a local copy of an N-dimensional variable, which may be prohibitive for highly dimensional problems. Hopefully, as is often the case in image processing problems, the operators $(A_j)_{1\leqslant j\leqslant J}$ have a sparse block structure (like gradient operators), which makes it possible to alleviate this problem.

More specifically, we assume in the sequel that

$$(\forall j \in \{1,\ldots,J\})\big(\forall x^j = ([x^j]_t)_{1\leqslant t\leqslant T} \in \mathbb{R}^N\big) \\ A_j x^j = \sum_{t\in\mathbb{T}_j} \mathcal{A}_{j,t}[x^j]_t \tag{6}$$

where, for every $j \in \{1,\ldots,J\}$, $[x^j]_t$ is a vector corresponding to a block of data of dimension L, T is the overall number of blocks (i.e., $N = TL$), and $\mathbb{T}_j \subset \{1,\ldots,T\}$ defines the reduced index subset of the components of vector x^j acting on the operator A_j. In the above equation, $(\mathcal{A}_{j,t})_{t\in\mathbb{T}_j}$ are the associated reduced-size matrices of dimensions $M_j \times L$. To avoid degenerate cases, we will assume that the sets $(\mathbb{T}_j)_{1\leqslant j\leqslant J}$ are nonempty and their union is equal to $\{1,\ldots,T\}$.

The specific form of the operators $(A_j)_{1\leqslant j\leqslant J}$ shows that it is necessary to define spaces Λ and $(\Lambda_\ell)_{1\leqslant \ell\leqslant L}$ in order to reach a consensus only for the components $([x^j]_t)_{1\leqslant j\leqslant J, t\in\mathbb{T}_j}$ of interest in vectors $(x^j)_{1\leqslant j\leqslant J}$. Under such a consensus constraint, Problem (4) reduces to the minimization, with respect to $([x^j]_t)_{1\leqslant j\leqslant J, t\in\mathbb{T}_j}$, of

$$\sum_{j=1}^J g_j\Big(\sum_{t\in\mathbb{T}_j}\mathcal{A}_{j,t}[x^j]_t\Big) + \frac{1}{2}\sum_{j=1}^J\sum_{t\in\mathbb{T}_j}\omega_{j,t}\big\|[x^j]_t - [\widetilde{x}]_t\big\|^2, \tag{7}$$

where $(\omega_{j,t})_{1\leqslant j\leqslant J, t\in\mathbb{T}_j}$ are positive constants. The application of the DBFB minimization method to the resolution of the above constrained problem yields Algorithm 1. Like its synchronous counterpart, Algorithm 1 benefits from the acceleration provided by variable metric methods through the introduction of preconditioning matrices $(B_j)_{1\leqslant j\leqslant J}$ in $\mathbb{R}^{M_j\times M_j}$ that must satisfy the majorization rule $B_j \succeq \sum_{t\in\mathbb{T}_j}\omega_{j,t}^{-1}\mathcal{A}_{j,t}\mathcal{A}_{j,t}^\top$. Under this condition, convergence guarantees on both the generated primal sequences $(x_n)_{n\in\mathbb{N}}$ and dual sequences $(y_n^j)_{n\in\mathbb{N}}$ with $j \in \{1,\ldots,J\}$ can be deduced from [22] under mild requirements on the sequence $(j_n)_{n\in\mathbb{N}}$ defining the variables to be updated at each iteration.

3.3. Case of practical interest

We now focus on the practical case when only $C \leqslant J$ processing units are available. An interesting instance of Algorithm 1 is when $L = C+1$ and when each hyperedge $\mathbb{V}_\ell$ with $\ell \in \{1,\ldots,C\}$ corresponds to a given computing unit where the computations are locally synchronized. Finally, the last hyperedge $\mathbb{V}_L$ is set to $\{1,\ldots,J\}$ in order to model the global synchronization steps. These consist of an averaging over all the available nodes. Then, at each iteration n, only a subset of the dual variable indices may be activated within the ℓ-th hyperedge. This update is followed by either local or global synchronization.

4. APPLICATION TO VIDEO DENOISING

4.1. Problem statement

In this section, we provide a validation of the proposed distributed algorithm for the denoising of video sequences. The original sequence $\overline{x} = ([\overline{x}]_t)_{1\leqslant t\leqslant T} \in \mathbb{R}^{TL}$ is naturally decomposed in T blocks of data, each corresponding to one frame containing L pixels (hence, $TL = N$). The sought sequence $\overline{x}$ is corrupted by additive zero-mean white Gaussian noise, so yielding the observed noisy sequence $\widetilde{x} = ([\widetilde{x}]_t)_{1\leqslant t\leqslant T} \in \mathbb{R}^{TL}$. An estimate of the unknown video can be inferred by solving Problem (2) where $J = T$. The last quadratic term in (2) is a least squares data fidelity term, and functions $(g_j)_{1\leqslant j\leqslant T}$ stand for regularization functions that incorporate both temporal and spatial prior knowledge on each video frame. For every $t \in \{1,\ldots T\}$, the regularization function $g_t\colon \mathbb{R}^{M_t} \to [0,+\infty[$ reads:

$$g_t\left(([x]_{t'})_{t'\in\mathbb{T}_t}\right) = \eta\ \mathrm{tgv}([x]_t) + \iota_{[x_{\min},x_{\max}]^L}([x]_t) \\ + h_t\left(([x]_{t'})_{t'\in\mathbb{T}_t}\right) \tag{8}$$

with, for every $t \in \{1,\ldots,T\}$), $\mathbb{T}_t = \big\{\max\{t-1,1\}, t, \min\{t+1,T\}\big\}$. Hereabove, $\iota_{[x_{\min},x_{\max}]^L}$ designates the indicator function of $[x_{\min},x_{\max}]^L$, equal to zero over this interval, and $+\infty$ everywhere else. tgv denotes the *Total Generalized Variation* regularization from [33]. Moreover, h_t is a temporal regularization term that takes into account motion estimation between consecutive frames similarly to the strategy used in [27].

4.2. Distributed implementation

We employ our proposed asynchronous distributed framework to address this denoising problem. The functions $(g_t)_{1\leqslant t\leqslant T}$ and their associated primal variables $([x^t]_{t'})_{t'\in\mathbb{T}_t}$ for $t \in \{1,\ldots,T\}$, are spread over C computing units, each of them handling the same number κ of nodes (i.e. $T = \kappa C$). The associated hyperedges are then given by: $(\forall c \in \{1,\ldots,C\})\ \mathbb{V}_c = \{(c-1)\kappa+1,\ldots,c\kappa\}$. In our case, at each local synchronization step, the update of each frame requires the averaging of at most 3 images. During the global synchronization step, local sums are transmitted to the next computing unit to compute an average which is sent back to the current unit. This workflow is illustrated by Figure 1 when $\kappa = 3$. In practice, we only activate global synchronization every 4 iterations. This frequency was chosen in order to achieve a good trade-off between the communication overhead and a satisfactory convergence speed.

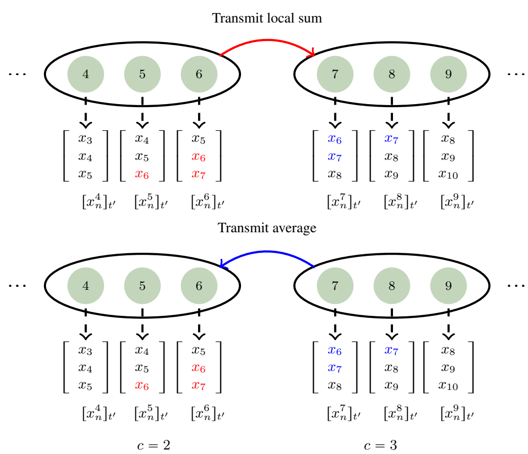

Fig. 1: Illustration of the global synchronization process between computing unit $c = 2$ and computing unit $c = 3$.

4.3. Simulation results

We now evaluate the performance of the proposed denoising method on the standard test video sequences **Foreman**, and **Irene**, composed of $T = 72$ frames of size 348×284 and 352×288 respectively, so that N is more than 6×10^7. The initial SNR (signal-to-noise ratio) values are 24.41 dB for the first sequence and 25.51 dB for the second. Our method is implemented in the Julia-0.4.6 language and a *Message Passing Interface* (MPI) wrapper for managing communication between cores [34, 35]. We use a multi-core architecture using 2 Intel(R) Xeon(R) E5-2670 v3 CPU @ 2.3 GHz processors, each with 12 cores, hence $C = 24$. We evaluate the proposed distributed approach in terms of restoration quality and acceleration provided by our algorithm with respect to the number of cores. The images composing the video sequences are partitioned in groups of equal size κ processed by the computing units.

Figure 2 displays the speedup in execution time with respect to the number of cores. The speedup is initially greater than 1:1 as the number of cores increases from 1 to 12. This effect is due to the fact that the cache memory increases with each new processor. This reduces main RAM access and lowers the global execution time despite the communication overhead. However, this speedup starts to saturate after 12 cores (as in Amdahl's law [36]). This may be due to the fact that inter CPU exchanges, rather than inter-core start to dominate. The former are significantly slower.

Figure 3 shows some frames extracted from the degraded and restored sequences, which allows us to evaluate the good visual quality of the performed denoising. The SNRs of the restored sequences are equal to 32.04 dB and 30.97 dB, respectively.

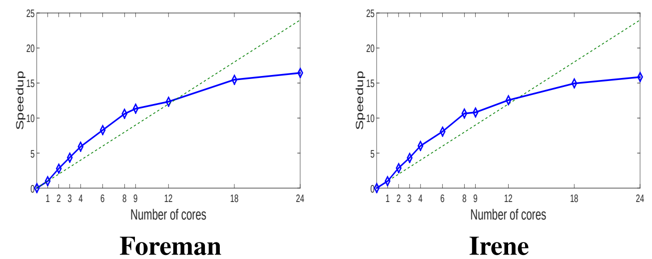

Fig. 2: Speedup with respect to the number of used cores: proposed method (solid, blue, diamond), linear speedup (dashed, green).

Fig. 3: Example of input degraded images (top) and associated restored images (bottom) for **Foreman** (left) and **Irene** (right) sequences.

5. CONCLUSION

In this paper, we introduced a distributed version of the preconditioned dual block-coordinate forward-backward algorithm for minimizing a strongly convex function. We mainly focused on an instance of the proposed approach when the involved linear operators have a block sparse structure. The experimental results we obtained for video denoising are quite promising and demonstrate the ability of our algorithm to take advantage of multiple cores. In future work, it would be interesting to evaluate the proposed optimization strategy on more distributed parallel computing systems.

6. REFERENCES

[1] P. L. Combettes and J.-C. Pesquet, “Proximal splitting methods in signal processing,” in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, H. H. Bauschke, R. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, Eds., pp. 185–212. Springer-Verlag, New York, 2010.

[2] N. Komodakis and J.-C. Pesquet, “Playing with duality : An overview of recent primal-dual approaches for solving large-scale optimization problems,” *IEEE Signal Process. Mag.*, vol. 32, no. 6, pp. 31–54, Nov. 2015.

[3] P. L. Combettes, D. Dung, and B. C. Vũ, “Dualization of signal recovery problems,” *Set-Valued Var. Anal.*, vol. 18, no. 3, pp. 373–404, Dec. 2010.

[4] L. Condat, “A primal-dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms,” *J. Optim. Theory App.*, vol. 158, no. 2, pp. 460–479, Aug. 2013.

[5] S. R. Becker and P. L. Combettes, “An algorithm for splitting parallel sums of linearly composed monotone operators, with applications to signal recovery,” *J. Nonlinear Convex Anal.*, vol. 15, no. 1, pp. 137–159, Jan. 2014.

[6] C. Couprie, L. Grady, L. Najman, J.-C Pesquet, and H. Talbot, “Dual constrained tv-based regularization on graphs,” *SIAM J. Imaging Sci.*, vol. 6, pp. 1246–1273, Jul. 2013.

[7] A. Jezierska, E. Chouzenoux, J.-C. Pesquet, and H. Talbot, “A primal-dual proximal splitting approach for restoring data corrupted with poisson-gaussian noise,” in *IEEE Int. Conf. Acoust. Speech and Signal Process. (ICASSP 2012)*, Kyoto, Japan, 25-30 Mar. 2012, pp. 1085–1088.

[8] A. Chambolle and T. Pock, “A first-order primal-dual algorithm for convex problems with applications to imaging,” *J. Math. Imag. Vision*, vol. 40, no. 1, pp. 120–145, May 2010.

[9] 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 *IEEE Int. Conf. Image Process. (ICIP 2014)*, Paris, France, 27-30 Oct. 2014, pp. 4141–4145.

[10] 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 Vision*, vol. 49, no. 3, pp. 551–568, Jul. 2014.

[11] 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.*, vol. 20, no. 2, pp. 307–330, Jun. 2012.

[12] R. I. Boţ and C. Hendrich, “A Douglas-Rachford type primal-dual method for solving inclusions with mixtures of composite and parallel-sum type monotone operators,” *SIAM J. Optim.*, vol. 23, no. 4, pp. 2541–2565, Dec. 2013.

[13] P. L. Combettes and J.-C. Pesquet, “Stochastic quasi-Fejér block-coordinate fixed point iterations with random sweeping,” *SIAM J. Optim.*, vol. 25, no. 2, pp. 1221–1248, Jul. 2015.

[14] 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.*, vol. 8, no. 1, pp. 1–122, Jan. 2011.

[15] L. M. Briceño Arias and P. L. Combettes, “A monotone+skew splitting model for composite monotone inclusions in duality,” *SIAM J. Optim.*, vol. 21, no. 4, pp. 1230–1250, Oct. 2011.

[16] D. Davis, “Convergence rate analysis of primal-dual splitting schemes,” *SIAM J. Optim.*, vol. 25, no. 3, pp. 1912–1943, Sep. 2015.

[17] L. Rosasco, S. Villa, and B. C. Vũ, “A first-order stochastic primal-dual algorithm with correction step,” *Numer. Funct. Anal. Optim.*, vol. 38, no. 5, pp. 602–626, Apr. 2017.

[18] S. Shalev-Shwartz and T. Zhang, “Stochastic dual coordinate ascent methods for regularized loss minimization,” *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 567–599, Feb. 2013.

[19] P. Bianchi, W. Hachem, and F. Iutzeler, “A coordinate descent primal-dual algorithm and application to distributed asynchronous optimization,” *IEEE Trans. Autom. Control*, vol. 61, no. 10, pp. 2947–2957, Oct. 2016.

[20] Z. Qu, P. Richtárik, and T. Zhang, “Randomized dual coordinate ascent with arbitrary sampling,” in *Adv. Neural Inf. Process. Syst. (NIPS 2015)*, Montréal, Canada, 7-12 Dec. 2015, pp. 865–873.

[21] M. Jaggi, V. Smith, M. Takac, J. Terhorst, S. Krishnan, T. Hofmann, and M. I. Jordan, “Communication-efficient distributed dual coordinate ascent,” in *Adv. Neural Inf. Process. Syst.*, Z. Ghahramani, M. Welling, C. Cortes, N.d. Lawrence, and K.q. Weinberger, Eds., pp. 3068–3076. Curran Associates, Inc., Sept. 2014.

[22] F. Abboud, E. Chouzenoux, J.-C. Pesquet, J.-H. Chenot, and L. Laborelli, “A dual block coordinate proximal algorithm with application to deconvolution of interlaced video sequences,” in *IEEE Int. Conf. Image Process. (ICIP 2015)*, Quebéc City, Canada, 27-30 Sep. 2015, pp. 4917–4921.

[23] A. Chambolle and T. Pock, “A remark on accelerated block coordinate descent for computing the proximity operators of a sum of convex functions,” *SIAM J. Comput. Math.*, vol. 1, pp. 29–57, 2015.

[24] J.-C. Pesquet and A. Repetti, “A class of randomized primal-dual algorithms for distributed optimization,” *J. Nonlinear Convex Anal.*, vol. 16, no. 12, pp. 2453–2490, Dec. 2015.

[25] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem, “Explicit convergence rate of a distributed alternating direction method of multipliers,” *IEEE Trans. Autom. Control*, vol. 61, no. 4, pp. 892–904, Apr. 2016.

[26] P. Richtárik and M. Takác, “Distributed coordinate descent method for learning with big data,” *J. Mach. Learn. Res.*, vol. 17, no. 75, pp. 1–25, Feb. 2016.

[27] F. Abboud, E. Chouzenoux, J.-C. Pesquet, J.-H. Chenot, and L. Laborelli, “Dual block coordinate forward-backward algorithm with application to deconvolution and deinterlacing of video sequences,” *J. Math. Imaging Vision*, vol. 59, no. 3, pp. 415–431, Nov. 2017.

[28] J. J. Moreau, “Proximité et dualité dans un espace hilbertien,” *Bull. Soc. Math. France*, vol. 93, pp. 273–299, 1965.

[29] E. Chouzenoux, A. Jezierska, J.-C. Pesquet, and H. Talbot, “A convex approach for image restoration with exact Poisson-Gaussian likelihood,” *SIAM Journal on Imaging Science*, vol. 8, no. 4, pp. 2662–2682, 2015.

[30] N. Pustelnik, A. Benazza-Benhayia, Y. Zheng, and J.-C. Pesquet, *Wavelet-based Image Deconvolution and Reconstruction*, John Wiley & Son, Feb. 2016.

[31] N. Pustelnik, C. Chaux, and J.-C. Pesquet, “Parallel proximal algorithm for image restoration using hybrid regularization,” *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2450–2462, Sep. 2011.

[32] F. Abboud, E. Chouzenoux, J.-C. Pesquet, J.-H. Chenot, and L. Laborelli, “A distributed strategy for computing proximity operators,” in *In Proceedings of 50th Asilomar Conference on Signals, Systems, and Computers (ASILOMAR 2015)*, Asilomar, Pacific Grove, CA, 8-11 Nov. 2015, pp. 396–400.

[33] K. Bredies, K. Kunisch, and T. Pock, “Total generalized variation,” *SIAM J. Imaging Sci.*, vol. 3, no. 3, pp. 492–526, Sep. 2010.

[34] Message Passing Interface Forum, “MPI: A Message-Passing Interface standard,” Tech. Rep., 1994.

[35] W. Gropp, E. Lusk, and A. Skjellum, *Using MPI: Portable Paral lel Programming with the Message Passing Interface*, MIT Press, 2 edition, 1999.

[36] G. M. Amdahl, “Validity of the single processor approach to achieving large scale computing capabilities,” in *American Fed. Inform. Process. Soc. (AFIPS 1967)*, Atlantic City, USA, 18-20 Apr. 1967, pp. 483–485.