Distributed Proximal Splitting Algorithms with Rates and Acceleration

Laurent Condat HTTPS://LCONDAT.GITHUB.IO/
Grigory Malinovsky
Peter Richtárik HTTPS://RICHTARIK.ORG/
King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia

Abstract

We propose new generic distributed proximal splitting algorithms, well suited for large-scale convex nonsmooth optimization. We derive sublinear and linear convergence results with new nonergodic rates, as well as new accelerated versions of the algorithms, using varying stepsizes.

1. Introduction

We propose new algorithms for the generic convex optimization problem[1]:

$$\underset{x\in\mathcal{X}}{\text{minimize}} \left\{ \Psi(x) := \frac{1}{M}\sum_{m=1}^{M}\Big(F_m(x) + H_m(K_m x)\Big) + R(x) \right\}, \tag{1}$$

where $M \geq 1$ is typically the number of parallel computing nodes in a distributed setting; the $K_m : \mathcal{X} \to \mathcal{U}_m$ are linear operators; $\mathcal{X}$ and $\mathcal{U}_m$ are real Hilbert spaces of finite dimension; R and H_m are proper, closed, convex functions with values in $\mathbb{R}\cup\{+\infty\}$, the proximity operators of which are easy to compute; and the F_m are convex L_{F_m}-smooth functions; that is, ∇F_m is L_{F_m}-Lipschitz continuous, for some $L_{F_m} > 0$. For any function G, we denote by $\mu_G \geq 0$ some constant such that G is μ_G-strongly convex; that is, $G - (\mu_G/2)\|\cdot\|^2$ is convex.

The template problem (1) covers most convex optimization problems met in machine learning, signal and image processing, operations research, control, and many other fields, and our goal is to propose new generic distributed algorithms able to deal with nonsmooth functions using their proximity operators, with acceleration in presence of strong convexity. This is important for distributed or federated learning [16, 18].

Proximal splitting algorithms [1, 2, 7, 10, 15, 20] are particularly well suited for large-scale convex nonsmooth optimization. They are generally designed as sequential algorithms, for $M = 1$, and then they can be extended by lifting in product spaces to parallel versions, suitable to minimize $F + R + \sum_m H_m \circ K_m$, see e.g. [10, Section 8]. However, it is not straightforward to adapt lifting to the case of a finite-sum $F = \frac{1}{M}\sum_m F_m$, with each function F_m handled by a different node, which is of primary importance in machine learning. This generalization is one of our contributions.

Also, there is a vast literature on distributed optimization to minimize $\frac{1}{M}\sum_m F_m + R$, with a focus on strategies based on (block-)coordinate or randomized activation, as well as replacing the

1. Our only assumption is that there exists a solution $x^\star \in \mathcal{X}$ such that $0 \in \partial R(x^\star) + \frac{1}{M}\sum_m \nabla F_m(x^\star) + K_m^* \partial H_m(K_m x^\star)$; see for instance [8, Proposition 4.3] for sufficient conditions on the functions for this property to hold.

gradients by cheaper stochastic estimates [3, 14, 21, 22]. Replacing the full gradient by a stochastic oracle in the accelerated algorithms with varying stepsizes we propose is not straightforward; we leave this direction for future research. In any case, the generalized setting, with the smooth functions F_m at the nodes supplemented or replaced by nonsmooth functions H_m, possibly composed with linear operators, seems to have received very little attention. We want to make up for that.

Our contributions are the following. They are detailed and illustrated by numerical experiments in the long version of the paper [11].

1. **New algorithms.** We propose the first distributed algorithms to solve (1) in whole generality, with proved convergence to an exact solution, and having the *full splitting*, or decoupling, property: ∇F_m, prox_{H_m}, K_m and K_m^* are applied at the m-th node, and the proximity operator of R is applied at the master node connected to all others. No other more complicated operation, like an inner loop or a linear system to solve, is involved.

2. **Unified framework.** The foundation of our distributed algorithms consists in 2 general principles, applied in a cascade, which are new contributions in themselves and could be used in other contexts:

   (a) We show that problem (1) with $M = 1$, i.e. the minimization of $F + R + H \circ K$, can be reformulated as the minimization of $\widetilde{F} + \widetilde{R} + \widetilde{H}$ in a different space, with preserved smoothness and strong convexity properties. Hence, the linear operator disappears and the Davis–Yin algorithm [12] can be applied to this new problem. Through this lens, we recover many algorithms as particular cases of this unified framework, like the PD3O, Chambolle–Pock, Loris–Verhoeven algorithms.

   (b) We design a non-straightforward lifting technique, so that the problem (1), with any M, is reformulated as the minimization of $\widehat{F} + \widehat{R} + \widehat{H} \circ \widehat{K}$ in some product space.

3. **New convergence analysis and acceleration.** Even when $M = 1$, we improve upon the state of the art in two ways:

   (a) For constant stepsizes, we recover existing algorithms, but we provide new, more precise, results about their convergence speed.

   (b) With a particular strategy of varying stepsizes, we exhibit new algorithms, which are accelerated versions of them. Even for the existing accelerated Chambolle–Pock algorithm [4, 5], which we recover as a particular case but with a different stepsize rule, our $O(1/k^2)$ rate is on the last iterate, improving upon the best known result, which is ergodic [5].

2. Deriving the Nonstationary PD3O and PDDY Algorithms

Let us first focus on the case $M = 1$; that is, we consider minimizing $F(x) + R(x) + H(Kx)$. The dual problem is to minimize $(F + R)^*(-K^*u) + H^*(u)$, where K^* is the adjoint operator of K and G^* denotes the convex conjugate of a function G.

Algorithm 1 Distributed PD3O Alg.

input: $(\gamma_k)_{k\in\mathbb{N}}$, $\eta \geq \|\widehat{K}\|^2$, $(\omega)_{m=1}^M$, $(q_m^0)_{m=1}^M \in \mathcal{X}^M$, $(u_m^0)_{m=1}^M \in \mathcal{U}^M$
initialize: $a_m^0 := q_m^0 - K_m^* u_m^0$, $m = 1...M$
for $k = 0, 1, \ldots$ **do**
  at master, **do**
    $x^{k+1} := \operatorname{prox}_{\gamma_k R}\big(\frac{\gamma_k}{M}\sum_{m=1}^M a_m^k\big)$
    broadcast x^{k+1} to all nodes
  at all nodes, for $m = 1, \ldots, M$, **do**
    $q_m^{k+1} := \frac{M\omega_m}{\gamma_{k+1}} x^{k+1} - \nabla F_m(x^{k+1})$
    $u_m^{k+1} := \operatorname{prox}_{M\omega_m H_m^*/(\gamma_{k+1}\eta)}\big(u_m^k + \frac{1}{\eta}K_m(\frac{M\omega_m}{\gamma_k}x^{k+1} + q_m^{k+1} - q_m^k)\big)$
    $a_m^{k+1} := q_m^{k+1} - K_m^* u_m^{k+1}$
    transmit a_m^{k+1} to master
end for

Algorithm 2 Distributed PDDY Alg.

input: $(\gamma_k)_{k\in\mathbb{N}}$, $\eta \geq \|\widehat{K}\|^2$, $(\omega_m)_{m=1}^M$, $x_R^0 \in \mathcal{X}$, $(u_m^0) \in \mathcal{U}^M$
initialize: $p_m^0 := K_m^* u_m^0$, $m = 1, ..., M$
for $k = 0, 1, \ldots$ **do**
  at all nodes, for $m = 1, \ldots, M$, **do**
    $u_m^{k+1} := \operatorname{prox}_{M\omega_m H_m^*/(\gamma_k\eta)}\big(u_m^k + \frac{M\omega_m}{\gamma_k\eta}K_m x_R^k\big)$
    $p_m^{k+1} := K_m^* u_m^{k+1}$
    $x_m^{k+1} := x_R^k - \frac{\gamma_k}{M\omega_m}(p_m^{k+1} - p_m^k)$
    $a_m^k := M\omega_m x_m^{k+1} - \gamma_{k+1}\nabla F_m(x_m^{k+1}) - \gamma_{k+1}p_m^{k+1}$
    transmit a_m^k to master
  at master, **do**
    $x_R^{k+1} := \operatorname{prox}_{\gamma_{k+1}R}\big(\frac{1}{M}\sum_{m=1}^M a_m^k\big)$
    broadcast x_R^{k+1} to all nodes
end for

If $K = I$, the Davis–Yin algorithm [12] is well suited to minimize the sum of three functions[2]. To make this algorithm applicable to $K \neq I$, we reformulate the problem as follows:

1) Choose $\eta \geq \|K\|^2$; we recommend to set $\eta = \|K\|^2$ in practice. Then there exists a real Hilbert space $\mathcal{W}$ and a linear operator $C : \mathcal{W} \to \mathcal{U}$ such that $KK^* + CC^* = \eta I$. C is not unique, for instance, $C = (\eta I - KK^*)^{1/2}$. We actually don't need to exhibit C, its existence is sufficient here!

2) Then the problem can be rewritten as:

$$\underset{x\in\mathcal{X}, w\in\mathcal{W}}{\text{minimize}}\ \widetilde{F}(x,w) + \widetilde{R}(x,w) + \widetilde{H}(x,w), \tag{2}$$

where $\widetilde{F} : (x,w) \mapsto F(x) + \frac{\mu_F}{2}\|w\|^2$, $\widetilde{R} : (x,w) \mapsto R(x) + \imath_0(w)$, where $\imath_0 : w \mapsto \{0 \text{ if } w = 0, +\infty \text{ else}\}$, and $\widetilde{H} : (x,w) = H(Kx+Cw)$. We have $\nabla\widetilde{F}(x,w) = (\nabla F(x), \mu_F w)$, $\operatorname{prox}_{\widetilde{R}}(x,w) = (\operatorname{prox}_R(x), 0)$. Importantly, for every $\gamma > 0$, we have [19]: $\operatorname{prox}_{\widetilde{H}^*/\gamma}(x,w) = (K^*u, C^*u)$, where $u = \operatorname{prox}_{H^*/(\gamma\eta)}\big((Kx + Cw)/\eta\big)$. Note that in [19], the authors use $\widetilde{F}(x,w) = F(x)$, whereas we add $\frac{\mu_F}{2}\|w\|^2$. This difference is essential, so that $\widetilde{F}$ is L_F-smooth and μ_F-strongly convex. Also, $\widetilde{R}$ is μ_R-strongly convex.

Then, we can apply the Davis–Yin algorithm to solve the problem (2). When we write the algorithm, some simplifications are made naturally; most notably, whenever CC^* appears, it is replaced by $\eta I - KK^*$. It turns out that the obtained algorithm is not new but is a nonstationary version of the PD3O algorithm [23]. On the other hand, if we exchange the roles of the two functions $\widetilde{H}$ and $\widetilde{R}$, we obtain a different algorithm: it is a nonstationary version of the PDDY algorithm proposed recently [22]. With constant stepsizes $\gamma_k \equiv \gamma \in (0, 2/L_F)$, for both the PD3O and PDDY algorithms, x^k and u^k converge to some primal and dual solutions $x^\star$ and $u^\star$, respectively.

2. There is a minor mistake in the way Algorithm 3 in [12] is initialized. This has been corrected in this work.

Particular cases of the PD3O and PDDY algorithms are the following:

1. If $K = I$ and $\eta = 1$, the PD3O algorithm reverts to the Davis–Yin algorithm; the PDDY algorithm too, but with H and R exchanged.
2. If $F = 0$, the PD3O and PDDY algorithms revert to the forms I and II [10] of the Chambolle–Pock algorithm [4], respectively.
3. If $R = 0$, the PD3O and PDDY algorithms revert to the Loris–Verhoeven algorithm [17], also discovered independently as the PDFP2O [6] and PAPC [13] algorithms; see also [9, 10] for an analysis as a primal-dual forward-backward algorithm.
4. If $F = 0$ in the Davis–Yin algorithm or $K = I$ and $\eta = 1$ in the Chambolle–Pock algorithm, we obtain the Douglas–Rachford algorihm; it is equivalent to the ADMM, see the discussion in [10].
5. If $H = 0$, with $L = I$ and $\eta = 1$, the Davis–Yin algorithm reverts to the forward–backward algorithm, a.k.a. proximal gradient descent. The Loris–Verhoeven algorithm with $L = I$ and $\eta = 1$, too.

To derive distributed versions of the algorithms, to solve (1) for any value of M, we developed a specific lifting technique in product spaces, which we don't detail here by lack of space. Let $(\omega_m)_{m=1}^M$ be a sequence of positive weights, whose sum is 1; they can be used to mitigate different $\|K_m\|$, by setting $\omega_m \propto 1/\|K_m\|^2$, or different L_{F_m}, by setting $\omega_m \propto L_{F_m}^2$, as a rule of thumb. So, we recast the minimization of $R(x) + \frac{1}{M}\sum_{m=1}^M \big(F_m(x) + H_m(K_m x)\big)$ as the minimization of $\widehat{R}(\hat{x}) + \widehat{F}(\hat{x}) + \widehat{H}(\widehat{K}\hat{x})$, for appropriate 'hat'-terms, which are defined using the weights ω_m. The constants $L_{\widehat{F}}$ and $\|\widehat{K}\|$ have closed forms.

3. Convergence Analysis

Theorem B.1 (convergence rate of the Distributed PD3O Algorithm) *In the Distributed PD3O Algorithm, suppose that $\gamma_k \equiv \gamma \in (0, 2/L_{\widehat{F}})$; if $F_m \equiv 0$, we can choose any $\gamma > 0$. Also, suppose that $\eta \geq \|\widehat{K}\|^2$. Then x^k converges to some solution $x^\star$ of (1). Also, u_m^k converges to some element $u_m^\star \in \mathcal{U}_m$, for every $m = 1, \ldots, M$. In addition, suppose that every H_m is continuous on a ball around $K_m x^\star$. Then the following hold:*

$$\text{(i)} \quad \Psi(x^k) - \Psi(x^\star) = o(1/\sqrt{k}). \tag{3}$$

Define the weighted ergodic iterate $\bar{x}^k = \frac{2}{k(k+1)}\sum_{i=1}^k i x^i$, for every $k \geq 1$. Then

$$\text{(ii)} \quad \Psi(\bar{x}^k) - \Psi(x^\star) = O(1/k). \tag{4}$$

Furthermore, if every H_m is L_m-smooth for some $L_m > 0$, we have a faster decay for the best iterate so far:

$$\text{(iii)} \quad \min_{i=1,\ldots,k} \Psi(x^i) - \Psi(x^\star) = o(1/k). \tag{5}$$

We now give accelerated convergence results using varying stepsizes, in presence of strong convexity. For this, we define $\mu_{\widehat{F}}$ as the strong convexity constant of the average function $\frac{1}{M}\sum_{m=1}^M F_m$. That is, $\mu_{\widehat{F}} \geq 0$ is such that the function $x \in \mathcal{X} \mapsto \frac{1}{M}\sum_{m=1}^M F_m(x) - \frac{\mu_{\widehat{F}}}{2}\|x\|^2$ is convex.

Theorem B.2 (Accelerated Distributed PD3O Algorithm) *Suppose that $\mu_{\widehat{F}} + \mu_R > 0$. Let $x^\star$ be the unique solution to (1). Let $\kappa \in (0,1)$ and $\gamma_0 \in (0, 2(1-\kappa)/L_{\widehat{F}})$. Set $\gamma_1 = \gamma_0$ and*

$$\gamma_{k+1} = \frac{-\gamma_k^2 \mu_{\widehat{F}} \kappa + \gamma_k \sqrt{(\gamma_k \mu_{\widehat{F}} \kappa)^2 + 1 + 2\gamma_k \mu_R}}{1 + 2\gamma_k \mu_R}, \quad \text{for every } k \geq 1. \tag{6}$$

Then in the Distributed PD3O Algorithm, there exists $\hat{c}_0 > 0$ such that, for every $k \geq 1$, $\|x^{k+1} - x^\star\|^2 \leq \frac{\gamma_{k+1}^2}{1-\gamma_{k+1}\mu_{\widehat{F}}\kappa} \hat{c}_0 = O\big(1/k^2\big)$.

Theorem B.4 (Accelerated Distributed PDDY Algorithm) *Suppose that $\mu_{\widehat{F}} > 0$. Let $x^\star$ be the unique solution to (1). Let $\kappa \in (0,1)$ and $\gamma_0 \in (0, 2(1-\kappa)/L_{\widehat{F}})$. Set $\gamma_1 = \gamma_0$ and*

$$\gamma_{k+1} = -\gamma_k^2 \mu_{\widehat{F}} \kappa + \gamma_k \sqrt{(\gamma_k \mu_{\widehat{F}} \kappa)^2 + 1}, \quad \text{for every } k \geq 1. \tag{7}$$

Then in the Distributed PDDY Algorithm, there exists $\hat{c}_0 > 0$ such that, for every $k \geq 1$,

$$\sum_{m=1}^{M} \omega_m \|x_m^{k+1} - x^\star\|^2 \leq \frac{\gamma_{k+1}^2}{1 - \gamma_{k+1}\mu_F \kappa} c_0 = O\big(1/k^2\big). \tag{8}$$

Consequently, for every $m = 1, \ldots, M$, $\|x_m^k - x^\star\|^2 = O\big(1/k^2\big)$.

Theorem B.3 (linear convergence of the Distributed PD3O Algorithm) *Suppose that $\mu_{\widehat{F}} + \mu_R > 0$ and that every H_m is L_m-smooth, for some $L_m > 0$. Let $x^\star$ be the unique solution to (1). We assume constant stepsizes: $\gamma_k \equiv \gamma$, for some $\gamma \in (0, 2/L_{\widehat{F}})$. Then the Distributed PD3O Algorithm converges linearly: there exists $\rho \in (0,1]$ and $\hat{c}_0 > 0$ such that, for every $k \in \mathbb{N}$, $\|x^{k+1} - x^\star\|^2 \leq (1-\rho)^k \hat{c}_0$.*

Theorem B.5 (linear convergence of the Distributed PDDY Algorithm) *Suppose that $\mu_{\widehat{F}} + \mu_R > 0$ and that every H_m is L_m-smooth, for some $L_m > 0$. Let $x^\star$ be the unique solution to (1). We assume constant stepsizes: $\gamma_k \equiv \gamma$, for some $\gamma \in (0, 2/L_{\widehat{F}})$. Then the Distributed PDDY Algorithm converges linearly: there exists $\rho \in (0,1]$ and $\hat{c}_0 > 0$ such that, for every $k \in \mathbb{N}$, $\|x_R^{k+1} - x^\star\|^2 \leq (1-\rho)^k \hat{c}_0$.*

Lower bounds for ρ in Theorems B.3 and B.5 can be derived from Theorem D.6 in the preprint version of [12]. We emphasize that linear convergence comes *for free* with the algorithms, if the conditions are met, without any modification. That is, there is no need to know the values of the strong convexity constants, since the conditions on the two parameters γ and η do not depend on them.

References

[1] A. Beck. *First-Order Methods in Optimization*. MOS-SIAM Series on Optimization. SIAM, 2017.

[2] R. I. Boţ, E. R. Csetnek, and C. Hendrich. Recent developments on primal–dual splitting methods with applications to convex minimization. In P. M. Pardalos and T. M. Rassias, editors, *Mathematics Without Boundaries: Surveys in Interdisciplinary Research*, pages 57–99. Springer New York, 2014.

[3] V. Cevher, S. Becker, and M. Schmidt. Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics. *IEEE Signal Process. Mag.*, 31(5):32–43, 2014.

[4] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vision*, 40(1):120–145, May 2011.

[5] A. Chambolle and T. Pock. On the ergodic convergence rates of a first-order primal–dual algorithm. *Math. Program.*, 159(1–2):253–287, September 2016.

[6] 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), 2013.

[7] P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In H. H. Bauschke, R. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, editors, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Springer-Verlag, New York, 2010.

[8] 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-Val. Var. Anal.*, 20(2):307–330, 2012.

[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 *Proc. of IEEE ICIP*, Paris, France, October 2014.

[10] L. Condat, D. Kitahara, A. Contreras, and A. Hirabayashi. Proximal splitting algorithms: A tour of recent advances, with new twists. preprint arXiv:1912.00137, 2019.

[11] L. Condat, G. Malinovsky, and P. Richtárik. Distributed proximal splitting algorithms with rates and acceleration. preprint arXiv:2010.00952, 2020.

[12] D. Davis and W. Yin. A three-operator splitting scheme and its optimization applications. *Set-Val. Var. Anal.*, 25:829–858, 2017.

[13] Y. Drori, S. Sabach, and M. Teboulle. A simple algorithm for a class of nonsmooth convex concave saddle-point problems. *Oper. Res. Lett.*, 43(2):209–214, 2015.

[14] E. Gorbunov, F. Hanzely, and P. Richtárik. A unified theory of SGD: Variance reduction, sampling, quantization and coordinate descent. In *Proc. of Int. Conf. Artif. Intell. Stat. (AISTATS)*, Palermo, Sicily, Italy, June 2020. to appear.

[15] 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.*, 32(6): 31–54, November 2015.

[16] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. Federated learning: Strategies for improving communication efficiency. In *NIPS Private Multi-Party Machine Learning Workshop*, 2016. arXiv preprint arXiv:1610.05492.

[17] I. Loris and C. Verhoeven. On a generalization of the iterative soft-thresholding algorithm for the case of non-separable penalty. *Inverse Problems*, 27(12), 2011.

[18] G. Malinovsky, D. Kovalev, E. Gasanov, L. Condat, and P. Richtárik. From local SGD to local fixed point methods for federated learning. In *Proc. of 37th Int. Conf. Machine Learning (ICML)*, 2020.

[19] D. O'Connor and L. Vandenberghe. On the equivalence of the primal-dual hybrid gradient method and Douglas–Rachford splitting. *Math. Program.*, 79:85–108, 2020.

[20] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 3(1): 127–239, 2014.

[21] P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Math. Program.*, 144(1–2):1–38, April 2014.

[22] A. Salim, L. Condat, K. Mishchenko, and P. Richtárik. Dualize, split, randomize: Fast nonsmooth optimization algorithms. preprint arXiv:2004.02635, 2020.

[23] M. Yan. A new Primal–Dual algorithm for minimizing the sum of three functions with a linear operator. *J. Sci. Comput.*, 76(3):1698–1717, September 2018.