RandProx: Primal–Dual Optimization Algorithms with Randomized Proximal Updates

Laurent Condat LAURENT.CONDAT@KAUST.EDU.SA
Peter Richtárik PETER.RICHTARIK@KAUST.EDU.SA
King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia

Abstract

Proximal splitting algorithms are well suited for large-scale nonsmooth optimization problems. We propose a primal–dual algorithm, in which the dual update is randomized, with the proximity operator of one of the function replaced by a stochastic oracle. A nonsmooth variance-reduction technique is implemented so that the algorithm finds an exact minimizer of the general problem. We derive linear convergence results in presence of strong convexity. Several existing randomized algorithms, like Point-SAGA, are recovered as particular cases. Randomness helps getting faster algorithms; this has long been known for stochastic-gradient-type algorithms, and our work shows that this fully applies in the more general primal–dual setting as well.

1. Introduction

Large-dimensional optimization problems arise virtually in all fields, including machine learning, data science, statistics [4, 13, 15, 17, 37, 62, 65, 72, 73]. When a function is smooth, an optimization algorithm typically makes calls to its **gradient**, whereas for a nonsmooth function, its **proximity operator** is called instead. Algorithms making use of proximity operators are called proximal (splitting) algorithms [63]. Over the past 10 years or so, several primal–dual proximal algorithms have been developed a [7, 10, 20, 21, 27, 29, 50, 63]. However, these deterministic algorithms are often too slow. Stochastic Gradient Descent (**SGD**)-type methods [11, 38, 40, 47, 61, 67], with the gradient of smooth functions replaced by stochastic estimates, are the striking demonstration that randomized algorithms can be cheaper than deterministic ones. But replacing the proximity operator of a function by a **stochastic proximity operator** estimate is a nearly virgin territory. This is important, because many functions of practical interest have a costly proximity operator. We can mention the nuclear norm of matrices, which requires singular value decompositions, indicator functions of sets on which it is difficult to project, or optimal transport costs [64].

In this paper, we propose RandProx (Algorithm 2), a randomized version of the Primal–Dual Davis–Yin (PDDY) method (Algorithm 1), which a proximal algorithm proposed recently [29, 70]. In RandProx, one proximity operator that appears in the PDDY algorithm is replaced by a stochastic estimate. RandProx is **variance-reduced** [38, 40, 42]; that is, by making use of control variates, the algorithm converges to an exact solution, just like its deterministic counterpart. We prove linear convergence of RandProx in the strongly convex setting, with additional results in the convex setting. We mention relationships between our results and related works in the literature throughout the paper. In special cases, RandProx reduces to Point-SAGA [32], the Stochastic Decoupling Method [58], ProxSkip, SplitSkip and Scaffnew [60], and randomized versions of the PAPC [34], PDHG [16] and ADMM [12] algorithms. They are all generalized and unified within our new framework. Thus, just like Point-SAGA [32] is the proximal counterpart of SAGA [33], our generic algorithm RandProx paves the way to a new world of proximal counterparts of variance-reduced SGD-type algorithms.

1.1. Problem formulation and proximal algorithms

Let $\mathcal{X}$ and $\mathcal{U}$ be finite-dimensional real Hilbert spaces. We consider the convex optimization problem:

$$\text{Find } x^\star \in \arg\min_{x\in\mathcal{X}} \Big(f(x) + g(x) + h(Kx)\Big), \tag{1}$$

where $K : \mathcal{X} \to \mathcal{U}$ is a nonzero linear operator; f is a convex L_f-smooth function, for some $L_f > 0$; that is, its gradient ∇f is L_f-Lipschitz continuous [6, Definition 1.47]; and $g : \mathcal{X} \to \mathbb{R} \cup \{+\infty\}$ and $h : \mathcal{U} \to \mathbb{R} \cup \{+\infty\}$ are proper closed convex functions whose proximity operator is easy to compute. We also introduce the dual problem to (1):

$$\text{Find } u^\star \in \arg\min_{u\in\mathcal{U}} \Big((f + g)^*(-K^*u) + h^*(u)\Big), \tag{2}$$

where $K^* : \mathcal{U} \to \mathcal{X}$ is the adjoint operator of K. For these problems to be well-posed, we suppose that there exists $x^\star \in \mathcal{X}$ such that $0 \in \nabla f(x^\star) + \partial g(x^\star) + K^*\partial h(Kx^\star)$, where $\partial(\cdot)$ denotes the subdifferential [6].

We will assume strong convexity of some functions: a convex function ϕ is said to be μ_ϕ-strongly convex, for some $\mu_\phi \geq 0$, if $\phi - \frac{\mu_\phi}{2}\|\cdot\|^2$ is convex.

We recall that for any function ϕ and parameter $\gamma > 0$, the proximity operator of $\gamma\phi$ is [6]: $\mathrm{prox}_{\gamma\phi} : x \in \mathcal{X} \mapsto \arg\min_{x'\in\mathcal{X}} \big(\phi(x') + \frac{1}{2}\|x' - x\|^2\big)$. This operator has a closed form for many functions of practical interest [36, 63, 66]. In addition, the Moreau identity holds: $\mathrm{prox}_{\gamma\phi^*}(x) = x - \gamma\, \mathrm{prox}_{\phi/\gamma}(x/\gamma)$, where $\phi^* : x \in \mathcal{X} \mapsto \sup_{x'\in\mathcal{X}} \big(\langle x, x'\rangle - \phi(x')\big)$ denotes the conjugate function of ϕ [6]. Thus, one can compute the proximity operator of ϕ from the one of ϕ^*, and conversely.

Proximal splitting algorithms, such as the forward–backward and the Douglas–Rachford algorithms [6], are well suited to minimizing the sum, $f + g$ or $g + h$ in our notation, of two functions. However, many problems take the form (1) with $K \neq \mathrm{Id}$, where Id denotes the identity, and the proximity operator of $h \circ K$ is intractable in most cases. A classical example is the total variation, widely used in image processing [14, 24, 25, 68] or for regularization on graphs [30], where h is some variant of the ℓ_1 norm and K takes differences between adjacent values. Another example is when h is the indicator function of some nonempty closed convex set $\Omega \subset \mathcal{U}$; that is, $h(u) = (0$ if $u \in \Omega$, $+\infty$ otherwise), in which case the problem (1) can be rewritten as

$$\text{Find } x^\star \in \arg\min_{x\in\mathcal{X}} \Big(f(x) + g(x)\Big) \quad \text{s.t.} \quad Kx \in \Omega.$$

If $g = 0$ and $\Omega = \{b\}$ for some $b \in \mathrm{ran}(K)$, where ran denotes the range, the problem can be further rewritten as the linearly constrained smooth minimization problem $\text{Find } x^\star \in \arg\min_{x\in\mathcal{X}} f(x) \quad \text{s.t.} \quad Kx = b$. This last problem has applications in decentralized optimization [52, 69, 79], for instance. Thus, the template problem (1) covers a wide range of optimization problems met in machine learning [4, 65], signal and image processing [17, 20], control [73], and many other fields.

Algorithm 1 PDDY algorithm [70]

input: initial points $x^0 \in \mathcal{X}$, $u^0 \in \mathcal{U}$;
stepsizes $\gamma > 0$, $\tau > 0$
$v^0 := K^* u^0$
for $t = 0, 1, \ldots$ **do**
  $\hat{x}^t := \mathrm{prox}_{\gamma g}\big(x^t - \gamma \nabla f(x^t) - \gamma v^t\big)$
  $u^{t+1} := \mathrm{prox}_{\tau h^*}\big(u^t + \tau K \hat{x}^t\big)$
  $v^{t+1} := K^* u^{t+1}$
  $x^{t+1} := \hat{x}^t - \gamma(v^{t+1} - v^t)$
end for

Algorithm 2 RandProx [new]

input: initial points $x^0 \in \mathcal{X}$, $u^0 \in \mathcal{U}$;
stepsizes $\gamma > 0$, $\tau > 0$; $\omega \geq 0$
$v^0 := K^* u^0$
for $t = 0, 1, \ldots$ **do**
  $\hat{x}^t := \mathrm{prox}_{\gamma g}\big(x^t - \gamma \nabla f(x^t) - \gamma v^t\big)$
  $u^{t+1} := u^t + \frac{1}{1+\omega}\mathcal{R}^t\big(\mathrm{prox}_{\tau h^*}(u^t + \tau K \hat{x}^t) - u^t\big)$
  $v^{t+1} := K^* u^{t+1}$
  $x^{t+1} := \hat{x}^t - \gamma\,(1+\omega)\,(v^{t+1} - v^t)$
end for

2. Proposed algorithm: RandProx

There exist several deterministic algorithms for solving the problem (1); see Condat et al. [27] for a recent overview. In this work, we focus on the PDDY algorithm (Algorithm 1) [29, 70]. In particular, our new algorithm RandProx (Algorithm 2) generalizes the PDDY algorithm with a stochastic estimate of the proximity operator of h^*.

We recall the general convergence result for the PDDY algorithm [29, Theorem 2]: *If* $\gamma \in (0, 2/L_f)$, $\tau > 0$, $\tau\gamma\|K\|^2 \leq 1$, *then* $(x^t)_{t\in\mathbb{N}}$ *converges to a primal solution* $x^\star$ *of* (1) *and* $(u^t)_{t\in\mathbb{N}}$ *converges to a dual solution* $u^\star$ *of* (2). The PDDY algorithm is similar and closely related to the PD3O algorithm [81], as discussed in Condat et al. [29], Salim et al. [70]. We can note that the popular Condat–Vũ algorithm [23, 75] can solve the same problem but has more restrictive conditions on γ and τ.

In the PDDY algorithm, the full gradient ∇f can be replaced by a stochastic estimator which is typically cheaper to compute [70]. Convergence rates and accelerations of the PDDY algorithm, as well as distributed versions of the algorithm, have been derived in Condat et al. [29]. In particular, if $\mu_f > 0$ or $\mu_g > 0$, the primal problem (1) is strongly convex. In this case, a varying stepsize strategy accelerates the algorithm, with a $\mathcal{O}(1/t^2)$ decay of $\|x^t - x^\star\|^2$, where $x^\star$ is the unique solution to (1). But strong convexity of the primal problem is not sufficient for the PDDY algorithm to converge linearly, and additional assumptions on h and K are needed. We will prove linear convergence when both the primal and dual problems are strongly convex; this is a natural condition for primal–dual algorithms. We can note that h is L_h-smooth, for some $L_h > 0$, if and only if h^* is μ_{h^*}-strongly convex, for some $\mu_{h^*} > 0$, with $\mu_{h^*} = 1/L_h$. In that case, the dual problem (2) is strongly convex.

We propose RandProx (Algorithm 2), a generalization of the PDDY algorithm (Algorithm 1) with a randomized update of the dual variable u. Let us formalize the random operations using random variables and stochastic processes. We introduce the underlying probability space $(\mathcal{S}, \mathcal{F}, P)$. Given a real Hilbert space $\mathcal{H}$, an $\mathcal{H}$-valued random variable is a measurable map from $(\mathcal{S}, \mathcal{F})$ to $(\mathcal{H}, \mathcal{B})$, where $\mathcal{B}$ is the Borel σ-algebra of $\mathcal{H}$. Formally, randomizing some steps in the PDDY algorithm amounts to defining $\big((x^t, u^t)\big)_{t\in\mathbb{N}}$ as a stochastic process, with x^t being a $\mathcal{X}$-valued random variable and u^t a $\mathcal{U}$-valued random variable, for every $t \geq 0$. We use light notations and write our randomized algorithm RandProx using stochastic operators $\mathcal{R}^t$ on $\mathcal{U}$; that is, for every $t \geq 0$ and any $r^t \in \mathcal{U}$, $\mathcal{R}^t(r^t)$ is a $\mathcal{U}$-valued random variable, which can be interpreted as r^t plus 'random noise' (formally,

r^t is itself a $\mathcal{U}$-valued random variable, but algorithmically, $\mathcal{R}^t$ is applied to a particular outcome in $\mathcal{U}$, hence the notation as an operator on $\mathcal{U}$). To fix the ideas, let us give two examples.

Example 1. The first example is compression [1–3, 9, 28, 43, 59]: $\mathcal{U} = \mathbb{R}^d$ for some $d \geq 1$ and $\mathcal{R}^t$ is the well known `rand`-k compressor or sparsifier, with $1 \leq k < d$: $\mathcal{R}^t$ multiplies k coordinates, chosen uniformly at random, of the vector r^t by d/k and sets the other ones to zero. An application to compressed communication is discussed in Section B.3.

Example 2. The second example, discussed in Section B.1, is the Bernoulli, or coin flip, operator

$$\mathcal{R}^t : r^t \mapsto \begin{cases} \frac{1}{p} r^t & \text{with probability } p, \\ 0 & \text{with probability } 1-p, \end{cases} \tag{3}$$

for some $p > 0$. In that case, with probability $1-p$, the outcome of $\mathcal{R}^t(r^t)$ is 0 and r^t does not need to be calculated; in particular, in the RandProx algorithm, $\mathrm{prox}_{\tau h^*}$ is not called, and this is why one can expect the iteration complexity of RandProx to decrease. Thus, in this example, $\mathcal{R}^t(r^t)$ does not really consist of applying the operator $\mathcal{R}^t$ to r^t; in general, the notation $\mathcal{R}^t(r^t)$ simply denotes a stochastic estimate of r^t.

Hereafter, we denote by $\mathcal{F}_t$ the σ-algebra generated by the collection of $(\mathcal{X} \times \mathcal{U})$-valued random variables $(x^0, u^0), \ldots, (x^t, u^t)$, for every $t \geq 0$. In this work, we consider **unbiased** random estimates: for every $t \geq 0$,

$$\mathbb{E}\big[\mathcal{R}^t(r^t) \mid \mathcal{F}_t\big] = r^t,$$

where $\mathbb{E}[\cdot]$ denotes the expectation, here conditionally on $\mathcal{F}_t$, and r^t is the random variable $r^t := \mathrm{prox}_{\tau h^*}(u^t + \tau K \hat{x}^t) - u^t$, as defined by RandProx. Note that our framework is general in that for $t \neq t'$, $\mathcal{R}^t$ and $\mathcal{R}^{t'}$ need not be independent nor have the same law. In simple words, at every iteration, the randomness is new but can have a different form and depend on the past, so that the operators $\mathcal{R}^t$ can be defined dynamically on the fly in RandProx.

We characterize the operators $\mathcal{R}^t$ by their *relative variance* $\omega \geq 0$ such that, for every $t \geq 0$,

$$\mathbb{E}\left[\left\|\mathcal{R}^t(r^t) - r^t\right\|^2 \mid \mathcal{F}_t\right] \leq \omega \left\|r^t\right\|^2. \tag{4}$$

The value of ω is supposed known and is used in the RandProx algorithm. Note that $\omega = 0$ if and only if $\mathcal{R}^t = \mathrm{Id}$, in which case there is no randomness and RandProx reverts to the original deterministic PDDY algorithm. To characterize how the error on the dual variable propagates to the primal variable after applying K^*, we also introduce the relative variance $\omega_{\mathrm{ran}} \geq 0$ in the range of K^* and the offset $\zeta \in [0,1]$ such that, for every $t \geq 0$,

$$\mathbb{E}\left[\left\|K^*\big(\mathcal{R}^t(r^t) - r^t\big)\right\|^2 \mid \mathcal{F}_t\right] \leq \omega_{\mathrm{ran}} \left\|r^t\right\|^2 - \zeta \left\|K^* r^t\right\|^2. \tag{5}$$

It is easy to see that (5) holds with $\omega_{\mathrm{ran}} = \|K\|^2 \omega$ and $\zeta = 0$, so this is the default choice without particular knowledge on K^*. But in some situations, e.g. sampling like in Section B.2, a much smaller value of ω_{ran} and a positive value of ζ can be derived.

3. Convergence analysis of RandProx

Our most general result, whose proof is in the Appendix, is the following:

Theorem 1. *Suppose that $\mu_f > 0$ or $\mu_g > 0$, and that $\mu_{h^*} > 0$. In RandProx, suppose that $0 < \gamma < \frac{2}{L_f}$, $\tau > 0$, and $\gamma\tau\big((1-\zeta)\|K\|^2 + \omega_{\mathrm{ran}}\big) \leq 1$, where ω_{ran} and ζ are defined in (5). For every $t \geq 0$, define the Lyapunov function*

$$\Psi^t := \frac{1}{\gamma}\left\|x^t - x^\star\right\|^2 + (1+\omega)\left(\frac{1}{\tau} + 2\mu_{h^*}\right)\left\|u^t - u^\star\right\|^2, \tag{6}$$

where $x^\star$ and $u^\star$ are the unique solutions to (1) and (2), respectively. Then RandProx converges linearly: for every $t \geq 0$,

$$\mathbb{E}\big[\Psi^t\big] \leq c^t \Psi^0, \tag{7}$$

where

$$c := \max\left(\frac{(1-\gamma\mu_f)^2}{1+\gamma\mu_g}, \frac{(\gamma L_f - 1)^2}{1+\gamma\mu_g}, 1 - \frac{2\tau\mu_{h^*}}{(1+\omega)(1+2\tau\mu_{h^*})}\right) < 1. \tag{8}$$

Also, $(x^t)_{t\in\mathbb{N}}$ and $(\hat{x}^t)_{t\in\mathbb{N}}$ both converge to $x^\star$ and $(u^t)_{t\in\mathbb{N}}$ converges to $u^\star$, almost surely.

In the Appendix, we provide other linear convergence results, *which do not require h to be smooth*, as well as many examples of applications.

References

[1] A. Albasyoni, M. Safaryan, L. Condat, and P. Richtárik. Optimal gradient compression for distributed and federated learning. preprint arXiv:2010.03246, 2020.

[2] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic. QSGD: Communication-efficient SGD via gradient quantization and encoding. In *Proc. of 31st Conf. Neural Information Processing Systems (NIPS)*, pages 1709–1720, 2017.

[3] Dan Alistarh, Torsten Hoefler, Mikael Johansson, Sarit Khirirat, Nikola Konstantinov, and C. Renggli. The convergence of sparsified gradient methods. In *Proc. of Conf. Neural Information Processing Systems (NeurIPS)*, 2018.

[4] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Found. Trends Mach. Learn.*, 4(1):1–106, 2012.

[5] D. Basu, D. Data, C. Karakus, and S. N. Diggavi. Qsparse-Local-SGD: Distributed SGD With Quantization, Sparsification, and Local Computations. *IEEE Journal on Selected Areas in Information Theory*, 1(1):217–226, 2020.

[6] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, New York, 2nd edition, 2017.

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

[8] D. P. Bertsekas. *Convex optimization algorithms*. Athena Scientific, Belmont, MA, USA, 2015.

[9] A. Beznosikov, S. Horváth, P. Richtárik, and M. Safaryan. On biased compression for distributed learning. preprint arXiv:2002.12410, 2020.

[10] 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.

[11] Léon Bottou. Stochastic gradient descent tricks. In Grégoire Montavon, Geneviève B. Orr, and Klaus-Robert Müller, editors, *Neural Networks: Tricks of the Trade*, pages 421–436. Springer Berlin Heidelberg, Berlin, Heidelberg, 2nd edition, 2012.

[12] 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 Mach. Learn.*, 3(1): 1–122, 2011.

[13] S. Bubeck. Convex optimization: Algorithms and complexity. *Found. Trends Mach. Learn.*, 8 (3–4):231–357, 2015.

[14] V. Caselles, A. Chambolle, and M. Novaga. Total variation in imaging. In O. Scherzer, editor, *Handbook of Mathematical Methods in Imaging*, volume 1016–1057. Springer New York, New York, NY, 2011.

[15] 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.

[16] 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.

[17] A. Chambolle and T. Pock. An introduction to continuous optimization for imaging. *Acta Numerica*, 25:161–319, 2016.

[18] A. Chambolle, M. J. Ehrhardt, P. Richtárik, and C.-B. Schönlieb. Stochastic primal-dual hybrid gradient algorithm with arbitrary sampling and imaging applications. *SIAM J. Optim.*, 28(4): 2783–2808, 2018.

[19] 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.

[20] 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.

[21] P. L. Combettes and J.-C. Pesquet. Fixed point strategies in data science. *IEEE Trans. Signal Process.*, 69:3878–3905, 2021.

[22] 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.

[23] 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, 2013.

[24] L. Condat. A generic proximal algorithm for convex optimization—Application to total variation minimization. *IEEE Signal Process. Lett.*, 21(8):1054–1057, August 2014.

[25] L. Condat. Discrete total variation: New definition and minimization. *SIAM J. Imaging Sci.*, 10 (3):1258–1290, 2017.

[26] L. Condat and P. Richtárik. MURANA: A generic framework for stochastic variance-reduced optimization. In *Proc. of the Mathematical and Scientific Machine Learning (MSML) conference*, 2022.

[27] L. Condat, D. Kitahara, A. Contreras, and A. Hirabayashi. Proximal splitting algorithms for convex optimization: A tour of recent advances, with new twists. *SIAM Review*, 2022. to appear.

[28] L. Condat, K. Li, and P. Richtárik. EF-BV: A unified theory of error feedback and variance reduction mechanisms for biased and unbiased compression in distributed optimization. In *Proc. of NeurIPS*, 2022.

[29] L. Condat, G. Malinovsky, and P. Richtárik. Distributed proximal splitting algorithms with rates and acceleration. *Frontiers in Signal Processing*, 1, January 2022.

[30] C. Couprie, L. Grady, L. Najman, J.-C. Pesquet, and H. Talbot. Dual constrained TV-based regularization on graphs. *SIAM J. Imaging Sci.*, 6(3):1246–1273, 2013.

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

[32] A. Defazio. A simple practical accelerated method for finite sums. In *Proc. of 30st Conf. Neural Information Processing Systems (NIPS)*, volume 29, pages 676–684, 2016.

[33] A. Defazio, F. Bach, and S. Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Proc. of 28th Conf. Neural Information Processing Systems (NIPS)*, pages 1646–1654, 2014.

[34] 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.

[35] A. Dutta, E. H. Bergou, A. M. Abdelmoniem, C. Y. Ho, A. N. Sahu, M. Canini, and P. Kalnis. On the discrepancy between the theoretical analysis and practical implementations of compressed communication for distributed deep learning. In *Proc. of AAAI Conf. Artificial Intelligence*, pages 3817–3824, 2020.

[36] Mireille El Gheche, Giovanni Chierchia, and J.-C. Pesquet. Proximity operators of discrete information divergences. *IEEE Transactions on Information Theory*, 64(2):1092–1104, 2018.

[37] R. Glowinski, S. J. Osher, and W. Yin, editors. *Splitting Methods in Communication, Imaging, Science, and Engineering*. Springer International Publishing, 2016.

[38] E. Gorbunov, F. Hanzely, and P. Richtárik. A unified theory of SGD: Variance reduction, sampling, quantization and coordinate descent. In *Proc. of 23rd Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, 2020.

[39] E. Gorbunov, F. Hanzely, and P. Richtárik. Local SGD: Unified theory and new efficient methods. In *Proc. of 24th Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, pages 3556–3564, 2021.

[40] R. M. Gower, M. Schmidt, F. Bach, and P. Richtárik. Variance-reduced methods for machine learning. *Proc. of the IEEE*, 108(11):1968–1983, November 2020.

[41] F. Hanzely, S. Hanzely, S. Horváth, and P. Richtárik. Lower bounds and optimal algorithms for personalized federated learning. In *Proc. of Conf. Neural Information Processing Systems (NeurIPS)*, volume 33, pages 2304–2315, 2020.

[42] Filip Hanzely and Peter Richtárik. One method to rule them all: Variance reduction for data, parameters and many new methods. *preprint arXiv:1905.11266*, 2019.

[43] Samuel Horváth, Chen-Yu Ho, Ludovít Horváth, Atal Narayan Sahu, Marco Canini, and Peter Richtárik. Natural compression for distributed deep learning. preprint arXiv:1905.10988, 2019.

[44] P. Kairouz et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2), 2021.

[45] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. U. Stich, and A. T. Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In *Proc. of 37th Int. Conf. Machine Learning (ICML)*, pages 5132–5143, 2020.

[46] A. Khaled, K. Mishchenko, and P. Richtárik. Tighter theory for local SGD on identical and heterogeneous data. In *Proc. of 23rd Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, 2020.

[47] A. Khaled, O. Sebbouh, N. Loizou, R. M. Gower, and P. Richtárik. Unified analysis of stochastic gradient methods for composite convex and smooth optimization. arXiv:2006.11573, 2020.

[48] Ahmed Khaled and Peter Richtárik. Gradient descent with compressed iterates. In *NeurIPS Workshop on Federated Learning for Data Privacy and Confidentiality*, 2019.

[49] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. First analysis of local GD on heterogeneous data. In *NeurIPS Workshop on Federated Learning for Data Privacy and Confidentiality*, 2019.

[50] 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.

[51] 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:1610.05492.

[52] Dmitry Kovalev, Adil Salim, and Peter Richtárik. Optimal and practical algorithms for smooth and strongly convex decentralized optimization. In *Proc. of Conf. on Neural Information Processing Systems (NeurIPS)*, 2020.

[53] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 3(37):50–60, 2020.

[54] 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.

[55] D. R. Luke and R. Shefi. A globally linearly convergent method for pointwise quadratically supportable convex-concave saddle point problems. *J. Math. Anal. Appl.*, 457:1568–1590, 2018.

[56] 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.

[57] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proc. of Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, volume PMLR 54, 2017.

[58] K. Mishchenko and P. Richtárik. A stochastic decoupling method for minimizing the sum of smooth and non-smooth functions. preprint arXiv:1905.11535v2, 2019.

[59] K. Mishchenko, E. Gorbunov, M. Takáč, and P. Richtárik. Distributed learning with compressed gradient differences. arXiv:1901.09269, 2019.

[60] K. Mishchenko, G. Malinovsky, S. Stich, and P. Richtárik. ProxSkip: Yes! Local Gradient Steps Provably Lead to Communication Acceleration! Finally! In *Proc. of the 39th International Conference on Machine Learning (ICML)*, July 2022.

[61] Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4): 1574–1609, 2009.

[62] D. P. Palomar and Y. C. Eldar, editors. *Convex Optimization in Signal Processing and Communications*. Cambridge University Press, 2009.

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

[64] G. Peyré and M. Cuturi. Computational optimal transport: With applications to data science. *Foundations and Trends in Machine Learning*, 11(5–6):355–607, 2019.

[65] N. G. Polson, J. G. Scott, and B. T. Willard. Proximal algorithms in statistics and machine learning. *Statist. Sci.*, 30(4):559–581, 2015.

[66] N. Pustelnik and L. Condat. Proximity operator of a sum of functions; application to depth map estimation. *IEEE Signal Process. Lett.*, 24(12):1827–1831, December 2017.

[67] Herbert Robbins and Sutton Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

[68] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1–4):259–268, 1992.

[69] A. Salim, L. Condat, D. Kovalev, and P. Richtárik. An optimal algorithm for strongly convex minimization under affine constraints. In *Proc. of Int. Conf. Artif. Intell. Stat. (AISTATS), PMLR 151*, pages 4482–4498, 2022.

[70] A. Salim, L. Condat, K. Mishchenko, and P. Richtárik. Dualize, split, randomize: Toward fast nonsmooth optimization algorithms. *J. Optim. Theory Appl.*, July 2022.

[71] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek. Robust and communication-efficient federated learning from non-i.i.d. data. *IEEE Trans. Neural Networks and Learning Systems*, 31(9):3400–3413, 2020.

[72] S. Sra, S. Nowozin, and S. J. Wright. *Optimization for Machine Learning*. The MIT Press, 2011.

[73] G. Stathopoulos, H. Shukla, A. Szucs, Y. Pu, and C. N. Jones. Operator splitting methods in control. *Foundations and Trends in Systems and Control*, 3(3):249–362, 2016.

[74] S. U. Stich. Local SGD converges fast and communicates little. In *Proc. of International Conference on Learning Representations (ICLR)*, 2019.

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

[76] J. Wangni, J. Wang, J. Liu, and T. Zhang. Gradient sparsification for communication-efficient distributed optimization. In *Proc. of 32nd Conf. Neural Information Processing Systems (NeurIPS)*, pages 1306–1316, 2018.

[77] W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, and H. Li. TernGrad: Ternary gradients to reduce communication in distributed deep learning. In *Proc. of 31st Conf. Neural Information Processing Systems (NIPS)*, pages 1509–1519, 2017.

[78] B. E. Woodworth, K. K. Patel, and N. Srebro. Minibatch vs Local SGD for heterogeneous distributed learning. In *Proc. of Conf. Neural Information Processing Systems (NeurIPS)*, 2020.

[79] R. Xin, S. Pu, A. Nedić, and U. A. Khan. A general framework for decentralized optimization with first-order methods. *Proceedings of the IEEE*, 108(11):1869–1889, November 2020.

[80] H. Xu, C.-Y. Ho, A. M. Abdelmoniem, A. Dutta, E. H. Bergou, K. Karatsenidis, M. Canini, and P. Kalnis. GRACE: A compressed communication framework for distributed machine learning. In *Proc. of 41st IEEE Int. Conf. Distributed Computing Systems (ICDCS)*, 2021.

[81] 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.

Table 1: The different particular cases of the problem (1) for which we derive an instance of RandProx, with the number of the theorem where its linear convergence is stated, and the corresponding condition on h and K. λ is a shorthand notation for $\lambda_{\min}(KK^*)$ and $\imath_{\{b\}} : x \mapsto (0 \text{ if } x = b, +\infty \text{ otherwise})$.

f	g	h	K	Deterministic algorithm	Randomized algorithm	Theorem	Condition ensuring linear convergence
any	any	any	any	PDDY	RandProx	1	$\mu_{h^*} > 0$
any	0	any	any	PAPC	RandProx	2	$\mu_{h^*} > 0$ or $\lambda > 0$
any	0	any	Id	forward-backward (FB)	RandProx-FB	3	—
any	0	$\imath_{\{b\}}$	any	PAPC	RandProx-LC	4	—
0	any	any	any	Chambolle–Pock (CP)	RandProx-CP	7	$\mu_{h^*} > 0$
0	any	any	Id	ADMM	RandProx-ADMM	8	$\mu_{h^*} > 0$
any	any	any	Id	Davis–Yin (DY)	RandProx-DY	9	$\mu_{h^*} > 0$

Appendix

Appendix A. More linear convergence results

Remark 1 (choice of τ, in the conditions of Theorem 1) Given γ, the rate c in (8) is smallest if τ is largest. So, there seems to be no reason to take $\tau\gamma\big((1-\zeta)\|K\|^2 + \omega_{\mathrm{ran}}\big) < 1$, and $\tau\gamma\big((1-\zeta)\|K\|^2 + \omega_{\mathrm{ran}}\big) = 1$ should be the best choice in most cases. Thus, one can set $\tau = \frac{1}{\gamma((1-\zeta)\|K\|^2+\omega_{\mathrm{ran}})}$ and keep γ as the only parameter to tune in RandProx.

In the rest of this section, we discuss some particular cases of (1), for which we derive stronger convergence guarantees than in Theorem 1 for RandProx. Other particular cases are studied in the Appendix; for instance, an instance of RandProx, called RandProx-ADMM, is a randomized version of the popular ADMM [12]. The different particular cases are summarized in Table 1.

A.1. Particular case $g = 0$

In this section, we assume that $g = 0$. Then the PDDY algorithm becomes an algorithm proposed for least-squares problems [54] and rediscovered independently as the PDFP2O algorithm [19] and as the Proximal Alternating Predictor-Corrector (PAPC) algorithm [34]; let us call it the PAPC algorithm. It has been shown to have a primal–dual forward–backward structure [22]. Thus, when $g = 0$, RandProx is a randomized version of the PAPC algorithm.

We can note that f^* is strongly convex, which is not the case of $(f+g)^*$ in general. Let us define $\lambda_{\min}(KK^*)$ as the smallest eigenvalue of KK^*. $\lambda_{\min}(KK^*) > 0$ if and only if $\ker(K^*) = \{0\}$, where $\ker$ denotes the kernel. If $\lambda_{\min}(KK^*) > 0$, $f^*(-K^*\cdot)$ is strongly convex. Thus, when $g = 0$, $\lambda_{\min}(KK^*) > 0$ and $\mu_{h^*} > 0$ are two sufficient conditions for the dual problem (2) to be strongly convex. We indeed get linear convergence of RandProx in that case:

Theorem 2. *Suppose that $g = 0$, $\mu_f > 0$, and that $\lambda_{\min}(KK^*) > 0$ or $\mu_{h^*} > 0$. In RandProx, suppose that $0 < \gamma < \frac{2}{L_f}$, $\tau > 0$ and $\gamma\tau\big((1-\zeta)\|K\|^2 + \omega_{\mathrm{ran}}\big) \le 1$. Then RandProx converges linearly: for every $t \ge 0$, $\mathbb{E}\big[\Psi^t\big] \le c^t \Psi^0$, where the Lyapunov function Ψ^t is defined in (6), and*

$$c := \max\left((1-\gamma\mu_f)^2, (\gamma L_f - 1)^2, 1 - \frac{2\tau\mu_{h^*} + \gamma\tau\lambda_{\min}(KK^*)}{(1+\omega)(1+2\tau\mu_{h^*})}\right) < 1. \quad (9)$$

Algorithm 3 RandProx-FB [new]

input: initial points $x^0 \in \mathcal{X}$, $u^0 \in \mathcal{X}$;
stepsize $\gamma > 0$; $\omega \geq 0$
for $t = 0, 1, \ldots$ **do**
  $\hat{x}^t := x^t - \gamma \nabla f(x^t) - \gamma u^t$
  $d^t := \mathcal{R}^t\big(\hat{x}^t - \mathrm{prox}_{\gamma(1+\omega)h}(\hat{x}^t + \gamma(1+\omega)u^t)\big)$
  $u^{t+1} := u^t + \frac{1}{\gamma(1+\omega)^2} d^t$
  $x^{t+1} := \hat{x}^t - \frac{1}{1+\omega} d^t$
end for

Algorithm 4 RandProx-LC [new]

input: initial points $x^0 \in \mathcal{X}$, $u^0 \in \mathcal{U}$;
stepsizes $\gamma > 0$, $\tau > 0$; $\omega \geq 0$
$v^0 := K^* u^0$
for $t = 0, 1, \ldots$ **do**
  $\hat{x}^t := x^t - \gamma \nabla f(x^t) - \gamma v^t$
  $u^{t+1} := u^t + \frac{\tau}{1+\omega} \mathcal{R}^t(K\hat{x}^t - b)$
  $v^{t+1} := K^* u^{t+1}$
  $x^{t+1} := \hat{x}^t - \gamma(1+\omega)(v^{t+1} - v^t)$
end for

Also, $(x^t)_{t\in\mathbb{N}}$ and $(\hat{x}^t)_{t\in\mathbb{N}}$ both converge to $x^\star$ and $(u^t)_{t\in\mathbb{N}}$ converges to $u^\star$, almost surely.

When $\mathcal{R}^t = \mathrm{Id}$ and $\omega = \omega_{\mathrm{ran}} = 0$, RandProx reverts to the PAPC algorithm. Even in this particular case, Theorem 2 proves linear convergence of the PAPC algorithm and is new. In Chen et al. [19, Theorem 3.7], linear convergence of an underrelaxed version of the algorithm was proved; underrelaxation slows down convergence. In Luke and Shefi [55], Theorem 3.1 is wrong, since it is based on the false assumption that if $\lambda_{\min}(K_i K_i^*) > 0$ for linear operators K_i, $i = 1, \ldots, p$, then $\lambda_{\min}(KK^*) > 0$, with $K : x \mapsto (K_1 x, \ldots, K_p x)$. Their theorem remains valid when $p = 1$, but their rate is complicated and worse than ours.

We now consider the even more particular case of $g = 0$ and $K = \mathrm{Id}$. Then the problems (1) and (2) consist in minimizing $f(x)+h(x)$ and $f^*(-u)+h^*(u)$, respectively. The dual problem is strongly convex and has a unique solution $u^\star = -\nabla f(x^\star)$, for any primal solution $x^\star$. By setting $\tau := 1/\gamma$ in the PAPC algorithm, with obtain the classical proximal gradient, a.k.a. forward-backward (FB), algorithm, which iterates $x^{t+1} := \mathrm{prox}_{\gamma h}\big(x^t - \gamma \nabla f(x^t)\big)$. Thus, when randomness is introduced, we set $\omega_{\mathrm{ran}} := \omega$, $\zeta := 0$ and, according to Remark 1, $\tau := \frac{1}{\gamma(1+\omega)}$ in RandProx. By noting that, for every $a > 0$, the abstract operators $\mathcal{R}^t$ and $a\mathcal{R}^t\big(\frac{1}{a}\,\cdot\,\big)$ have the same properties, we can put the constant $\gamma(1+\omega)$ outside $\mathcal{R}^t$ to simplify the algorithm, and rewrite RandProx as RandProx-FB, shown above. As a corollary of Theorem 2, we have:

Theorem 3. *Suppose that $\mu_f > 0$. In RandProx-FB, suppose that $0 < \gamma < \frac{2}{L_f}$. For every $t \geq 0$, define the Lyapunov function*

$$\Psi^t := \frac{1}{\gamma}\left\|x^t - x^\star\right\|^2 + (1+\omega)\big(\gamma(1+\omega) + 2\mu_{h^*}\big)\left\|u^t - u^\star\right\|^2, \tag{10}$$

where $x^\star$ is the unique minimizer of $f+h$ and $u^\star = -\nabla f(x^\star)$ is the unique minimizer of $f^*(-\cdot)+h^*$. Then RandProx-FB converges linearly: for every $t \geq 0$,

$$\mathbb{E}\big[\Psi^t\big] \leq c^t \Psi^0,$$

where

$$c := \max\left((1-\gamma\mu_f)^2, (\gamma L_f - 1)^2, 1 - \frac{1 + \frac{2}{\gamma}\mu_{h^*}}{(1+\omega)\big(1+\omega+\frac{2}{\gamma}\mu_{h^*}\big)}\right) < 1. \tag{11}$$

Also, $(x^t)_{t\in\mathbb{N}}$ and $(\hat{x}^t)_{t\in\mathbb{N}}$ both converge to $x^\star$ and $(u^t)_{t\in\mathbb{N}}$ converges to $u^\star$, almost surely.

It is important to note that it is not necessary to have $\mu_{h^*} > 0$ in Theorem 3. If we ignore the properties of h^*, the third factor in (11) can be replaced by its upper bound $1 - \frac{1}{(1+\omega)^2}$.

A.2. Linearly constrained smooth minimization

Let $b \in \mathrm{ran}(K)$. In this section, we consider the linearly constrained (LC) minimization problem

$$\text{Find } x^\star \in \arg\min_{x \in \mathcal{X}} f(x) \quad \text{s.t.} \quad Kx = b, \tag{12}$$

which is a particular case of (1) with $g = 0$ and $h : u \in \mathcal{U} \mapsto (0 \text{ if } u = b, +\infty \text{ otherwise})$. We have $h^* : u \in \mathcal{U} \mapsto \langle u, b\rangle$ and $\mathrm{prox}_{\tau h^*} : u \in \mathcal{U} \mapsto u - \tau b$. The dual problem to (12) is

$$\text{Find } u^\star \in \arg\min_{u \in \mathcal{U}} \Big(f^*(-K^* u) + \langle u, b\rangle \Big). \tag{13}$$

We denote by $u_0^\star$ the unique solution to (13) in $\mathrm{ran}(K)$. Then the set of solutions of (13) is the affine subspace $u_0^\star + \ker(K^*)$. Thus, the dual problem is not strongly convex, unless $\ker(K^*) = \{0\}$. Yet, we will see that strong convexity of f is sufficient to have linear convergence of RandProx, without any condition on K.

We rewrite RandProx in this setting as RandProx-LC, shown above. We observe that u^t does not appear in the argument of $\mathcal{R}^t$ any more, so that the iteration can be rewritten with the variable $v^t = K^* u^t$, and u^t can be removed if we are not interested in estimating a dual solution. In any case, we denote by $P_{\mathrm{ran}(K)}$ the orthogonal projector onto $\mathrm{ran}(K)$ and by $\lambda_{\min}^+(KK^*) > 0$ the smallest *nonzero* eigenvalue of KK^*. Then:

Theorem 4. *In the setup (12)–(13), suppose that $\mu_f > 0$. In RandProx-LC, suppose that $0 < \gamma < \frac{2}{L_f}$, $\tau > 0$ and $\gamma\tau\big((1-\zeta)\|K\|^2 + \omega_{\mathrm{ran}}\big) \leq 1$. Define the Lyapunov function, for every $t \geq 0$,*

$$\Psi^t := \frac{1}{\gamma} \left\| x^t - x^\star \right\|^2 + \frac{1+\omega}{\tau} \left\| u_0^t - u_0^\star \right\|^2, \tag{14}$$

where $u_0^t := P_{\mathrm{ran}(K)}(u^t)$ is also the unique element in $\mathrm{ran}(K)$ such that $v^t = K^* u_0^t$, $x^\star$ is the unique solution of (12) and $u_0^\star$ is the unique solution in $\mathrm{ran}(K)$ of (13). Then RandProx-LC converges linearly: for every $t \geq 0$,

$$\mathbb{E}\big[\Psi^t\big] \leq c^t \Psi^0,$$

where

$$c := \max\left((1-\gamma\mu_f)^2, (\gamma L_f - 1)^2, 1 - \frac{\gamma\tau\lambda_{\min}^+(KK^*)}{1+\omega} \right) < 1. \tag{15}$$

Also, $(x^t)_{t\in\mathbb{N}}$ and $(\hat{x}^t)_{t\in\mathbb{N}}$ both converge to $x^\star$ and $(u_0^t)_{t\in\mathbb{N}}$ converges to $u_0^\star$, almost surely.

Theorem 4 is new even for the PAPC algorithm when $\omega = 0$: its linear convergence under the stronger condition $\gamma\tau\|K\|^2 < 1$ has been shown in Salim et al. [70, Theorem 6.2], but our rate in (15) is better.

We further discuss RandProx-LC, which can be used for decentralized optimization, in the Appendix. Another example of application is when $\mathcal{X} = \mathbb{R}^d$, for some $d \geq 1$, and K is a matrix; one can solve (12) by activating one row of K chosen uniformly at random at every iteration.

Algorithm 5 RandProx-Skip [new]

input: initial points $x^0 \in \mathcal{X}$, $u^0 \in \mathcal{U}$;
stepsizes $\gamma > 0$, $\tau > 0$; $p \in (0, 1]$
$v^0 := K^* u^0$
for $t = 0, 1, \ldots$ **do**
  $\hat{x}^t := \mathrm{prox}_{\gamma g}\big(x^t - \gamma \nabla f(x^t) - \gamma v^t\big)$
  Flip a coin $\theta^t =$ (1 with probability p, 0 else)
  if $\theta^t = 1$ **then**
    $u^{t+1} := \mathrm{prox}_{\tau h^*}(u^t + \tau K \hat{x}^t)$
    $v^{t+1} := K^* u^{t+1}$
    $x^{t+1} := \hat{x}^t - \frac{\gamma}{p}(v^{t+1} - v^t)$
  else
    $u^{t+1} := u^t$, $v^{t+1} := v^t$, $x^{t+1} := \hat{x}^t$
  end if
end for

Algorithm 6 RandProx-Minibatch [new]

input: initial points $x^0 \in \mathcal{X}$, $(u_i^0)_{i=1}^n \in \mathcal{X}^n$;
stepsize $\gamma > 0$; $k \in \{1, \ldots, n\}$
$v^0 := \sum_{i=1}^n u_i^0$
for $t = 0, 1, \ldots$ **do**
  $\hat{x}^t := \mathrm{prox}_{\gamma g}\big(x^t - \gamma \nabla f(x^t) - \gamma v^t\big)$
  pick $\Omega^t \subset \{1, \ldots, n\}$ of size k unif. at random
  for $i \in \Omega^t$ **do**
    $u_i^{t+1} := \mathrm{prox}_{\frac{1}{\gamma n} h_i^*}(u_i^t + \frac{1}{\gamma n}\hat{x}^t)$
  end for
  for $i \in \{1, \ldots, n\} \backslash \Omega^t$ **do**
    $u_i^{t+1} := u_i^t$
  end for
  $v^{t+1} := \sum_{i=1}^n u_i^{t+1}$
  $x^{t+1} := \hat{x}^t - \frac{\gamma n}{k}(v^{t+1} - v^t)$
end for

Appendix B. Examples

B.1. Skipping the proximity operator

In this section, we consider the case of Bernoulli operators $\mathcal{R}^t$ defined in (3), which compute and return their argument only with probability $p > 0$. RandProx becomes RandProx-Skip, shown above. Then $\omega = \frac{1}{p} - 1$, $\omega_{\mathrm{ran}} = \|K\|^2 \omega$, and $\zeta = 0$.

If $g = 0$, RandProx-Skip reverts to the SplitSkip algorithm proposed recently [60]. Our Theorems 1 and 4 recover the same rate as given for SplitSkip in Mishchenko et al. [60, Theorem D.1], if smoothness of h is ignored. If in addition $K = \mathrm{Id}$ and $\tau = \frac{1}{\gamma(1+\omega)} = \frac{p}{\gamma}$, RandProx-Skip reverts to ProxSkip, a particular case of SplitSkip [60]. Our Theorem 3 applies to this case and allows us to exploit the possible smoothness of h in RandProx-Skip = ProxSkip, which is not the case of the results of [60]. As a practical application of our new results, let us consider *personalized federated learning (FL)* [41]: given a client-server architecture with a master and $n \geq 1$ users, each with local cost function f_i, $i = 1, \ldots, n$, the goal is to

$$\underset{(x_i)_{i=1}^n \in (\mathbb{R}^d)^n}{\mathrm{minimize}} \; \sum_{i=1}^n f_i(x_i) + \frac{\lambda}{2}\sum_{i=1}^n \|x_i - \bar{x}\|^2, \tag{16}$$

where $\bar{x} := \frac{1}{n}\sum_{i=1}^n x_i$. Each f_i is supposed L_f-smooth and μ_f-strongly convex. We set $\mathcal{X} := (\mathbb{R}^d)^n$, $f : x = (x_i)_{i=1}^n \mapsto \sum_{i=1}^n f_i(x_i)$, $h : x \mapsto \frac{\lambda}{2}\sum_{i=1}^n \|x_i - \bar{x}\|^2$. f is L_f-smooth and μ_f-strongly convex, h is λ-smooth, so that $\mu_{h^*} = \frac{1}{\lambda}$. Thus, with $\gamma = \frac{1}{L_f}$, we have in (11):

$$c \leq 1 - \min\left(\frac{\mu_f}{L_f}, \frac{1 + \frac{2L_f}{\lambda}}{\frac{1}{p}\big(\frac{1}{p} + \frac{2L_f}{\lambda}\big)}\right) < 1.$$

Hence, with $p = \frac{\sqrt{\mu_f \min(L_f,\lambda)}}{L_f}$, the communication complexity in terms of the expected number of communication rounds to reach ϵ-accuracy is $\mathcal{O}\left(\sqrt{\frac{\min(L_f,\lambda)}{\mu_f}} \log \frac{1}{\epsilon}\right)$, which is optimal [41]. This shows that in personalized FL with $\lambda < L_f$, the complexity can be decreased in comparison with non-personalized FL, which corresponds to $\lambda = +\infty$. This is achieved by properly setting p in ProxSkip, according to our new theory, which exploits the smoothness of h.

B.2. Sampling among several functions

We first remark that we can extend Problem (1) with the term $h(Kx)$ replaced by the sum $\sum_{i=1}^n h_i(K_i x)$ of $n \geq 2$ proper closed convex functions h_i composed with linear operators $K_i : \mathcal{X} \rightarrow \mathcal{U}_i$, for some real Hilbert spaces $\mathcal{U}_i$, by using the classical product-space trick: by defining $\mathcal{U} := \mathcal{U}_1 \times \cdots \mathcal{U}_n$, $h : u = (u_i)_{i=1}^n \in \mathcal{U} \mapsto \sum_{i=1}^n h_i(u_i)$, $K : x \in \mathcal{X} \mapsto (K_i x)_{i=1}^n \in \mathcal{U}$, we have $h(Kx) = \sum_{i=1}^n h_i(K_i x)$. In particular, by setting $K_i := \mathrm{Id}$ and $\mathcal{U}_i := \mathcal{X}$, we consider in this section the problem:

$$\text{Find } x^\star \in \operatorname*{arg\,min}_{x \in \mathcal{X}} \left(f(x) + g(x) + \sum_{i=1}^n h_i(x) \right). \tag{17}$$

We have $h^* : (u_i)_{i=1}^n \in \mathcal{X}^n \mapsto \sum_{i=1}^n h_i^*(u_i)$ and we suppose that every function h_i^* is μ_{h^*}-strongly convex, for some $\mu_{h^*} \geq 0$; then h^* is μ_{h^*}-strongly convex. Thus, the dual problem to (17) is

$$\text{Find } (u_i^\star)_{i=1}^n \in \operatorname*{arg\,min}_{(u_i)_{i=1}^n \in \mathcal{X}^n} \left((f+g)^* \Big(-\sum_{i=1}^n u_i \Big) + \sum_{i=1}^n h_i^*(u_i) \right). \tag{18}$$

Since $K^*K = n\mathrm{Id}$, $\|K\|^2 = n$. Now, we choose $\mathcal{R}^t$ as the rand-k sampling operator, for some $k \in \{1, \ldots, n\}$: $\mathcal{R}^t$ multiplies k elements out of the n of its argument sequence, chosen uniformly at random, by n/k and sets the other ones to zero. It is known [26, Proposition 1] that we can set

$$\omega := \frac{n}{k} - 1, \quad \omega_{\mathrm{ran}} := \frac{n(n-k)}{k(n-1)}, \quad \zeta := \frac{n-k}{k(n-1)}.$$

Note that this value of ω_{ran} is $n-1$ times smaller than the naive bound $\|K\|^2\omega = \frac{n(n-k)}{k}$. We have $(1-\zeta)\|K\|^2 + \omega_{\mathrm{ran}} = n$. RandProx in this setting, with $\tau := \frac{1}{\gamma n}$, becomes RandProx-Minibatch, shown above, and Theorem 1 yields:

Theorem 5. *Suppose that $\mu_f > 0$ or $\mu_g > 0$, and that $\mu_{h^*} > 0$. In RandProx-Minibatch, suppose that $0 < \gamma < \frac{2}{L_f}$. Define the Lyapunov function, for every $t \geq 0$,*

$$\Psi^t := \frac{1}{\gamma} \left\| x^t - x^\star \right\|^2 + \frac{n}{k} (\gamma n + 2\mu_{h^*}) \sum_{i=1}^n \left\| u_i^t - u_i^\star \right\|^2, \tag{19}$$

where $x^\star$ and $(u_i^\star)_{i=1}^n$ are the unique solutions to (17) and (18), respectively. Then RandProx-Minibatch converges linearly: for every $t \geq 0$, $\mathbb{E}\big[\Psi^t\big] \leq c^t \Psi^0$, where

$$c := \max\left(\frac{(1-\gamma\mu_f)^2}{1+\gamma\mu_g}, \frac{(\gamma L_f - 1)^2}{1+\gamma\mu_g}, 1 - \frac{2k\mu_{h^*}}{n(\gamma n + 2\mu_{h^*})} \right). \tag{20}$$

Algorithm 7 SDM [58]

input: initial points $x^0 \in \mathcal{X}$, $(u_i^0)_{i=1}^n \in \mathcal{X}^n$; stepsize $\gamma > 0$
$v^0 := \sum_{i=1}^n u_i^0$
for $t = 0, 1, \ldots$ **do**
 $\hat{x}^t := \mathrm{prox}_{\gamma g}\big(x^t - \gamma \nabla f(x^t) - \gamma v^t\big)$
 pick $i^t \in \{1, \ldots, n\}$ uniformly at random
 $x^{t+1} := \mathrm{prox}_{\gamma n h_i}(\gamma n u_{i^t}^t + \hat{x}^t)$
 $u_{i^t}^{t+1} := u_{i^t}^t + \frac{1}{\gamma n}(\hat{x}^t - x^{t+1})$
 for every $i \in \{1, \ldots, n\} \backslash \{i^t\}$, $u_i^{t+1} := u_i^t$
 $v^{t+1} := \sum_{i=1}^n u_i^{t+1}$ // $= v^t + u_{i^t}^{t+1} - u_{i^t}^t$
end for

Algorithm 8 Point-SAGA [32]

input: initial points $x^0 \in \mathcal{X}$, $(u_i^0)_{i=1}^n \in \mathcal{X}^n$; stepsize $\gamma > 0$
$v^0 := \sum_{i=1}^n u_i^0$
for $t = 0, 1, \ldots$ **do**
 $\hat{x}^t := x^t - \gamma v^t$
 pick $i^t \in \{1, \ldots, n\}$ uniformly at random
 $x^{t+1} := \mathrm{prox}_{\gamma n h_i}(\gamma n u_{i^t}^t + \hat{x}^t)$
 $u_{i^t}^{t+1} := u_{i^t}^t + \frac{1}{\gamma n}(\hat{x}^t - x^{t+1})$
 for every $i \in \{1, \ldots, n\} \backslash \{i^t\}$, $u_i^{t+1} := u_i^t$
 $v^{t+1} := \sum_{i=1}^n u_i^{t+1}$ // $= v^t + u_{i^t}^{t+1} - u_{i^t}^t$
end for

Also, $(x^t)_{t\in\mathbb{N}}$ and $(\hat{x}^t)_{t\in\mathbb{N}}$ both converge to $x^\star$ and $(u_i^t)_{t\in\mathbb{N}}$ converges to $u_i^\star$, $\forall i$, almost surely.

RandProx-Minibatch with $k = 1$ becomes the Stochastic Decoupling Method (SDM) proposed in Mishchenko and Richtárik [58], where strong convexity of g is not exploited, but similar guarantees are derived as in Theorem 5 if $\mu_g = 0$. Linear convergence of SDM is also proved in Mishchenko and Richtárik [58] in conditions related to ours in Theorems 2 and 4. Thus, RandProx-Minibatch extends SDM to larger minibatch size k and exploits possible strong convexity of g.

When $f = 0$ and $g = 0$, SDM further simplifies to Point-SAGA [32]. In that case, our results do not apply directly, since there is no strong convexity in f and g any more, but when minimizing the average of functions h_i, with each function supposed to be L-smooth and μ-strongly convex, for some $L \geq \mu > 0$, we can transfer the strong convexity to g by subtracting $\frac{\mu}{2}\|\cdot\|^2$ to each h_i and setting $g = \frac{\mu}{2}\|\cdot\|^2$. This does not change the problem and the algorithm but our Theorem 5 now applies, and with the right choice of γ, we recover the result in Defazio [32], that the asymptotic complexity of Point-SAGA to reach ϵ-accuracy is $\mathcal{O}\left(\left(n + \sqrt{\frac{nL}{\mu}}\right) \log \frac{1}{\epsilon}\right)$, which is conjectured to be optimal.

Thus, RandProx-Minibatch extends Point-SAGA to larger minibatch size and to the more general problem (17) with nonzero f or g.

When $n = 1$, there is no randomness and SDM reverts to the DY algorithm discussed in Appendix H.

B.3. Distributed and federated learning with compression

We consider in this section distributed optimization within the client-server model, with a master node communicating back and forth with $n \geq 1$ parallel workers. This is particularly relevant for federated learning (FL) [44, 51, 53, 57], where a potentially huge number of devices, with their owners' data stored on each of them, are involved in the collaborative process of training a global machine learning model. The goal is to exploit the wealth of useful information lying in the heterogeneous data stored across the devices. Communication between the devices and the distant server, which can be costly and slow, is the main bottleneck in this framework. So, it is of primary importance to devise novel algorithmic strategies, which are efficient in terms of computation and

Algorithm 9 RandProx-FL [new]

input: initial estimates $(x_i^0)_{i=1}^n \in \mathcal{X}^n$, $(u_i^0)_{i=1}^n \in \mathcal{X}^n$ such that $\sum_{i=1}^n u_i^0 = 0$; stepsize $\gamma > 0$; $\omega \geq 0$
for $t = 0, 1, \ldots$ **do**
  for $i = 1, \ldots, n$ at nodes in parallel **do**
    $\hat{x}_i^t := x_i^t - \gamma \nabla f_i(x_i^t) - \gamma u_i^t$
    $a_i^t := \mathcal{R}^t(\hat{x}_i^t)$
    // send compressed vector a_i^t to master
  end for
  $a^t := \frac{1}{n}\sum_{i=1}^n a_i^t$ // aggregation at master
  // broadcast a^t to all nodes
  for $i = 1, \ldots, n$ at nodes in parallel **do**
    $d_i^t := a_i^t - a^t$
    $u_i^{t+1} := u_i^t + \frac{1}{\gamma(1+\omega)^2} d_i^t$
    $x_i^{t+1} := \hat{x}_i^t - \frac{1}{1+\omega} d_i^t$
  end for
end for

communication complexities. A natural and widely used idea is to make use of (lossy) *compression*, to reduce the size of the communicated message [1, 2, 5, 35, 48, 71, 76, 77, 80]. Another popular idea is to make use of *local steps* [39, 45, 46, 49, 56, 57, 60, 74, 78]; that is, communication with the server does not occur at every iteration but only every few iterations, for instance communication is triggered randomly with a small probability at every iteration. Between communication rounds, the workers perform multiple local steps independently, based on their local objectives. Our proposed algorithm RandProx-FL unifies the two strategies, in the sense that depending on the choice of the randomization process $\mathcal{R}^t$, we obtain a method with local steps or with compression, or both.

Thus, we consider the problem

$$\text{Find } x^\star \in \arg\min_{x \in \mathbb{R}^d} \left(\sum_{i=1}^n f_i(x) \right), \tag{21}$$

where $d \geq 1$ is the model dimension and $n \geq 1$ is the number of parallel workers, each having its own objective function f_i. Every function $f_i : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex and L-smooth, for some $L \geq \mu > 0$. We define $\kappa := L/\mu$.

Now, we can observe that (21) can be recast as (1) with $K = \mathrm{Id}, \mathcal{U} = \mathcal{X}$, $g = 0$; that is, as the minimization of $f + h$, as studied in Section A.1, with

$$\mathcal{X} = (\mathbb{R}^d)^n, \quad f : x = (x_i)_{i=1}^n \mapsto \sum_{i=1}^n f_i(x_i), \tag{22}$$

$$h : x = (x_i)_{i=1}^n \mapsto (0 \text{ if } x_1 = \cdots = x_n,\ +\infty \text{ otherwise}). \tag{23}$$

We can note that f is μ-strongly convex and L-smooth, and $\mu_{h^*} = 0$. Making these substitutions in RandProx-FB yields RandProx-FL, a distributed algorithm well suited for FL, shown above. In

RandProx-FL, randomization takes the form of *linear* random unbiased operators $\mathcal{R}^t$ applied to the vectors sent to the server. Note that at every iteration, the same operator $\mathcal{R}^t$ is applied at every node; that is, its randomness is shared. We can easily check that RandProx-FL is an instance of RandProx-FB, because of the linearity of the $\mathcal{R}^t$ and because the property $\sum_{i=1}^n u_i^t = 0$ is maintained at every iteration. Formally, $\mathcal{R}^t$ applied as a whole in RandProx-FB consists of n copies of $\mathcal{R}^t$ applied individually at every node in RandProx-FL, that is why we keep the same notation; in particular, the value of ω is the same in both interpretations.

Interestingly, in RandProx-FL, information about the functions f_i or their gradients is never communicated and is exploited completely locally. This is ideal in terms of privacy.

As an application of Theorem 3, we obtain:

Theorem 10. *In RandProx-FL, suppose that $0 < \gamma < \frac{2}{L_f}$. Define the Lyapunov function, for every $t \geq 0$,*

$$\Psi^t \coloneqq \sum_{i=1}^n \left(\frac{1}{\gamma} \left\| x_i^t - x^\star \right\|^2 + \gamma(1+\omega)^2 \left\| u_i^t - u_i^\star \right\|^2 \right), \tag{24}$$

where $x^\star$ is the unique solution of (21) and $u_i^\star \coloneqq -\nabla f_i(x^\star)$. Then RandProx-FL converges linearly: for every $t \geq 0$, $\mathbb{E}\left[\Psi^t\right] \leq c^t \Psi^0$, where

$$c \coloneqq \max\left((1-\gamma\mu_f)^2, (\gamma L_f - 1)^2, 1 - \frac{1}{(1+\omega)^2} \right) < 1. \tag{25}$$

Also, the $(x_i^t)_{t\in\mathbb{N}}$ and $(\hat{x}_i^t)_{t\in\mathbb{N}}$ all converge to $x^\star$ and every $(u_i^t)_{t\in\mathbb{N}}$ converges to $u_i^\star$, almost surely.

If $\mathcal{R}^t$ is the Bernoulli compressor we have seen before in (3) and in Section B.1, RandProx-FL reverts to the Scaffnew algorithm proposed in Mishchenko et al. [60], which communicates at every iteration with probability $p \in (0, 1]$ and performs in average $1/p$ local steps between successive communication rounds. We have $\omega = \frac{1}{p} - 1$. The analysis of Scaffnew in Theorem 10 is the same as in Mishchenko et al. [60]. With $\gamma = \frac{1}{L}$, the iteration complexity of Scaffnew is $\mathcal{O}\big((\kappa + \frac{1}{p^2}) \log \frac{1}{\epsilon}\big)$, and since the algorithm communicates with probability p, its average communication complexity is $\mathcal{O}\big((p\kappa + \frac{1}{p}) \log \frac{1}{\epsilon}\big)$. In particular, with $p = \frac{1}{\sqrt{\kappa}}$, the average communication complexity of Scaffnew is $\mathcal{O}\big(\sqrt{\kappa} \log \frac{1}{\epsilon}\big)$.

We now propose a new algorithm with compressed communication: in RandProx-FL we choose, for every $t \geq 0$, $\mathcal{R}^t$ as the well-known `rand-`k compressor, for some $k \in \{1, \dots, d\}$: $\mathcal{R}^t$ multiplies k coordinates, chosen uniformly at random, of its vector argument by d/k and sets the other ones to zero. We have $\omega = \frac{d}{k} - 1$. The iteration complexity with $\gamma = \frac{1}{L}$ is $\mathcal{O}\big((\kappa + \frac{d^2}{k^2}) \log \frac{1}{\epsilon}\big)$ and the communication complexity, in terms of average number of floats sent by every worker to the master, is $\mathcal{O}\big((k\kappa + \frac{d^2}{k}) \log \frac{1}{\epsilon}\big)$, since k floats are sent by every worker at every iteration. Thus, by choosing $k = \lceil d/\sqrt{\kappa} \rceil$, as long as $d \geq \sqrt{\kappa}$, the communication complexity in terms of floats is $\mathcal{O}\left(d\sqrt{\kappa} \log \frac{1}{\epsilon}\right)$; this is the same as the one of Scaffnew with $\gamma = \frac{1}{L}$ and $p = \frac{1}{\sqrt{\kappa}}$, but RandProx-FL with `rand-`k compressors removes the necessity to communicate full d-dimensional vectors periodically.

Appendix C. Contraction of gradient descent

Lemma 1. *For every $\gamma > 0$, the gradient descent operator $\mathrm{Id} - \gamma\nabla f$ is c_γ-Lipschitz continuous, with $c_\gamma \coloneqq \max(1 - \gamma\mu_f, \gamma L_f - 1)$. That is, for every $(x, x') \in \mathcal{X}^2$,*

$$\|(\mathrm{Id} - \gamma\nabla f)x - (\mathrm{Id} - \gamma\nabla f)x'\| \leq c_\gamma \|x - x'\|.$$

Proof Let $(x,x') \in \mathcal{X}^2$. By cocoercivity of $\nabla f - \mu_f \mathrm{Id}$, we have [13, Lemma 3.11] $\langle \nabla f(x) - \nabla f(x'), x - x' \rangle \geq \frac{L_f \mu_f}{L_f+\mu_f}\|x-x'\|^2 + \frac{1}{L_f+\mu_f}\|\nabla f(x) - \nabla f(x')\|^2$. Hence,

$$\|(\mathrm{Id}-\gamma\nabla f)x - (\mathrm{Id}-\gamma\nabla f)x'\|^2 \leq \big(1 - \tfrac{2\gamma L_f \mu_f}{L_f+\mu_f}\big)\|x-x'\|^2 + \big(\gamma^2 - \tfrac{2\gamma}{L_f+\mu_f}\big)\|\nabla f(x) - \nabla f(x')\|^2.$$

Thus, if $\gamma \leq \frac{2}{L_f+\mu_f}$, since $\|\nabla f(x) - \nabla f(x')\| \geq \mu_f \|x-x'\|$,

$$\begin{aligned}\|(\mathrm{Id}-\gamma\nabla f)x - (\mathrm{Id}-\gamma\nabla f)x'\|^2 &\leq \Big(1 - \tfrac{2\gamma L_f \mu_f}{L_f+\mu_f} + (\gamma^2 - \tfrac{2\gamma}{L_f+\mu_f})\mu_f^2\Big)\|x-x'\|^2 \\ &= (1-\gamma\mu_f)^2\|x-x'\|^2.\end{aligned}$$

On the other hand, if $\gamma \geq \frac{2}{L_f+\mu_f}$, since $\|\nabla f(x) - \nabla f(x')\| \leq L_f \|x-x'\|$,

$$\begin{aligned}\|(\mathrm{Id}-\gamma\nabla f)x - (\mathrm{Id}-\gamma\nabla f)x'\|^2 &\leq \Big(1 - \tfrac{2\gamma L_f \mu_f}{L_f+\mu_f} + (\gamma^2 - \tfrac{2\gamma}{L_f+\mu_f})L_f^2\Big)\|x-x'\|^2 \\ &= (\gamma L_f - 1)^2\|x-x'\|^2.\end{aligned}$$

Since $\max(1-\gamma\mu_f, \gamma L_f - 1) = (1-\gamma\mu_f$ if $\gamma \leq \frac{2}{L_f+\mu_f}$, $\gamma L_f - 1$ otherwise$) \geq 0$, we arrive at the given expression of c_γ. ☐

We can note that if $\gamma < \frac{2}{L_f}$ and $\mu_f > 0$, $c_\gamma < 1$.

Appendix D. Proof of Theorem 1

Let $t \in \mathbb{N}$. Let $p^t \in \partial g(\hat{x}^t)$ be such that $\hat{x}^t = x^t - \gamma\nabla f(x^t) - \gamma p^t - \gamma K^* u^t$; p^t exists and is unique, by properties of the proximity operator. We also define $p^\star := -\nabla f(x^\star) - K^* u^\star$; we have $p^\star \in \partial g(x^\star)$. Let $q^t := p^t - \mu_g \hat{x}^t$ and $q^\star := p^\star - \mu_g x^\star$. We have $(1+\gamma\mu_g)\hat{x}^t = x^t - \gamma\nabla f(x^t) - \gamma q^t - \gamma K^* u^t$. Let $w^t := x^t - \gamma\nabla f(x^t)$ and $w^\star := x^\star - \gamma\nabla f(x^\star)$.

We define

$$\hat{u}^{t+1} := \mathrm{prox}_{\tau h^*}\big(u^t + \tau K\hat{x}^t\big).$$

Then,

$$\begin{aligned}\mathbb{E}\Big[\big\|x^{t+1} - x^\star\big\|^2 \mid \mathcal{F}_t\Big] &= \big\|\mathbb{E}\big[x^{t+1} \mid \mathcal{F}_t\big] - x^\star\big\|^2 + \mathbb{E}\Big[\big\|x^{t+1} - \mathbb{E}\big[x^{t+1} \mid \mathcal{F}_t\big]\big\|^2 \mid \mathcal{F}_t\Big] \\ &\leq \big\|\hat{x}^t - x^\star - \gamma K^*(\hat{u}^{t+1} - u^t)\big\|^2 + \gamma^2\omega_{\mathrm{ran}}\big\|\hat{u}^{t+1} - u^t\big\|^2 \\ &\quad - \gamma^2\zeta\big\|K^*(\hat{u}^{t+1} - u^t)\big\|^2 .\end{aligned}$$

Moreover,

$$\begin{aligned}
\left\|\hat{x}^t - x^\star - \gamma K^*(\hat{u}^{t+1} - u^t)\right\|^2 &= \left\|\hat{x}^t - x^\star\right\|^2 + \gamma^2 \left\|K^*(\hat{u}^{t+1} - u^t)\right\|^2 \\
&\quad - 2\gamma\langle \hat{x}^t - x^\star, K^*(\hat{u}^{t+1} - u^t)\rangle \\
&\leq (1 + \gamma\mu_g) \left\|\hat{x}^t - x^\star\right\|^2 + \gamma^2 \left\|K^*(\hat{u}^{t+1} - u^t)\right\|^2 \\
&\quad - 2\gamma\langle \hat{x}^t - x^\star, K^*(\hat{u}^{t+1} - u^\star)\rangle + 2\gamma\langle \hat{x}^t - x^\star, K^*(u^t - u^\star)\rangle \\
&= \langle w^t - w^\star - \gamma(q^t - q^\star) - \gamma K^*(u^t - u^\star), \hat{x}^t - x^\star\rangle \\
&\quad + \gamma^2 \left\|K^*(\hat{u}^{t+1} - u^t)\right\|^2 \\
&\quad - 2\gamma\langle \hat{x}^t - x^\star, K^*(\hat{u}^{t+1} - u^\star)\rangle + 2\gamma\langle \hat{x}^t - x^\star, K^*(u^t - u^\star)\rangle \\
&= -2\gamma\langle q^t - q^\star, \hat{x}^t - x^\star\rangle \\
&\quad + \langle w^t - w^\star + \gamma(q^t - q^\star) + \gamma K^*(u^t - u^\star), \hat{x}^t - x^\star\rangle \\
&\quad + \gamma^2 \left\|K^*(\hat{u}^{t+1} - u^t)\right\|^2 - 2\gamma\langle \hat{x}^t - x^\star, K^*(\hat{u}^{t+1} - u^\star)\rangle \\
&= -2\gamma\langle q^t - q^\star, \hat{x}^t - x^\star\rangle \\
&\quad + \frac{1}{1 + \gamma\mu_g}\langle w^t - w^\star + \gamma(q^t - q^\star) + \gamma K^*(u^t - u^\star), \\
&\qquad w^t - w^\star - \gamma(q^t - q^\star) - \gamma K^*(u^t - u^\star)\rangle \\
&\quad + \gamma^2 \left\|K^*(\hat{u}^{t+1} - u^t)\right\|^2 - 2\gamma\langle \hat{x}^t - x^\star, K^*(\hat{u}^{t+1} - u^\star)\rangle \\
&= -2\gamma\langle q^t - q^\star, \hat{x}^t - x^\star\rangle + \frac{1}{1 + \gamma\mu_g} \left\|w^t - w^\star\right\|^2 \\
&\quad - \frac{\gamma^2}{1 + \gamma\mu_g} \left\|q^t - q^\star + K^*(u^t - u^\star)\right\|^2 \\
&\quad + \gamma^2 \left\|K^*(\hat{u}^{t+1} - u^t)\right\|^2 - 2\gamma\langle \hat{x}^t - x^\star, K^*(\hat{u}^{t+1} - u^\star)\rangle.
\end{aligned}$$

We have $\langle q^t - q^\star, \hat{x}^t - x^\star\rangle \geq 0$. Hence,

$$\begin{aligned}
\left\|\hat{x}^t - x^\star - \gamma K^*(\hat{u}^{t+1} - u^t)\right\|^2 &\leq \frac{1}{1 + \gamma\mu_g} \left\|w^t - w^\star\right\|^2 - \frac{\gamma^2}{1 + \gamma\mu_g} \left\|q^t - q^\star + K^*(u^t - u^\star)\right\|^2 \\
&\quad + \gamma^2 \left\|K^*(\hat{u}^{t+1} - u^t)\right\|^2 - 2\gamma\langle \hat{x}^t - x^\star, K^*(\hat{u}^{t+1} - u^\star)\rangle,
\end{aligned}$$

so that

$$\begin{aligned}
\mathbb{E}\left[\left\|x^{t+1} - x^\star\right\|^2 \;\middle|\; \mathcal{F}_t\right] &\leq \frac{1}{1 + \gamma\mu_g} \left\|w^t - w^\star\right\|^2 - \frac{\gamma^2}{1 + \gamma\mu_g} \left\|q^t - q^\star + K^*(u^t - u^\star)\right\|^2 \\
&\quad + \gamma^2(1 - \zeta) \left\|K^*(\hat{u}^{t+1} - u^t)\right\|^2 - 2\gamma\langle \hat{x}^t - x^\star, K^*(\hat{u}^{t+1} - u^\star)\rangle \\
&\quad + \gamma^2\omega_{\mathrm{ran}} \left\|\hat{u}^{t+1} - u^t\right\|^2.
\end{aligned}$$

On the other hand,

$$\begin{aligned}
\mathbb{E}\Big[\big\|u^{t+1}-u^\star\big\|^2 \mid \mathcal{F}_t\Big] &\le \Big\|u^t-u^\star+\frac{1}{1+\omega}\big(\hat{u}^{t+1}-u^t\big)\Big\|^2+\frac{\omega}{(1+\omega)^2}\big\|\hat{u}^{t+1}-u^t\big\|^2\\
&=\frac{\omega^2}{(1+\omega)^2}\big\|u^t-u^\star\big\|^2+\frac{1}{(1+\omega)^2}\big\|\hat{u}^{t+1}-u^\star\big\|^2\\
&\quad+\frac{2\omega}{(1+\omega)^2}\langle u^t-u^\star,\hat{u}^{t+1}-u^\star\rangle+\frac{\omega}{(1+\omega)^2}\big\|\hat{u}^{t+1}-u^\star\big\|^2\\
&\quad+\frac{\omega}{(1+\omega)^2}\big\|u^t-u^\star\big\|^2-\frac{2\omega}{(1+\omega)^2}\langle u^t-u^\star,\hat{u}^{t+1}-u^\star\rangle\\
&=\frac{1}{1+\omega}\big\|\hat{u}^{t+1}-u^\star\big\|^2+\frac{\omega}{1+\omega}\big\|u^t-u^\star\big\|^2. \qquad (26)
\end{aligned}$$

Let $s^{t+1}\in\partial h^*(\hat{u}^{t+1})$ be such that $\hat{u}^{t+1}=u^t+\tau K\hat{x}^t-\tau s^{t+1}$; s^{t+1} exists and is unique. We also define $s^\star := Kx^\star$; we have $s^\star\in\partial h^*(u^\star)$. Therefore,

$$\begin{aligned}
\big\|\hat{u}^{t+1}-u^\star\big\|^2 &= \big\|(u^t-u^\star)+(\hat{u}^{t+1}-u^t)\big\|^2\\
&=\big\|u^t-u^\star\big\|^2+\big\|\hat{u}^{t+1}-u^t\big\|^2+2\langle u^t-u^\star,\hat{u}^{t+1}-u^t\rangle\\
&=\big\|u^t-u^\star\big\|^2+2\langle \hat{u}^{t+1}-u^\star,\hat{u}^{t+1}-u^t\rangle-\big\|\hat{u}^{t+1}-u^t\big\|^2\\
&=\big\|u^t-u^\star\big\|^2-\big\|\hat{u}^{t+1}-u^t\big\|^2+2\tau\langle \hat{u}^{t+1}-u^\star,K(\hat{x}^t-x^\star)\rangle\\
&\quad-2\tau\langle \hat{u}^{t+1}-u^\star,s^{t+1}-s^\star\rangle.
\end{aligned}$$

Hence,

$$\begin{aligned}
&\frac{1}{\gamma}\mathbb{E}\Big[\big\|x^{t+1}-x^\star\big\|^2 \mid \mathcal{F}_t\Big]+\frac{1+\omega}{\tau}\mathbb{E}\Big[\big\|u^{t+1}-u^\star\big\|^2 \mid \mathcal{F}_t\Big]\\
&\le \frac{1}{\gamma(1+\gamma\mu_g)}\big\|w^t-w^\star\big\|^2-\frac{\gamma}{1+\gamma\mu_g}\big\|q^t-q^\star+K^*(u^t-u^\star)\big\|^2\\
&\quad+\gamma(1-\zeta)\big\|K^*(\hat{u}^{t+1}-u^t)\big\|^2-2\langle \hat{x}^t-x^\star,K^*(\hat{u}^{t+1}-u^\star)\rangle\\
&\quad+\gamma\omega_{\mathrm{ran}}\big\|\hat{u}^{t+1}-u^t\big\|^2+\frac{1}{\tau}\big\|u^t-u^\star\big\|^2-\frac{1}{\tau}\big\|\hat{u}^{t+1}-u^t\big\|^2\\
&\quad+2\langle \hat{u}^{t+1}-u^\star,K(\hat{x}^t-x^\star)\rangle-2\langle \hat{u}^{t+1}-u^\star,s^{t+1}-s^\star\rangle\\
&\quad+\frac{\omega}{\tau}\big\|u^t-u^\star\big\|^2\\
&\le \frac{1}{\gamma(1+\gamma\mu_g)}\big\|w^t-w^\star\big\|^2-\frac{\gamma}{1+\gamma\mu_g}\big\|q^t-q^\star+K^*(u^t-u^\star)\big\|^2\\
&\quad+\frac{1+\omega}{\tau}\big\|u^t-u^\star\big\|^2+\left(\gamma\big((1-\zeta)\|K\|^2+\omega_{\mathrm{ran}}\big)-\frac{1}{\tau}\right)\big\|\hat{u}^{t+1}-u^t\big\|^2\\
&\quad-2\langle \hat{u}^{t+1}-u^\star,s^{t+1}-s^\star\rangle\\
&\le \frac{1}{\gamma(1+\gamma\mu_g)}\big\|w^t-w^\star\big\|^2-\frac{\gamma}{1+\gamma\mu_g}\big\|q^t-q^\star+K^*(u^t-u^\star)\big\|^2\\
&\quad+\frac{1+\omega}{\tau}\big\|u^t-u^\star\big\|^2-2\langle \hat{u}^{t+1}-u^\star,s^{t+1}-s^\star\rangle.
\end{aligned}$$

By μ_{h^*}-strong monotonicity of ∂h^*, $\langle \hat{u}^{t+1} - u^\star, s^{t+1} - s^\star \rangle \geq \mu_{h^*} \left\| \hat{u}^{t+1} - u^\star \right\|^2$, and using (26),

$$\langle \hat{u}^{t+1} - u^\star, s^{t+1} - s^\star \rangle \geq \mu_{h^*} \left((1+\omega)\mathbb{E}\Big[\left\| u^{t+1} - u^\star \right\|^2 \mid \mathcal{F}_t \Big] - \omega \left\| u^t - u^\star \right\|^2 \right).$$

Hence,

$$\begin{aligned} \frac{1}{\gamma}\mathbb{E}\Big[\left\| x^{t+1} - x^\star \right\|^2 \mid \mathcal{F}_t \Big] + (1+\omega)\left(\frac{1}{\tau} + 2\mu_{h^*} \right) \mathbb{E}\Big[\left\| u^{t+1} - u^\star \right\|^2 \mid \mathcal{F}_t \Big] \\ \leq \frac{1}{\gamma(1+\gamma\mu_g)} \left\| w^t - w^\star \right\|^2 - \frac{\gamma}{1+\gamma\mu_g} \left\| q^t - q^\star + K^*(u^t - u^\star) \right\|^2 \\ + \left(\frac{1+\omega}{\tau} + 2\omega\mu_{h^*} \right) \left\| u^t - u^\star \right\|^2. \end{aligned} \tag{27}$$

After Lemma 1,

$$\begin{aligned} \left\| w^t - w^\star \right\|^2 &= \left\| (\mathrm{Id} - \gamma\nabla f)x^t - (\mathrm{Id} - \gamma\nabla f)x^\star \right\|^2 \\ &\leq \max(1-\gamma\mu_f, \gamma L_f - 1)^2 \left\| x^t - x^\star \right\|^2. \end{aligned}$$

Plugging this inequality in (27) yields

$$\begin{aligned} \mathbb{E}\big[\Psi^{t+1} \mid \mathcal{F}_t \big] \leq \frac{1}{\gamma(1+\gamma\mu_g)} \max(1-\gamma\mu_f, \gamma L_f - 1)^2 \left\| x^t - x^\star \right\|^2 \\ + \left(\frac{1+\omega}{\tau} + 2\omega\mu_{h^*} \right) \left\| u^t - u^\star \right\|^2 - \frac{\gamma}{1+\gamma\mu_g} \left\| q^t - q^\star + K^*(u^t - u^\star) \right\|^2. \end{aligned} \tag{28}$$

Ignoring the last term in (28), we obtain:

$$\mathbb{E}\big[\Psi^{t+1} \mid \mathcal{F}_t \big] \leq \max\left(\frac{(1-\gamma\mu_f)^2}{1+\gamma\mu_g}, \frac{(\gamma L_f - 1)^2}{1+\gamma\mu_g}, 1 - \frac{2\tau\mu_{h^*}}{(1+\omega)(1+2\tau\mu_{h^*})} \right) \Psi^t. \tag{29}$$

Using the tower rule, we can unroll the recursion in (29) to obtain the unconditional expectation of Ψ^{t+1}. Since $\mathbb{E}\big[\Psi^t\big] \to 0$, we have $\mathbb{E}\Big[\left\| x^t - x^\star \right\|^2\Big] \to 0$ and $\mathbb{E}\Big[\left\| u^t - u^\star \right\|^2\Big] \to 0$. Moreover, using classical results on supermartingale convergence [8, Proposition A.4.5], it follows from (29) that $\Psi^t \to 0$ almost surely. Almost sure convergence of x^t and u^t follows. Finally, by Lipschitz continuity of ∇f, K^*, prox_g, we can upper bound $\|\hat{x}^t - x^\star\|^2$ by a linear combination of $\|x^t - x^\star\|^2$ and $\|u^t - u^\star\|^2$. It follows that $\mathbb{E}\Big[\left\| \hat{x}^t - x^\star \right\|^2\Big] \to 0$ linearly with the same rate c and that $\hat{x}^t \to x^\star$ almost surely, as well. □

Appendix E. Proof of Theorem 2

Let us go back to (28). Since $g = 0$, we have $q^t = q^\star = 0$ and $\mu_g = 0$, so that

$$\begin{aligned} \mathbb{E}\big[\Psi^{t+1} \mid \mathcal{F}_t \big] \leq \frac{1}{\gamma} \max(1-\gamma\mu_f, \gamma L_f - 1)^2 \left\| x^t - x^\star \right\|^2 + \left(\frac{1+\omega}{\tau} + 2\omega\mu_{h^*} \right) \left\| u^t - u^\star \right\|^2 \\ - \gamma \left\| K^*(u^t - u^\star) \right\|^2. \end{aligned}$$

Algorithm 10 RandPriLiCo [new]

input: initial points $x^0 \in \mathcal{X}$, $v^0 \in \operatorname{ran}(W)$;
stepsizes $\gamma > 0$, $\tau > 0$; $\omega \geq 0$
for $t = 0, 1, \dots$ **do**
  $\hat{x}^t := x^t - \gamma \nabla f(x^t) - \gamma v^t$
  $d^{t+1} := \tau \mathcal{S}^t(W\hat{x}^t - a)$
  $v^{t+1} := v^t + \frac{1}{1+\omega} d^{t+1}$
  $x^{t+1} := \hat{x}^t - \gamma d^{t+1}$
end for

We have $\left\|K^*(u^t - u^\star)\right\|^2 \geq \lambda_{\min}(KK^*) \left\|u^t - u^\star\right\|^2$. This yields

$$\begin{aligned}\mathbb{E}\left[\Psi^{t+1} \mid \mathcal{F}_t\right] &\leq \frac{1}{\gamma} \max(1 - \gamma\mu_f, \gamma L_f - 1)^2 \left\|x^t - x^\star\right\|^2 \\ &\quad + \left(\frac{1+\omega}{\tau} + 2\omega\mu_{h^*} - \gamma\lambda_{\min}(KK^*)\right) \left\|u^t - u^\star\right\|^2 \\ &\leq \max\left((1 - \gamma\mu_f)^2, (\gamma L_f - 1)^2, 1 - \frac{2\tau\mu_{h^*} + \gamma\tau\lambda_{\min}(KK^*)}{(1+\omega)(1+2\tau\mu_{h^*})}\right) \Psi^t. \end{aligned} \tag{30}$$

The end of the proof is the same as the one of Theorem 1. □

Let us add here a remark on the PAPC algorithm, which is the particular case of RandProx when $\omega = 0$, in the conditions of Theorem 2:

Remark 2 (PAPC vs. proximal gradient descent on the dual problem) If $\mu_f > 0$, f^* is μ^{-1}-smooth and L_f^{-1}-strongly convex. Then $f^* \circ -K^*$ is $\mu_f^{-1}\|K\|^2$-smooth and $L_f^{-1}\lambda_{\min}(KK^*)$-strongly convex. So, if ∇f^* is computable, one can apply the proximal gradient algorithm on the dual problem (2), which iterates $u^{t+1} = \operatorname{prox}_{\tau h^*}\big(u^t + \tau K \nabla f^*(-K^* u^t)\big)$, with $\tau \in \big(0, \frac{2\mu_f}{\|K\|^2}\big)$. If $\lambda_{\min}(KK^*) > 0$, this algorithm converges linearly: $\|u^{t+1} - u^\star\|^2 \leq c^2 \|u^t - u^\star\|^2$ with $c = \max\big(1 - \tau L_f^{-1}\lambda_{\min}(KK^*), \tau\mu_f^{-1}\|K\|^2 - 1\big)$. c is smallest with $\tau = 2/\big(\mu_f^{-1}\|K\|^2 + L_f^{-1}\lambda_{\min}(KK^*)\big)$, in which case

$$c = \frac{1 - \frac{\mu_f}{L_f}\frac{\lambda_{\min}(KK^*)}{\|K\|^2}}{1 + \frac{\mu_f}{L_f}\frac{\lambda_{\min}(KK^*)}{\|K\|^2}}.$$

This is much worse than the rate of the PAPC algorithm, since it involves the product of the condition numbers L_f/μ_f and $\|K\|^2/\lambda_{\min}(KK^*)$, instead of their maximum. This is due to calling gradients of $f^* \circ -K^*$, whereas f and K are split, or decoupled, in the PAPC algorithm.

Appendix F. Proof of Theorem 4 and further discussion

We observe that in RandProx-LC and Theorem 4, it is as if the sequence $(u_0^t)_{t\in\mathbb{N}}$ had been computed by the following iteration, initialized with $x^0 \in \mathcal{X}$ and $u_0^0 := P_{\mathrm{ran}(K)}(u^0)$:

$$\left\lfloor \begin{array}{l} \hat{x}^t := x^t - \gamma\nabla f(x^t) - \gamma v^t \\ u_0^{t+1} := u_0^t + \frac{1}{1+\omega} P_{\mathrm{ran}(K)}\mathcal{R}^t\big(\tau(K\hat{x}^t - b)\big) \\ v^{t+1} := K^* u_0^{t+1} \\ x^{t+1} := \hat{x}^t - \gamma(1+\omega)(v^{t+1} - v^t) \end{array} \right. .$$

Then we remark that this is simply the iteration of RandProx, with $\mathcal{R}^t$ replaced by $\widetilde{\mathcal{R}}^t := P_{\mathrm{ran}(K)}\mathcal{R}^t$. Since its argument $r^t = \tau(K\hat{x}^t - b)$ is always in $\mathrm{ran}(K)$, $\widetilde{\mathcal{R}}^t$ is unbiased, and we have, for every $t \geq 0$,

$$\mathbb{E}\left[\left\|\widetilde{\mathcal{R}}^t(r^t) - r^t\right\|^2 \mid \widetilde{\mathcal{F}}_t\right] \leq \mathbb{E}\left[\left\|\mathcal{R}^t(r^t) - r^t\right\|^2 \mid \widetilde{\mathcal{F}}_t\right] \leq \omega \left\|r^t\right\|^2,$$

where $\widetilde{\mathcal{F}}_t$ the σ-algebra generated by the collection of random variables $(x^0, u_0^0), \dots, (x^t, u_0^t)$. Also, ω_{ran} is unchanged. Therefore, the analysis of RandProx in Theorem 2 applies, with u^t replaced by u_0^t and $u^\star$ by $u_0^\star$. Now, for every $u \in \mathrm{ran}(K)$,

$$\|K^*u\|^2 \geq \lambda_{\min}^+(KK^*) \|u\|^2,$$

and using this lower bound in the proof of Theorem 2, with $\mu_{h^*} = 0$, we obtain Theorem 4. □

Furthermore, the constraint $Kx = b$ is equivalent to the constraint $K^*Kx = K^*b$; so, let us consider problems where we are given K^*K and not K in the first place:

Let W be a linear operator on $\mathcal{X}$, which is self-adjoint, i.e. $W^* = W$, and positive, i.e. $\langle Wx, x\rangle \geq 0$ for every $x \in \mathcal{X}$. Let $a \in \mathrm{ran}(W)$. We consider the linearly constrained minimization problem

$$\text{Find } x^\star \in \operatorname*{arg\,min}_{x\in\mathcal{X}} \; f(x) \quad \text{s.t.} \quad Wx = a. \tag{31}$$

Now, we let $\mathcal{U} := \mathcal{X}$ and $K = K^* := \sqrt{W}$, where $\sqrt{W}$ is the unique positive self-adjoint linear operator on $\mathcal{X}$ such that $\sqrt{W}\sqrt{W} = W$. Also, b is defined as the unique element in $\mathrm{ran}(W) = \mathrm{ran}(K)$ such that $\sqrt{W}b = a$. Then (31) is equivalent to (12) and the dual problem is (13). We consider the Randomized Primal Linearly Constrained minimization algorithm (RandPriLiCo), shown above. We suppose that the stochastic operators $\mathcal{S}^t$ in RandPriLiCo satisfy, for every $t \geq 0$,

$$\mathbb{E}\left[\mathcal{S}^t(r^t) \mid \widetilde{\mathcal{F}}_t\right] = r^t \quad \text{and} \quad \mathbb{E}\left[\left\|\mathcal{S}^t(r^t) - r^t\right\|^2 \mid \widetilde{\mathcal{F}}_t\right] \leq \omega \left\|r^t\right\|^2, \tag{32}$$

for some $\omega \geq 0$, where $r^t := \tau W\hat{x}^t - \tau a$.

In addition, we suppose that the $\mathcal{S}^t$ commute with $\sqrt{W}$: for every $t \geq 0$ and $x \in \mathcal{X}$,

$$\sqrt{W}\mathcal{S}^t(x) = \mathcal{S}^t(\sqrt{W}x).$$

This is satisfied with the Bernoulli operators or some linear sketching operators, for instance. Then RandPriLiCo is equivalent to RandProx-LC, with $\mathcal{S}^t$ playing the role of $\mathcal{R}^t$ and $\omega_{\mathrm{ran}} = \|W\|\omega$, $\zeta = 0$. Applying Theorem 4 with these equivalences, we obtain:

Theorem 6. *In the setting of* (31)*, suppose that* $\mu_f > 0$*. In RandPriLiCo, suppose that* $0 < \gamma < \frac{2}{L_f}$*,* $\tau > 0$ *and* $\gamma\tau\|W\|(1+\omega) \leq 1$*. Define the Lyapunov function, for every* $t \geq 0$*,*

$$\Psi^t := \frac{1}{\gamma}\left\|x^t - x^\star\right\|^2 + \frac{1+\omega}{\tau}\left\|u_0^t - u_0^\star\right\|^2, \tag{33}$$

Algorithm 11 CP algorithm [16]

input: initial points $x^0 \in \mathcal{X}$, $u^0 \in \mathcal{U}$;
stepsizes $\gamma > 0$, $\tau > 0$
$\hat{x}^0 := \mathrm{prox}_{\gamma g}\big(x^0 - \gamma K^* u^0\big)$
for $t = 0, 1, \ldots$ **do**
  $u^{t+1} := \mathrm{prox}_{\tau h^*}\big(u^t + \tau K \hat{x}^t\big)$
  // $x^{t+1} := \hat{x}^t - \gamma K^*(u^{t+1} - u^t)$
  $\hat{x}^{t+1} := \mathrm{prox}_{\gamma g}\big(\hat{x}^t - \gamma K^*(2u^{t+1} - u^t)\big)$
end for

Algorithm 12 RandProx-CP [new]

input: initial points $x^0 \in \mathcal{X}$, $u^0 \in \mathcal{U}$;
stepsizes $\gamma > 0$, $\tau > 0$; $\omega \geq 0$
$\hat{x}^0 := \mathrm{prox}_{\gamma g}\big(x^0 - \gamma K^* u^0\big)$
for $t = 0, 1, \ldots$ **do**
  $d^t := \mathcal{R}^t\big(\mathrm{prox}_{\tau h^*}(u^t + \tau K \hat{x}^t) - u^t\big)$
  $u^{t+1} := u^t + \frac{1}{1+\omega} d^t$
  // $x^{t+1} := \hat{x}^t - \gamma K^* d^t$
  $\hat{x}^{t+1} := \mathrm{prox}_{\gamma g}\big(\hat{x}^t - \gamma K^*(u^{t+1} + d^t)\big)$
end for

where u_0^t is the unique element in $\mathrm{ran}(W)$ such that $v^t = \sqrt{W} u_0^t$, $x^\star$ is the unique solution of (31) and $u_0^\star$ is the unique element in $\mathrm{ran}(W)$ such that $-\nabla f(x^\star) = \sqrt{W} u_0^\star$. Then RandPriLiCo converges linearly: for every $t \geq 0$,

$$\mathbb{E}\big[\Psi^t\big] \leq c^t \Psi^0, \tag{34}$$

where

$$c := \max\left((1 - \gamma\mu_f)^2, (\gamma L_f - 1)^2, 1 - \frac{\gamma\tau\lambda_{\min}^{+}(W)}{1+\omega}\right) < 1. \tag{35}$$

Also, $(x^t)_{t\in\mathbb{N}}$ and $(\hat{x}^t)_{t\in\mathbb{N}}$ both converge to $x^\star$ almost surely.

RandPriLiCo can be applied to decentralized optimization, like in Kovalev et al. [52], Salim et al. [69] but with randomized communication; we leave the detailed study of this setting for future work.

Appendix G. Particular case $f = 0$: randomized Chambolle–Pock algorithm

In this section, we suppose that $f = 0$. The primal problem (1) becomes:

$$\text{Find } x^\star \in \operatorname*{arg\,min}_{x\in\mathcal{X}} \Big(g(x) + h(Kx)\Big), \tag{36}$$

and the dual problem (2) becomes:

$$\text{Find } u^\star \in \operatorname*{arg\,min}_{u\in\mathcal{U}} \Big(g^*(-K^*u) + h^*(u)\Big). \tag{37}$$

The PDDY algorithm becomes the Chambolle-Pock (CP), a.k.a. PDHG, algorithm [16], shown above. RandProx can be rewritten as RandProx-CP, shown above, too. In both algorithms, the variable x^t is not needed any more and can be removed.

Since $f = 0$, $L_f > 0$ can be set arbitrarily close to zero, so that Theorem 1 can be rewritten as:

Theorem 7. *Suppose that $\mu_g > 0$ and $\mu_{h^*} > 0$. In RandProx-CP, suppose that $\gamma > 0$, $\tau > 0$, $\gamma\tau\big((1-\zeta)\|K\|^2 + \omega_{\mathrm{ran}}\big) \leq 1$. Define the Lyapunov function, for every $t \geq 0$,*

$$\Psi^t := \frac{1}{\gamma}\left\|x^t - x^\star\right\|^2 + (1+\omega)\left(\frac{1}{\tau} + 2\mu_{h^*}\right)\left\|u^t - u^\star\right\|^2, \tag{38}$$

Algorithm 13 ADMM

input: initial points $x^0 \in \mathcal{X}$, $u^0 \in \mathcal{U}$;
stepsize $\gamma > 0$
for $t = 0, 1, \ldots$ **do**
$\hat{x}^t := \mathrm{prox}_{\gamma g}(x^t - \gamma u^t)$
$x^{t+1} := \mathrm{prox}_{\gamma h}(\hat{x}^t + \gamma u^t)$
$u^{t+1} := u^t + \frac{1}{\gamma}(\hat{x}^t - x^{t+1})$
end for

Algorithm 14 RandProx-ADMM [new]

input: initial points $x^0 \in \mathcal{X}$, $u^0 \in \mathcal{U}$;
stepsize $\gamma > 0$; $\omega \geq 0$
for $t = 0, 1, \ldots$ **do**
$\hat{x}^t := \mathrm{prox}_{\gamma g}\big(x^t - \gamma u^t\big)$
$d^t := \mathcal{R}^t\big(\hat{x}^t - \mathrm{prox}_{\gamma(1+\omega)h}(\hat{x}^t + \gamma(1+\omega)u^t)\big)$
$x^{t+1} := \hat{x}^t - \frac{1}{1+\omega} d^t$
$u^{t+1} := u^t + \frac{1}{\gamma(1+\omega)^2} d^t$
end for

where $x^\star$ and $u^\star$ are the unique solutions to (36) and (37), respectively. Then RandProx-CP converges linearly: for every $t \geq 0$,

$$\mathbb{E}\big[\Psi^t\big] \leq c^t \Psi^0, \tag{39}$$

where

$$c := \max\left(\frac{1}{1+\gamma\mu_g}, 1 - \frac{2\tau\mu_{h^*}}{(1+\omega)(1+2\tau\mu_{h^*})}\right) \tag{40}$$

$$= 1 - \min\left(\frac{\gamma\mu_g}{1+\gamma\mu_g}, \frac{2\tau\mu_{h^*}}{(1+\omega)(1+2\tau\mu_{h^*})}\right) < 1. \tag{41}$$

Also, $(x^t)_{t\in\mathbb{N}}$ and $(\hat{x}^t)_{t\in\mathbb{N}}$ both converge to $x^\star$ and $(u^t)_{t\in\mathbb{N}}$ converges to $u^\star$, almost surely.

It would be interesting to study whether the mechanism in the stochastic PDHG algorithm proposed in Chambolle et al. [18] can be viewed as a particular case of RandProx-CP; we leave the analysis of this connection for future work. In any case, the strong convexity constants μ_g and μ_{h^*} need to be known in the linearly converging version of the stochastic PDHG algorithm, which is not the case here; this is an important advantage of RandProx-CP.

Now, let us look at the particular case $K = \mathrm{Id}$ in (36) and (37). The primal problem becomes:

$$\text{Find } x^\star \in \mathop{\arg\min}_{x\in\mathcal{X}} \Big(g(x) + h(x)\Big), \tag{42}$$

and the dual problem becomes:

$$\text{Find } u^\star \in \mathop{\arg\min}_{u\in\mathcal{U}} \Big(g^*(-u) + h^*(u)\Big). \tag{43}$$

When $K = \mathrm{Id}$, the CP algorithm with $\tau = \frac{1}{\gamma}$ reverts to the Douglas–Rachford algorithm, which is equivalent to the Alternating Direction Method of Multipliers (ADMM) [12, 27], shown above. Therefore, in that case, with $\omega_{\mathrm{ran}} = \omega$, $\zeta = 0$ and $\tau = \frac{1}{\gamma(1+\omega)}$, RandProx-CP can be rewritten as RandProx-ADMM, shown above. Theorem 7 becomes:

Theorem 8. *Suppose that $\mu_g > 0$ and $\mu_{h^*} > 0$. In RandProx-ADMM, suppose that $\gamma > 0$. For every $t \geq 0$, define the Lyapunov function*

$$\Psi^t := \frac{1}{\gamma}\left\|x^t - x^\star\right\|^2 + (1+\omega)\big(\gamma(1+\omega) + 2\mu_{h^*}\big)\left\|u^t - u^\star\right\|^2, \tag{44}$$

Algorithm 15 DY algorithm [31]

input: initial points $x^0 \in \mathcal{X}$, $u^0 \in \mathcal{X}$;
stepsize $\gamma > 0$
for $t = 0, 1, \dots$ **do**
$\hat{x}^t := \mathrm{prox}_{\gamma g}\big(x^t - \gamma \nabla f(x^t) - \gamma u^t\big)$
$x^{t+1} := \mathrm{prox}_{\gamma h}(\hat{x}^t + \gamma u^t)$
$u^{t+1} := u^t + \frac{1}{\gamma}(\hat{x}^t - x^{t+1})$
end for

Algorithm 16 RandProx-DY [new]

input: initial points $x^0 \in \mathcal{X}$, $u^0 \in \mathcal{X}$;
stepsize $\gamma > 0$; $\omega \geq 0$
for $t = 0, 1, \dots$ **do**
$\hat{x}^t := \mathrm{prox}_{\gamma g}\big(x^t - \gamma \nabla f(x^t) - \gamma u^t\big)$
$d^t := \mathcal{R}^t\big(\hat{x}^t - \mathrm{prox}_{\gamma(1+\omega)h}(\hat{x}^t + \gamma(1+\omega)u^t)\big)$
$x^{t+1} := \hat{x}^t - \frac{1}{1+\omega} d^t$
$u^{t+1} := u^t + \frac{1}{\gamma(1+\omega)^2} d^t$
end for

where $x^\star$ and $u^\star$ are the unique solutions to (42) and (43), respectively. Then RandProx-ADMM converges linearly: for every $t \geq 0$,

$$\mathbb{E}\big[\Psi^t\big] \leq c^t \Psi^0, \tag{45}$$

where

$$c := \max\left(\frac{1}{1+\gamma\mu_g}, 1 - \frac{2\tau\mu_{h^*}}{(1+\omega)(1+2\tau\mu_{h^*})}\right) \tag{46}$$

$$= 1 - \min\left(\frac{\gamma\mu_g}{1+\gamma\mu_g}, \frac{2\tau\mu_{h^*}}{(1+\omega)(1+2\tau\mu_{h^*})}\right) < 1. \tag{47}$$

Also, $(x^t)_{t\in\mathbb{N}}$ and $(\hat{x}^t)_{t\in\mathbb{N}}$ both converge to $x^\star$ and $(u^t)_{t\in\mathbb{N}}$ converges to $u^\star$, almost surely.

Appendix H. Particular case $K = \mathrm{Id}$: randomized Davis–Yin algorithm

After the particular case $g = 0$ discussed in Section A.1 and the particular case $f = 0$ discussed in Section G, we discuss in this section the third particular case $K = \mathrm{Id}$ in (1) and (2). The primal problem becomes:

$$\text{Find } x^\star \in \underset{x\in\mathcal{X}}{\arg\min}\Big(f(x) + g(x) + h(x)\Big), \tag{48}$$

and the dual problem becomes:

$$\text{Find } u^\star \in \underset{u\in\mathcal{U}}{\arg\min}\Big((f+g)^*(-u) + h^*(u)\Big). \tag{49}$$

When $K = \mathrm{Id}$, the PDDY algorithm with $\tau = \frac{1}{\gamma}$ reverts to the Davis–Yin (DY) algorithm [31], shown above. Therefore, in that case, with $\omega_{\mathrm{ran}} = \omega$, $\zeta = 0$ and $\tau = \frac{1}{\gamma(1+\omega)}$, RandProx can be rewritten as RandProx-DY, shown above, too. When $g = 0$, RandProx-DY reverts to RandProx-FB and when $f = 0$, RandProx-DY reverts to RandProx-ADMM; in other words, RandProx-DY generalizes RandProx-FB and RandProx-ADMM into a single algorithm. Theorem 1 yields:

Theorem 9. *Suppose that $\mu_f > 0$ or $\mu_g > 0$, and that $\mu_{h^*} > 0$. In RandProx-DY, suppose that $0 < \gamma < \frac{2}{L_f}$. For every $t \geq 0$, define the Lyapunov function,*

$$\Psi^t := \frac{1}{\gamma}\left\|x^t - x^\star\right\|^2 + (1+\omega)\big(\gamma(1+\omega) + 2\mu_{h^*}\big)\left\|u^t - u^\star\right\|^2, \tag{50}$$

where $x^\star$ and $u^\star$ are the unique solutions to (48) and (49), respectively. Then RandProx-DY converges linearly: for every $t \geq 0$,

$$\mathbb{E}\big[\Psi^t\big] \leq c^t \Psi^0, \tag{51}$$

where

$$c := \max\left(\frac{(1-\gamma\mu_f)^2}{1+\gamma\mu_g}, \frac{(\gamma L_f - 1)^2}{1+\gamma\mu_g}, 1 - \frac{\frac{2}{\gamma}\mu_{h^*}}{(1+\omega)\big(1+\omega+\frac{2}{\gamma}\mu_{h^*}\big)}\right) < 1. \tag{52}$$

Also, $(x^t)_{t\in\mathbb{N}}$ and $(\hat{x}^t)_{t\in\mathbb{N}}$ both converge to $x^\star$ and $(u^t)_{t\in\mathbb{N}}$ converges to $u^\star$, almost surely.

We can note that in Theorem 9, $\mu_{h^*} > 0$ is required. It is only in the case $g = 0$, when RandProx-DY reverts to RandProx-FB, that one can apply Theorem 3, which does not require strong convexity of h^*.

Appendix I. Convergence in the merely convex case

In all theorems, strong convexity of f or g is assumed; that is, $\mu_f > 0$ or $\mu_g > 0$. In this section, we remove this hypothesis, so that the primal problem is not necessarily strongly convex any more. But $\nabla f(x^\star)$ is the same for every solution $x^\star$ of (1), and we denote by $\nabla f(x^\star)$ this element.

We define the Bregman divergence of f at points $(x, x') \in \mathcal{X}^2$ as

$$D_f(x, x') := f(x) - f(x') - \langle \nabla f(x'), x - x'\rangle \geq 0.$$

For every $t \geq 0$, $D_f(x^t, x^\star)$ is the same for every solution $x^\star$ of (1), and we denote by $D_f(x^t, x^\star)$ this element. $D_f(x^t, x^\star)$ can be viewed as a generalization of the objective gap $f(x^t) - f(x^\star)$ to the case when $\nabla f(x^\star) \neq 0$. $D_f(x^t, x^\star)$ is a loose kind of distance between x^t and the solution set, but under some additional assumptions on f, for instance strict convexity, $D_f(x^t, x^\star) \to 0$ implies that the distance from x^t to the solution set tends to zero. Also, $D_f(x^t, x^\star) \geq \frac{1}{2L_f}\|\nabla f(x^t) - \nabla f(x^\star)\|^2$, so that $D_f(x^t, x^\star) \to 0$ implies that $\big(\nabla f(x^t)\big)_{t\in\mathbb{N}}$ converges to $\nabla f(x^\star)$.

Theorem 11. *In RandProx, suppose that $0 < \gamma < \frac{2}{L_f}$, $\tau > 0$, and $\gamma\tau\big((1-\zeta)\|K\|^2 + \omega_{\mathrm{ran}}\big) \leq 1$. Then $D_f(x^t, x^\star) \to 0$, almost surely and in quadratic mean. Moreover, for every $t \geq 0$, we define $\bar{x}^t := \frac{1}{t}\sum_{i=1}^t x^i$. Then, for every $t \geq 0$,*

$$\mathbb{E}\big[D_f(\bar{x}^t, x^*)\big] \leq \frac{\Psi^0}{(2\gamma - \gamma^2 L_f)t} = \mathcal{O}(1/t). \tag{53}$$

If, in addition, $\mu_{h^*} > 0$, there is a unique dual solution $u^\star$ to (2) and $(u^t)_{t\in\mathbb{N}}$ converges to $u^\star$, in quadratic mean.

The counterpart of Theorem 2 in the convex case is:

Theorem 12. *Suppose that $g = 0$, and that $\lambda_{\min}(KK^*) > 0$ or $\mu_{h^*} > 0$. In RandProx, suppose that $0 < \gamma < \frac{2}{L_f}$, $\tau > 0$, and $\gamma\tau\big((1-\zeta)\|K\|^2 + \omega_{\mathrm{ran}}\big) \leq 1$. Then there is a unique dual solution $u^\star$ to (2) and $(u^t)_{t\in\mathbb{N}}$ converges to $u^\star$, in quadratic mean.*

We can derive counterparts of the other theorems in the same way. These theorems apply to all algorithms presented in the paper. For instance, Theorems 11 and 12 apply to Scaffnew [60], a particular case of RandProx-FL seen in Section B.3, and provide for it the first convergence results in the non-strongly convex case.

I.1. Proof of Theorems 11 and 12

Proof of Theorem 11 We have, for every $(x, x') \in \mathcal{X}^2$,

$$\begin{aligned}\|(\mathrm{Id} - \gamma\nabla f)x - (\mathrm{Id} - \gamma\nabla f)x'\|^2 &= \|x - x'\|^2 - 2\gamma\langle\nabla f(x) - \nabla f(x'), x - x'\rangle \\ &\quad + \gamma^2\|\nabla f(x) - \nabla f(x')\|^2 \\ &\leq \|x - x'\|^2 - (2\gamma - \gamma^2 L_f)\langle\nabla f(x) - \nabla f(x'), x - x'\rangle,\end{aligned}$$

where the second inequality follows from cocoercivity of the gradient. Moreover, for every $(x, x') \in \mathcal{X}^2$, $D_f(x, x') \leq \langle\nabla f(x) - \nabla f(x'), x - x'\rangle$. Therefore, in the proof of Theorem 1, for every primal–dual solution $(x^\star, u^\star)$ and $t \geq 0$, since $\|w^t - w^\star\|^2 = \|(\mathrm{Id} - \gamma\nabla f)x^t - (\mathrm{Id} - \gamma\nabla f)x^\star\|^2$, (27) yields

$$\begin{aligned}\mathbb{E}\big[\Psi^{t+1} \mid \mathcal{F}_t\big] &\leq \frac{1}{\gamma}\left\|x^t - x^\star\right\|^2 - (2\gamma - \gamma^2 L_f) D_f(x^t, x^*) \\ &\quad + \left(\frac{1+\omega}{\tau} + 2\omega\mu_{h^*}\right)\left\|u^t - u^\star\right\|^2 - \gamma\left\|q^t - q^\star + K^*(u^t - u^\star)\right\|^2.\end{aligned}$$

Ignoring the last term, this yields

$$\begin{aligned}\mathbb{E}\big[\Psi^{t+1} \mid \mathcal{F}_t\big] &\leq \frac{1}{\gamma}\left\|x^t - x^\star\right\|^2 + c(1+\omega)\left(\frac{1}{\tau} + 2\mu_{h^*}\right)\left\|u^t - u^\star\right\|^2 \qquad (54)\\ &\quad - (2\gamma - \gamma^2 L_f) D_f(x^t, x^*) \\ &\leq \Psi^t - (2\gamma - \gamma^2 L_f) D_f(x^t, x^*), \qquad (55)\end{aligned}$$

with $c = 1 - \frac{2\tau\mu_{h^*}}{(1+\omega)(1+2\tau\mu_{h^*})}$ in (54). Using classical results on supermartingale convergence [8, Proposition A.4.5], it follows from (55) that Ψ^t converges almost surely to a random variable Ψ^∞ and that

$$\sum_{t=0}^{\infty} D_f(x^t, x^*) < +\infty \quad \text{almost surely.}$$

Hence, $D_f(x^t, x^*) \to 0$ almost surely. Moreover, for every $T \geq 0$,

$$(2\gamma - \gamma^2 L_f)\sum_{t=0}^{T}\mathbb{E}\big[D_f(x^t, x^*)\big] \leq \Psi^0 - \mathbb{E}\big[\Psi^{T+1}\big] \leq \Psi^0 \qquad (56)$$

and

$$(2\gamma - \gamma^2 L_f)\sum_{t=0}^{\infty}\mathbb{E}\big[D_f(x^t, x^*)\big] \leq \Psi^0.$$

Therefore, $\mathbb{E}\big[D_f(x^t, x^*)\big] \to 0$; that is, $D_f(x^t, x^*) \to 0$ in quadratic mean.

The Bregman divergence is convex in its first argument, so that for every $T \geq 0$,

$$D_f(\bar{x}^T, x^*) \leq \frac{1}{T}\sum_{t=0}^{T} D_f(x^t, x^*).$$

Combining this last inequality with (56) yields

$$T(2\gamma - \gamma^2 L_f)\mathbb{E}\big[D_f(\bar{x}^T, x^*)\big] \leq \Psi^0.$$

Now, if $\mu_{h^*} > 0$, then $c < 1$ in (54), and since Ψ^t converges almost surely to Ψ^∞, it must be that $\mathbb{E}\left[\left\|u^t - u^\star\right\|^2\right] \to 0$. □

Proof of Theorem 12 Considering the proof of Theorem 2, the same arguments as in the proof of Theorem 11 apply, with c in (54) now equal to

$$c = 1 - \frac{2\tau\mu_{h^*} + \gamma\tau\lambda_{\min}(KK^*)}{(1+\omega)(1+2\tau\mu_{h^*})} < 1.$$

Hence, $\mathbb{E}\left[\left\|u^t - u^\star\right\|^2\right] \to 0$. □