SIAM J. OPTIM.
Vol. 29, No. 1, pp. 100–134

© 2019 Society for Industrial and Applied Mathematics

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

A COORDINATE-DESCENT PRIMAL-DUAL ALGORITHM WITH LARGE STEP SIZE AND POSSIBLY NONSEPARABLE FUNCTIONS[*]

OLIVIER FERCOQ[†] AND PASCAL BIANCHI[†]

Abstract. This paper introduces a randomized coordinate-descent version of the Vũ–Condat algorithm. By coordinate descent, we mean that only a subset of the coordinates of the primal and dual iterates is updated at each iteration, the other coordinates being maintained to their past value. Our method allows us to solve optimization problems with a combination of differentiable functions and constraints as well as nonseparable and nondifferentiable regularizers. We show that the sequences generated by our algorithm almost surely converge to a saddle point of the problem at stake, for a wider range of parameter values than previous methods. In particular, the condition on the step sizes depends on the coordinatewise Lipschitz constant of the differentiable function's gradient, which is a major feature allowing classical coordinate descent to perform so well when it is applicable. We then prove a sublinear rate of convergence in general and a linear rate of convergence if the objective enjoys strong convexity properties. We illustrate the performances of the algorithm on a total-variation regularized least squares regression problem and on large-scale support vector machine problems.

Key words. convex optimization, coordinate descent, primal-dual algorithm, proximal method

AMS subject classifications. 90C25, 49M25, 90C06

DOI. 10.1137/18M1168480

1. Introduction.

1.1. Motivation.

We consider the optimization problem

$$\inf_{x\in\mathcal{X}} f(x) + g(x) + h(Mx), \tag{1}$$

where $\mathcal{X}$ is a Euclidean space, $M : \mathcal{X} \to \mathcal{Y}$ is a linear operator onto a second Euclidean space $\mathcal{Y}$, functions $f : \mathcal{X} \to \mathbb{R}$, $g : \mathcal{X} \to \,]-\infty, +\infty]$ and $h : \mathcal{Y} \to \,]-\infty, +\infty]$ are assumed convex, proper, and lower-semicontinuous, and the function f is moreover assumed differentiable. We assume that $\mathcal{X}$ and $\mathcal{Y}$ are product spaces of the form $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ and $\mathcal{Y} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_p$ for some integers n, p. For any $x \in \mathcal{X}$, we use the notation $x = (x^{(1)}, \dots, x^{(n)})$ to represent the (block of) coordinates of x (similarly for $y = (y^{(1)}, \dots, y^{(p)})$ in $\mathcal{Y}$). Problem (1) has numerous applications, e.g., in machine learning [9], image processing [10], or distributed optimization [8].

Under the standard qualification condition $0 \in \mathrm{ri}(M\mathrm{dom}g - \mathrm{dom}h)$ (where dom and ri stand for domain and relative interior, respectively), a point $x \in \mathcal{X}$ is a minimizer of (1) if and only if there exists $y \in \mathcal{Y}$ such that (x, y) is a saddle point of the Lagrangian function

$$L(x, y) = f(x) + g(x) + \langle y, Mx\rangle - h^\star(y),$$

where $\langle\cdot,\cdot\rangle$ is the inner product and $h^\star : y \mapsto \sup_{z\in\mathcal{Y}}\langle y, z\rangle - h(z)$ is the Fenchel–Legendre transform of h. There is a rich literature on *primal-dual* algorithms searching

[*]Received by the editors February 2, 2018; accepted for publication (in revised form) October 2, 2018; published electronically January 8, 2019. A summary of the results of this paper has been published in the Proceedings of the 2016 Conference on Decision and Control [6].

http://www.siam.org/journals/siopt/29-1/M116848.html

Funding: This work has been supported by the Orange/Telecom ParisTech think tank Phi-TAB.

[†]LTCI, Télécom ParisTech, Université Paris-Saclay, 75013, Paris, France (olivier.fercoq@telecom-paristech.fr, pascal.bianchi@telecom-paristech.fr).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

for a saddle point of L (see [47] and references therein). In the special case where $f = 0$, the alternating direction method of multipliers (ADMM) proposed by Glowinsky and Marroco [27], Gabay and Mercier [24] and the algorithm of Chambolle and Pock [13] are amongst the most celebrated ones. Based on an elegant idea also used in [29], Vũ [53] and Condat [17] separately proposed a primal-dual algorithm also allowing ∇f to be handled explicitly, and requiring one evaluation of the gradient of f at each iteration. Hence, the ∇f is handled explicitly in the sense that the algorithm does *not* involve, for instance, the call of a proximity operator associated with f. A convergence rate analysis is provided in [14] (see also [47]). A related splitting method has recently been introduced by [18].

This paper introduces a *coordinate-descent* (CD) version of the Vũ–Condat algorithm. By coordinate descent, we mean that only a subset of the coordinates of the primal and dual iterates is updated at each iteration, the other coordinates being maintained to their past value. Coordinate descent was historically used in the context of coordinatewise minimization of a unique function in a Gauss–Seidel sense [54, 5, 50]. Tseng and coworkers [35, 51, 52] and Nesterov [38] developed CD versions of the gradient descent. The convergence speed of cyclic coordinate gradient descent was analyzed in [3, 30]. In [38] as well as in this paper, the updated coordinates are randomly chosen at each iteration. The algorithm of [38] has at least two interesting features. Not only it is often easier to evaluate a single coordinate of the gradient vector rather than the whole vector, but the conditions under which the CD version of the algorithm is provably convergent are generally weaker than in the case of standard gradient descent. The key point is that the *step size* used in the algorithm when updating a given coordinate i can be chosen to be inversely proportional to the *coordinatewise* Lipschitz constant of ∇f along its ith coordinate, rather than the global Lipschitz constant of ∇f (as would be the case in a standard gradient descent). Hence, the introduction of coordinate descent allows us to use *longer step sizes*, which potentially results in a more attractive performance. The random CD gradient descent of [38] was later generalized by Richtárik and Takáč [41] to the minimization of a sum of two convex functions $f + g$ (that is, $h = 0$ in problem (1)). The algorithm of [41] is analyzed under the additional assumption that function g is *separable* in the sense that, for each $x \in \mathcal{X}$, $g(x) = \sum_{i=1}^n g_i(x^{(i)})$ for some functions $g_i : \mathcal{X}_i \to \,]-\infty, +\infty]$. Accelerated and parallel versions of the algorithm were later developed by [43, 42, 21, 34], always assuming the separability of g.

In the literature, several papers seek to apply the principle of coordinate descent to primal-dual algorithms. In the case where $f = 0$, h is separable and smooth, and g is strongly convex, Zhang and Xiao [55] introduce a stochastic CD primal-dual algorithm and analyze its convergence rate (see also [46] for related works). In 2013, Iutzeler et al. [32] proved that random coordinate descent can be successfully applied to fixed-point iterations of firmly nonexpansive (FNE) operators. According to [23], the ADMM can be written as a fixed-point algorithm of a FNE operator, which led Iutzeler et al. [32] to propose a coordinate-descent version of ADMM with application to distributed optimization. The key idea behind the convergence proof of [32] is to establish the so-called stochastic Fejér monotonicity of the sequence of iterates as noted by Combettes and Pesquet [16]. In a more general setting than [32], Combettes and Pesquet [16] and Bianchi et al. [7] extend the proof to the so-called α-averaged operators, which include FNE operators as a special case. This generalization allows the coordinate-descent principle to be applied to a broader class of primal-dual algorithms which is no longer restricted to the ADMM or the Douglas–Rachford algorithm. For instance, forward-backward splitting is considered in [16] and

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

particular cases of the Vũ–Condat algorithm are considered in [7, 40]. Nevertheless, the above approach has two major limitations.

First, in order to derive a converging coordinate-descent version of a given deterministic algorithm, the latter must be written as a fixed-point algorithm over some product Hilbert space of the form $H = H_1 \times \cdots \times H_q$ where the inner product in H is the sum of the inner products in the H_i's. Unfortunately, this condition does *not* hold in general for the Vũ–Condat method, because the inner product over H involves the coupling linear operator M. A work-around was proposed in [7], but for a particular example only.

Second, and even more importantly, the approach in [32, 16, 7, 40] needs "small" step sizes. More precisely, the convergence conditions are identical to those of the brute method, without coordinate descent. These conditions involve the global Lipschitz constant of the gradient ∇f instead of its coordinatewise Lipschitz constants. In practice, this means that the application of coordinate descent to the primal-dual algorithm as suggested in [16, 7] is restricted to the use of potentially small step sizes. One of the major benefits of coordinate descent is lost.

Some recent works also focused on designing primal-dual coordinate-descent methods with a guaranteed convergence rate. In [25, 12], an $O(1/k)$ rate is obtained for the ergodic mean of the sequences. The rates are given in terms of feasibility and optimality or Bregman distance. These two papers require all the dual variables to be updated at each iteration, which may not be efficient if there are more than a few dual variables. In the present paper, we will have much more flexibility in the variables we choose to update at each iteration, while retaining a provable convergence rate.

1.2. Contribution.

- Our main contribution is to provide a CD primal-dual algorithm with a broad range of admissible step sizes. Our numerical experiments show that remarkable performance gains can be obtained when using larger step sizes.
- We identify two setups for which the structure of the problem is favorable to coordinate-descent algorithms.
- We prove a sublinear rate of convergence in general and a linear rate of convergence if the objective enjoys strong convexity properties.

1.3. Organization of the paper. The algorithm is introduced in section 2. At each iteration k, an index i is randomly chosen w.r.t. the uniform distribution in $\{1, \dots, n\}$, where n is, as we recall, the number of primal coordinates. The coordinate $x_k^{(i)}$ of the current primal iterate x_k is updated, as is a set of associated dual iterates. Under some assumptions involving the coordinatewise Lipschitz constants of ∇f, the primal-dual iterates converge to a saddle point of the Lagrangian. As a remarkable feature, our CD algorithm makes no assumption of separability of the functions f, g, or h. In the special case where $h = 0$ and g is separable, the algorithm reduces to the CD proximal gradient algorithm of [41].

The convergence proof is provided in section 3. It is worth noting that, under the stated assumption on the step size, the stochastic Fejér monotonicity of the sequence of iterates, which is the key idea in [32, 16, 7], does not hold (a counterexample is provided). Our proof relies on the introduction of an adequate Lyapunov function. In section 4, we prove a sublinear rate of convergence in general, and a linear rate of convergence if the objective enjoys strong convexity properties. In section 5, the proposed algorithm is instantiated in the case of total-variation regularization and support vector machines. Numerical results performed on real MRI and text data establish the

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

attractive behavior of the proposed algorithm and emphasize the importance of using primal-dual CD with large step sizes.

2. Coordinate-descent primal-dual algorithm.

2.1. Notation. We note $M = (M_{j,i} : j \in \{1,\dots,p\}, i \in \{1,\dots,n\})$, where $M_{j,i} : \mathcal{X}_i \to \mathcal{Y}_j$ are the block components of M. For each $j \in \{1,\dots,p\}$, we introduce the set

$$I(j) := \Big\{ i \in \{1,\dots,n\} \,:\, M_{j,i} \neq 0 \Big\}.$$

In other words, the jth component of vector Mx only depends on x through the coordinates $x^{(i)}$ such that $i \in I(j)$. We denote by

$$m_j := \mathrm{card}(I(j))$$

the number of such coordinates. Without loss of generality, we assume that $m_j \neq 0 \,\forall j$. We also define

$$\pi_j := \frac{1}{\mathrm{card}(I(j))}.$$

For all $i \in \{1,\dots,n\}$, we define

$$J(i) := \Big\{ j \in \{1,\dots,p\} \,:\, M_{j,i} \neq 0 \Big\}.$$

Note that, for every pair (i,j), the statements $i \in I(j)$ and $j \in J(i)$ are equivalent.

If ℓ is an integer, $\gamma = (\gamma_1,\dots,\gamma_\ell)$ is a collection of positive real numbers, and $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_\ell$ is a product of Euclidean spaces, we introduce the weighted norm $\|\cdot\|_\gamma$ on $\mathcal{A}$ given by $\|u\|_\gamma^2 = \sum_{i=1}^{\ell} \gamma_i \|u^{(i)}\|_{\mathcal{A}_i}^2$ for every $u = (u^{(1)},\dots,u^{(\ell)})$, where $\|\cdot\|_{\mathcal{A}_i}$ stands for the norm on $\mathcal{A}_i$. If $F : \mathcal{A} \to]-\infty,+\infty]$ denotes a convex proper lower-semicontinuous function, we introduce the proximity operator $\mathrm{prox}_{\gamma,F} : \mathcal{A} \to \mathcal{A}$ defined for any $u \in \mathcal{A}$ by

$$\mathrm{prox}_{\gamma,F}(u) := \arg\min_{w \in \mathcal{A}} \left[F(w) + \frac{1}{2}\|w-u\|_{\gamma^{-1}}^2 \right]$$

where we use the notation $\gamma^{-1} = (\gamma_1^{-1},\dots,\gamma_\ell^{-1})$. We denote by $\mathrm{prox}_{\gamma,F}^{(i)} : \mathcal{A} \to \mathcal{A}_i$ the ith coordinate mapping of $\mathrm{prox}_{\gamma,F}$, that is, $\mathrm{prox}_{\gamma,F}(u) = (\mathrm{prox}_{\gamma,F}^{(1)}(u),\dots,\mathrm{prox}_{\gamma,F}^{(\ell)}(u))$ for any $u \in \mathcal{A}$. The notation $D_{\mathcal{A}}(\gamma)$ (or simply $D(\gamma)$ when no ambiguity occurs) stands for the diagonal operator on $\mathcal{A} \to \mathcal{A}$ given by $D_{\mathcal{A}}(\gamma)(u) = (\gamma_1 u^{(1)},\dots,\gamma_\ell u^{(\ell)})$ for every $u = (u^{(1)},\dots,u^{(\ell)})$.

Finally, the adjoint of a linear operator B is denoted by $B^\star$. The spectral radius of a square matrix A is denoted by $\rho(A)$. The number of nonzero elements of a matrix A is denoted by $\mathrm{nnz}(A)$.

2.2. Main algorithm. Consider problem (1). Let $\sigma = (\sigma_1,\dots,\sigma_p)$ and $\tau = (\tau_1,\dots,\tau_n)$ be two tuples of positive real numbers. Consider an independent and identically distributed sequence $(i_k : k \in \mathbb{N}^*)$ with uniform distribution on $\{1,\dots,n\}$.[1] Algorithm 1, the proposed primal-dual CD algorithm, consists in updating two sequences $x_k \in \mathcal{X}$, $y_k \in \mathcal{Y}$.

For every $i \in \{1,\dots,n\}$, we denote by $U_i : \mathcal{X}_i \to \mathcal{X}$ the linear operator such that all coordinates of $U_i(u)$ are zero except the ith coordinate which coincides with u: $U_i(u) = (0,\dots,0,u,0,\dots,0)$. Our convergence result holds under the following assumptions.

[1] The results of this paper easily extend to the selection of several primal coordinates at each iteration with a uniform samplings of the coordinates, using the techniques introduced in [42].

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Algorithm 1 Coordinate-descent primal-dual algorithm.

```
Initialization: Choose $x_0 \in \mathcal{X}$, $y_0 \in \mathcal{Y}$.
Iteration $k$: Define
    $\overline{y}_{k+1} = \mathrm{prox}_{\sigma,h^\star}\big(y_k + D(\sigma)Mx_k\big)$,
    $\overline{x}_{k+1} = \mathrm{prox}_{\tau,g}\Big(x_k - D(\tau)\big(\nabla f(x_k) + 2M^\star\overline{y}_{k+1} - M^\star y_k\big)\Big)$.
For $i = i_{k+1}$ and for each $j \in J(i_{k+1})$, update as follows:
    $x^{(i)}_{k+1} = \overline{x}^{(i)}_{k+1}$,
    $y^{(j)}_{k+1} = y^{(j)}_k + \pi_j(\overline{y}^{(j)}_{k+1} - y^{(j)}_k)$.
Otherwise, set $x^{(i')}_{k+1} = x^{(i')}_k$ and $y^{(j')}_{k+1} = y^{(j')}_k$.
```

Assumption 1.

(a) The functions f, g, h are convex, proper, and lower-semicontinuous.

(b) The function f is differentiable on $\mathcal{X}$.

(c) For every $i \in \{1,\dots,n\}$, there exists $\beta_i \geq 0$ such that, for any $x \in \mathcal{X}$, any $u \in \mathcal{X}_i$,
$$f(x + U_i u) \leq f(x) + \langle \nabla f(x), U_i u\rangle + \frac{\beta_i}{2}\|u\|^2_{\mathcal{X}_i}.$$

(d) Random sequence $(i_k)_{k\in\mathbb{N}^*}$ is independent, uniformly distributed on $\{1,\dots,n\}$.

(e) The step sizes $\tau = (\tau_1,\dots,\tau_n)$ and $\sigma = (\sigma_1,\dots,\sigma_p)$ satisfy, for all $i \in \{1,\dots,n\}$,
$$\tau_i < \frac{1}{\beta_i + \rho\Big(\sum_{j\in J(i)}(2-\pi_j)m_j\sigma_j M^\star_{j,i}M_{j,i}\Big)}.$$

We denote by $\mathcal{S}$ the set of saddle points of the Lagrangian function L. In other words, a couple $(x_*, y_*) \in \mathcal{X}\times\mathcal{Y}$ lies in $\mathcal{S}$ if and only if it satisfies the following inclusions:

$$0 \in \nabla f(x_*) + \partial g(x_*) + M^\star y_*, \tag{2}$$
$$0 \in -Mx_* + \partial h^\star(y_*). \tag{3}$$

We shall also refer to elements of $\mathcal{S}$ as primal-dual solutions.

THEOREM 1. *Let Assumption* 1 *hold true and suppose that* $\mathcal{S} \neq \emptyset$. *Let* (x_k, y_k) *be a sequence generated by Algorithm* 1. *Almost surely, there exists* $(x_*, y_*) \in \mathcal{S}$ *such that*
$$\lim_{k\to\infty} x_k = x_*,$$
$$\lim_{k\to\infty} y_k = y_*.$$

2.3. Efficient implementation using problem structure. In Algorithm 1, it is worth noting that quantities $(\overline{x}_{k+1}, \overline{y}_{k+1})$ do not need to be explicitly calculated. At iteration k, only the coordinates
$$\overline{x}^{(i_{k+1})}_{k+1} \text{ and } \overline{y}^{(j)}_{k+1} \quad \forall j \in J(i_{k+1})$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

are needed to perform the update. From a computational point of view, it is often the case that the evaluation of the above coordinates is less demanding than the computation of the whole vectors $\overline{x}_{k+1}$, $\overline{y}_{k+1}$. Two situations have been reported in the literature.

- If g is separable, to perform the kth iteration one only needs to compute the quantities $\nabla_{i_{k+1}} f(x_k)$, $(2M^\star \bar{y}_{k+1} - M^\star y_k)^{(i_{k+1})}$, and $\mathrm{prox}_{\tau_{i_{k+1}}, g_{i_{k+1}}}$. A classical example of such a smart residual update [39] can be found in the proximal coordinate-descent gradient algorithm (the g separable and $h = 0$ case) [41]. More generally, if g (resp., $h^\star$) is block-separable, we can use this structure in the algorithm, even if this block structure does not match $\mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ (resp., $\mathcal{Y}_1 \times \cdots \times \mathcal{Y}_p$). We use this idea in section 5.1 to deal efficiently with the proximal operator of the $\ell_{2,1}$ norm.
- If g is the indicator of the consensus constraint $\{x_1 = \cdots = x_n\}$, f is separable, and $h = 0$, we recover MISO [36]. In that case, we can store $\nabla f(x_k)$ and update its average. Thanks to the separability of f, only one coordinate of $\nabla f(x_k)$ needs to be updated at each iteration. We use similar ideas in section 5.2 to deal efficiently with the projection onto the subspace orthogonal to a vector.

To illustrate the importance of these implementation tricks, in Table 1 we compare the number of operations required to compute the updates of the standard Vũ–Condat method against those for the proposed algorithm.

Table 1
Number of operations per iteration for the proposed algorithm and for the standard Vũ–Condat algorithm: The use cases are described in section 5. *The numbers* 6 *and* 12 *highlight the (mild) overhead of duplication in the total variation* + ℓ_1*-regularized least squares problem.*

Problem/dimension of data	Vũ–Condat	Our algorithm
Total variation + ℓ_1-regularization $A \in \mathbb{R}^{m\times n}$: dense; $M \in \mathbb{R}^{3n\times n}$: $\mathrm{nnz}(M) = 6n$	$O(mn + 6n)$	$O(m + 12)$
Support vector machines $A \in \mathbb{R}^{m\times n}$: sparse	$O(\mathrm{nnz}(A) + n)$	$O(\mathrm{nnz}(Ae_i) + 1)$

2.4. Primal-dual coordinate descent with duplicated dual variables.

In this section, we present a generalization of Algorithm 1 that allows for more flexibility in the update rule for dual variables. It will also be a convenient formulation for the analysis.

Recall that $\mathcal{Y} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_p$. For every $j \in \{1, \dots, p\}$, we use the notation $\boldsymbol{\mathcal{Y}}_j := \mathcal{Y}_j^{I(j)}$, which means that $\boldsymbol{\mathcal{Y}}_j$ consists of $|I(j)|$ copies of $\mathcal{Y}_j$ indexed by $I(j)$. An arbitrary element $\boldsymbol{u}$ in $\boldsymbol{\mathcal{Y}}_j$ will be represented by $\boldsymbol{u} = (\boldsymbol{u}(i) : i \in I(j))$. We define $\boldsymbol{\mathcal{Y}} := \boldsymbol{\mathcal{Y}}_1 \times \cdots \times \boldsymbol{\mathcal{Y}}_p$. An arbitrary element $\boldsymbol{y}$ in $\boldsymbol{\mathcal{Y}}$ will be represented as $\boldsymbol{y} = (\boldsymbol{y}^{(1)}, \dots, \boldsymbol{y}^{(p)})$ and we shall call such an element a duplicated dual variable. This notation is recalled in Table 2.

In our algorithm, we will stack a collection of primal variables $(x_k^{(i)} : i \in \{1, \dots, n\})$ at iteration k, and a set of (duplicated) dual variables $(\boldsymbol{y}_k^{(j)}(i) : i \in \{1, \dots, n\}, j \in J(i))$. In a coordinate-descent spirit, however, we update only a subset of these variables at every iteration k. First, we choose uniformly at random a block of primal coordinates i_{k+1}: Eventually, only the primal variable $x_k^{(i_{k+1})}$ will be updated. As far as the dual variables are concerned, a natural choice is to update the dual variables $(\boldsymbol{y}_k^{(j)}(i_{k+1}) : j \in J(i_{k+1}))$ associated to the primal variable $x_k^{(i_{k+1})}$. This case will be investigated in section 2.5.1. For reasons that will be made clear later on, it may be

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

TABLE 2
Standing notation.

Space	Element	Dimension (if blocks of size 1)
$\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$	$x = (x^{(i)} : i \in \{1, \dots, n\})$	n
$\mathcal{Y} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_p$	$y = (y^{(j)} : j \in \{1, \dots, p\})$	p
$\boldsymbol{\mathcal{Y}}_j = \mathcal{Y}_j^{I(j)}$	$\boldsymbol{u} = (\boldsymbol{u}(i) : i \in I(j))$	$\lvert I(j)\rvert$
$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{Y}}_1 \times \cdots \times \boldsymbol{\mathcal{Y}}_p$	$\boldsymbol{y} = (\boldsymbol{y}^{(j)} : j \in \{1, \dots, p\})$, where $\boldsymbol{y}^{(j)} = (\boldsymbol{y}^{(j)}(i) : i \in I(j))$ $\forall j$	$\mathrm{nnz}(M)$

interesting in some situations to update a larger set of duplicated dual variables at iteration k, namely $(\boldsymbol{y}_k^{(j)}(l) : (l, j) \in \mathcal{J}(i_{k+1}))$, where, for every $i \in \{1, \dots, n\}$, $\mathcal{J}(i)$ is a subset of $\{1, \dots, n\} \times \{1, \dots, p\}$ chosen in such a way that

$$\{i\} \times J(i) \subset \mathcal{J}(i) \subset \{(l, j) : j \in J(l)\} \,. \tag{4}$$

We shall also define the probability that $j \in J(i_{k+1})$, knowing that $(l, j) \in \mathcal{J}(i_{k+1})$ as

$$\boldsymbol{\pi}_j(i) = \frac{1}{\mathrm{card}(\{l \, : \, (i, j) \in \mathcal{J}(l)\})}. \tag{5}$$

Note that $0 < \boldsymbol{\pi}_j(i) \leq 1$. In the special case where $\mathcal{J}(i) = \{i\} \times J(i)$, note also that $\boldsymbol{\pi}_j(i) = 1$ for every $j \in J(i)$.

As for Algorithm 1, we consider an independent and identically distributed sequence $(i_k : k \in \mathbb{N}^*)$ with uniform distribution on $\{1, \dots, n\}$. Algorithm 2 consists in updating four sequences $x_k \in \mathcal{X}$, $w_k \in \mathcal{X}$, $z_k \in \mathcal{Y}$, and $\boldsymbol{y}_k \in \boldsymbol{\mathcal{Y}}$.

THEOREM 2. *Let Assumption* 1 *hold true and*

$$\tau_i < \frac{1}{\beta_i + \rho \left(\sum_{j \in J(i)} (2 - \boldsymbol{\pi}_j(i)) m_j \sigma_j M_{j,i}^\star M_{j,i} \right)}. \tag{6}$$

Suppose that (4) *holds and that $\mathcal{S} \neq \emptyset$. Let $(x_k, \boldsymbol{y}_k)$ be a sequence generated by Algorithm* 2. *Almost surely, there exists $(x_*, y_*) \in \mathcal{S}$ such that*

$$\begin{aligned} &\lim_{k \to \infty} x_k = x_*, \\ &\lim_{k \to \infty} \boldsymbol{y}_k^{(j)}(i) = y_*^{(j)} \qquad (\forall j \in \{1, \dots, p\}, \ \forall i \in I(j)). \end{aligned}$$

2.5. Special cases.

2.5.1. The case in which $\boldsymbol{\mathcal{J}(i) = \{i\} \times J(i)}$ $\forall \boldsymbol{i}$. According to (4), the smallest possible choice for $\mathcal{J}(i)$ is $\mathcal{J}(i) = \{i\} \times J(i)$. In that case, $\boldsymbol{\pi}_j(i) = 1$ $\forall j \in J(i)$ and the update of the dual variable simplifies to

$$\forall j \in J(i_{k+1}), \quad \boldsymbol{y}_{k+1}^{(j)}(i_{k+1}) = \overline{y}_{k+1}^{j}.$$

This choice of dual sampling also implies that the primal and dual variables are grouped into n disjoint primal-dual blocks of the type $(x^{(i)}, (\boldsymbol{y}_{k+1}^{(j)}(i))_{j \in J(i)})$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Algorithm 2 Coordinate-descent primal-dual algorithm with duplicated variables.

```
Initialization: Choose $x_0 \in \mathcal{X}$, $\boldsymbol{y}_0 \in \boldsymbol{\mathcal{Y}}$.
For all $i \in \{1, \dots, n\}$, set $w_0^{(i)} = \sum_{j \in J(i)} M^\star_{j,i} \, \boldsymbol{y}_0^{(j)}(i)$.
For all $j \in \{1, \dots, p\}$, set $z_0^{(j)} = \frac{1}{m_j} \sum_{i \in I(j)} \boldsymbol{y}_0^{(j)}(i)$.
Iteration $k$: Define
    $\overline{y}_{k+1} = \mathrm{prox}_{\sigma, h^\star}\big(z_k + D(\sigma) M x_k\big)$,
    $\overline{x}_{k+1} = \mathrm{prox}_{\tau, g}\Big(x_k - D(\tau)\left(\nabla f(x_k) + 2M^\star \overline{y}_{k+1} - w_k\right)\Big)$.
For $i = i_{k+1}$ and for each $(l, j) \in \mathcal{J}(i_{k+1})$, update as follows:
    $x_{k+1}^{(i)} = \overline{x}_{k+1}^{(i)}$,
    $\boldsymbol{y}_{k+1}^{(j)}(l) = \boldsymbol{y}_k^{(j)}(l) + \boldsymbol{\pi}_j(l)(\overline{y}_{k+1}^{(j)} - \boldsymbol{y}_k^{(j)}(l))$,
    $w_{k+1}^{(l)} = w_k^{(l)} + \sum_{(l,j) \in \mathcal{J}(i)} M^\star_{j,l} \, (\boldsymbol{y}_{k+1}^{(j)}(l) - \boldsymbol{y}_k^{(j)}(l))$,
    $z_{k+1}^{(j)} = z_k^{(j)} + \frac{1}{m_j} \sum_{l : (l,j) \in \mathcal{J}(i)} (\boldsymbol{y}_{k+1}^{(j)}(l) - \boldsymbol{y}_k^{(j)}(l))$.
Otherwise, set $x_{k+1}^{(i')} = x_k^{(i')}$, $w_{k+1}^{(l')} = w_k^{(l')}$, $z_{k+1}^{(j')} = z_k^{(j')}$, and $\boldsymbol{y}_{k+1}^{(j')}(l') = \boldsymbol{y}_k^{(j')}(l')$.
```

2.5.2. The case in which $\boldsymbol{\mathcal{J}(i) = \bigcup_{j \in J(i)} I(j) \times J(i)}$ $\boldsymbol{\forall i}$.

With this update scheme for dual variables, given i_{k+1}, we update $\boldsymbol{y}_{k+1}^{(j)}(l)$ $\forall j \in J(i_{k+1})$ and all $l \in I(j)$. In other words, we update all the copies of $y_{k+1}^{(j)}$ as soon as one of them has to be updated.

We have

$$\boldsymbol{\pi}_j(l) = \frac{1}{|I(j)|} = \frac{1}{m_j} \quad \text{for all } l \in I(j).$$

The advantage of this update scheme is that, provided there exists y' such that $\boldsymbol{y}_0^{(j)}(l) = {y'}_0^{(j)}$ $\forall l \in I(j)$, we have, $\forall l \in I(j), \forall k \geq 0$,

$$\boldsymbol{y}_{k+1}^{(j)}(l) = {y'}_{k+1}^{(j)} = \frac{1}{m_j} \overline{y}_{k+1}^{(j)} + \left(1 - \frac{1}{m_j}\right) {y'}_k^{(j)}.$$

Hence, choosing $\mathcal{J}(i) = \bigcup_{j \in J(i)} I(j) \times J(i)$ allows us to undo the duplication of dual variables and reduce the size of the vector of dual variables from the number of nonzero elements in M, $\mathrm{nnz}(M)$, to its number of rows, p.

This shows the following equivalence result.

Proposition 3. *Algorithm* 1 *with initial point* y'_0 *is equivalent to Algorithm* 2 *with the choice of dual sampling* $\mathcal{J}(i) = \bigcup_{j \in J(i)} I(j) \times J(i)$ $\forall i \in \{1, \dots, n\}$ *and initial point* $\boldsymbol{y}_0^{(j)}(l) = {y'_0}^{(j)}$ $\forall j \in \{1, \dots, p\}, \forall l \in I(j)$.

So, a byproduct of the proof of Theorem 2 will be a proof for Theorem 1.

2.5.3. The case in which $\boldsymbol{m_1 = \cdots = m_p = 1}$.

We consider the special case in which $m_1 = \cdots = m_p = 1$. In other words, the linear operator M has a single nonzero component $M_{j,i}$ per row $j \in \{1, \dots, p\}$. This happens, for instance, in the

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

context of distributed optimization [7]. This case will also be extensively used in the proofs.

In this scenario, the notations can be drastically simplified. Indeed, for every $j \in \{1, \dots, p\}$, $I(j)$ is a singleton. The corresponding set of duplicated dual variables $(\boldsymbol{y}_k^{(j)}(i) : i \in I(j))$ is reduced to a single variable $\boldsymbol{y}_k^{(j)}(I(j))$, which we shall simply denote as $y_k^{(j)}$. According to (4), $\mathcal{J}(i)$ is a subset of $\{(l, j) : l \in I(j)\}$ which simply coincides with the set $\{(I(j), j) : j \in \{1, \dots, p\}\}$. Therefore, the set $\mathcal{J}(i)$ is uniquely determined by its projection onto the second set of indices. In other words, the selection of $\mathcal{J}(i)$ for a given i is equivalent to the selection of a subset of $\{1, \dots, p\}$ which we abusively denote by $\mathcal{J}(i)$ in this paragraph.

Then, Algorithm 2 simplifies to Algorithm 3. Note that Algorithm 3 has a range of applicability which is different from Algorithm 1. We make an additional assumption on M but we have more freedom on the dual sampling $\mathcal{J}$.

Algorithm 3 Coordinate-descent primal-dual algorithm: The case in which $m_1 = \cdots = m_p = 1$.

Initialization: Choose $x_0 \in \mathcal{X}$, $y_0 \in \mathcal{Y}$.
Iteration k: Define

$$\begin{aligned}\overline{y}_{k+1} &= \operatorname{prox}_{\sigma, h^\star}\big(y_k + D(\sigma) M x_k\big),\\ \overline{x}_{k+1} &= \operatorname{prox}_{\tau, g}\Big(x_k - D(\tau)\big(\nabla f(x_k) + M^\star(2\overline{y}_{k+1} - y_k)\big)\Big).\end{aligned}$$

For $i = i_{k+1}$ and for each $j \in \mathcal{J}(i_{k+1})$, update as follows:

$$\begin{aligned}x_{k+1}^{(i)} &= \overline{x}_{k+1}^{(i)},\\ y_{k+1}^{(j)} &= y_k^{(j)} + \pi_j(\overline{y}_{k+1}^{(j)} - y_k^{(j)}).\end{aligned}$$

Otherwise, set $x_{k+1}^{(i')} = x_k^{(i')}$, $y_{k+1}^{(j')} = y_k^{(j')}$.

2.5.4. The case in which $h = 0$. Instantiating Algorithm 2 in the special case in which $h = 0$, it boils down to the following CD forward-backward algorithm:

$$x_{k+1}^{(i)} = \begin{cases} \operatorname{prox}_{\tau, g}^{(i)}\big(x_k - D(\tau)\nabla f(x_k)\big) & \text{if } i = i_{k+1},\\ x_k^{(i)} & \text{otherwise.}\end{cases} \tag{7}$$

As a consequence, Algorithm 2 allows us to recover the CD proximal gradient algorithm of [41] with the notable difference that we do *not* assume the separability of g. On the other hand, Assumption 1(e) becomes $\tau_i < 1/\beta_i$, whereas in the separable case Richtárik and Takáč [41] assume $\tau_i = 1/\beta_i$. This remark leads us to conjecture that, even though Assumption 1(e) generally allows for the use of larger step sizes than the ones suggested by the approach in [16, 7], one might be able to use even larger step sizes than those allowed by Theorem 2.

Note that a similar CD forward-backward algorithm can be found in [16] with no need to require the separability of g. However, the algorithm in [16] assumes that the step size τ_i (there assumed to be independent of i) is less than $2/\beta$, where β is the *global* Lipschitz constant of ∇f. As discussed in the introduction, an attractive feature of our algorithm is the fact that our convergence condition $\tau_i < 1/\beta_i$ only involves the coordinatewise Lipschitz constant of ∇f.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

2.6. Failure of stochastic Fejér monotonicity. As discussed in the introduction, an existing approach to prove convergence of CD algorithm in a general setting (that is, not restricted to $h = 0$ and separable g) is to establish the stochastic Fejér monotonicity of the iterates. The idea was used in [32] and extended in [16, 7] to a more general setting. Unfortunately, this approach implies to selecting a "small" step size as noted in the previous section. The use of a small step size is unfortunate in practice, as it may significantly affect the convergence rate.

It is natural to ask whether the existing convergence proof based on stochastic Fejér monotonicity can be extended to the use of larger step sizes. The answer is negative, as shown by the following example.

Example 2. Consider the toy problem

$$\min_{x\in\mathbb{R}^3} \frac{1}{2}(x^{(1)} + x^{(2)} + x^{(3)} - 1)^2,$$

i.e., we take $f(x) = \frac{1}{2}(x^{(1)} + x^{(2)} + x^{(3)} - 1)^2$ and $g = h = M = 0$. One of the minimizers is $x_* = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. The global Lipschitz constant of ∇f is equal to 3 and the coordinatewise Lipschitz constants are equal to 1. The CD proximal gradient algorithm (7) writes

$$x_{k+1}^{(i)} = \begin{cases} x_k^{(i)} - \tau(x_k^{(1)} + x_k^{(2)} + x_k^{(3)} - 1) & \text{if } i = i_{k+1}, \\ x_k^{(i)} & \text{otherwise,} \end{cases}$$

where we used $\tau_1 = \tau_2 = \tau_3 \triangleq \tau$ for simplicity. By Theorem 2, x_k converges almost surely to x_* whenever $\tau < 1$. Setting $x_0 = 0$, one has $\|x_0 - x_*\|^2 = \frac{1}{3}$. It can immediately be seen that $\mathbb{E}\|x_1 - x_*\|^2 = (\tau - \frac{1}{3})^2 + \frac{1}{9} + \frac{1}{9}$, where $\mathbb{E}$ represents the expectation. In particular, $\mathbb{E}\|x_1 - x_*\|^2 > \|x_0 - x_*\|^2$ as soon as $\tau > \frac{2}{3}$. Therefore, the sequence $\mathbb{E}\|x_k - x_*\|^2$ is not decreasing. This example shows that the proof techniques based on monotone operators and Fejér monotonicity are not directly applicable in the case of long step sizes. Indeed, as shown in Lemma 6, one needs to make use of another Lyapunov function, defined in the inequality (19), which shows that the sequence exhibits a stochastic monotonicity property in the Bregman divergence sense [1].

3. Proof of Theorem 2.

3.1. Preliminary lemma. For every $(x, y) \in \mathcal{X} \times \mathcal{Y}$, we define

$$V(x,y) := \frac{1}{2}\|x\|^2_{\tau^{-1}} + \langle y, Mx\rangle + \frac{1}{2}\|y\|^2_{\sigma^{-1}}. \tag{8}$$

Lemma 4. *Let Assumption* 1(a) and (b) *hold true. Let* $(x, y) \in \mathcal{X} \times \mathcal{Y}$ *and* $(x_*, y_*) \in \mathcal{S}$. *Define*

$$\overline{y} = \operatorname{prox}_{\sigma,h^\star}\big(y + D(\sigma)Mx\big),$$
$$\overline{x} = \operatorname{prox}_{\tau,g}\Big(x - D(\tau)\big(\nabla f(x) + M^\star(2\overline{y} - y)\big)\Big),$$

and set $z = (x, y)$, $z_* = (x_*, y_*)$, $\overline{z} = (\overline{x}, \overline{y})$. *Then,*

$$\langle \nabla f(x_*) - \nabla f(x), x_* - \overline{x}\rangle + V(\overline{z} - z) \le V(z - z_*) - V(\overline{z} - z_*).$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Proof. The inclusions (3) also read

$$\begin{aligned} &\forall u \in \mathcal{X}, \quad g(u) \geq g(x_*) + \langle -\nabla f(x_*) - M^\star y_*, u - x_* \rangle, \\ &\forall v \in \mathcal{Y}, \quad h^\star(v) \geq h^\star(y_*) + \langle M x_*, v - y_* \rangle. \end{aligned}$$

Setting $u = \overline{x}$ and $v = \overline{y}$ in the above inequalities, we obtain

$$g(\overline{x}) \geq g(x_*) + \langle \nabla f(x_*) + M^\star y_*, x_* - \overline{x} \rangle, \tag{9}$$

$$h^\star(\overline{y}) \geq h^\star(y_*) + \langle M x_*, \overline{y} - y_* \rangle. \tag{10}$$

By definition of the proximal operator,

$$\overline{y} = \arg\min_{v \in \mathcal{Y}} h^\star(v) - \langle v, Mx \rangle + \frac{1}{2}\|v - y\|^2_{\sigma^{-1}}, \tag{11}$$

$$\overline{x} = \arg\min_{u \in \mathcal{X}} g(u) + \langle u, \nabla f(x) + M^\star(2\overline{y} - y) \rangle + \frac{1}{2}\|u - x\|^2_{\tau^{-1}}. \tag{12}$$

Consider equality (11) above. It classically implies [49, Property 1] the following three-point identity: For any $v \in \mathcal{Y}$,

$$h^\star(\overline{y}) - \langle \overline{y}, Mx \rangle + \frac{1}{2}\|\overline{y} - y\|^2_{\sigma^{-1}} \leq h^\star(v) - \langle v, Mx \rangle + \frac{1}{2}\|v - y\|^2_{\sigma^{-1}} - \frac{1}{2}\|\overline{y} - v\|^2_{\sigma^{-1}}. \tag{13}$$

Setting $v = y_*$, we obtain

$$h^\star(\overline{y}) \leq h^\star(y_*) + \langle \overline{y} - y_*, Mx \rangle + \frac{1}{2}\|y_* - y\|^2_{\sigma^{-1}} - \frac{1}{2}\|\overline{y} - y_*\|^2_{\sigma^{-1}} - \frac{1}{2}\|\overline{y} - y\|^2_{\sigma^{-1}}, \tag{14}$$

and using (10) we finally have

$$\langle M(x_* - x), \overline{y} - y_* \rangle \leq \frac{1}{2}\|y_* - y\|^2_{\sigma^{-1}} - \frac{1}{2}\|\overline{y} - y_*\|^2_{\sigma^{-1}} - \frac{1}{2}\|\overline{y} - y\|^2_{\sigma^{-1}}. \tag{15}$$

Similarly, equality (12) implies that, for any $u \in \mathcal{X}$,

$$\begin{aligned} g(\overline{x}) + \langle \overline{x}, \nabla f(x) + M^\star(2\overline{y} - y) \rangle + \frac{1}{2}\|\overline{x} - x\|^2_{\tau^{-1}} \\ \leq g(u) + \langle u, \nabla f(x) + M^\star(2\overline{y} - y) \rangle + \frac{1}{2}\|u - x\|^2_{\tau^{-1}} - \frac{1}{2}\|\overline{x} - u\|^2_{\tau^{-1}}. \end{aligned} \tag{16}$$

We set $u = x_*$. This yields

$$\begin{aligned} g(\overline{x}) \leq g(x_*) + \langle x_* - \overline{x}, \nabla f(x) + M^\star(2\overline{y} - y) \rangle \\ + \frac{1}{2}\|x_* - x\|^2_{\tau^{-1}} - \frac{1}{2}\|\overline{x} - x_*\|^2_{\tau^{-1}} - \frac{1}{2}\|\overline{x} - x\|^2_{\tau^{-1}}. \end{aligned}$$

Moreover, using inequality (9), we obtain

$$\begin{aligned} \langle \nabla f(x_*) + M^\star y_*, x_* - \overline{x} \rangle \ \leq \langle x_* - \overline{x}, \nabla f(x) + M^\star(2\overline{y} - y) \rangle \\ + \frac{1}{2}\|x_* - x\|^2_{\tau^{-1}} - \frac{1}{2}\|\overline{x} - x_*\|^2_{\tau^{-1}} - \frac{1}{2}\|\overline{x} - x\|^2_{\tau^{-1}}; \end{aligned}$$

hence, rearranging the terms yields

$$\begin{aligned} \langle \nabla f(x_*) - \nabla f(x), x_* - \overline{x} \rangle - \frac{1}{2}\|x_* - x\|^2_{\tau^{-1}} + \frac{1}{2}\|\overline{x} - x_*\|^2_{\tau^{-1}} + \frac{1}{2}\|\overline{x} - x\|^2_{\tau^{-1}} \\ \leq \langle 2\overline{y} - y - y_*, M(x_* - \overline{x}) \rangle. \end{aligned}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Summing the above inequality with (15), we get

$$\begin{aligned}\langle \nabla f(x_*) - \nabla f(x), x_* - \overline{x}\rangle + \frac{1}{2}\|\overline{x} - x\|^2_{\tau^{-1}} + \langle \overline{y} - y, M(\overline{x} - x)\rangle + \frac{1}{2}\|\overline{y} - y\|^2_{\sigma^{-1}} \\ \leq \frac{1}{2}\|x - x_*\|^2_{\tau^{-1}} + \langle y - y_*, M(x - x_*)\rangle + \frac{1}{2}\|y - y_*\|^2_{\sigma^{-1}} \\ - \frac{1}{2}\|\overline{x} - x_*\|^2_{\tau^{-1}} - \langle \overline{y} - y_*, M(\overline{x} - x_*)\rangle - \frac{1}{2}\|\overline{y} - y_*\|^2_{\sigma^{-1}}.\end{aligned}$$

This completes the proof of the lemma thanks to the definition of V. ∎

3.2. Study of Algorithm 3. We first prove Theorem 2 in the special case in which $m_1 = \cdots = m_p = 1$. In that case, Algorithm 2 boils down to Algorithm 3. We recall that in this case the vector $\boldsymbol{y}_k^{(j)}$ is reduced to a single value $\boldsymbol{y}_k^{(j)}(i) \in \mathcal{Y}_j$, where i is the unique index such that $M_{j,i} \neq 0$. We simply denote this value by $y_k^{(j)}$.

We denote by $\mathcal{F}_k$ the filtration generated by the random variable (r.v.) $i_1, \ldots, i_k$. We denote by $\mathbb{E}_k(\cdot) = \mathbb{E}(\cdot \mid \mathcal{F}_k)$ the conditional expectation w.r.t. $\mathcal{F}_k$.

Lemma 5. *Let Assumptions* 1(a), (b), and (d) *hold true. Suppose* $m_1 = \cdots = m_p = 1$. *Consider Algorithm* 3 *and let* $\gamma_1, \ldots, \gamma_n, \gamma'_1, \ldots, \gamma'_p$ *be arbitrary positive coefficients. For every* $k \geq 1$ *and every* $\mathcal{F}_k$*-measurable pair of random variables* (X, Y) *on* $\mathcal{X} \times \mathcal{Y}$,

$$\begin{aligned}&\mathbb{E}_k(x_{k+1}) = \frac{1}{n}\overline{x}_{k+1} + \left(1 - \frac{1}{n}\right)x_k, \\ &\mathbb{E}_k(\|x_{k+1} - X\|^2_\gamma) = \frac{1}{n}\|\overline{x}_{k+1} - X\|^2_\gamma + \left(1 - \frac{1}{n}\right)\|x_k - X\|^2_\gamma, \\ &\mathbb{E}_k(\|y_{k+1} - Y\|^2_{\gamma'}) = \frac{1}{n}\|\overline{y}_{k+1} - Y\|^2_{\gamma'} + \left(1 - \frac{1}{n}\right)\|y_k - Y\|^2_{\gamma'} - \frac{1}{n}\|\overline{y}_{k+1} - y_k\|^2_{D(1-\pi)\gamma'}, \\ &\mathbb{E}_k(\langle y_{k+1} - Y, M(x_{k+1} - X)\rangle) = \frac{1}{n}\langle \overline{y}_{k+1} - Y, M(\overline{x}_{k+1} - X)\rangle \\ &\qquad + \left(1 - \frac{1}{n}\right)\langle y_k - Y, M(x_k - X)\rangle \\ &\qquad - \frac{1}{n}\langle D(1-\pi)(\overline{y}_{k+1} - y_k), M(\overline{x}_{k+1} - x_k)\rangle.\end{aligned}$$

Proof. The first equality is immediate.

Consider the second one: $\mathbb{E}_k(\|x_{k+1} - X\|^2_\gamma) = \sum_{i=1}^n \gamma_i \mathbb{E}_k(\|x_{k+1}^{(i)} - X^{(i)}\|^2)$, which coincides with

$$\sum_{i=1}^n \gamma_i \left(\frac{1}{n}\|\overline{x}_{k+1}^{(i)} - X^{(i)}\|^2 + \left(1 - \frac{1}{n}\right)\|x_k^{(i)} - X^{(i)}\|^2\right),$$

and the second equality is proved.

Similarly, for the third equality, $\mathbb{E}_k(\|y_{k+1} - Y\|^2_{\gamma'}) = \sum_{j=1}^p \gamma'_j \mathbb{E}_k(\|y_{k+1}^{(j)} - Y^{(j)}\|^2)$ and, for every j,

$$\begin{aligned}\mathbb{E}_k(\|y_{k+1}^{(j)} - Y^{(j)}\|^2) = \|y_k^{(j)} + \pi_j(\overline{y}_{k+1}^{(j)} - y_k^{(j)}) - Y^{(j)}\|^2 \mathbb{P}(j \in \mathcal{J}(i_{k+1})) \\ + \|y_k^{(j)} - Y^{(j)}\|^2 \mathbb{P}(j \notin \mathcal{J}(i_{k+1})).\end{aligned}$$

As $j \in J(i_{k+1}) \Leftrightarrow i_{k+1} \in I(j)$, we get

$$\mathbb{P}(j \in J(i_{k+1})) = \mathbb{P}(i_{k+1} \in I(j)) = \mathrm{card}(I(j))/n = 1/n.$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

From (5),

$$\pi_j = \mathbb{P}(j \in J(i_{k+1}) \mid j \in \mathcal{J}(i_{k+1})) = \frac{\mathbb{P}(j \in J(i_{k+1}) \,\&\, j \in \mathcal{J}(i_{k+1}))}{\mathbb{P}(j \in \mathcal{J}(i_{k+1}))} = \frac{\mathbb{P}(j \in J(i_{k+1}))}{\mathbb{P}(j \in \mathcal{J}(i_{k+1}))}$$

and so

$$\mathbb{P}(j \in \mathcal{J}(i_{k+1})) = \frac{1}{n\pi_j} = \frac{|\{i \,:\, j \in \mathcal{J}(i)\}|}{n}.$$

We also have

$$\begin{aligned}&\|y_k^{(j)} + \pi_j(\overline{y}_{k+1}^{(j)} - y_k^{(j)}) - Y^{(j)}\|^2 \\ &\qquad = \pi_j \|\overline{y}_{k+1}^{(j)} - Y^{(j)}\|^2 + (1-\pi_j)\|y_k^{(j)} - Y^{(j)}\|^2 - \pi_j(1-\pi_j)\|\overline{y}_{k+1}^{(j)} - y_k^{(j)}\|^2.\end{aligned}$$

This leads to

$$\mathbb{E}_k(\|y_{k+1}^{(j)} - Y^{(j)}\|^2) = \frac{1}{n}\|\overline{y}_{k+1}^{(j)} - Y^{(j)}\|^2 + \left(1 - \frac{1}{n}\right)\|y_k^{(j)} - Y^{(j)}\|^2 - \frac{1-\pi_j}{n}\|\overline{y}_{k+1}^{(j)} - y_k^{(j)}\|^2.$$

This proves the third equality.

Consider the fourth equality. Note that

$$\langle y_{k+1} - Y, M(x_{k+1} - X)\rangle = \sum_{i=1}^{n} \sum_{j \in J(i)} \langle y_{k+1}^{(j)} - Y^{(j)}, M_{j,i}(x_{k+1}^{(i)} - X^{(i)})\rangle.$$

For any pair (i, j) such that $j \in J(i)$, the conditional expectation of each term in the sum is equal to

$$\begin{aligned}&\frac{1}{n}\langle \pi_j \overline{y}_{k+1}^{(j)} + (1-\pi_j) y_k^{(j)} - Y^{(j)}, M_{j,i}(\overline{x}_{k+1}^{(i)} - X^{(i)})\rangle \\ &\quad + \left(\frac{1}{n\pi_j} - \frac{1}{n}\right)\langle \pi_j \overline{y}_{k+1}^{(j)} + (1-\pi_j) y_k^{(j)} - Y^{(j)}, M_{j,i}(x_k^{(i)} - X^{(i)})\rangle \\ &\quad + \left(1 - \frac{1}{n\pi_j}\right)\langle y_k^{(j)} - Y^{(j)}, M_{j,i}(x_k^{(i)} - X^{(i)})\rangle \\ &= \frac{\pi_j}{n}\langle \overline{y}_{k+1}^{(j)} - Y^{(j)}, M_{j,i}(\overline{x}_{k+1}^{(i)} - X^{(i)})\rangle \\ &\quad + \left(1 - \frac{2}{n} + \frac{\pi_j}{n}\right)\langle y_k^{(j)} - Y^{(j)}, M_{j,i}(x_k^{(i)} - X^{(i)})\rangle \\ &\quad + \left(\frac{1}{n} - \frac{\pi_j}{n}\right)\langle y_k^{(j)} - Y^{(j)}, M_{j,i}(\overline{x}_{k+1}^{(i)} - X^{(i)})\rangle \\ &\quad + \left(\frac{1}{n} - \frac{\pi_j}{n}\right)\langle \overline{y}_{k+1}^{(j)} - Y^{(j)}, M_{j,i}(x_k^{(i)} - X^{(i)})\rangle \\ &= \frac{1}{n}\langle \overline{y}_{k+1}^{(j)} - Y^{(j)}, M_{j,i}(\overline{x}_{k+1}^{(i)} - X^{(i)})\rangle \\ &\quad + \left(1 - \frac{1}{n}\right)\langle y_k^{(j)} - Y^{(j)}, M_{j,i}(x_k^{(i)} - X^{(i)})\rangle \\ &\quad + \left(\frac{1}{n} - \frac{\pi_j}{n}\right)\langle y_k^{(j)} - \overline{y}_{k+1}^{(j)}, M_{j,i}(\overline{x}_{k+1}^{(i)} - X^{(i)})\rangle \\ &\quad + \gamma_j'\left(\frac{1}{n} - \frac{\pi_j}{n}\right)\langle \overline{y}_{k+1}^{(j)} - y_k^{(j)}, M_{j,i}(x_k^{(i)} - X^{(i)})\rangle\end{aligned}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

$$\begin{aligned}
&= \frac{1}{n}\langle \overline{y}_{k+1}^{(j)} - Y^{(j)}, M_{j,i}(\overline{x}_{k+1}^{(i)} - X^{(i)})\rangle \\
&\quad + \left(1 - \frac{1}{n}\right)\langle y_k^{(j)} - Y^{(j)}, M_{j,i}(x_k^{(i)} - X^{(i)})\rangle \\
&\quad + \left(\frac{1}{n} - \frac{\pi_j}{n}\right)\langle y_k^{(j)} - \overline{y}_{k+1}^{(j)}, M_{j,i}(\overline{x}_{k+1}^{(i)} - x_k^{(i)})\rangle.
\end{aligned}$$

Finally, we obtain

$$\begin{aligned}
\mathbb{E}(\langle y_{k+1} - Y, M(x_{k+1} - X)\rangle) &= \frac{1}{n}\langle \overline{y}_{k+1} - Y, M(\overline{x}_{k+1} - X)\rangle \\
&\quad + \left(1 - \frac{1}{n}\right)\langle y_k - Y, M(x_k - X)\rangle - \frac{1}{n}\langle D(1-\pi)(\overline{y}_{k+1} - y_k), M(\overline{x}_{k+1} - x_k)\rangle,
\end{aligned}$$

which in turn implies the fourth equality in the lemma. □

Assume that $\tau_i^{-1} > \beta_i$ for each $i \in \{1, \dots, n\}$. Define, for every $z = (x, y) \in \mathcal{X} \times \mathcal{Y}$,

$$\tilde{V}(z) = \tilde{V}(x, y) := \frac{1}{2}\|x\|^2_{\tau^{-1}-\beta} + \langle D(2-\pi)y, Mx\rangle + \frac{1}{2}\|y\|^2_{\sigma^{-1}(2-\pi)}. \tag{17}$$

Lemma 6. *Let Assumptions* 1(a)–(d) *hold true. Suppose* $m_1 = \cdots = m_p = 1$ *and assume that* $\tau_i^{-1} > \beta_i$ *for each* $i \in \{1, \dots, n\}$*. Consider Algorithm* 3 *and define, for every* $k \in \mathbb{N}$,

$$S_{k,*} := f(x_k) - f(x_*) - \langle \nabla f(x_*), x_k - x_*\rangle. \tag{18}$$

Then the following inequality holds:

$$\mathbb{E}_k\left[S_{k+1,*} + V(z_{k+1} - z_*)\right] \le \left(1 - \frac{1}{n}\right)S_{k,*} + V(z_k - z_*) - \frac{1}{n}\tilde{V}(\overline{z}_{k+1} - z_k), \tag{19}$$

where $\overline{z}_{k+1} = (\overline{x}_{k+1}, \overline{y}_{k+1})$.

Proof. We can write the relations of Lemma 5 as

$$\begin{aligned}
\|\overline{x}_{k+1} - X\|^2_{\tau^{-1}} &= n\mathbb{E}_k(\|x_{k+1} - X\|^2_{\tau^{-1}}) - (n-1)\|x_k - X\|^2_{\tau^{-1}}, \\
\|\overline{y}_{k+1} - Y\|^2_{\sigma^{-1}} &= n\mathbb{E}_k(\|y_{k+1} - Y\|^2_{\sigma^{-1}}) \\
&\quad - (n-1)\|y_k - Y\|^2_{\sigma^{-1}} + \|\overline{y}_{k+1} - y_k\|^2_{\sigma^{-1}(1-\pi)}, \\
\langle \overline{y}_{k+1} - Y, M(\overline{x}_{k+1} - X)\rangle &= n\mathbb{E}_k(\langle y_{k+1} - Y, M(x_{k+1} - X)\rangle) \\
&\quad - (n-1)\langle y_k - Y, M(x_k - X)\rangle \\
&\quad + \langle D(1-\pi)(\overline{y}_{k+1} - y_k), M(\overline{x}_{k+1} - x_k)\rangle.
\end{aligned}$$

Choosing $Z = (X, Y)$ and defining $z_k = (x_k, y_k)$ and $\overline{z}_k = (\overline{x}_k, \overline{y}_k)$, we obtain

$$\begin{aligned}
V(\overline{z}_{k+1} - Z) &= n\mathbb{E}_k(V(z_{k+1} - Z)) - nV(z_k - Z) + V(z_k - Z) \\
&\quad + \frac{1}{2}\|\overline{y}_{k+1} - y_k\|^2_{\sigma^{-1}(1-\pi)} + \langle D(1-\pi)(\overline{y}_{k+1} - y_k), M(\overline{x}_{k+1} - x_k)\rangle.
\end{aligned} \tag{20}$$

We shall define

$$R_\pi := \frac{1}{2}\|\overline{y}_{k+1} - y_k\|^2_{\sigma^{-1}(1-\pi)} + \langle D(1-\pi)(\overline{y}_{k+1} - y_k), M(\overline{x}_{k+1} - x_k)\rangle. \tag{21}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Let $z_* = (x_*, y_*) \in \mathcal{S}$. By Lemma 4,

$$\langle \nabla f(x_*) - \nabla f(x_k), x_* - \overline{x}_{k+1}\rangle + V(\overline{z}_{k+1} - z_k) \le V(z_k - z_*) - V(\overline{z}_{k+1} - z_*).$$

Identifying Z in (20) to z_* and z_k successively, we obtain

$$\begin{aligned}\langle \nabla f(x_*) - \nabla f(x_k), x_* - \overline{x}_{k+1}\rangle &+ n\mathbb{E}_k(V(z_{k+1} - z_k)) \\ &\le nV(z_k - z_*) - n\mathbb{E}_k(V(z_{k+1} - z_*)) - 2R_\pi.\end{aligned}$$

Dividing both sides of the above inequality by n and using that $\overline{x}_{k+1} = n\mathbb{E}_k(x_{k+1}) - (n-1)x_k$, we obtain

$$\begin{aligned}\langle \nabla f(x_*) - \nabla f(x_k), x_* - \mathbb{E}_k(x_{k+1}) + \left(1 - \frac{1}{n}\right)(x_k - x_*)\rangle + \mathbb{E}_k(V(z_{k+1} - z_k)) \\ \le V(z_k - z_*) - \mathbb{E}_k(V(z_{k+1} - z_*)) - \frac{2}{n}R_\pi.\end{aligned}$$

Rearranging the terms,

$$\begin{aligned}(22)\quad & \mathbb{E}_k\left[\langle \nabla f(x_k) - \nabla f(x_*), x_{k+1} - x_k\rangle + V(z_{k+1} - z_*)\right] \\ &\le -\frac{1}{n}\langle \nabla f(x_k) - \nabla f(x_*), x_k - x_*\rangle + V(z_k - z_*) - \mathbb{E}_k(V(z_{k+1} - z_k)) - \frac{2}{n}R_\pi.\end{aligned}$$

We now use Assumption 1(c), knowing that x_{k+1} only differs from x_k along coordinate i_{k+1}:

$$\begin{aligned}f(x_{k+1}) &\le f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k\rangle + \frac{\beta_{i_{k+1}}}{2}\|x_{k+1} - x_k\|^2 \\ &= f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k\rangle + \frac{1}{2}\|x_{k+1} - x_k\|_\beta^2, \qquad (23)\end{aligned}$$

which implies that $\langle \nabla f(x_k), x_{k+1} - x_k\rangle \ge f(x_{k+1}) - f(x_k) - \frac{1}{2}\|x_{k+1} - x_k\|_\beta^2$. Thus, plugging this into (22),

$$\begin{aligned}\mathbb{E}_k & \left[f(x_{k+1}) - f(x_k) - \frac{1}{2}\|x_{k+1} - x_k\|_\beta^2 - \langle \nabla f(x_*), x_{k+1} - x_k\rangle + V(z_{k+1} - z_*)\right] \\ &\le -\frac{1}{n}\langle \nabla f(x_k) - \nabla f(x_*), x_k - x_*\rangle + V(z_k - z_*) - \mathbb{E}_k(V(z_{k+1} - z_k)) - \frac{2}{n}R_\pi.\end{aligned}$$

Introducing the quantity $S_{k,*}$ as in (18), the inequality simplifies to

$$\begin{aligned}\mathbb{E}_k & \Big[S_{k+1,*} + V(z_{k+1} - z_*) - \frac{1}{2}\|x_{k+1} - x_k\|_\beta^2\Big] \\ &\le f(x_k) - f(x_*) - \left(1 - \frac{1}{n}\right)\langle \nabla f(x_*), x_k - x_*\rangle - \frac{1}{n}\langle \nabla f(x_k), x_k - x_*\rangle \\ &\quad + V(z_k - z_*) - \mathbb{E}_k(V(z_{k+1} - z_k)) - \frac{2}{n}R_\pi.\end{aligned}$$

An estimate of the right-hand side is obtained upon noticing that $\langle \nabla f(x_k), x_k - x_*\rangle \ge f(x_k) - f(x_*)$. Therefore,

$$\begin{aligned}\mathbb{E}_k & \big[S_{k+1,*} + V(z_{k+1} - z_*) - \frac{1}{2}\|x_{k+1} - x_k\|_\beta^2\big] \\ &\le \left(1 - \frac{1}{n}\right)S_{k,*} + V(z_k - z_*) - \mathbb{E}_k(V(z_{k+1} - z_k)) - \frac{2}{n}R_\pi.\end{aligned}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Using Lemma 5, (17), and (21), it is immediate that

$$\begin{aligned}
&\mathbb{E}_k(V(z_{k+1}-z_k) - \frac{1}{2}\|x_{k+1}-x_k\|_\beta^2) + \frac{2}{n}R_\pi \\
&\qquad = \frac{1}{n}V(\overline{z}_{k+1}-z_k) - \frac{1}{n}R_\pi - \frac{1}{2n}\left\|\overline{x}_{k+1}^{(j)} - x_k\right\|_\beta^2 + \frac{2}{n}R_\pi \\
&\qquad = \frac{1}{n}\tilde{V}(\overline{z}_{k+1}-z_k)
\end{aligned}$$

and the proof is complete. □

Recall that we denote by $\rho(A)$ the spectral radius of a matrix A.

LEMMA 7. *Suppose that $m_1 = \cdots = m_p = 1$ and assume that the following condition holds for every $i \in \{1,\dots,n\}$:*

$$\tau_i < \frac{1}{\beta_i + \rho\left(\sum_{j\in J(i)}(2-\pi_j)\sigma_j M_{j,i}^\star M_{j,i}\right)}. \tag{24}$$

Then $\tilde{V}^{1/2}$ is a norm on $\mathcal{X}\times\mathcal{Y}$.

Note that under the assumptions of Lemma 7, $V^{1/2}$ is also, a fortiori, a norm, but that $V^{1/2}$ need not be a norm.

Proof. Let $\gamma^{-1} = \tau^{-1} - \beta$. Denote by $\sigma'_j = (2-\pi_j)\sigma_j$ $\forall j$ and by $D(\sigma')$ the diagonal matrix on $\mathcal{Y}\to\mathcal{Y}$ defined by $D(\sigma')(y) := (\sigma'_1 y^{(1)},\dots,\sigma'_p y^{(p)})$ for every $y = (y^{(1)},\dots,y^{(p)})$. We define $D(\gamma)$ similarly on $\mathcal{X}\to\mathcal{X}$. By [31, Theorem 7.7.6], a sufficient (and necessary) condition for $\tilde{V}$ to be a squared norm is that $D(\gamma^{-1}) \succ M^\star D(\sigma')M$ (where the notation $A \succ B$ means that $A - B$ is a positive definite matrix). Defining $R = D({\sigma'}^{1/2})MD(\gamma^{1/2})$ (that is, $R_{j,i} = (\gamma_i\sigma'_j)^{1/2}M_{j,i}$ for every j,i), the condition equivalently reads $\rho(R^\star R) < 1$. As the set $I(j)$ is reduced to a unique element for all j, the matrix $R^\star R$ is (block) diagonal. Precisely, for any $1 \le i,\ell \le n$, the (i,ℓ)-component $(R^\star R)_{i,\ell}$ is zero whenever $i \neq \ell$ and is equal to $(R^\star R)_{i,i} = \gamma_i \sum_{j\in J(i)} \sigma'_j M_{j,i}^\star M_{j,i}$ otherwise. The condition $\rho(R^\star R) < 1$ yields $\gamma_i \rho(\sum_{j\in J(i)}\sigma'_j M_{j,i}^\star M_{j,i}) < 1$ for each $i \in \{1,\dots,n\}$, which is in turn equivalent to (24). □

Proof of Theorem 1 *in the case in which $m_1 = \cdots = m_p = 1$.* Let z_* be an arbitrary point in $\mathcal{S}$. Whenever condition (24) is met, the r.v.'s $V(z_k - z_*)$ and $\tilde{V}(\overline{z}_{k+1}-z_k)$ are nonnegative. The r.v. $S_{k,*}$ is nonnegative as well, by convexity of f. We review two important consequences of Lemma 6.

- Define $U_k := S_{k,*} + V(z_k - z_*)$. A first consequence of Lemma 6 is that, for all k,

$$\mathbb{E}_k(U_{k+1}) \le U_k - \frac{1}{n}S_{k,*}.$$

Recalling that U_k and S_k are nonnegative r.v.'s, the Robbins–Siegmund lemma [44] implies that almost surely $\lim_{k\to\infty} U_k$ exists and $\sum_k S_{k,*} < \infty$. In particular, $S_{k,*}$ converges almost surely to zero. By definition of U_k, this implies that $\lim_{k\to\infty} V(z_k - z_*)$ exists almost surely.

We proceed as in [4, Proposition 9] (see also [32], [16, Proposition 2.3]). As the reasoning depends on z_*, we can be more precise and write that there exists an event A_{z_*} of probability 1 such that, for all $\omega \in A_{z_*}$, $\lim_{k\to\infty} V(z_k(\omega) - z_*)$ exists.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Let $\{v_i\}_{i\in\mathbb{N}}$ be a countable subset of the relative interior $\mathrm{ri}(\mathcal{S})$ that is dense in $\mathcal{S}$. From what we proved beforehand, $\forall i \in \mathbb{N}, \forall \omega \in A_{v_i}$, $\lim_{k\to\infty} V(z_k(\omega)-v_i)$ exists. The intersection $A = \bigcap_{i\in\mathbb{N}} A_{v_i}$ has probability 1 since it is the intersection of a countable number of set of probability 1.

Let us now fix $\check z \in \mathcal{S}$. There exists a subsequence $(v_{\phi(i)})$ such that $v_{\phi(i)} \to \check z$. We also fix $\omega \in A$. By the triangle inequality, $\forall k \in \mathbb{N}, \forall i \in \mathbb{N}$,

$$\begin{aligned}
-V^{1/2}(v_{\phi(i)} - \check z) &\le V^{1/2}(z_k(\omega) - \check z) - V^{1/2}(z_k(\omega) - v_{\phi(i)}) \\
&\le V^{1/2}(v_{\phi(i)} - \check z).
\end{aligned}$$

Therefore,

$$\begin{aligned}
-V^{1/2}(v_{\phi(i)} - \check z) &\le \liminf_{k\to+\infty} V^{1/2}(z_k(\omega) - \check z) - \lim_{k\to+\infty} V^{1/2}(z_k(\omega) - v_{\phi(i)}) \\
&\le \limsup_{k\to+\infty} V^{1/2}(z_k(\omega) - \check z) - \lim_{k\to+\infty} V^{1/2}(z_k(\omega) - v_{\phi(i)}) \\
&\le V^{1/2}(v_{\phi(i)} - \check z).
\end{aligned}$$

Taking limits as $i \to +\infty$, we obtain that

$$\begin{aligned}
\liminf_{k\to+\infty} V^{1/2}(z_k(\omega) - \check z) &= \limsup_{k\to+\infty} V^{1/2}(z_k(\omega) - \check z) \\
&= \lim_{i\to+\infty} \lim_{k\to+\infty} V^{1/2}(z_k(\omega) - v_{\phi(i)}),
\end{aligned}$$

and so the limit exists.

Summing up, there exists an event A of probability 1 such that for every $\omega \in A$ and every $\check z \in \mathcal{S}$, $\lim_{k\to\infty} V^{1/2}(z_k(\omega) - \check z)$ exists.

• A second consequence of Lemma 6 is that, by taking the expectation $\mathbb{E}$ of both sides of (19),

$$\mathbb{E}\left[S_{k+1,*} + V(z_{k+1} - z_*)\right] \le \mathbb{E}[S_{k,*} + V(z_k - z_*)] - \frac{1}{n}\mathbb{E}(\tilde V(\overline{z}_{k+1} - z_k))$$

and by summing these inequalities, we obtain

$$0 \le S_{0,*} + V(z_0 - z_*) - \frac{1}{n}\sum_{i=0}^{k} \mathbb{E}(\tilde V(\overline{z}_{i+1} - z_i)). \tag{25}$$

Thus, $\mathbb{E}(\sum_{i=0}^{\infty} \tilde V(\overline{z}_{i+1} - z_i)) < \infty$. The integrand is nonnegative by Lemma 7. It is therefore finite almost everywhere. In particular, the sequence $\tilde V(\overline{z}_{k+1} - z_k)$ converges almost surely to zero. By Lemma 7, $\overline{z}_{k+1} - z_k$ converges to zero almost surely. Say $\overline{z}_{k+1}(\omega) - z_k(\omega) \to 0$ for every $\omega \in B$, where B is a probability event of probability 1.

We introduce the mapping $T : \mathcal{X} \times \mathcal{Y} \to \mathcal{X} \times \mathcal{Y}$ such that for any $(x, y) \in \mathcal{X} \times \mathcal{Y}$, the quantity $T(x, y)$ coincides with the couple $(\overline{x}, \overline{y})$ given by

$$\begin{aligned}
\overline{y} &= \mathrm{prox}_{\sigma,h^\star}\big(y + D(\sigma)Mx\big), \\
\overline{x} &= \mathrm{prox}_{\tau,g}\big(x - D(\tau)\nabla f(x) - D(\tau)M^\star(2\overline{y} - y)\big).
\end{aligned}$$

With this definition, $\overline{z}_{k+1} = T(z_k)$. By nonexpansiveness of the proximity operator, it is straightforward to show that T is continuous. It is also straightforward to verify that its set of fixed points coincides with $\mathcal{S}$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

We select a fixed $\omega \in A \cap B$. Note that $z_k(\omega)$ is a bounded sequence. Let $\tilde{z}$ be a cluster point of the latter. We have shown that $T(z_k(\omega)) - z_k(\omega) \to 0$, which implies that $T(\tilde{z}) - \tilde{z} = 0$ by continuity of T. Thus, $\tilde{z} \in \mathcal{S}$. This implies that $\lim_{k\to\infty} V^{1/2}(z_k(\omega) - \tilde{z})$ exists. Since $V^{1/2}(z_k(\omega) - \tilde{z})$ tends to zero at least on some subsequence, we conclude that $\lim_{k\to\infty} V^{1/2}(z_k(\omega) - \tilde{z}) = 0$. In other words, the sequence $z_k(\omega)$ converges to some point $\tilde{z} \in \mathcal{S}$. This completes the proof of Theorem 2 in the case in which $m_1 = \cdots = m_p = 1$. $\square$

3.3. General case. For every $j \in \{1, \dots, p\}$, $\boldsymbol{\mathcal{Y}}_j = \mathcal{Y}_j^{I(j)}$ is equipped with the inner product $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \sum_{i \in I(j)} \langle \boldsymbol{u}(i), \boldsymbol{v}(i) \rangle$. The space $\boldsymbol{\mathcal{Y}}_j$ stores $I(j)$ duplicates of the original problem's jth dual variable y_j. We introduce the averaging operator $S_j : \boldsymbol{\mathcal{Y}}_j \to \mathcal{Y}_j$ defined for every $\boldsymbol{u} \in \boldsymbol{\mathcal{Y}}_j$ by

$$S_j(\boldsymbol{u}) := \frac{1}{m_j} \sum_{i \in I(j)} \boldsymbol{u}(i).$$

The averaging operators allow us to come back from duplicated dual variables to actual dual variables. For any $u \in \mathcal{Y}_j$, we denote by $\mathbf{1}_{m_j} \otimes u = (u, \dots, u)$ the vector of $\boldsymbol{\mathcal{Y}}_j$ whose components all coincide with u.

We introduce the linear operator $K_j : \mathcal{X} \to \boldsymbol{\mathcal{Y}}_j$ by

$$K_j(x) = (M_{j,i}(x^{(i)}) \,:\, i \in I(j)).$$

The operators $S : \boldsymbol{\mathcal{Y}} \to \mathcal{Y}$, $K : \mathcal{X} \to \boldsymbol{\mathcal{Y}}$ are respectively defined by $S(\boldsymbol{y}) := (S_1(\boldsymbol{y}^{(1)}), \dots, S_p(\boldsymbol{y}^{(p)}))$ and $K(x) := (K_1(x), \dots, K_p(x))$. It is immediate to verify that

$$M = D(m)SK, \tag{26}$$

where $m = (m_1, \dots, m_p)$. In order to gain some insights, the following example illustrates the construction of K for a given M.

Example 3. Let $\mathcal{X} = \mathcal{Y} = \mathbb{R}^3$ and define $M : \mathcal{X} \to \mathcal{Y}$ as the 3×3 matrix

$$M = \begin{pmatrix} M_{1,1} & M_{1,2} & 0 \\ 0 & M_{2,2} & 0 \\ M_{3,1} & M_{3,2} & M_{3,3} \end{pmatrix}.$$

Here, $I(1) = \{1, 2\}$ is the set of nonzero coefficients of the first row of M and it cardinal is $m_1 = 2$. Similarly, $m_2 = 1$, $m_3 = 3$, and $\boldsymbol{\mathcal{Y}} = \mathbb{R}^6$. Then $K : \mathbb{R}^3 \to \mathbb{R}^6$ coincides with the matrix

$$K = \begin{pmatrix} M_{1,1} & 0 & 0 \\ 0 & M_{1,2} & 0 \\ 0 & M_{2,2} & 0 \\ M_{3,1} & 0 & 0 \\ 0 & M_{3,2} & 0 \\ 0 & 0 & M_{3,3} \end{pmatrix}$$

and each row of K contains exactly one nonzero coefficient. On the other hand, S and $D(m)$ respectively coincide with

$$S = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix} \qquad \text{and} \qquad D(m) = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 3 \end{pmatrix}$$

and obviously $D(m)SK = M$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

We define the function $\overline{h} := h \circ (D(m)S)$. By (26), problem (1) is equivalent to

$$\min_{x \in \mathcal{X}} f(x) + g(x) + \overline{h}(Kx). \tag{27}$$

We denote by $\boldsymbol{\mathcal{S}}$ the set of primal-dual solutions of the above problem, i.e., the set of pairs $(x_*, \boldsymbol{y}_*) \in \mathcal{X} \times \boldsymbol{\mathcal{Y}}$ satisfying

$$\begin{aligned} &0 \in \nabla f(x_*) + \partial g(x_*) + K^\star \boldsymbol{y}_*, \\ &0 \in -Kx_* + \partial \overline{h}^\star(\boldsymbol{y}_*). \end{aligned}$$

Substituting M with K, we may now apply Algorithm 3 to (27). For a fixed parameter $\sigma = (\sigma_1, \dots, \sigma_p)$, we define $\tilde{\sigma}_j := m_j \sigma_j$ and we define $\tilde{\sigma} \in \mathbb{R}^{\sum_{j=1}^p m_j}$ as the vector $\tilde{\sigma} := (\tilde{\sigma}_1 \mathbf{1}_{m_1}, \dots, \tilde{\sigma}_p \mathbf{1}_{m_p})$, where $\mathbf{1}_{m_j}$ is a vector of size m_j whose components are all equal to 1. Algorithm 3 gives the following.

Initialization: Choose $x_0 \in \mathcal{X}$, $\boldsymbol{y}_0 \in \boldsymbol{\mathcal{Y}}$.
Iteration k: Define

$$\overline{\boldsymbol{y}}_{k+1} = \operatorname{prox}_{\tilde{\sigma}, \overline{h}^\star}\big(\boldsymbol{y}_k + D(\tilde{\sigma})Kx_k\big), \tag{28}$$

$$\overline{x}_{k+1} = \operatorname{prox}_{\tau, g}\Big(x_k - D(\tau)\big(\nabla f(x_k) + K^\star(2\overline{\boldsymbol{y}}_{k+1} - \boldsymbol{y}_k)\big)\Big). \tag{29}$$

For $i = i_{k+1}$ and for each $(l, j) \in \mathcal{J}(i_{k+1})$, update as follows:

$$x_{k+1}^{(i)} = \overline{x}_{k+1}^{(i)}, \tag{30}$$

$$\boldsymbol{y}_{k+1}^{(j)}(l) = \boldsymbol{y}_k^{(j)}(l) + \boldsymbol{\pi}_j(l)(\overline{\boldsymbol{y}}_{k+1}^{(j)}(l) - \boldsymbol{y}_{k+1}^{(j)}(l)). \tag{31}$$

Otherwise, set $x_{k+1}^{(i)} = x_k^{(i)}$, $\boldsymbol{y}_{k+1}^{(j)}(l) = \boldsymbol{y}_k^{(j)}(l)$.

Using the result of section 3.2 and the properties of K, the sequence $(x_k, \boldsymbol{y}_k)$ converges almost surely to a primal-dual point of problem (27), provided that such a point exists and that the following condition holds:

$$\begin{aligned} \tau_i &< \frac{1}{\beta_i + \rho\Big(\sum_{(l,j) \in \{i\} \times J(i)} (2 - \boldsymbol{\pi}_j(l)) \tilde{\sigma}_j K^\star_{(l,j),i} K_{(l,j),i}\Big)} \\ &= \frac{1}{\beta_i + \rho\Big(\sum_{j \in J(i)} (2 - \boldsymbol{\pi}_j(i)) \tilde{\sigma}_j M^\star_{j,i} M_{j,i}\Big)}, \end{aligned}$$

which is equivalent to (6). It remains to prove that the algorithm given by the iterations (28)–(31) coincides with Algorithm 2. To that end, we need the following lemma.

Lemma 8. *For any* $\boldsymbol{y} \in \boldsymbol{\mathcal{Y}}$,

$$\operatorname{prox}_{\tilde{\sigma}, \overline{h}^\star}(\boldsymbol{y}) = (\mathbf{1}_{m_1} \otimes \operatorname{prox}^{(1)}_{\sigma, h^\star}(S(\boldsymbol{y})), \dots, \mathbf{1}_{m_p} \otimes \operatorname{prox}^{(p)}_{\sigma, h^\star}(S(\boldsymbol{y}))).$$

Proof. We have $\overline{h}(\boldsymbol{y}) = h(m_1 S_1(\boldsymbol{y}^{(1)}), \dots, m_p S_p(\boldsymbol{y}^{(p)}))$. Thus,

$$\overline{h}^\star(\boldsymbol{\varphi}) = \sup_{\boldsymbol{y} \in \boldsymbol{\mathcal{Y}}} \langle \boldsymbol{\varphi}, \boldsymbol{y} \rangle - h(m_1 S_1(\boldsymbol{y}^{(1)}), \dots, m_p S_p(\boldsymbol{y}^{(p)})).$$

For all $j \in \{1, \dots, p\}$, denote by $\boldsymbol{\mathcal{C}}_j$ the subset of $\boldsymbol{\mathcal{Y}}_j$ formed by the vectors of the form $(u, \dots, u)$ for some $u \in \mathcal{Y}_j$, and define $\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{C}}_1 \times \cdots \times \boldsymbol{\mathcal{C}}_p$. Clearly, $\overline{h}^\star(\boldsymbol{\varphi}) = +\infty$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

whenever $\boldsymbol{\varphi} \notin \boldsymbol{\mathcal{C}}$ and $\partial \overline{h}^\star(\boldsymbol{\varphi}) = \emptyset$ in that case. If on the other hand $\boldsymbol{\varphi} \in \boldsymbol{\mathcal{C}}$, one can write $\boldsymbol{\varphi}$ in the form $\boldsymbol{\varphi} = (\mathbf{1}_{m_1} \otimes \varphi^{(1)}, \dots, \mathbf{1}_{m_p} \otimes \varphi^{(p)})$ for some $\varphi \in \mathcal{Y}$. In that case,

$$\begin{aligned}\overline{h}^\star(\boldsymbol{\varphi}) &= \sup_{y\in\mathcal{Y}} \sum_{j=1}^p \langle \mathbf{1}_{m_j} \otimes \varphi^{(j)}, \mathbf{1}_{m_j} \otimes y^{(j)}\rangle - h(m_1 y^{(1)}, \dots, m_p y^{(p)}) \\ &= \sup_{y\in\mathcal{Y}} \sum_{j=1}^p \langle \varphi^{(j)}, m_j y^{(j)}\rangle - h(m_1 y^{(1)}, \dots, m_p y^{(p)}) \ = h^\star(\varphi).\end{aligned}$$

Then, $\boldsymbol{u} \in \partial \overline{h}^\star(\boldsymbol{\varphi})$ if and only if, for every $\psi \in \mathcal{Y}$,

$$h^\star(\psi) \geq h^\star(\varphi) + \sum_{j=1}^p \langle \boldsymbol{u}^{(j)}, \mathbf{1}_{m_j} \otimes (\psi^{(j)} - \varphi^{(j)})\rangle$$

or, equivalently,

$$h^\star(\psi) \geq h^\star(\varphi) + \sum_{j=1}^p \langle m_j S_j(\boldsymbol{u}^{(j)}), \psi^{(j)} - \varphi^{(j)}\rangle.$$

Therefore, $\boldsymbol{u} \in \partial \overline{h}^\star(\boldsymbol{\varphi})$ if and only if $D(m)S(\boldsymbol{u}) \in \partial h^\star(\varphi)$.

Now consider an arbitrary $\boldsymbol{y} \in \boldsymbol{\mathcal{Y}}$ and set $\boldsymbol{q} = \operatorname{prox}_{\tilde{\sigma}, \overline{h}^\star}(\boldsymbol{y})$. This is equivalent to

$$D(\tilde{\sigma}^{-1})(\boldsymbol{y} - \boldsymbol{q}) \in \partial \overline{h}^\star(\boldsymbol{q}). \tag{32}$$

In particular, $\boldsymbol{q} \in \operatorname{dom}(\partial \overline{h}^\star)$ and thus $\boldsymbol{q}$ has the form $\boldsymbol{q} = (\mathbf{1}_{m_1} \otimes q^{(1)}, \dots, \mathbf{1}_{m_p} \otimes q^{(p)})$ for some $q \in \mathcal{Y}$. The inclusion (32) reads $D(m)SD(\tilde{\sigma}^{-1})(\boldsymbol{y} - \boldsymbol{q}) \in \partial h^\star(q)$. Since $D(m)SD(\tilde{\sigma}^{-1}) = D(\sigma^{-1})S$, we obtain $D(\sigma^{-1})(S(\boldsymbol{y}) - q) \in \partial h^\star(q)$ which is equivalent to $q = \operatorname{prox}_{\sigma, h^\star}(S(\boldsymbol{y}))$. This completes the proof. ∎

The proof of the following lemma is immediate.

LEMMA 9. *For any $\boldsymbol{y} \in \boldsymbol{\mathcal{Y}}$,*

$$K^\star(\boldsymbol{y}) = \Big(\sum_{j \in J(1)} M_{j1}^\star(\boldsymbol{y}^{(j)}(1)), \dots, \sum_{j\in J(n)} M_{jn}^\star(\boldsymbol{y}^{(j)}(n)) \Big).$$

In particular, for any $y \in \mathcal{Y}$,

$$K^\star(\mathbf{1}_{m_1} \otimes y^{(1)}, \dots, \mathbf{1}_{m_p} \otimes y^{(p)}) = M^\star y.$$

The following example shows how we are going to use the concept of duplication.

Example 4 (total variation). Let us consider $\mathcal{X} = \mathbb{R}^{n_1\times n_2\times n_3}$, $\mathcal{Y} = \mathbb{R}^{3\times n_1\times n_2\times n_3}$, and the total variation regularizer defined as $h \circ M$, where

$$h(y) = \sum_{i_1=1}^{n_1}\sum_{i_2=1}^{n_2}\sum_{i_3=1}^{n_3} \sqrt{\sum_{j=1}^3 y_{j,i_1,i_2,i_3}^2} = \sum_{i_1=1}^{n_1}\sum_{i_2=1}^{n_2}\sum_{i_3=1}^{n_3} h_{i_1,i_2,i_3}(y_{:,i_1,i_2,i_3})$$

and M is defined by blocks of the type

$$M_{(i_1,i_2,i_3)} = \left(\begin{array}{cccc} (x_{i_1,i_2,i_3}) & (x_{i_1+1,i_2,i_3}) & (x_{i_1,i_2+1,i_3}) & (x_{i_1,i_2,i_3+1}) \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{array}\right) \begin{array}{c} \\ (y_{1,i_1,i_2,i_3}) \\ (y_{2,i_1,i_2,i_3}) \\ (y_{3,i_1,i_2,i_3}) \end{array}.$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Each line has two nonzero elements so we duplicate dual variables as

$$K_{(i_1,i_2,i_3)} = \begin{pmatrix} (x_{i_1,i_2,i_3}) & (x_{i_1+1,i_2,i_3}) & (x_{i_1,i_2+1,i_3}) & (x_{i_1,i_2,i_3+1}) \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{array}{l} \\ (\boldsymbol{y}_{1,i_1,i_2,i_3}(1)) \\ (\boldsymbol{y}_{1,i_1,i_2,i_3}(2)) \\ (\boldsymbol{y}_{2,i_1,i_2,i_3}(1)). \\ (\boldsymbol{y}_{2,i_1,i_2,i_3}(2)) \\ (\boldsymbol{y}_{3,i_1,i_2,i_3}(1)) \\ (\boldsymbol{y}_{3,i_1,i_2,i_3}(2)) \end{array}$$

Hence, we can write

$$\bar{h}_{i_1,i_2,i_3}(\boldsymbol{y}_{i_1,i_2,i_3,:}) = \sqrt{\sum_{j=1}^{3}(\boldsymbol{y}_{i_1,i_2,i_3,j}(1) + \boldsymbol{y}_{i_1,i_2,i_3,j}(2))^2}.$$

Denoting by e_l the lth coordinate vector,

$$\mathrm{prox}_{m\sigma,\overline{h}^*}(\boldsymbol{y}) = (\mathbf{1}_{m_1} \otimes \mathrm{prox}^{(1)}_{\sigma,h^*}(S(\boldsymbol{y})), \dots, \mathbf{1}_{m_p} \otimes \mathrm{prox}^{(p)}_{\sigma,h^*}(S(\boldsymbol{y})))$$

becomes

$$\mathrm{prox}_{2\sigma,\overline{h}^*_{i_1,i_2,i_3}}(\boldsymbol{y}_{i_1,i_2,i_3,:}) = \begin{pmatrix} e_1^\top \mathrm{prox}_{\sigma,h^*_{i_1,i_2,i_3}}(\boldsymbol{y}_{i_1,i_2,i_3,:}(1) + \boldsymbol{y}_{i_1,i_2,i_3,:}(2)) \\ e_1^\top \mathrm{prox}_{\sigma,h^*_{i_1,i_2,i_3}}(\boldsymbol{y}_{i_1,i_2,i_3,:}(1) + \boldsymbol{y}_{i_1,i_2,i_3,:}(2)) \\ e_2^\top \mathrm{prox}_{\sigma,h^*_{i_1,i_2,i_3}}(\boldsymbol{y}_{i_1,i_2,i_3,:}(1) + \boldsymbol{y}_{i_1,i_2,i_3,:}(2)) \\ e_2^\top \mathrm{prox}_{\sigma,h^*_{i_1,i_2,i_3}}(\boldsymbol{y}_{i_1,i_2,i_3,:}(1) + \boldsymbol{y}_{i_1,i_2,i_3,:}(2)) \\ e_3^\top \mathrm{prox}_{\sigma,h^*_{i_1,i_2,i_3}}(\boldsymbol{y}_{i_1,i_2,i_3,:}(1) + \boldsymbol{y}_{i_1,i_2,i_3,:}(2)) \\ e_3^\top \mathrm{prox}_{\sigma,h^*_{i_1,i_2,i_3}}(\boldsymbol{y}_{i_1,i_2,i_3,:}(1) + \boldsymbol{y}_{i_1,i_2,i_3,:}(2)) \end{pmatrix}.$$

Suppose we would like to update $x_{3,4,5}$:

- the dual variables corresponding to $x_{3,4,5}$ are $\boldsymbol{y}_{3,4,5,1}(1)$, $\boldsymbol{y}_{3,4,5,2}(1)$, $\boldsymbol{y}_{3,4,5,3}(1)$, $\boldsymbol{y}_{2,4,5,1}(2)$, $\boldsymbol{y}_{3,3,5,2}(2)$, and $\boldsymbol{y}_{3,4,4,3}(2)$;
- next compute $\mathrm{prox}_{2\sigma,\overline{h}^*_{3,4,5}}(\boldsymbol{y}_{3,4,5,:})$, $\mathrm{prox}_{2\sigma,\overline{h}^*_{2,4,5}}(\boldsymbol{y}_{2,4,5,:})$, $\mathrm{prox}_{2\sigma,\overline{h}^*_{3,3,5}}(\boldsymbol{y}_{3,3,5,:})$, and $\mathrm{prox}_{2\sigma,\overline{h}^*_{3,4,4}}(\boldsymbol{y}_{3,4,4,:})$, which amount to 12 real numbers;
- update only the six useful dual values.

We are now in a position to simplify the iterations (28)–(31). For every k, we define the vectors

$$\overline{y}_{k+1} = \mathrm{prox}_{\sigma,h^\star}(S(\boldsymbol{y}_k + D(\tilde{\sigma})Kx_k)) \quad \text{and} \quad \overline{\boldsymbol{y}}_{k+1} = (\mathbf{1}_{m_1} \otimes \overline{y}^{(1)}_{k+1}, \dots, \mathbf{1}_{m_p} \otimes \overline{y}^{(p)}_{k+1}).$$

Upon noting that $SD(\tilde{\sigma})K = D(\sigma)D(m)SK = D(\sigma)M$ we obtain

$$\overline{y}_{k+1} = \mathrm{prox}_{\sigma,h^\star}(z_k + D(\sigma)Mx_k), \tag{33}$$

where we define $z_k = S(\boldsymbol{y}_k)$. In other words, for each $j \in \{1, \dots, p\}$,

$$z_k^{(j)} = \frac{1}{m_j} \sum_{i \in I(j)} \boldsymbol{y}_k^{(j)}(i).$$

Note that z_{k+1} differs from z_k only along the components j for which $\boldsymbol{y}^{(j)}_{k+1}(i)$ differs from $\boldsymbol{y}^{(j)}_k(i)$ for some i. That is, $z^{(j)}_{k+1} = z^{(j)}_k$ for each j such that $(i,j) \notin \mathcal{J}(i_{k+1})$ $\forall i$, while, for any j such that there exists i such that $(i,j) \in \mathcal{J}(i_{k+1})$,

$$z^{(j)}_{k+1} = z^{(j)}_k + \frac{1}{m_j} \sum_{i:(i,j) \in \mathcal{J}(i_{k+1})} (\boldsymbol{y}^{(j)}_{k+1}(i) - \boldsymbol{y}^{(j)}_k(i)). \tag{34}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Now consider (29). By Lemma 9, $K^\star \overline{\boldsymbol{y}}_{k+1} = M^\star \overline{y}_{k+1}$. Thus, setting $w_k = K^\star \boldsymbol{y}_k$, (29) simplifies to

$$\overline{x}_{k+1} = \operatorname{prox}_{\tau,g}\Big(x_k - D(\tau)\big(\nabla f(x_k) + (2M^\star \overline{y}_{k+1} - w_k)\big)\Big). \tag{35}$$

By Lemma 9 again, $w_k = (\sum_{j\in J(1)} M^\star_{j1}\boldsymbol{y}_k^{(j)}(1), \dots, \sum_{j \in J(n)} M^\star_{jn}\boldsymbol{y}_k^{(j)}(n))$. Therefore, w_{k+1} only differs from w_k along the coordinates i such that there exists $(i,j) \in \mathcal{J}(i_{k+1})$ and the update reads

$$w_{k+1}^{(i)} = w_k^{(i)} + \sum_{(i,j)\in\mathcal{J}(i_{k+1})} M^\star_{j,i}(\boldsymbol{y}_{k+1}^{(j)}(i) - \boldsymbol{y}_k^{(j)}(i)). \tag{36}$$

Putting all pieces together, the updated equations (33)–(36) coincide with Algorithm 2. We have thus proved that Algorithm 2 is such that $(x_k, \boldsymbol{y}_k)$ converges to a primal-dual point of problem (27) provided that such a point exists. To complete the proof, the final step is to relate the primal-dual solutions of problem (27) to the primal-dual solutions of the initial problem (1).

Consider the mapping $G : \mathcal{X} \times \mathcal{Y} \to \mathcal{X} \times \boldsymbol{\mathcal{Y}}$ defined by

$$G(x,y) := (x, (\mathbf{1}_{m_1} \otimes y^{(1)}, \dots, \mathbf{1}_{m_p} \otimes y^{(p)})).$$

Lemma 10. $\boldsymbol{\mathcal{S}} = G(\mathcal{S})$.

Proof. Let $(x,y) \in \mathcal{X} \times \mathcal{Y}$ and set $\boldsymbol{y} = (\mathbf{1}_{m_1} \otimes y^{(1)}, \dots, \mathbf{1}_{m_p} \otimes y^{(p)})$. Then $M^\star y = K^\star \boldsymbol{y}$; therefore,

$$0 \in \nabla f(x) + \partial g(x) + K^\star \boldsymbol{y} \ \Leftrightarrow\ 0 \in \nabla f(x) + \partial g(x) + M^\star y.$$

Moreover,

$$\begin{aligned} 0 \in -Kx + \partial \overline{h}^\star(\boldsymbol{y}) &\Leftrightarrow Kx \in \partial \overline{h}^\star(\boldsymbol{y}) \\ &\Leftrightarrow D(m)S(Kx) \in \partial h^\star(y) \\ &\Leftrightarrow Mx \in \partial h^\star(y), \end{aligned}$$

where we used Lemma 8 along with the identities $D(m)SK = M$ and $S(\boldsymbol{y}) = y$. The proof is completed upon noting that if $(x, \boldsymbol{y}) \in \boldsymbol{\mathcal{S}}$, then there exists $y \in \mathcal{Y}$ such that $\boldsymbol{y}$ has the form $\boldsymbol{y} = (\mathbf{1}_{m_1} \otimes y^{(1)}, \dots, \mathbf{1}_{m_p} \otimes y^{(p)})$. ▯

We have shown that, almost surely, $(x_k, \boldsymbol{y}_k)$ converges to some point in $G(\mathcal{S})$. This completes the proof of Theorem 2.

4. Convergence rate.

In this section, we are interested in the rate of convergence of the method. We consider three cases.

- h is Lipschitz continuous: We prove an $O(1/\sqrt{k})$ decrease for the function value (Theorem 11).
- $h = I_{\{b\}}$, i.e., $h(y) = 0$ if $y = b$ and $h(y) = +\infty$ otherwise: This corresponds to an optimization problem under the affine constraints $Mx = b$. We prove an $O(1/\sqrt{k})$ decrease for the function value and the feasibility (Theorem 11).
- $f + g$ is strongly convex and ∇h is Lipschitz continuous: We prove an $O(e^{-\mu k})$ rate for the distance to the optimum (Theorem 12).

These convergence guarantees are of the same order as those that can be obtained by other primal-dual methods such as the ADMM [19], i.e., $O(1/\sqrt{k})$ in general and linear rate of convergence under strong convexity assumptions. Note that the $O(1/\sqrt{k})$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

worst-case rate is not optimal and that some algorithms trade linear convergence in favorable cases for an $O(1/k)$ worst-case rate. We may cite, for instance, averaged ADMM [19, 26] or the coordinate-descent primal-dual method in [25].

THEOREM 11. *Define, for $\alpha \geq 1$,*

$$C_{1,\alpha} = \max_{1\leq i\leq n} \frac{\tau_i^{-1} + \tau_i^{-1/2}\rho(\sum_{j\in J(i)} m_j\sigma_j M_{j,i}^\star M_{j,i})^{1/2}}{\tau_i^{-1} - \rho(\sum_{j\in J(i)} m_j\sigma_j M_{j,i}^\star M_{j,i})}\Big(1+\frac{n}{\alpha}\Big)$$
$$C_{2,\alpha} = \bigg(1 + \max_{1\leq i\leq n} \frac{\alpha^{-1}(n(n-1)+1)+1}{\tau_i^{-1} - \beta_i - \rho(\sum_{j\in J(i)}(2-\boldsymbol{\pi}_j(i))m_j\sigma_j M_{j,i}^\star M_{j,i})}\beta_i\bigg).$$

We have that $C_{1,\alpha}$ and $C_{2,\alpha}$ are nonincreasing with respect to α, and are thus bounded.

Define the number of iterations $\hat K \in \{1,\dots,k\}$ as a random variable, independent of $\{i_1,\dots,i_k\}$ and such that $\Pr(\hat K = l) = \frac{1}{k}$ $\forall l \in \{1,\dots,k\}$.

If h is $L(h)$-Lipschitz in the norm $\|\cdot\|_{D(m)\sigma}$, then, for all $k \geq 0$,

$$\mathbb{E}(f(\bar x_{\hat K}) + g(\bar x_{\hat K}) + h(M\bar x_{\hat K}) - f(x_*) - g(x_*) - h(Mx_*)) \\ \leq \frac{C_{2,\sqrt{k}} + 2C_{1,k}}{\sqrt{k}} n(S_{0,*} + V(z_0 - z_*)) + \frac{4}{\sqrt{k}}L(h)^2,$$

where V is defined in (8) *and $S_{0,*}$ is defined in* (18).

If $h = I_{\{b\}}$, then for all $k \geq 0$,

$$\mathbb{E}(f(\bar x_{\hat K}) + g(\bar x_{\hat K}) - f(x_*) - g(x_*)) \\ \leq \frac{C_{2,\sqrt{k}} + 2C_{1,k}}{\sqrt{k}} n\big(S_{0,*} + V(z_0 - z_*)\big) + \|y_*\|\,\mathbb{E}(\big\|M\bar x_{\hat K} - b\big\|),$$
$$\mathbb{E}(\big\|M\bar x_{\hat K} - b\big\|_{D(m)\sigma}) \leq \frac{2}{\sqrt{k}}\Big(\sqrt{C_{2,\sqrt{k}} + 2C_{1,k}} + \sqrt{2C_{1,k}}\Big)\big(n(S_{0,*} + V(z_0 - z_*))\big)^{1/2}.$$

Proof. We begin with the proof for Algorithm 3, that is, the case in which $m_1 = \cdots = m_p = 1$.

We combine the following inequalities proved in the previous sections and that are valid for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$:

$$g(\bar x_{k+1}) + \langle \bar x_{k+1}, \nabla f(x_k) + M^\star(2\bar y_{k+1} - y_k)\rangle + \frac{1}{2}\|\bar x_{k+1} - x_k\|^2_{\tau^{-1}} \\ \overset{(16)}{\leq} g(x) + \langle x, \nabla f(x_k) + M^\star(2\bar y_{k+1} - y_k)\rangle + \frac{1}{2}\|x - x_k\|^2_{\tau^{-1}} - \frac{1}{2}\|\bar x_{k+1} - x\|^2_{\tau^{-1}},$$
$$h^\star(\bar y_{k+1}) - \langle \bar y_{k+1}, Mx_k\rangle + \frac{1}{2}\|\bar y_{k+1} - y_k\|^2_{\sigma^{-1}} \\ \overset{(13)}{\leq} h^\star(y) - \langle y, Mx_k\rangle + \frac{1}{2}\|y - y_k\|^2_{\sigma^{-1}} - \frac{1}{2}\|\bar y_{k+1} - y\|^2_{\sigma^{-1}},$$
$$\mathbb{E}_k(f(x_{k+1})) \overset{(23)+\text{Lemma 5}}{\leq} f(x_k) + \frac{1}{n}\langle \nabla f(x_k), \bar x_{k+1} - x_k\rangle + \frac{1}{2n}\|\bar x_{k+1} - x_k\|^2_\beta,$$
$$f(x) \geq f(x_k) + \langle \nabla f(x_k), x - x_k\rangle.$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

We obtain that, for all $z \in \mathcal{X} \times \mathcal{Y}$ such that z is measurable with respect to $\mathcal{F}_k$,

$$\begin{aligned} g(\bar{x}_{k+1}) &+ n\mathbb{E}_k(f(x_{k+1})) - (n-1)f(x_k) + \langle M\bar{x}_{k+1}, y\rangle \\ &\quad - h^\star(y) + h^\star(\bar{y}_{k+1}) - \langle M^\top \bar{y}_{k+1}, x\rangle - g(x) - f(x) \\ &\qquad \le V(z_k - z) - V(\bar{z}_{k+1} - z) - V(\bar{z}_{k+1} - z_k) + \frac{1}{2}\|\bar{x}_{k+1} - x_k\|_\beta^2. \end{aligned}$$

As ∇f is n-Lipschitz in the norm $\|\cdot\|_\beta$ [42] and $n\mathbb{E}(x_{k+1}) - (n-1)x_k - \bar{x}_{k+1} = 0$,

$$\begin{aligned} n\mathbb{E}_k(f(x_{k+1})) &- (n-1)f(x_k) \\ &\ge n\mathbb{E}_k\big(f(\bar{x}_{k+1}) + \langle \nabla f(\bar{x}_{k+1}), x_{k+1} - \bar{x}_{k+1}\rangle\big) \\ &\quad - (n-1)\bigg(f(\bar{x}_{k+1}) + \langle \nabla f(\bar{x}_{k+1}), x_k - \bar{x}_{k+1}\rangle + \frac{n}{2}\left\|x_k - \bar{x}_{k+1}\right\|_\beta^2\bigg) \\ &\ge f(\bar{x}_{k+1}) - \frac{n(n-1)}{2}\left\|x_k - \bar{x}_{k+1}\right\|_\beta^2. \end{aligned}$$

We also have, for all $\alpha > 0$,

$$\begin{aligned} V(z_k - z) - V(\bar{z}_{k+1} - z) - V(\bar{z}_{k+1} - z_k) &= \langle z_k - \bar{z}_{k+1}, \bar{z}_{k+1} - z\rangle_V \\ &\le 2V(z_k - \bar{z}_{k+1})^{1/2} V(\bar{z}_{k+1} - z)^{1/2} \\ &\le \alpha V(z_k - \bar{z}_{k+1}) + \frac{1}{\alpha} V(\bar{z}_{k+1} - z). \end{aligned}$$

Gathering everything, we get

$$\begin{aligned} g(\bar{x}_{k+1}) + f(\bar{x}_{k+1}) &+ \langle M\bar{x}_{k+1}, y\rangle - h^\star(y) + h^\star(\bar{y}_{k+1}) - \langle M^\top \bar{y}_{k+1}, x\rangle \\ &\quad - g(x) - f(x) - \frac{1}{\alpha} V(\bar{z}_{k+1} - z) \\ &\qquad \le \alpha V(z_k - \bar{z}_{k+1}) + \frac{n(n-1)+1}{2}\|\bar{x}_{k+1} - x_k\|_\beta^2. \end{aligned}$$

We can show by tedious but straightforward algebra that the norms $V^{1/2}$, $\tilde{V}^{1/2}$, and $(1/2(\|x\|_{\tau^{-1}}^2 + \|y\|_{\sigma^{-1}}^2))^{1/2}$ are equivalent with constants given by

$$\begin{aligned} V(z) &\le \Bigg(\max_{1\le i\le n} 1 + \sqrt{\tau_i \rho\bigg(\sum_{j\in J(i)} \sigma_j M_{j,i}^\star M_{j,i}\bigg)}\Bigg)\frac{1}{2}(\|x\|_{\tau^{-1}}^2 + \|y\|_{\sigma^{-1}}^2) \\ &\le 2 \times \frac{1}{2}(\|x\|_{\tau^{-1}}^2 + \|y\|_{\sigma^{-1}}^2), \\ \frac{1}{2}(\|x\|_{\tau^{-1}}^2 + \|y\|_{\sigma^{-1}}^2) &\le \max_{1\le i\le n} \frac{\tau_i^{-1} + \tau_i^{-1/2}\rho(\sum_{j\in J(i)} \sigma_j^{1/2} M_{j,i}^\star M_{j,i})^{1/2}}{\tau_i^{-1} - \rho(\sum_{j\in J(i)} \sigma_j M_{j,i}^\star M_{j,i})} V(z) \\ &= C_{1,\infty} V(z), \\ \alpha V(z) &+ \frac{n(n-1)+1}{2}\|\bar{x}_{k+1} - x_k\|_\beta^2 \\ &\le \Big(\alpha + \max_{1\le i\le n} \frac{n(n-1)+1+\alpha}{\tau_i^{-1} - \beta_i - \rho(\sum_{j\in J(i)}(2-\pi_j)\sigma_j M_{j,i}^\star M_{j,i})}\beta_i\Big)\tilde{V}(z) \\ &= \alpha C_{2,\alpha} \tilde{V}(z), \end{aligned}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

where $C_{2,\alpha} \in O(1)$ for $\alpha \to \infty$. Denoting the smoothed gap [48] as

$$\mathcal{G}_{\frac{2}{\alpha}}(\bar z_k, \bar z_k) = \sup_z g(\bar x_k) + f(\bar x_k) + \langle M\bar x_k, y\rangle - h^\star(y) + h^\star(\bar y_k)$$
$$- \langle M^\top \bar y_k, x\rangle - g(x) - f(x) - \frac{2}{2\alpha}\left\|\bar x_k - x\right\|^2_{\tau^{-1}} - \frac{2}{2\alpha}\left\|\bar y_k - y\right\|^2_{\sigma^{-1}},$$

we have

$$\mathcal{G}_{\frac{2}{\alpha}}(\bar z_k, \bar z_k) \leq \alpha C_{2,\alpha}\tilde V(\bar z_k - z_{k-1}).$$

Now, by (25) and the fact that $\hat K$ is independent of the coordinate selection process,

$$\mathbb{E}(\tilde V(\overline{z}_{\hat K} - z_{\hat K -1})) \leq \sum_{i=1}^k \frac{1}{k}\mathbb{E}(\tilde V(\overline{z}_i - z_{i-1})) \overset{(25)}{\leq} \frac{n}{k}(S_{0,*} + V(z_0 - z_*))$$

so

$$\mathbb{E}\big(\mathcal{G}_{\frac{2}{\alpha}}(\bar z_{\hat K}, \bar z_{\hat K})\big) \leq \frac{\alpha C_{2,\alpha}}{k} n(S_{0,*} + V(z_0 - z_*)).$$

Taking $\alpha = \sqrt{k}$ as in [19], we get

$$\mathbb{E}\big(\mathcal{G}_{\frac{2}{\sqrt{k}}}(\bar z_{\hat K}, \bar z_{\hat K})\big) \leq \frac{C_{2,\sqrt{k}}}{\sqrt{k}} n(S_{0,*} + V(z_0 - z_*)).$$

We can also bound

$$\begin{aligned}
&\frac{1}{2}\mathbb{E}(\left\|\bar x_{\hat K} - x_*\right\|^2_{\tau^{-1}}) \leq C_{1,\infty}\mathbb{E}(V(\bar z_{\hat K} - z_*)) \\
&\overset{(20)+(21)}{=} C_{1,\infty}\mathbb{E}(nV(z_{\hat K} - z_*) - nV(z_{\hat K -1} - z_*) + V(z_{\hat K -1} - z_*) + R_\pi^{(\hat K)}) \\
&\quad = \frac{C_{1,\infty}}{k}\sum_{i=1}^k \mathbb{E}(nV(z_i - z_*) - nV(z_{i-1} - z_*) + V(z_{i-1} - z_*) + R_\pi^{(i)}) \\
&\quad = \frac{C_{1,\infty}}{k}\mathbb{E}\left(nV(z_k - z_*) - nV(z_0 - z_*) + \sum_{i=1}^k V(z_{i-1} - z_*) + R_\pi^{(i)}\right) \\
&\quad \overset{(19)}{\leq} C_{1,\infty}\frac{n+k}{k}(S_{0,*} + V(z_0 - z_*)) + \frac{C_{1,\infty}}{k}\sum_{i=1}^k R_\pi^{(i)} - \tilde V(\bar z_i - z_{i-1}) \\
&\quad \leq C_{1,k}(S_{0,*} + V(z_0 - z_*)),
\end{aligned}$$

where the last inequality follows from

$$R_\pi^{(i)} - \tilde V(\bar z_i - z_{i-1}) = \frac{1}{2}\left\|\bar x_i - x_{i-1}\right\|^2_\beta - V(\bar z_i - z_{i-1}) \leq 0.$$

If h is $L(h)$-Lipschitz in the norm $\|\cdot\|_\sigma$, we can choose $y \in \partial h(M\bar x_k) \neq \emptyset$ so that $\langle M\bar x_k, y\rangle - h^*(y) = h(M\bar x_k)$, and $x = x^\star$ so that $h^*(\bar y_k) - \langle M^\top \bar y_k, x^\star\rangle \geq -h(Mx^\star)$.

We then use the inequality

$$\mathcal{G}_{\frac{2}{\sqrt{k}}}(\bar z_{\hat K}, \bar z_{\hat K}) \geq f(\bar x_{\hat K}) + g(\bar x_{\hat K}) + h(M\bar x_{\hat K})$$
$$- \frac{4}{\sqrt{k}}L(h)^2 - f(x_*) - g(x_*) - h(Mx_*) - \frac{1}{\sqrt{k}}\left\|\bar x_{\hat K} - x_*\right\|^2_{\tau^{-1}}$$

to conclude.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

If $h = I_{\{b\}}$, then using Lemma 1 in [48] we get that

$$\begin{aligned}
&\mathbb{E}(f(\bar{x}_{\hat{K}}) + g(\bar{x}_{\hat{K}}) - f(x_*) - g(x_*)) \\
&\qquad \leq \frac{C_{2,\sqrt{k}}}{\sqrt{k}} n(S_{0,*} + V(z_0 - z_*)) \\
&\qquad\quad + \mathbb{E}\bigg(\frac{1}{\sqrt{k}} \left\|\bar{x}_{\hat{K}} - x_*\right\|^2_{\tau^{-1}} + \frac{1}{\sqrt{k}} \left\|\bar{y}_{\hat{K}} - y_*\right\|^2_{\sigma^{-1}} - \langle y_*, M\bar{x}_{\hat{K}} - b\rangle \bigg), \\
&\mathbb{E}(\left\|M\bar{x}_{\hat{K}} - b\right\|_\sigma) \\
&\qquad \leq \frac{2}{\sqrt{k}} \Big(\mathbb{E}(\left\|\bar{y}_{\hat{K}} - y_\star\right\|_{\sigma^{-1}}) + \big[\mathbb{E}(\left\|\bar{y}_{\hat{K}} - y_\star\right\|^2_{\sigma^{-1}}) \\
&\qquad\qquad\quad + \frac{2}{2/\sqrt{k}} \frac{C_{2,\sqrt{k}}}{\sqrt{k}} n(S_{0,*} + V(z_0 - z_*)) + \mathbb{E}(\left\|\bar{x}_{\hat{K}} - x_\star\right\|^2_{\tau^{-1}})\big]^{1/2} \Big).
\end{aligned}$$

To obtain the result for Algorithm 2, we only need to remark that when we need to duplicate dual variables we have $h^\star(\bar{y}_k) = \overline{h}^\star(\bar{\boldsymbol{y}}_k)$. We then just need to replace σ_j by $m_j\sigma_j$ in the conditions. ∎

Remark 5. To prove the result of Theorem 11, we use a random number of iterations. This has also been proposed, for instance, in [45] for the stochastic dual coordinate ascent algorithm. Note that the number of iterations can be sampled beforehand, which means that the procedure comes with no computational cost. When $\hat{K}$ iterations have taken place, one just needs to compute $\bar{x}_{\hat{K}+1}$ once in order to obtain the guarantee.

We also have a fast rate if the problem has particular properties. We prove that if the Lagrangian function satisfies a strong convexity and strong concavity assumption, then Algorithm 2 converges exponentially fast with a rate that depends on the step size.

Assumption 6. There exist nonnegative constants μ_g and μ_f such that $\mu_f + \mu_g > 0$, and a constant $\mu_{h^\star} > 0$ such that g is μ_g-strongly convex in the norm $\|\cdot\|_{\tau^{-1}}$, f is μ_f-strongly convex in the norm $\|\cdot\|_{\tau^{-1}}$, and $h^\star$ is $\mu_{h^\star}$-strongly convex in the norm $\|\cdot\|_{\sigma^{-1}}$.

THEOREM 12. *For $z = (x, \boldsymbol{y})$, define $V^\mu(z) = V(z) + \mu_g\|x\|^2_{\tau^{-1}} + \mu'_{h^\star}\|\boldsymbol{y}\|^2_{(D(m)\sigma)^{-1}}$, where*

$$\begin{aligned}
\mu'_{h^\star} = \min\Bigg(\mu_{h^\star}, \sup\bigg\{\mu > 0 : \forall i,& \\
\tau_i^{-1} > \beta_i + \rho\bigg(\sum_{j\in J(i)} \frac{(2-\boldsymbol{\pi}_j(i))^2\sigma_j m_j}{2 - \boldsymbol{\pi}_j(i) - \mu(1 - \boldsymbol{\pi}_j(i))} M^\star_{j,i} M_{j,i}\bigg)&\bigg\}\Bigg)
\end{aligned}$$

(note that if $\boldsymbol{\pi}_j(i) = 1$ $\forall i$ and $\forall j$, then $\mu'_{h^\star} = \mu_{h^\star}$). If Assumption 6 holds then the iterates of Algorithm 2 satisfy

$$\mathbb{E}\left[S_{k,*} + V^\mu(z_k - z_*)\right] \leq \left(1 - \frac{1}{n}\frac{(\mu_f + 2\mu_g)\mu'_{h^\star}}{\mu_f + 2\mu_g + \mu'_{h^\star}}\right)^k \left[S_{0,*} + V^\mu(z_0 - z_*)\right].$$

In order to prove this theorem, we begin with a lemma that generalizes Lemma 4.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

LEMMA 13. *If Assumption* 6 *holds, then*

$$\begin{aligned}\langle \nabla f(x_*) - \nabla f(x), x_* - \overline{x}\rangle + V(\overline{z} - z)& \\ &\le V(z - z_*) - V(\overline{z} - z_*) - \mu_g \|\overline{x} - x\|^2_{\tau^{-1}} - \mu_{h^\star}\|\overline{y} - y\|^2_{\sigma^{-1}}.\end{aligned}$$

Proof. Assumption 6 gives us the following: For $(x_*, y_*) \in \mathcal{S}$,

$$g(\overline{x}) \ge f(x_*) + g(x_*) + \langle \nabla f(x_*) + M^\star y_*, x_* - \overline{x}\rangle + \frac{\mu_g}{2}\|x - x_*\|^2_{\tau^{-1}}\,, \tag{37}$$

$$h^\star(\overline{y}) \ge h^\star(y_*) + \langle M x_*, \overline{y} - y_*\rangle + \frac{\mu_{h^\star}}{2}\|\overline{y} - y_*\|^2_{\sigma^{-1}}. \tag{38}$$

With the same argument as in (14), we have

$$h^\star(\overline{y}) \le h^\star(y_*) + \langle \overline{y} - y_*, Mx\rangle + \frac{1}{2}\|y_* - y\|^2_{\sigma^{-1}} - \frac{1 + \mu_{h^\star}}{2}\|\overline{y} - y_*\|^2_{\sigma^{-1}} - \frac{1}{2}\|\overline{y} - y\|^2_{\sigma^{-1}}$$

and so, using (38),

$$\langle M(x_* - x), \overline{y} - y_*\rangle \le \frac{1}{2}\|y - y_*\|^2_{\sigma^{-1}} - \frac{1 + 2\mu_{h^\star}}{2}\|\overline{y} - y_*\|^2_{\sigma^{-1}} - \frac{1}{2}\|\overline{y} - y\|^2_{\sigma^{-1}}. \tag{39}$$

Similarly, we have

$$\begin{aligned}\langle \nabla f(x_*) - \nabla f(x), x_* - \overline{x}\rangle - \frac{1 + 2\mu_g}{2}\|x - x_*\|^2_{\tau^{-1}} + \frac{1}{2}\|\overline{x} - x_*\|^2_{\tau^{-1}} + \frac{1}{2}\|\overline{x} - x\|^2_{\tau^{-1}}& \\ &\le \langle 2\overline{y} - y - y_*, M(x_* - \overline{x})\rangle.\end{aligned}$$

Summing the above inequality with (39), and recalling the definition

$$V(z) = V(x, y) = \frac{1}{2}\|x\|^2_{\tau^{-1}} + \langle y, Mx\rangle + \frac{1}{2}\|y\|^2_{\sigma^{-1}},$$

we get

$$\begin{aligned}\langle \nabla f(x_*) - \nabla f(x), x_* - \overline{x}\rangle + V(\overline{z} - z)& \\ &\le V(z - z_*) - V(\overline{z} - z_*) - \mu_g \|\overline{x} - x\|^2_{\tau^{-1}} - \mu_{h^\star}\|\overline{y} - y\|^2_{\sigma^{-1}}. \qquad \square\end{aligned}$$

Proof of Theorem 12. We begin with the case in which $m_1 = \cdots = m_p$.

By Assumption 1(a), if $\mu_{h^\star} > 0$, then $\mu'_{h^\star} > 0$ and if $h^\star$ is $\mu_{h^\star}$-strongly convex, it is also $\mu'_{h^\star}$-strongly convex. Then, by a straightforward adaptation of the proof of Lemma 6 to the strongly convex case, we have

$$\begin{aligned}&\mathbb{E}_k\left[S_{k+1,*} + V(z_{k+1} - z_*) + \frac{2\mu_g}{2}\|x_{k+1} - x_*\|^2_{\tau^{-1}} + \frac{2\mu'_{h^\star}}{2}\|y_{k+1} - y_*\|^2_{\sigma^{-1}}\right] \\ &\quad\le \left(1 - \frac{1}{n}\right)S_{k,*} + V(z_k - z_*) + \frac{2(n-1)\mu_g - \mu_f}{2n}\|x_k - x_*\|^2_{\tau^{-1}} \\ &\qquad + \frac{2(n-1)\mu'_{h^\star}}{2n}\|y_k - y_*\|^2_{\sigma^{-1}} - \frac{1}{n}\tilde{V}(\overline{z}_{k+1} - z_k) + \frac{\mu'_{h^\star}}{n}\|\overline{y}_{k+1} - y_k\|^2_{\sigma^{-1}(1-\pi)}.\end{aligned}$$

As soon as

$$\tau_i^{-1} > \beta_i + \rho\left(\sum_{j \in J(i)} \frac{(2 - \pi_j)^2}{2 - \pi_j - \mu'_{h^\star}(1 - \pi_j)}\sigma_j M^\star_{j,i} M_{j,i}\right),$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

we can remove the term $-\frac{1}{n}\tilde{V}(\overline{z}_{k+1} - z_k) + \frac{\mu'_{h^\star}}{n}\|\overline{y}_{k+1} - y_k\|^2_{\sigma^{-1}(1-\pi)} \leq 0$. This is indeed guaranteed by the definition of $\mu'_{h^\star}$.

In order to prove a linear convergence rate $(1-\eta)$, it suffices to prove that $(1-\frac{1}{n}) \leq (1-\eta)$ and that, with respect to the order of semi-definite matrices,

$$\begin{bmatrix} \tau^{-1}(1 + \frac{2(n-1)\mu_g - \mu_f}{n}) & M^\star \\ M & \sigma^{-1}(1 + \frac{2(n-1)\mu'_{h^\star}}{n}) \end{bmatrix} \preceq (1-\eta) \begin{bmatrix} \tau^{-1}(1+2\mu_g) & M^\star \\ M & \sigma^{-1}(1+2\mu'_{h^\star}) \end{bmatrix}.$$

Using the fact that M is block-diagonal, this gives, for all i, the conditions

$$\begin{aligned} 1 + \frac{2(n-1)\mu_g - \mu_f}{n} &\leq (1-\eta)(1+2\mu_g), \\ 1 + \frac{2(n-1)\mu'_{h^\star}}{n} &\leq (1-\eta)(1+2\mu'_{h^\star}), \\ \tau_i^{-1}\Big(-\eta(1+2\mu_g) + \frac{\mu_f + 2\mu_g}{n}\Big) &\geq \sum_{j \in J(i)} \frac{\sigma_j}{-\eta(1+2\mu'_{h^\star}) + \frac{2\mu'_{h^\star}}{n}} \eta^2 M^\star_{j,i} M_{j,i}. \end{aligned}$$

Using the second condition we can multiply the third one by $-\eta(1+2\mu'_{h^\star}) + \frac{2\mu'_{h^\star}}{n} \geq 0$ and we obtain the condition

$$\begin{aligned} &\eta^2\Big(\tau_i^{-1} - \sum_{j\in J(i)} \sigma_j M^\star_{j,i} M_{j,i}\Big) \\ &\quad + \tau_i^{-1}\Bigg(\eta^2(2\mu_g + 2\mu'_{h^\star}) \\ &\qquad - \eta\Big(\frac{\mu_f + 2\mu_g + 2\mu'_{h^\star}}{n} - \frac{4\mu_g\mu'_{h^\star}}{n} - \frac{2\mu_f\mu'_{h^\star} + 4\mu_g\mu'_{h^\star}}{n}\Big) \\ &\qquad\qquad + \frac{(\mu_f + 2\mu_g)\mu'_{h^\star}}{n^2}\Bigg) \geq 0. \end{aligned}$$

The first term is nonnegative thanks to Assumption 1(a). The second term is nonnegative as soon as

$$\eta \leq \frac{1}{n}\frac{(\mu_f + 2\mu_g)\mu'_{h^\star}}{\mu_f + 2\mu_g + \mu'_{h^\star}}.$$

To conclude, we remark that

$$\frac{1}{n}\frac{(\mu_f + 2\mu_g)\mu'_{h^\star}}{\mu_f + 2\mu_g + \mu'_{h^\star}} \leq \min\left(\frac{1}{n}\frac{\mu_f + 2\mu_g}{1+2\mu_g}, \frac{1}{n}\frac{2\mu'_{h^\star}}{1+2\mu'_{h^\star}}\right) \leq \frac{1}{n}.$$

This result also implies the same rate for the iterates of Algorithm 2 because h^* is $\mu_{h^\star}$-strongly convex in the norm $\|\cdot\|_{\sigma^{-1}}$ if and only if $\overline{h}^\star$ is $\mu_{h^\star}$-strongly convex in the norm $\|\cdot\|_{\tilde{\sigma}^{-1}}$. □

Remark 7. It is worth noting that the algorithm does not depend on the strong convexity constants, which means that it automatically adapts to local strong convex-concave parameters of the Lagrangian. Moreover, as can be seen in Figure 2, we do observe linear convergence in some cases, even when Assumption 6 is not satisfied. Thus, we think that Theorem 12 can give an indication of how the algorithm behaves in favorable cases.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Remark 8. Of particular interest is the relation between the rate proved in Theorem 12 and the size of the steps. Having longer step sizes improves the rate greatly since μ_f, μ_g, and $\mu_{h^\star}$, measured in the weighted norm, are "proportional" to the step sizes: As $\mu_g \|x\|^2_{\tau^{-1}} = (\alpha\mu_g)\|x\|^2_{(\alpha\tau)^{-1}}$ $\forall \alpha > 0$, multiplying the step sizes by $\alpha > 1$ also multiplies μ_f, μ_g, and $\mu_{h^\star}$ by α, which leads to an improved rate

$$1 - \frac{1}{n}\frac{(\alpha\mu_f + 2\alpha\mu_g)\alpha\mu_{h^\star}}{\alpha\mu_f + 2\alpha\mu_g + \alpha\mu_{h^\star}} = 1 - \alpha\frac{1}{n}\frac{(\mu_f + 2\mu_g)\mu_{h^\star}}{\mu_f + 2\mu_g + \mu_{h^\star}} < 1 - \frac{1}{n}\frac{(\mu_f + 2\mu_g)\mu_{h^\star}}{\mu_f + 2\mu_g + \mu_{h^\star}}.$$

As shown in section 5.2, in large-scale applications one can expect step sizes to be much more than twice as large and so we can expect a much faster algorithm by using large steps than by using the steps proposed in [32].

5. Numerical experiments.

For all the experiments, we used one processor of a computer with Intel Xeon CPUs at 2.80 GHz.

5.1. Total variation + ℓ_1-regularized least squares regression.

For given regularization parameters $\alpha > 0$ and $r \in [0,1]$, we would like to solve the following regression problem with regularization given by the sum of total variation (TV) and the ℓ_1 norm:

$$\min_{x \in \mathbb{R}^n} \frac{1}{2}\|Ax - b\|_2^2 + \alpha\big(r\|x\|_1 + (1-r)\|Mx\|_{2,1}\big).$$

The problem takes place on a 3D image of the brains of size 40 × 48 × 34. The optimization variable x is a real vector with one entry in each voxel, that is $n =$ 65,280. Matrix M is the discretized three-dimensional (3D) gradient. This is a sparse matrix of size $195,840 \times 65,280$ with two nonzero elements in each row. The matrix $A \in \mathbb{R}^{768\times 65,280}$ and the vector $b \in \mathbb{R}^{768}$ correspond to 768 labeled experiments where each line of A gathers brains activity for the corresponding experiment. Parameter r tunes the tradeoff between the two regularization terms. If $r = 1$, one gets a Lasso problem for which coordinate descent has been reported to be very efficient [22]. For $r < 1$, classical (primal) coordinate descent cannot be applied but primal-dual coordinate descent can.

In this scenario, we set the objective as $f(x) = \frac{1}{2}\|Ax - b\|_2^2$, $g(x) = \alpha r\|x\|_1$, and $h(y) = \alpha(1-r)\|y\|_{2,1}$. We coded Algorithm 2 in Cython and duplicated each dual variable two times.[2] Note that, as $h = \alpha(1-r)\|\cdot\|_{2,1}$ is not separable, we must compute 12 dual components of $\bar{\boldsymbol{y}}_{k+1}$ for each updated primal variable $x^{(i)}_{k+1}$ but then use only six of them to update $z^{(j)}_{k+1}$ for $j \in J(i_{k+1})$. This procedure is explained in detail in section 3.3. We chose σ_j such that $\rho(\sum_{j\in J(i)} \sigma_j M^\star_{j,i} M_{j,i})$ is of the same order of magnitude as β_i and τ_i equal to 0.95 times its upper bound in Assumption 1. We compared Algorithm 2 against

- Vũ and Condat's algorithm [53, 17],
- Chambolle and Pock's algorithm [13],
- a fast iterative shrinkage-thresholding algorithm (FISTA) [2] with an inexact resolution of the proximal operator of TV and a momentum factor ensuring convergence [11],
- the limited-memory Broyden—Fletcher—Goldfarb—Shanno (L-BFGS) algorithm [56] with a smoothing of the nonsmooth functions and continuation.

Figure 1 indicates that our primal coordinate descent is a competitive algorithm for a wide range of regularization parameters (see online version for all color figures).

[2] The Cython code is available at http://perso.telecom-paristech.fr/~ofercoq/Software.html.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

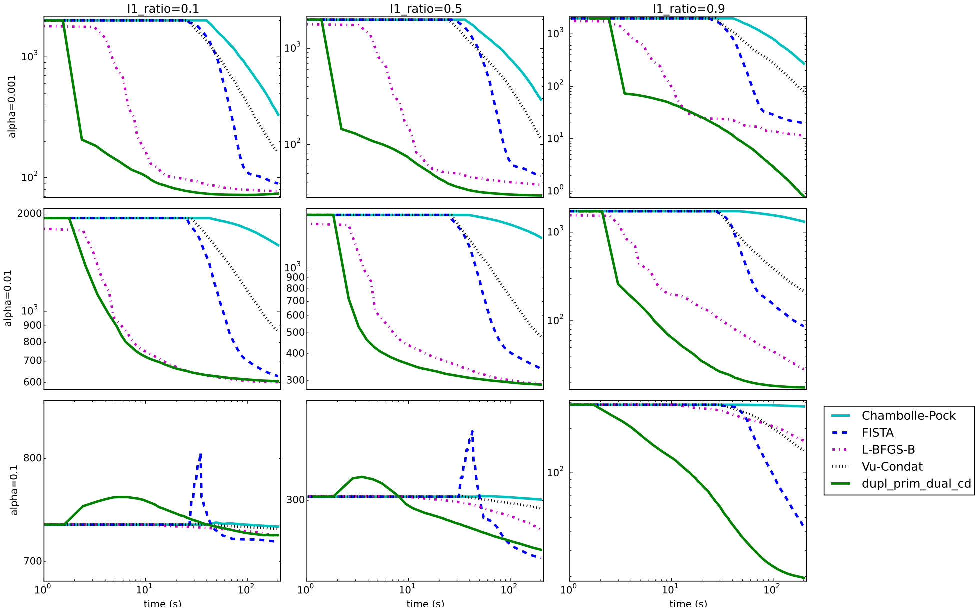

Fig. 1. *Comparison of algorithms for $TV + \ell_1$-regularized regression at various regularization parameters. For each problem, we compute the dual function at the last iterate (this amounts to solving a Lasso problem). Then we compare the primal objective curves to this reference value and we plot them on a logarithmic scale. Note that, for the choices of regularization parameters such that $\alpha(1-r)$ is larger, the problem is more difficult to solve because the total variation regularizer is dominant. This is in fact the most challenging part of the objective because it is nondifferentiable and nonseparable.*

Note that the Chambolle–Pock algorithm needs to compute the singular values decomposition of A (which explains the flat shape of the performance curve when the algorithm starts). FISTA and Vũ–Condat need to estimate its largest singular value. If only a low accuracy is required, Algorithm 2 may have reached this low accuracy even before these preprocessing steps are completed.

L-BFGS has similar behavior to Algorithm 2 except for $\alpha = 0.1$, $r = 0.9$, where it suffers from the nonsmoothness of the objective, while Algorithm 2 deals with it directly by the proximal operators. FISTA is the fastest algorithm for problems with a heavy TV regularization.

5.2. Linear support vector machines. We now present a second application for our algorithm. We consider a set of n observations gathered into a data matrix $A \in \mathbb{R}^{m\times n}$ and labels $b \in \mathbb{R}^n$ and we intend to solve the following support vector machine (SVM) problem:

$$\min_{w\in\mathbb{R}^m, w_0\in\mathbb{R}} \sum_{i=1}^n C_i \max\big(0, 1 - b_i((A^\top w)_i + w_0)\big) + \frac{\lambda}{2}\|w\|_2^2 .$$

As is common practice for this problem, we solve instead the dual support vector machine problem:

$$\max_{x\in\mathbb{R}^n} -\frac{1}{2\lambda}\|AD(b)x\|_2^2 + e^\top x - \sum_{i=1}^n I_{[0,C_i]}(x_i) - I_{\{0\}}(\langle b, x\rangle)$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

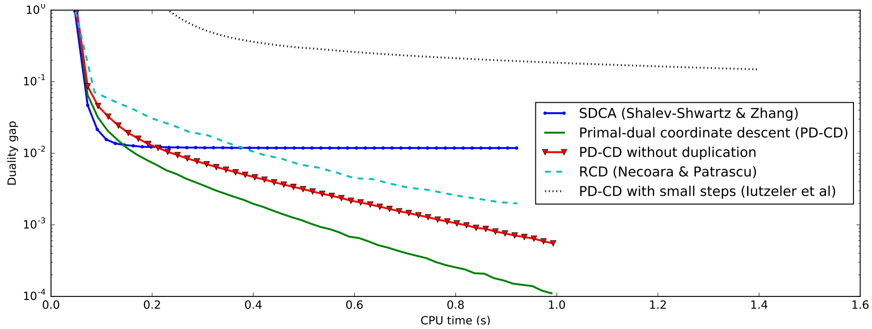

FIG. 2. *Comparison of dual algorithms for the resolution of linear SVM on the RCV*1 *dataset. We report the value of the duality gap after a postprocessing to recover feasible primal and dual variables. Primal variables are recovered as suggested in* [45] *and the intercept is recovered by exact minimization of the primal objective given the other primal variables. When dual iterates are not feasible, we project them onto the dual feasible set before computing the dual objective. We stopped each algorithm after* 100 *passes through the data: Note that the cost per iteration of the* 5 *algorithms is similar but that the algorithm of* [32] *needs first to compute the Lipschitz constant of the gradient.*

Here, we are considering a nonzero bias. Therefore, the primal SVM problem is not strongly convex and the dual SVM problem has a coupling constraint. Some authors [45] proposed to fix the bias to 0 in order to make the problem easier to solve, but we show that our method can solve the original SVM problem nearly as fast.

In the experiments,[3] we consider the following datasets.

- The RCV1 dataset [33], where A is a sparse $m \times n$ matrix with $m = 20{,}242$, $n = 47{,}236$, and 0.157% of nonzero entries and we take $C_i = \frac{1}{n}$ $\forall i$ and $\lambda = \frac{1}{4n}$. For this dataset, $\|A\|^2 \approx 450 \max_i \|Ae_i\|^2$, which means that using small step sizes leads to a roughly 450 times slower algorithm. This situation is not uncommon and is one of the reasons why coordinate-descent methods are attractive.
- The KDD Cup 2009 dataset [28]: The data is a mix of 14,740 numerical values and 260 categorical values from Orange Labs. We preprocessed the data by adding a feature for each column containing missing values and binarizing the categorical values. We obtained a sparse matrix with $m = 86{,}825$, $n = 50{,}000$, and 1.79% of nonzero entries. We divided the columns by their standard deviation and removed columns with too small a standard deviation. There are three tasks with this dataset: Estimate the appetency, churn, and up-selling probability of customers. As the classes are unbalanced, we compensate this with values of C_i proportional to the class weight and we chose $\max_i C_i = \lambda = \frac{1}{n}$. We also chose a value of σ_i depending on the class.

Here $f(x) = \frac{1}{2}\|AD(b)x\|_2^2 - e^\top x$, $g(x) = \sum_{i=1}^n I_{[0,C_i]}(x_i)$, $h(y) = I_{\{0\}}(y)$ (h is the indicator for $\{0\} \subset \mathbb{R}$, i.e., $h(y) = 0$ if $y = 0$, $h(y) = +\infty$ otherwise) and $M = b^\top$. We compare the following methods.

- Stochastic dual coordinate ascent (SDCA) [45]: Note that SDCA simply forgets $I_{\{0\}}(y)$ in order to be able to apply the classical coordinate-descent method a thus will not converge to an optimal solution.
- Randomized coordinate descent (RCD) [37]: At each iteration, the algorithm selects two coordinates randomly and performs a coordinate-descent step ac-

[3] Code available at http://perso.telecom-paristech.fr/~ofercoq/Software.html.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

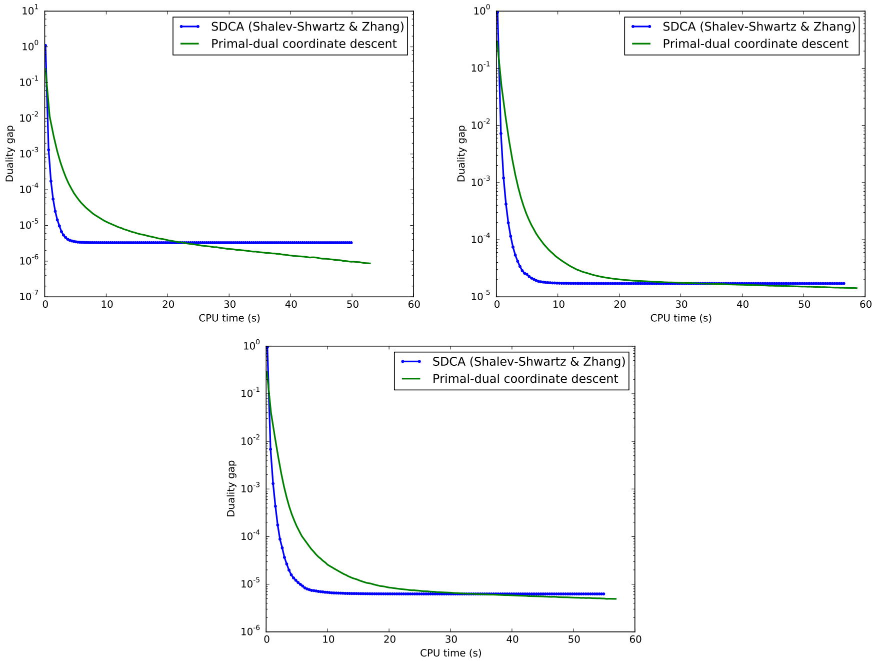

FIG. 3. *Comparison of dual algorithms for the resolution of linear SVM on the KDD Cup* 2009 *dataset for the appetency, churn, and up-selling tasks (one plot for each). We did the same postprocessing as in Figure* 2. *We stopped each algorithm after* 300 *passes through the data. We can see here also that dealing with the intercept allows us to find more accurate solutions for a similar computational cost as with SDCA.*

cording to these two variables. Updating two variables at a time allows us to satisfy the linear constraint at each iteration.

- Primal-dual coordinate descent (PD-CD) with small steps using the step size $\tau < \frac{1}{\beta(f)/2+\sigma\rho(K^\star K)}$ as in [32].
- Algorithm 2 with $\mathcal{J}(i) = \{i\} \times J(i)\ \forall i$ (PC-CD).
- Algorithm 1, i.e., Algorithm 2 with

$$\mathcal{J}(i) = \bigcup_{1 \le j \le n} \{j\} \times J(j) \quad \text{for all } i$$

in order to maintain a single Lagrange multiplier (PD-CD without duplication).

We can see in Figures 2 and 3 the decrease of the SVM duality gap for each algorithm. SDCA is very efficient in the beginning and converges quickly. However, as the method does not take into account the intercept, it does not converge to the optimal solution and stagnates after a few passes on the data. Algorithm 2 allows step sizes nearly as long as SDCA's and, taking into account, the coupling constraint represents only marginal additional work. Hence, the objective value decreases nearly as fast for SDCA in the beginning without sacrificing the intercept, leading to a smaller objective value in the end. The RCD method in [37] does work but is not competitive in terms of the rate of convergence. Also, as expected, using small steps [32] leads to a very slow algorithm in this context. Finally, for this problem, the additional memory requirement induced by duplication is negligible compared to the size of the problem

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

data, but the slightly stricter step-size condition may explain why PD-CD without duplication is slower. We also tried the C implementation of LIBSVM [15] but it needed 175 s to solve the (medium-size) RCV1 problem.

6. Conclusion.

In this work, we combined features of two seemingly incompatible versions of coordinate descent: One based on Fejér monotonicity [16], which allows nonseparable nonsmooth functions, and one based on the decrease of the function value [41], which allows a large step size. We proved the convergence of the algorithm and demonstrated its efficiency on two large-scale problems.

Our future work will focus on the limits of Theorem 2. We believe that the restriction to uniform sampling probabilities can be removed. Also, by analogy with the Vũ–Condat method, one should be able to replace β_i by $\beta_i/2$ in the step-size condition. A more prospective research, motivated by [36], consists in studying the impact of the nonsmooth functions on the range of step sizes ensuring convergence.

Acknowledgment. We are grateful to Elvis Dohmatob for letting us use his benchmarking tool [20].

REFERENCES

[1] H. H. Bauschke, J. M. Borwein, and P. L. Combettes, *Bregman monotone optimization algorithms*, SIAM J. Control Optim, 42 (2003), pp. 596–636.

[2] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2 (2009), pp. 183–202.

[3] A. Beck and L. Tetruashvili, *On the convergence of block coordinate descent type methods*, SIAM J. Optim., 23 (2013), pp. 2037–2060.

[4] D. P. Bertsekas, *Incremental proximal methods for large scale convex optimization*, Math. Program., 129 (2011), pp. 163–195.

[5] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[6] P. Bianchi and O. Fercoq, *Using big steps in coordinate descent primal-dual algorithms*, in Proceedings of 2016 IEEE 55th Conference on Decision and Control, Las Vegas, NV, IEEE Press, Piscataway, NJ, 2016, pp. 1895–1899.

[7] P. Bianchi, W. Hachem, and F. Iutzeler, *A Stochastic Coordinate Descent Primal-Dual Algorithm and Applications to Large-Scale Composite Optimization*, preprint, https://arxiv.org/abs/1407.0898v1, 2014.

[8] 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 (2011), pp. 1–122.

[9] 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 (2014), pp. 32–43.

[10] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, and T. Pock, *An introduction to total variation for image analysis*, in Theoretical Foundations and Numerical Methods for Sparse Recovery, Radon Series on Comput. Appl. Math. 9, 2010, Walter De Gruyter, Berlin, pp. 263–340.

[11] A. Chambolle and C. Dossal, *On the Convergence of the Iterates of the "Fast Iterative Shrinkage/Thresholding Algorithm,"* J. Optim. Theory Appl, 166 (2015), pp. 968–982.

[12] A. Chambolle, M. J. Ehrhardt, P. Richtárik, and C.-B. Schönlieb, *Stochastic Primal-Dual Hybrid Gradient Algorithm with Arbitrary Sampling and Imaging Application*, preprint, https://arxiv.org/abs/1706.04957, 2017.

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

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

[15] C.-C. Chang and C.-J. Lin, *LIBSVM: A library for support vector machines*, ACM Trans. Intell. Syst. Technol., 2 (2011), 27.

[16] P. L. Combettes and J.-C. Pesquet, *Stochastic quasi-Fejér block-coordinate fixed point iterations with random sweeping*, SIAM J. Optim., 25 (2015), pp. 1221–1248.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

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

[18] D. Davis and W. Yin, *A Three-Operator Splitting Scheme and Its Optimization Applications*, preprint, https://arxiv.org/abs/1504.01032, 2015.

[19] D. Davis and W. Yin, *Faster convergence rates of relaxed Peaceman–Rachford and ADMM under regularity assumptions*, Math. Oper. Res., 42 (2017), pp. 783–805.

[20] E. Dohmatob, A. Gramfort, B. Thirion, and G. Varoquaux, *Benchmarking solvers for TV-ℓ_1 least-squares and logistic regression in brain imaging*, in Proceedings of the 2014 International Workshop on Pattern Recognition in Neuroimaging, IEEE Press, Piscataway, NJ, 2014, DOI: 10.1109/PRNI.2014.6858516.

[21] O. Fercoq and P. Richtárik, *Accelerated, parallel and proximal coordinate descent*, SIAM J. Optim., 25 (2015), pp. 1997–2023.

[22] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani, *Pathwise coordinate optimization*, Ann. Appl. Stat., 1 (2007), pp. 302–332.

[23] D. Gabay, *Applications of the method of multipliers to variational inequalities*, in Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems, Stud. Math. Applic. 15, M. Fortin and R. Glowinski, eds., Elsevier, New York, 1983, pp. 299–331.

[24] D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite element approximation*, Comput. Math. Appl., 2 (1976), pp. 17–40.

[25] X. Gao, Y. Xu, and S. Zhang, *Randomized Primal-Dual Proximal Block Coordinate Updates*, preprint, https://arxiv.org/abs/1605.05969v1, 2016.

[26] X. Gao and S.-Z. Zhang, *First-order algorithms for convex optimization with nonseparable objective and coupled constraints*, J. Oper. Res. Soc. China, 5 (2017), pp. 131–159.

[27] R. Glowinski and A. Marroco, *Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires*,RAIRO Analyse Numérique, 9 (1975), pp. 41–76.

[28] I. Guyon, V. Lemaire, M. Boullé, G. Dror, and D. Vogel, *Analysis of the KDD Cup* 2009*: Fast scoring on a large Orange customer database*, Proc. Mach. Learn. Res. 7, pp. 1–22; available at http://proceedings.mlr.press/v7/.

[29] B. He and X. Yuan, *Convergence analysis of primal-dual algorithms for a saddle-point problem: From contraction perspective*, SIAM J. Imaging Sci., 5 (2012), pp. 119–149.

[30] M. Hong, X. Wang, M. Razaviyayn, and Z.-Q. Luo, *Iteration complexity analysis of block coordinate descent methods*, Math. Program., 163 (2017), pp. 85–114.

[31] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 2012.

[32] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem, *Asynchronous distributed optimization using a randomized alternating direction method of multipliers*, in Proceedings of IEEE 52nd Annual Conference on Decision and Control, IEEE Press, Piscataway, NJ, 2013, pp. 3671–3676.

[33] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, *RCV*1*: A new benchmark collection for text categorization research*, J. Mach. Learn. Res., 5 (2004), pp. 361–397.

[34] Q. Lin, Z. Lu, and L. Xiao, *An accelerated proximal coordinate gradient method*, in Adv. Neural Inf. Process. Syst. 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds., Curran Associates, Red Hook, NY, 2014, pp. 3059–3067.

[35] Z. Q. Luo and P. Tseng, *On the convergence of the coordinate descent method for convex differentiable minimization*, J. Optim. Theory Appl., 72 (1992), pp. 7–35.

[36] J. Mairal, *Incremental majorization-minimization optimization with application to large-scale machine learning*, SIAM J. Optim., 25 (2015), pp. 829–855.

[37] I. Necoara and A. Patrascu, *A Random Coordinate Descent Algorithm for Optimization Problems with Composite Objective Function and Linear Coupled Constraints*, Technical report, Politehnica University of Bucharest, 2012.

[38] Y. Nesterov, *Efficiency of coordinate descent methods on huge-scale optimization problems*, SIAM J. Optim., 22 (2012), pp. 341–362.

[39] Y. Nesterov, *Subgradient methods for huge-scale optimization problems*, Math. Program., 146 (2014), pp. 275–297.

[40] J.-C. Pesquet and A. Repetti, *A class of randomized primal-dual algorithms for distributed optimization*, J. Nonlinear Convex Anal., 16 (2015), pp. 2453–2490.

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

[42] P. Richtárik and M. Takáč, *Parallel coordinate descent methods for big data optimization*, Math. Program., 156 (2016), pp. 433–484.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/08/19 to 129.81.226.78. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

[43] P. Richtárik and M. Takáč, *Efficient Serial and Parallel Coordinate Descent Method for Huge-Scale Truss Topology Design*, Oper. Res. Proc. 2011, Springer, 2012, pp. 27–32.

[44] H. Robbins and D. Siegmund, *A convergence theorem for non negative almost supermartingales and some applications*, in Optimizing Methods in Statistics, Academic Press, New York, 1971, pp. 233–257.

[45] S. Shalev-Shwartz and T. Zhang, *Stochastic dual coordinate ascent methods for regularized loss minimization*, J. Mach. Learn. Res., 14 (2013), pp. 567–599.

[46] T. Suzuki, *Stochastic dual coordinate ascent with alternating direction method of multipliers*, Proc. Mach. Learn. Res. 32, 2014, pp. 736–744; available at http://proceedings.mlr.press/v32/.

[47] Q. Tran-Dinh and V. Cevher, *A Primal-Dual Algorithmic Framework for Constrained Convex Minimization*, preprint, https://arxiv.org/abs/1406.5403v1, 2014.

[48] Q. Tran-Dinh, O. Fercoq, and V. Cevher, *A Smooth Primal-Dual Optimization Framework for Nonsmooth Composite Convex Minimization*, preprint, https://arxiv.org/abs/1507.06243v4, 2016.

[49] P. Tseng, *On accelerated proximal gradient methods for convex-concave optimization*, SIAM J. Optim., submitted.

[50] P. Tseng and C. O. L. Mangasarian, *Convergence of a block coordinate descent method for nondifferentiable minimization*, J. Optim. Theory Appl., (2001), pp. 475–494.

[51] P. Tseng and S. Yun, *A coordinate gradient descent method for nonsmooth separable minimization*, Math. Program., 117 (2009), pp. 387–423.

[52] P. Tseng and S. Yun, *A coordinate gradient descent method for linearly constrained smooth optimization and support vector machines training*, Comput. Optim. Appl., 47 (2010), pp. 179–206.

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

[54] J. Warga, *Minimizing certain convex functions*, J. Soc. Indust. Appl. Math., 11 (1963), pp. 588–593.

[55] Y. Zhang and L. Xiao, *Stochastic Primal-Dual Coordinate Method for Regularized Empirical Risk Minimization*, preprint, https://arxiv.org/abs/1409.3257v1, 2014.

[56] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, *Algorithm* 778: *L-BFGS-B: FORTRAN subroutines for large-scale bound-constrained optimization*, ACM Trans. Math. Software, 23 (1997), pp. 550–560.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.