



HAL
open science

A Coordinate Descent Primal-Dual Algorithm with Large Step Size and Possibly Non Separable Functions

Olivier Fercoq, Pascal Bianchi

► **To cite this version:**

Olivier Fercoq, Pascal Bianchi. A Coordinate Descent Primal-Dual Algorithm with Large Step Size and Possibly Non Separable Functions. *SIAM Journal on Optimization*, 2019, 29 (1), pp.100-134. hal-01497104v2

HAL Id: hal-01497104

<https://hal.science/hal-01497104v2>

Submitted on 5 Feb 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Coordinate Descent Primal-Dual Algorithm with Large Step Size and Possibly Non-Separable Functions *

Olivier Fercoq[†]Pascal Bianchi[‡]

February 5, 2018

Abstract

This paper introduces a randomized coordinate descent version of the Vū-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, constraints as well as non-separable and non-differentiable 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 coordinate-wise 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.

1 Introduction

1.1 Motivation

We consider the optimization problem

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

where \mathcal{X} is a Euclidean space, $M : \mathcal{X} \rightarrow \mathcal{Y}$ is a linear operator onto a second Euclidean space \mathcal{Y} ; functions $f : \mathcal{X} \rightarrow \mathbb{R}$, $g : \mathcal{X} \rightarrow (-\infty, +\infty]$ and $h : \mathcal{Y} \rightarrow]-\infty, +\infty]$ are assumed proper, closed and convex; the function f is moreover assumed differentiable. We assume that \mathcal{X} and \mathcal{Y} are product spaces of the form $\mathcal{X} = \mathcal{X}_1 \times \dots \times \mathcal{X}_n$ and $\mathcal{Y} = \mathcal{Y}_1 \times \dots \times \mathcal{Y}_p$ 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 [8], image processing [9] or distributed optimization [7].

Under the standard qualification condition $0 \in \text{ri}(M\text{dom}g - \text{dom}h)$ (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

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

*A summary of the results of this paper has been published in the proceedings of the 2016 Conference on Decision and Control [5]. This work has been supported by the Orange/Telecom ParisTech think tank Phi-TAB.

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

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

where $\langle \cdot, \cdot \rangle$ is the inner product and $h^* : y \mapsto \sup_{z \in \mathcal{Y}} \langle y, z \rangle - h(z)$ is the Fenchel-Legendre transform of h . There is a rich literature on *primal-dual* algorithms searching for a saddle point of L (see [45] and references therein). In the special case where $f = 0$, the alternating direction method of multipliers (ADMM) proposed by Glowinsky and Marroco [25], Gabay and Mercier [23] and the algorithm of Chambolle and Pock [12] are amongst the most celebrated ones. Based on an elegant idea also used in [27], Vũ [51] and Condat [16] separately proposed a primal-dual algorithm allowing as well to handle ∇f 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 [13] (see also [45]). A related splitting method has been recently introduced by [17].

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 coordinate-wise minimization of a unique function in a Gauss-Seidel sense [52, 4, 48]. Tseng *et al.* [32, 49, 50] and Nesterov [35] developed CD versions of the gradient descent. In [35] as well as in this paper, the updated coordinates are randomly chosen at each iteration. The algorithm of [35] 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 *coordinate-wise* Lipschitz constant of ∇f along its i th 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 to use *longer step sizes* which potentially results in a more attractive performance. The random CD gradient descent of [35] was later generalized by Richtárik and Takáč [38] to the minimization of a sum of two convex functions $f + g$ (that is, $h = 0$ in problem (1)). The algorithm of [38] 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 $g_i : \mathcal{X}_i \rightarrow]-\infty, +\infty]$. Accelerated and parallel versions of the algorithm have been later developed by [41, 40, 20, 31], 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 [53] introduce a stochastic CD primal-dual algorithm and analyze its convergence rate (see also [44] for related works). In 2013, Iutzeler *et al.* [29] proved that random coordinate descent can be successfully applied to fixed point iterations of firmly non-expansive (FNE) operators. According to [22], the ADMM can be written as a fixed point algorithm of a FNE operator, which led the authors of [29] to propose a coordinate descent version of ADMM with application to distributed optimization. The key idea behind the convergence proof of [29] is to establish the so-called stochastic Fejér monotonicity of the sequence of iterates as noted by [15]. In a more general setting than [29], Combettes *et al.* in [15] and Bianchi *et al.* [6] extend the proof to the so-called α -averaged operators, which include FNE operators as a special case. This generalization allows to apply the coordinate descent principle 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 [15] and particular cases of the Vũ-Condat algorithm are considered in [6, 37]. 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 write as a fixed point algorithm over some product Hilbert space of the form $H = H_1 \times \dots \times H_q$ 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 workaround was proposed in [6] but for a particular example only.

Second and even more importantly, the approach of [29, 15, 6, 37] needs “small” step sizes. More precisely, the convergence conditions are identical to the ones of the brute method, the one without coordinate descent. These conditions involve the global Lipschitz constant of the gradient ∇f instead than its coordinate-wise Lipschitz constants. In practice, it means that the application of coordinate descent to primal-dual algorithm

as suggested by [15] and [6] 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 [24] and [11], a $O(1/k)$ rate is obtained for the ergodic mean of the sequences. The rates are given in terms of feasibility and optimality or Bregman distance. Those 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 well as a set of associated dual iterates. Under some assumptions involving the coordinate-wise Lipschitz constants of ∇f , the primal-dual iterates converges 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 [38].

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 [29, 15, 6], does not hold (a counter-example 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 to the case of total-variation regularization and support vector machines. Numerical results performed on real MRI and text data establish the 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 $M_{j,i} : \mathcal{X}_i \rightarrow \mathcal{Y}_j$ are the block components of M . For each $j \in \{1, \dots, p\}$, we introduce the set

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

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

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

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

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

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

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

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 $\mathcal{A} = \mathcal{A}_1 \times \dots \times \mathcal{A}_\ell$ 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 $\|\cdot\|_{\mathcal{A}_i}$ stand for the norm on \mathcal{A}_i . If $F : \mathcal{A} \rightarrow]-\infty, +\infty]$ denotes a convex proper lower-semicontinuous function, we introduce the proximity operator $\text{prox}_{\gamma, F} : \mathcal{A} \rightarrow \mathcal{A}$ defined for any $u \in \mathcal{A}$ by

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

where we use the notation $\gamma^{-1} = (\gamma_1^{-1}, \dots, \gamma_\ell^{-1})$. We denote by $\text{prox}_{\gamma, F}^{(i)} : \mathcal{A} \rightarrow \mathcal{A}_i$ the i th coordinate mapping of $\text{prox}_{\gamma, F}$ that is, $\text{prox}_{\gamma, F}(u) = (\text{prox}_{\gamma, F}^{(1)}(u), \dots, \text{prox}_{\gamma, F}^{(\ell)}(u))$ 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 $\mathcal{A} \rightarrow \mathcal{A}$ 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 B^* . 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 $\text{nnz}(A)$.

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 $\{1, \dots, n\}^1$. The proposed primal-dual CD algorithm consists in updating two sequences $x_k \in \mathcal{X}$, $y_k \in \mathcal{Y}$. It is provided in Algorithm 1 below.

Algorithm 1 Coordinate-descent primal-dual algorithm

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

Iteration k : Define:

$$\begin{aligned} \bar{y}_{k+1} &= \text{prox}_{\sigma, h^*}(y_k + D(\sigma)Mx_k) \\ \bar{x}_{k+1} &= \text{prox}_{\tau, g}\left(x_k - D(\tau)(\nabla f(x_k) + 2M^*\bar{y}_{k+1} - M^*y_k)\right). \end{aligned}$$

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

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

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

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

¹The 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 [40].

Assumption 2.1. a) The functions f, g, h are closed proper and convex.

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 $u \in \mathcal{X}_i$,

$$f(x + U_i u) \leq f(x) + \langle \nabla f(x), U_i u \rangle + \frac{\beta_i}{2} \|u\|_{\mathcal{X}_i}^2.$$

d) The 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 $i \in \{1, \dots, n\}$,

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

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

$$0 \in \nabla f(x_*) + \partial g(x_*) + M^* y_* \quad (2)$$

$$0 \in -M x_* + \partial h^*(y_*). \quad (3)$$

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

Theorem 1. *Let Assumption 2.1 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*

$$\begin{aligned} \lim_{k \rightarrow \infty} x_k &= x_* \\ \lim_{k \rightarrow \infty} y_k &= y_* . \end{aligned}$$

2.3 Efficient implementation using problem structure

In Algorithm 1, it is worth noting that quantities $(\bar{x}_{k+1}, \bar{y}_{k+1})$ do not need to be explicitly calculated. At iteration k , only the coordinates

$$\bar{x}_{k+1}^{(i_{k+1})} \text{ and } \bar{y}_{k+1}^{(j)} \text{, } \forall j \in J(i_{k+1})$$

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 $\bar{x}_{k+1}, \bar{y}_{k+1}$. Two situations have been reported in the literature:

- If g is separable, one only needs to compute the quantities $\nabla_{i_{k+1}} f(x_k)$, $(2M^* \bar{y}_{k+1} - M^* y_k)^{(i_{k+1})}$ and $\text{prox}_{\tau_{i_{k+1}}, g_{i_{k+1}}}$ to perform the k th iteration. A classical example of such smart residual update [36] can be found in the proximal coordinate descent gradient algorithm (case g separable and $h = 0$) [39]. More generally, if g (resp. h^*) is block-separable, we can use this structure in the algorithm, even if this block structure does not match $\mathcal{X}_1 \times \dots \times \mathcal{X}_n$ (resp. $\mathcal{Y}_1 \times \dots \times \mathcal{Y}_p$).

We used 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 $\{x_1 = \dots = x_n\}$, f is separable and $h = 0$, we recover MISO [33]. 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 used 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, we give in the following table a comparison of the number of operations to compute the updates of the standard Vü-Condat method against the proposed algorithm.

Problem / Dimension of data	Vũ-Condat	Our algorithm
Total Variation + ℓ_1 regularization $A \in \mathbb{R}^{m \times n}$: dense; $M \in \mathbb{R}^{3n \times n}$: $\text{nnz}(M) = 6n$	$O(mn + 6n)$	$O(m + 12)$
Support Vector Machines $A \in \mathbb{R}^{m \times n}$: sparse	$O(\text{nnz}(A) + n)$	$O(\text{nnz}(Ae_i) + 1)$

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

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

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
$\mathcal{Y}_j = \mathcal{Y}_j^{I(j)}$	$\mathbf{u} = (\mathbf{u}(i) : i \in I(j))$	$ I(j) $
$\mathcal{Y} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_p$	$\mathbf{y} = (\mathbf{y}^{(j)} : j \in \{1, \dots, p\})$ where $\mathbf{y}^{(j)} = (\mathbf{y}^{(j)}(i) : i \in I(j)) \forall j$	$\text{nnz}(M)$

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 $(\mathbf{y}_k^{(j)}(i) : i \in \{1, \dots, n\}, j \in J(i))$. In a coordinate descent spirit, we however 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 $(\mathbf{y}_k^{(j)}(i_{k+1}) : j \in J(i_{k+1}))$ 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 interesting in some situations to update a larger set of duplicated dual variables at iteration k , namely $(\mathbf{y}_k^{(j)}(l) : (l, j) \in \mathcal{J}(i_{k+1}))$ 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

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

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

$$\pi_j(i) = \frac{1}{\text{card}\{(l, j) \in \mathcal{J}(l)\}}. \quad (5)$$

Note that $0 < \pi_j(i) \leq 1$. In the special case where $\mathcal{J}(i) = \{i\} \times J(i)$, note also that $\pi_j(i) = 1$ 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 $\{1, \dots, n\}$. The algorithm consists in updating four sequences $x_k \in \mathcal{X}$, $w_k \in \mathcal{X}$, $z_k \in \mathcal{Y}$ and $\mathbf{y}_k \in \mathcal{Y}$. It is provided in Algorithm 2 below.

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

Initialization: Choose $x_0 \in \mathcal{X}$, $\mathbf{y}_0 \in \mathcal{Y}$.

For all $i \in \{1, \dots, n\}$, set $w_0^{(i)} = \sum_{j \in J(i)} M_{j,i}^* \mathbf{y}_0^{(j)}(i)$.

For all $j \in \{1, \dots, p\}$, set $z_0^{(j)} = \frac{1}{m_j} \sum_{i \in I(j)} \mathbf{y}_0^{(j)}(i)$.

Iteration k : Define:

$$\begin{aligned} \bar{\mathbf{y}}_{k+1} &= \text{prox}_{\sigma, h^*}(z_k + D(\sigma)Mx_k) \\ \bar{x}_{k+1} &= \text{prox}_{\tau, g}\left(x_k - D(\tau)(\nabla f(x_k) + 2M^*\bar{\mathbf{y}}_{k+1} - w_k)\right). \end{aligned}$$

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

$$\begin{aligned} x_{k+1}^{(i)} &= \bar{x}_{k+1}^{(i)} \\ \mathbf{y}_{k+1}^{(j)}(l) &= \mathbf{y}_k^{(j)}(l) + \pi_j(l)(\bar{\mathbf{y}}_{k+1}^{(j)} - \mathbf{y}_k^{(j)}(l)) \\ w_{k+1}^{(l)} &= w_k^{(l)} + \sum_{(l,j) \in \mathcal{J}(i)} M_{j,l}^* (\mathbf{y}_{k+1}^{(j)}(l) - \mathbf{y}_k^{(j)}(l)) \\ z_{k+1}^{(j)} &= z_k^{(j)} + \frac{1}{m_j} \sum_{l: (l,j) \in \mathcal{J}(i)} (\mathbf{y}_{k+1}^{(j)}(l) - \mathbf{y}_k^{(j)}(l)). \end{aligned}$$

Otherwise, set $x_{k+1}^{(i')} = x_k^{(i')}$, $w_{k+1}^{(l')} = w_k^{(l')}$, $z_{k+1}^{(j')} = z_k^{(j')}$ and $\mathbf{y}_{k+1}^{(j')}(l') = \mathbf{y}_k^{(j')}(l')$.

Theorem 2. *Let Assumption 2.1 hold true and*

$$\tau_i < \frac{1}{\beta_i + \rho \left(\sum_{j \in J(i)} (2 - \pi_j(i)) m_j \sigma_j M_{j,i}^* M_{j,i} \right)}. \quad (6)$$

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

$$\begin{aligned} \lim_{k \rightarrow \infty} x_k &= x_* \\ \lim_{k \rightarrow \infty} \mathbf{y}_k^{(j)}(i) &= \mathbf{y}_*^{(j)}(i) \quad (\forall j \in \{1, \dots, p\}, \forall i \in I(j)). \end{aligned}$$

2.5 Special Cases

2.5.1 The case $\mathcal{J}(i) = \{i\} \times J(i)$ for all i

According to (4), the smallest possible choice for $\mathcal{J}(i)$ is $\mathcal{J}(i) = \{i\} \times J(i)$. In that case, $\pi_j(i) = 1$ for all $j \in J(i)$ and the update of the dual variable simplifies to:

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

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

2.5.2 The case $\mathcal{J}(i) = \cup_{j \in J(i)} I(j) \times J(i)$ for all i

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

We have $\pi_j(l) = \frac{1}{|I(j)|} = \frac{1}{m_j}$ for all $l \in I(j)$. The advantage of this update scheme is that, provided there exists y' such that $\mathbf{y}_0^{(j)}(l) = y'_0{}^{(j)}$ for all $l \in I(j)$, we have for all $l \in I(j)$ and all $k \geq 0$,

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

Hence, choosing $\mathcal{J}(i) = \cup_{j \in J(i)} I(j) \times J(i)$ 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 , $\text{nnz}(M)$, to its number of rows p .

This shows the following equivalence result.

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

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

2.5.3 The Case $m_1 = \dots = m_p = 1$

We consider the special case $m_1 = \dots = m_p = 1$. Otherwise stated, 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 context of distributed optimization [6]. 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 $(\mathbf{y}_k^{(j)}(i) : i \in I(j))$ is reduced to a single variable $\mathbf{y}_k^{(j)}(I(j))$, 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. Otherwise stated, 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 below. 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 - Case $m_1 = \dots = m_p = 1$.

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

Iteration k : Define:

$$\begin{aligned} \bar{y}_{k+1} &= \text{prox}_{\sigma, h^*}(y_k + D(\sigma)Mx_k) \\ \bar{x}_{k+1} &= \text{prox}_{\tau, g}\left(x_k - D(\tau)(\nabla f(x_k) + M^*(2\bar{y}_{k+1} - y_k))\right). \end{aligned}$$

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

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

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

2.5.4 The Case $h = 0$

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

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

As a consequence, Algorithm 2 allows to recover the CD proximal gradient algorithm of [38] with the notable difference that we do *not* assume the separability of g . On the other hand, Assumption 2.1(e) becomes $\tau_i < 1/\beta_i$ whereas in the separable case, [38] assumes $\tau_i = 1/\beta_i$. This remark leads us to conjecture that, even though Assumption 2.1(e) generally allows for the use of larger step sizes than the ones suggested by the approach of [15, 6], one might be able to use even larger step sizes than the ones allowed by Theorem 2.

Note that a similar CD forward-backward algorithm can be found in [15] with no need to require the separability of g . However, the algorithm of [15] 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 coordinate-wise Lipschitz constant of ∇f .

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 [29] and extended by [15] and [6] to a more general setting. Unfortunately, this approach implies to select a “small” step size as noticed in the previous section. The use of 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 1. Consider the toy problem

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

that is 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 coordinate-wise 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 is immediately 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 $\tau > 2/3$. 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 3 below, one needs to make use of another Lyapunov function, defined in (19). That inequality 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

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

Lemma 1. *Let Assumption 2.1(a-b) hold true. Let $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and $(x_*, y_*) \in \mathcal{S}$. Define*

$$\begin{aligned} \bar{y} &= \text{prox}_{\sigma, h^*}(y + D(\sigma)Mx) \\ \bar{x} &= \text{prox}_{\tau, g}\left(x - D(\tau)(\nabla f(x) + M^*(2\bar{y} - y))\right) \end{aligned}$$

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

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

Proof. The inclusions (3) also read

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

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

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

$$h^*(\bar{y}) \geq h^*(y_*) + \langle Mx_*, \bar{y} - y_* \rangle. \quad (10)$$

By definition of the proximal operator,

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

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

Consider Equality (11) above. It classically implies [47] that for any $v \in \mathcal{Y}$,

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

Setting $v = y_*$, we obtain

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

and using (10), we finally have

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

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

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

We set $u = x_*$. This yields

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

Using moreover Inequality (9), we obtain

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

hence, rearranging the terms,

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

Summing the above inequality with (15),

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

This completes the proof of the lemma thanks to the definition of V . \square

3.2 Study of Algorithm 3

We first prove Theorem 2 in the special case $m_1 = \dots = m_p = 1$. In that case, Algorithm 2 boils down to Algorithm 3. We recall that in this case, the vector $\mathbf{y}_k^{(j)}$ is reduced to a single value $\mathbf{y}_k^{(j)}(i) \in \mathcal{Y}_j$ 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, \dots, i_k . We denote by $\mathbb{E}_k(\cdot) = \mathbb{E}(\cdot | \mathcal{F}_k)$ the conditional expectation w.r.t. \mathcal{F}_k .

Lemma 2. *Let Assumptions 2.1(a,b,d) hold true. Suppose $m_1 = \dots = m_p = 1$. Consider Algorithm 3 and let $\gamma_1, \dots, \gamma_n, \gamma'_1, \dots, \gamma'_p$ be arbitrary positive coefficients. For every $k \geq 1$ and every \mathcal{F}_k -measurable pair of random variables (X, Y) on $\mathcal{X} \times \mathcal{Y}$,*

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

Proof. The first equality is immediate.

Consider the second one. $\mathbb{E}_k(\|x_{k+1} - X\|_{\gamma}^2) = \sum_{i=1}^n \gamma_i \mathbb{E}_k(\|x_{k+1}^{(i)} - X^{(i)}\|^2)$ which coincides with $\sum_{i=1}^n \gamma_i (\frac{1}{n} \|\bar{x}_{k+1}^{(i)} - X^{(i)}\|^2 + (1 - \frac{1}{n}) \|x_k^{(i)} - X^{(i)}\|^2)$ and the second equality is proved.

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

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

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

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

From (5),

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

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

$$\|y_k^{(j)} + \pi_j(\bar{y}_{k+1}^{(j)} - y_k^{(j)}) - Y^{(j)}\|^2 = \pi_j \|\bar{y}_{k+1}^{(j)} - Y^{(j)}\|^2 + (1 - \pi_j) \|y_k^{(j)} - Y^{(j)}\|^2 - \pi_j(1 - \pi_j) \|\bar{y}_{k+1}^{(j)} - y_k^{(j)}\|^2$$

This leads to

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

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

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

Finally, we obtain

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

which in turn implies the fourth equality in the Lemma. \square

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

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

Lemma 3. *Let Assumptions 2.1(a,b,c,d) hold true. Suppose $m_1 = \dots = m_p = 1$ and assume that $\tau_i^{-1} > \beta_i$ for each $i \in \{1, \dots, n\}$. Consider Algorithm 3 and define for every $k \in \mathbb{N}$,*

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

Then the following inequality holds:

$$\mathbb{E}_k [S_{k+1,*} + V(z_{k+1} - z_*)] \leq (1 - \frac{1}{n})S_{k,*} + V(z_k - z_*) - \frac{1}{n}\tilde{V}(\bar{z}_{k+1} - z_k) \quad (19)$$

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

Proof. We can write the relations of Lemma 2 as

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

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

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

We shall denote

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

Let $z_* = (x_*, y_*) \in \mathcal{S}$. By Lemma 1,

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

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

$$\langle \nabla f(x_*) - \nabla f(x_k), x_* - \bar{x}_{k+1} \rangle + n\mathbb{E}_k(V(z_{k+1} - z_k)) \leq nV(z_k - z_*) - n\mathbb{E}_k(V(z_{k+1} - z_*)) - 2R_\pi$$

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

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

Rearranging the terms,

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

We now use Assumption 2.1(c), knowing that x_{k+1} only differs from x_k along coordinate i_{k+1}

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

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

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

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

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

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

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

Using Lemma 2, (17) and (21), it is immediate that

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

and the proof is complete. \square

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

Lemma 4. *Suppose that $m_1 = \dots = m_p = 1$ and assume that the following condition holds for every $i \in \{1, \dots, n\}$:*

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

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

Note that under the assumptions of Lemma 4, $V^{1/2}$ 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 $\sigma'_j = (2 - \pi_j) \sigma_j$ for all j and by $D(\sigma')$ the diagonal matrix on $\mathcal{Y} \rightarrow \mathcal{Y}$ 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 $\mathcal{X} \rightarrow \mathcal{X}$. By [28, Theorem 7.7.6], a sufficient (and necessary) condition for \tilde{V} to be a squared norm is that $D(\gamma^{-1}) \succ M^* D(\sigma') M$ (where notation $A \succ B$ means that $A - B$ is a positive definite matrix). Defining $R = D(\sigma'^{1/2}) M D(\gamma^{1/2})$ (that is, $R_{j,i} = \sqrt{\gamma_i \sigma'_j} M_{j,i}$ for every j, i), the condition reads equivalently $\rho(R^* R) < 1$. As the set $I(j)$ is reduced to a unique element for all j , the matrix $R^* R$ is (block) diagonal. Precisely, for any $1 \leq i, \ell \leq n$, the (i, ℓ) -component $(R^* R)_{i,\ell}$ is zero whenever $i \neq \ell$ and is equal to $(R^* R)_{i,i} = \gamma_i \sum_{j \in J(i)} \sigma'_j M_{j,i}^* M_{j,i}$ otherwise. The condition $\rho(R^* R) < 1$ yields $\gamma_i \rho\left(\sum_{j \in J(i)} \sigma'_j M_{j,i}^* M_{j,i}\right) < 1$ for each $i \in \{1, \dots, n\}$ which is in turn equivalent to (24). \square

Proof of Theorem 1 in the case $m_1 = \dots = m_p = 1$. Let z_* be an arbitrary point in \mathcal{S} . Whenever condition (24) is met, the r.v. $V(z_k - z_*)$ and $\tilde{V}(\bar{z}_{k+1} - z_k)$ are non-negative. The r.v. $S_{k,*}$ is non-negative as well by convexity of f . We review two important consequences of Lemma 3.

- Define $U_k := S_{k,*} + V(z_k - z_*)$. A first consequence of Lemma 3 is that for all k ,

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

Recalling that U_k and S_k are non-negative r.v., the Robbins-Siegmund Lemma [42] implies that almost surely, $\lim_{k \rightarrow \infty} U_k$ exists and $\sum_k S_{k,*} < \infty$. In particular, $S_{k,*}$ converges almost surely to zero. By definition of U_k , this implies that $\lim_{k \rightarrow \infty} V(z_k - z_*)$ exists almost surely. Following the argument of [3, Prop. 9] (see also [29], [15, Prop. 2.3]), this implies that there exists an event A of probability one such that for every $\omega \in A$ and every $\tilde{z} \in \mathcal{S}$, $\lim_{k \rightarrow \infty} V^{1/2}(z_k(\omega) - \tilde{z})$ exists.

- A second consequence of Lemma 3 is that, by taking the expectation \mathbb{E} of both handsides of (19),

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

and by summing these inequalities, we obtain

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

Thus $\mathbb{E}(\sum_{i=0}^{\infty} \tilde{V}(\bar{z}_{i+1} - z_i)) < \infty$. The integrand is non-negative by Lemma 4. It is therefore finite almost everywhere. In particular, the sequence $\tilde{V}(\bar{z}_{k+1} - z_k)$ converges almost surely to zero. By Lemma 4, $\bar{z}_{k+1} - z_k$ converges to zero almost surely. Say $\bar{z}_{k+1}(\omega) - z_k(\omega) \rightarrow 0$ for every $\omega \in B$ where B is a probability event of probability one.

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

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

With this definition, $\bar{z}_{k+1} = T(z_k)$. By non-expansiveness 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} .

From now on to the end of this paragraph, 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 $T(z_k(\omega)) - z_k(\omega) \rightarrow 0$ which implies that $T(\tilde{z}) - \tilde{z} = 0$ by continuity of T . Thus, $\tilde{z} \in \mathcal{S}$. This implies that $\lim_{k \rightarrow \infty} V^{1/2}(z_k(\omega) - \tilde{z})$ exists. Since $V^{1/2}(z_k(\omega) - \tilde{z})$ tends to zero at least on some subsequence, we conclude that $\lim_{k \rightarrow \infty} V^{1/2}(z_k(\omega) - \tilde{z}) = 0$. Otherwise stated, the sequence $z_k(\omega)$ converges to some point $\tilde{z} \in \mathcal{S}$. This completes the proof of Theorem 2 in the case $m_1 = \dots = m_p = 1$. \square

3.3 General Case

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

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

The averaging operators allows 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 \mathcal{Y}_j whose components all coincide with u .

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

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

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

$$M = D(m)SK \quad (26)$$

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

Example 2. Let $\mathcal{X} = \mathcal{Y} = \mathbb{R}^3$ and define $M : \mathcal{X} \rightarrow \mathcal{Y}$ 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 non-zero coefficients of the first row of M and its cardinal is $m_1 = 2$. Similarly $m_2 = 1$, $m_3 = 3$ and $\mathcal{Y} = \mathbb{R}^6$. Then $K : \mathbb{R}^3 \rightarrow \mathbb{R}^6$ 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 non-zero coefficient. On the other hand, S and $D(m)$ respectively coincide with

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

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

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

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

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

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

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 one. Algorithm 3 writes

Initialization: Choose $x_0 \in \mathcal{X}$, $\mathbf{y}_0 \in \mathcal{Y}$.

Iteration k : Define:

$$\bar{\mathbf{y}}_{k+1} = \text{prox}_{\tilde{\sigma}, \bar{h}^*}(\mathbf{y}_k + D(\tilde{\sigma})Kx_k) \quad (28)$$

$$\bar{x}_{k+1} = \text{prox}_{\tau, g}\left(x_k - D(\tau)(\nabla f(x_k) + K^*(2\bar{\mathbf{y}}_{k+1} - \mathbf{y}_k))\right). \quad (29)$$

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

$$x_{k+1}^{(i)} = \bar{x}_{k+1}^{(i)} \quad (30)$$

$$\mathbf{y}_{k+1}^{(j)}(l) = \mathbf{y}_k^{(j)}(l) + \pi_j(l)(\bar{\mathbf{y}}_{k+1}^{(j)}(l) - \mathbf{y}_{k+1}^{(j)}(l)). \quad (31)$$

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

Using the result of the Section 3.2 and the properties of K , the sequence (x_k, \mathbf{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:

$$\tau_i < \frac{1}{\beta_i + \rho \left(\sum_{(l,j) \in \{i\} \times J(i)} (2 - \pi_j(l)) \tilde{\sigma}_j K_{(l,j),i}^* K_{(l,j),i} \right)} = \frac{1}{\beta_i + \rho \left(\sum_{j \in J(i)} (2 - \pi_j(i)) \tilde{\sigma}_j M_{j,i}^* M_{j,i} \right)}$$

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 5. *For any $\mathbf{y} \in \mathcal{Y}$,*

$$\text{prox}_{\bar{\sigma}, \bar{h}^*}(\mathbf{y}) = (\mathbf{1}_{m_1} \otimes \text{prox}_{\sigma, h^*}^{(1)}(S(\mathbf{y})), \dots, \mathbf{1}_{m_p} \otimes \text{prox}_{\sigma, h^*}^{(p)}(S(\mathbf{y}))).$$

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

$$\bar{h}^*(\varphi) = \sup_{\mathbf{y} \in \mathcal{Y}} \langle \varphi, \mathbf{y} \rangle - h(m_1 S_1(\mathbf{y}^{(1)}), \dots, m_p S_p(\mathbf{y}^{(p)}))$$

For all $j \in \{1, \dots, p\}$, denote by \mathcal{C}_j the subset of \mathcal{Y}_j formed by the vectors of the form (u, \dots, u) for some $u \in \mathcal{Y}_j$, and define $\mathcal{C} = \mathcal{C}_1 \times \dots \times \mathcal{C}_p$. Clearly, $\bar{h}^*(\varphi) = +\infty$ whenever $\varphi \notin \mathcal{C}$ and $\partial \bar{h}^*(\varphi) = \emptyset$ in that case. If on the other hand $\varphi \in \mathcal{C}$, one can write φ under the form $\varphi = (\mathbf{1}_{m_1} \otimes \varphi^{(1)}, \dots, \mathbf{1}_{m_p} \otimes \varphi^{(p)})$ for some $\varphi \in \mathcal{Y}$. In that case,

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

Then, $\mathbf{u} \in \partial \bar{h}^*(\varphi)$ if and only if for every $\psi \in \mathcal{Y}$, $h^*(\psi) \geq h^*(\varphi) + \sum_{j=1}^p \langle \mathbf{u}^{(j)}, \mathbf{1}_{m_j} \otimes (\psi^{(j)} - \varphi^{(j)}) \rangle$ or equivalently, $h^*(\psi) \geq h^*(\varphi) + \sum_{j=1}^p \langle m_j S_j(\mathbf{u}^{(j)}), \psi^{(j)} - \varphi^{(j)} \rangle$. Therefore, $\mathbf{u} \in \partial \bar{h}^*(\varphi)$ if and only if $D(m)S(\mathbf{u}) \in \partial h^*(\varphi)$.

Now consider an arbitrary $\mathbf{y} \in \mathcal{Y}$ and set $\mathbf{q} = \text{prox}_{\bar{\sigma}, \bar{h}^*}(\mathbf{y})$. This is equivalent to

$$D(\bar{\sigma}^{-1})(\mathbf{y} - \mathbf{q}) \in \partial \bar{h}^*(\mathbf{q}). \quad (32)$$

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

The proof of the following Lemma is immediate.

Lemma 6. *For any $\mathbf{y} \in \mathcal{Y}$,*

$$K^*(\mathbf{y}) = \left(\sum_{j \in J(1)} M_{j1}^*(\mathbf{y}^{(j)}(1)), \dots, \sum_{j \in J(n)} M_{jn}^*(\mathbf{y}^{(j)}(n)) \right).$$

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

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

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

Example 3 (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

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

and M defined by blocks of the type

$$M_{(i_1, i_2, i_3)} = \begin{pmatrix} (x_{i_1, i_2, i_3}) & (x_{i_1+1, i_2, i_3}) & (x_{i_1, i_2+1, i_3}) & (x_{i_1, i_2, i_3+1}) \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} (\mathbf{y}_{1, i_1, i_2, i_3}) \\ (\mathbf{y}_{2, i_1, i_2, i_3}) \\ (\mathbf{y}_{3, i_1, i_2, i_3}) \end{pmatrix}$$

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

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

Hence, we can write $\bar{h}_{i_1, i_2, i_3}(\mathbf{y}_{i_1, i_2, i_3, :}) = \sqrt{\sum_{j=1}^3 (\mathbf{y}_{i_1, i_2, i_3, j} (1) + \mathbf{y}_{i_1, i_2, i_3, j} (2))^2}$

$\text{prox}_{m\sigma, \bar{h}^*}(\mathbf{y}) = (\mathbf{1}_{m_1} \otimes \text{prox}_{\sigma, h^*}^{(1)}(S(\mathbf{y})), \dots, \mathbf{1}_{m_p} \otimes \text{prox}_{\sigma, h^*}^{(p)}(S(\mathbf{y})))$ becomes, denoting e_l the l th coordinate vector,

$$\text{prox}_{2\sigma, \bar{h}_{i_1, i_2, i_3}^*}(\mathbf{y}_{i_1, i_2, i_3, :}) = \begin{pmatrix} e_1^\top \text{prox}_{\sigma, h_{i_1, i_2, i_3}^*}(\mathbf{y}_{i_1, i_2, i_3, :} (1) + \mathbf{y}_{i_1, i_2, i_3, :} (2)) \\ e_1^\top \text{prox}_{\sigma, h_{i_1, i_2, i_3}^*}(\mathbf{y}_{i_1, i_2, i_3, :} (1) + \mathbf{y}_{i_1, i_2, i_3, :} (2)) \\ e_2^\top \text{prox}_{\sigma, h_{i_1, i_2, i_3}^*}(\mathbf{y}_{i_1, i_2, i_3, :} (1) + \mathbf{y}_{i_1, i_2, i_3, :} (2)) \\ e_2^\top \text{prox}_{\sigma, h_{i_1, i_2, i_3}^*}(\mathbf{y}_{i_1, i_2, i_3, :} (1) + \mathbf{y}_{i_1, i_2, i_3, :} (2)) \\ e_3^\top \text{prox}_{\sigma, h_{i_1, i_2, i_3}^*}(\mathbf{y}_{i_1, i_2, i_3, :} (1) + \mathbf{y}_{i_1, i_2, i_3, :} (2)) \\ e_3^\top \text{prox}_{\sigma, h_{i_1, i_2, i_3}^*}(\mathbf{y}_{i_1, i_2, i_3, :} (1) + \mathbf{y}_{i_1, i_2, i_3, :} (2)) \end{pmatrix}$$

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

- The dual variables corresponding to $x_{3,4,5}$ are $\mathbf{y}_{3,4,5,1}(1)$, $\mathbf{y}_{3,4,5,2}(1)$, $\mathbf{y}_{3,4,5,3}(1)$, $\mathbf{y}_{2,4,5,1}(2)$, $\mathbf{y}_{3,3,5,2}(2)$ and $\mathbf{y}_{3,4,4,3}(2)$.
- We compute $\text{prox}_{2\sigma, \bar{h}_{3,4,5}^*}(\mathbf{y}_{3,4,5, :})$, $\text{prox}_{2\sigma, \bar{h}_{2,4,5}^*}(\mathbf{y}_{2,4,5, :})$, $\text{prox}_{2\sigma, \bar{h}_{3,3,5}^*}(\mathbf{y}_{3,3,5, :})$ and $\text{prox}_{2\sigma, \bar{h}_{3,4,4}^*}(\mathbf{y}_{3,4,4, :})$, which amounts to 12 real numbers.
- We update only the 6 useful dual values.

We are now in a position to simplify the iterations (28)–(31). For every k , we define the vectors $\bar{\mathbf{y}}_{k+1} = \text{prox}_{\sigma, h^*}(S(\mathbf{y}_k + D(\tilde{\sigma})Kx_k))$ and $\bar{\mathbf{y}}_{k+1} = (\mathbf{1}_{m_1} \otimes \bar{\mathbf{y}}_{k+1}^{(1)}, \dots, \mathbf{1}_{m_p} \otimes \bar{\mathbf{y}}_{k+1}^{(p)})$. Upon noting that $SD(\tilde{\sigma})K = D(\sigma)D(m)SK = D(\sigma)M$, we obtain

$$\bar{\mathbf{y}}_{k+1} = \text{prox}_{\sigma, h^*}(z_k + D(\sigma)Mx_k) \quad (33)$$

where we defined $z_k = S(\mathbf{y}_k)$, otherwise stated, for each $j \in \{1, \dots, p\}$,

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

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

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

Now consider equation (29). By Lemma 6, $K^*\bar{\mathbf{y}}_{k+1} = M^*\bar{\mathbf{y}}_{k+1}$. Thus, setting $w_k = K^*\mathbf{y}_k$, equation (29) simplifies to:

$$\bar{x}_{k+1} = \text{prox}_{\tau, g} \left(x_k - D(\tau) (\nabla f(x_k) + (2M^*\bar{\mathbf{y}}_{k+1} - w_k)) \right). \quad (35)$$

By Lemma 6 again, $w_k = (\sum_{j \in J(1)} M_{j1}^* \mathbf{y}_k^{(j)}(1), \dots, \sum_{j \in J(n)} M_{jn}^* \mathbf{y}_k^{(j)}(n))$. 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:

$$w_{k+1}^{(i)} = w_k^{(i)} + \sum_{(i,j) \in \mathcal{J}(i_{k+1})} M_{j,i}^* (\mathbf{y}_{k+1}^{(j)}(i) - \mathbf{y}_k^{(j)}(i)). \quad (36)$$

Putting all pieces together, the update equations (33)–(36) coincide with Algorithm 2. We have thus proved that Algorithm 2 is such that (x_k, \mathbf{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 $G : \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{X} \times \mathcal{Y}$ defined by

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

Lemma 7. $\mathcal{S} = G(\mathcal{S})$.

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

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

Moreover,

$$\begin{aligned} 0 \in -Kx + \partial \bar{h}^*(\mathbf{y}) &\Leftrightarrow Kx \in \partial \bar{h}^*(\mathbf{y}) \\ &\Leftrightarrow D(m)S(Kx) \in \partial h^*(y) \\ &\Leftrightarrow Mx \in \partial h^*(y) \end{aligned}$$

where we used Lemma 5 along with the identities $D(m)SK = M$ and $S(\mathbf{y}) = y$. The proof is completed upon noting that if $(x, \mathbf{y}) \in \mathcal{S}$, then there exists $y \in \mathcal{Y}$ such that \mathbf{y} has the form $\mathbf{y} = (\mathbf{1}_{m_1} \otimes y^{(1)}, \dots, \mathbf{1}_{m_p} \otimes y^{(p)})$. \square

We have shown that, almost surely, (x_k, \mathbf{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 a $O(1/\sqrt{k})$ decrease for the function value (Theorem 3).
- $h = I_{\{b\}}$, *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 a $O(1/\sqrt{k})$ decrease for the function value and the feasibility (Theorem 3).
- $f + g$ is strongly convex and ∇h is Lipschitz continuous: we prove a $O(e^{-\mu k})$ rate for the distance to the optimum (Theorem 4).

These convergence guarantees are of the same order as what can be obtained by other primal-dual methods like the ADMM [18], *i.e.* $O(1/\sqrt{k})$ in general and linear rate of convergence under strong convexity assumptions.

Theorem 3. Define for $\alpha \geq 1$,

$$C_{1,\alpha} = \max_{1 \leq i \leq n} \frac{\tau_i^{-1} + \tau_i^{-1/2} \rho(\sum_{j \in J(i)} m_j \sigma_j M_{j,i}^* M_{j,i})^{1/2}}{\tau_i^{-1} - \rho(\sum_{j \in J(i)} m_j \sigma_j M_{j,i}^* M_{j,i})} \left(1 + \frac{n}{\alpha}\right)$$

$$C_{2,\alpha} = \left(1 + \max_{1 \leq i \leq n} \frac{\alpha^{-1}(n(n-1) + 1) + 1}{\tau_i^{-1} - \beta_i - \rho(\sum_{j \in J(i)} (2 - \pi_j(i)) m_j \sigma_j M_{j,i}^* M_{j,i})} \beta_i\right).$$

We have that $C_{1,\alpha}$ and $C_{2,\alpha}$ are nonincreasing with respect to α , and 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 $\Pr(\hat{K} = l) = \frac{1}{k}$ for all $l \in \{1, \dots, k\}$.

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 $k \geq 0$,

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

$$\mathbb{E}(\|M\bar{x}_{\hat{K}} - b\|_{D(m)\sigma}) \leq \frac{2}{\sqrt{k}} \left(\sqrt{C_{2,\sqrt{k}} + 2C_{1,k}} + \sqrt{2C_{1,k}}\right) (n(S_{0,*} + V(z_0 - z_*)))^{1/2}$$

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

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

$$g(\bar{x}_{k+1}) + \langle \bar{x}_{k+1}, \nabla f(x_k) + M^*(2\bar{y}_{k+1} - y_k) \rangle + \frac{1}{2} \|\bar{x}_{k+1} - x_k\|_{\tau^{-1}}^2$$

$$\stackrel{(16)}{\leq} g(x) + \langle x, \nabla f(x_k) + M^*(2\bar{y}_{k+1} - y_k) \rangle + \frac{1}{2} \|x - x_k\|_{\tau^{-1}}^2 - \frac{1}{2} \|\bar{x}_{k+1} - x\|_{\tau^{-1}}^2$$

$$h^*(\bar{y}_{k+1}) - \langle \bar{y}_{k+1}, Mx_k \rangle + \frac{1}{2} \|\bar{y}_{k+1} - y_k\|_{\sigma^{-1}}^2 \stackrel{(13)}{\leq} h^*(y) - \langle y, Mx_k \rangle + \frac{1}{2} \|y - y_k\|_{\sigma^{-1}}^2 - \frac{1}{2} \|\bar{y}_{k+1} - y\|_{\sigma^{-1}}^2$$

$$\mathbb{E}_k(f(x_{k+1})) \stackrel{(23)+Lem. 2}{\leq} f(x_k) + \frac{1}{n} \langle \nabla f(x_k), \bar{x}_{k+1} - x_k \rangle + \frac{1}{2n} \|\bar{x}_{k+1} - x_k\|_{\beta}^2$$

$$f(x) \geq f(x_k) + \langle \nabla f(x_k), x - x_k \rangle$$

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

$$g(\bar{x}_{k+1}) + n\mathbb{E}_k(f(x_{k+1})) - (n-1)f(x_k) + \langle M\bar{x}_{k+1}, y \rangle - h^*(y) + h^*(\bar{y}_{k+1})$$

$$- \langle M^\top \bar{y}_{k+1}, x \rangle - g(x) - f(x) \leq V(z_k - z) - V(\bar{z}_{k+1} - z) - V(\bar{z}_{k+1} - z_k) + \frac{1}{2} \|\bar{x}_{k+1} - x_k\|_{\beta}^2$$

As ∇f is n -Lipschitz in the norm $\|\cdot\|_{\beta}$ [40] and $n\mathbb{E}(x_{k+1}) - (n-1)x_k - \bar{x}_{k+1} = 0$,

$$n\mathbb{E}_k(f(x_{k+1})) - (n-1)f(x_k) \geq n\mathbb{E}_k(f(\bar{x}_{k+1}) + \langle \nabla f(\bar{x}_{k+1}), x_{k+1} - \bar{x}_{k+1} \rangle)$$

$$- (n-1)(f(\bar{x}_{k+1}) + \langle \nabla f(\bar{x}_{k+1}), x_k - \bar{x}_{k+1} \rangle) + \frac{n}{2} \|x_k - \bar{x}_{k+1}\|_{\beta}^2$$

$$\geq f(\bar{x}_{k+1}) - \frac{n(n-1)}{2} \|x_k - \bar{x}_{k+1}\|_{\beta}^2.$$

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

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

Gathering everything, we get

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

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

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

where $C_{2,\alpha} \in O(1)$ for $\alpha \rightarrow \infty$. Denoting the smoothed gap [46] as

$$\begin{aligned} \mathcal{G}_\alpha(\bar{z}_k, \bar{z}_k) &= \sup_z g(\bar{x}_k) + f(\bar{x}_k) + \langle M\bar{x}_k, y \rangle - h^*(y) + h^*(\bar{y}_k) \\ &\quad - \langle M^\top \bar{y}_k, x \rangle - g(x) - f(x) - \frac{2}{2\alpha} \|\bar{x}_k - x\|_{\tau-1}^2 - \frac{2}{2\alpha} \|\bar{y}_k - y\|_{\sigma-1}^2, \end{aligned}$$

we have

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

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

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

so

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

Taking $\alpha = \sqrt{k}$ as in [18], we get

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

We can also bound

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

where the last inequality follows from $R_{\pi}^{(i)} - \tilde{V}(\bar{z}_i - z_{i-1}) = \frac{1}{2} \|\bar{x}_i - x_{i-1}\|_{\beta}^2 - V(\bar{z}_i - z_{i-1}) \leq 0$.

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 $x = x^*$ so that $h^*(\bar{y}_k) - \langle M^{\top} \bar{y}_k, x^* \rangle \geq -h(Mx^*)$

We then use the inequality

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

to conclude.

If $h = I_{\{b\}}$, then using Lemma 1 in [46], we get that

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

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

Remark 1. To prove the result of Theorem 3, we use a random number of iterations. This has also been proposed for instance in [43] 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 4.1. There exists non-negative constants μ_g and μ_f such that $\mu_f + \mu_g > 0$ and a constant $\mu_{h^*} > 0$ 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^* is μ_{h^*} -strongly convex in the norm $\|\cdot\|_{\sigma-1}$.

Theorem 4. For $z = (x, \mathbf{y})$, denote $V^\mu(z) = V(z) + \mu_g \|x\|_{\tau-1}^2 + \mu'_{h^*} \|\mathbf{y}\|_{(D(m)\sigma)^{-1}}^2$ where $\mu'_{h^*} = \min(\mu_{h^*}, \sup\{\mu > 0 : \forall i, \tau_i^{-1} > \beta_i + \rho(\sum_{j \in J(i)} \frac{(2-\pi_j(i))^2 \sigma_j m_j}{2-\pi_j(i)-\mu(1-\pi_j(i))} M_{j,i}^* M_{j,i})\}$ (note that if $\pi_j(i) = 1$ for all i and j , then $\mu'_{h^*} = \mu_{h^*}$). If Assumption 4.1 holds then the iterates of Algorithm 2 satisfy

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

In order to prove this theorem, we begin with a lemma that generalizes Lemma 1.

Lemma 8. If Assumption 4.1 holds, then

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

Proof. Assumption 4.1 gives us: for $(x_*, y_*) \in \mathcal{S}$,

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

$$h^*(\bar{y}) \geq h^*(y_*) + \langle M x_*, \bar{y} - y_* \rangle + \frac{\mu_{h^*}}{2} \|\bar{y} - y_*\|_{\sigma-1}^2. \quad (38)$$

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

$$h^*(\bar{y}) \leq h^*(y_*) + \langle \bar{y} - y_*, M x \rangle + \frac{1}{2} \|y_* - y\|_{\sigma-1}^2 - \frac{1 + \mu_{h^*}}{2} \|\bar{y} - y_*\|_{\sigma-1}^2 - \frac{1}{2} \|\bar{y} - y\|_{\sigma-1}^2$$

and so using (38)

$$\langle M(x_* - x), \bar{y} - y_* \rangle \leq \frac{1}{2} \|y - y_*\|_{\sigma-1}^2 - \frac{1 + 2\mu_{h^*}}{2} \|\bar{y} - y_*\|_{\sigma-1}^2 - \frac{1}{2} \|\bar{y} - y\|_{\sigma-1}^2 \quad (39)$$

Similarly, we have

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

Summing the above inequality with (39), and recalling the definition of $V(z) = V(x, y) = \frac{1}{2} \|x\|_{\tau-1}^2 + \langle y, Mx \rangle + \frac{1}{2} \|y\|_{\sigma-1}^2$, we get

$$\langle \nabla f(x_*) - \nabla f(x), x_* - \bar{x} \rangle + V(\bar{z} - z) \leq V(z - z_*) - V(\bar{z} - z_*) - \mu_g \|\bar{x} - x\|_{\tau-1}^2 - \mu_{h^*} \|\bar{y} - y\|_{\sigma-1}^2$$

□

Proof of Theorem 4. We begin with the case $m_1 = \dots = m_p$.

By Assumption 2.1(e), if $\mu_{h^*} > 0$, then $\mu'_{h^*} > 0$ and if h^* is μ_{h^*} -strongly convex, it is also μ'_{h^*} -strongly convex. Then, by a straightforward adaptation of the proof of Lemma 3 to the strongly convex case, we have

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

As soon as $\tau_i^{-1} > \beta_i + \rho\left(\sum_{j \in J(i)} \frac{(2-\pi_j)^2}{2-\pi_j-\mu'_{h^*}(1-\pi_j)} \sigma_j M_{j,i}^* M_{j,i}\right)$, we can remove the term $-\frac{1}{n} \tilde{V}(\bar{z}_{k+1} - z_k) + \frac{\mu'_{h^*}}{n} \|\bar{y}_{k+1} - y_k\|_{\sigma-1(1-\pi)}^2 \leq 0$. This is indeed guaranteed by the definition of μ'_{h^*} .

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,

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

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

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

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

$$\begin{aligned} \eta^2 \left(\tau_i^{-1} - \sum_{j \in J(i)} \sigma_j M_{j,i}^* M_{j,i} \right) + \tau_i^{-1} \left(\eta^2 (2\mu_g + 2\mu'_{h^*}) \right. \\ \left. - \eta \left(\frac{\mu_f + 2\mu_g + 2\mu'_{h^*}}{n} - \frac{4\mu_g \mu'_{h^*}}{n} - \frac{2\mu_f \mu'_{h^*} + 4\mu_g \mu'_{h^*}}{n} \right) + \frac{(\mu_f + 2\mu_g)\mu'_{h^*}}{n^2} \right) \geq 0. \end{aligned}$$

The first term is nonnegative thanks to Assumption 2.1(e). The second term is nonnegative as soon as

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

To conclude, we remark that

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

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

Remark 2. 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 on Figure 2 we do observe linear convergence in some cases, even when Assumption 4.1 is not satisfied. Thus we think that Theorem 4 can give an indication of how the algorithm behaves in favorable cases.

Remark 3. Of particular interest is the relation between the rate proved in Theorem 4 and the size of the steps. Having longer step sizes improves the rate greatly since μ_f , μ_g and μ_{h^*} , measured in the weighted norm, are “proportional” to the step-sizes: as $\mu_g \|x\|_{\tau^{-1}}^2 = (\alpha\mu_g) \|x\|_{(\alpha\tau)^{-1}}^2$ for all $\alpha > 0$, multiplying the step-sizes by $\alpha > 1$ also multiplies μ_f , μ_g and μ_{h^*} by α , which leads to an improved rate $1 - \frac{1}{n} \frac{(\alpha\mu_f + 2\alpha\mu_g)\alpha\mu_{h^*}}{\alpha\mu_f + 2\alpha\mu_g + \alpha\mu_{h^*}} = 1 - \alpha \frac{1}{n} \frac{(\mu_f + 2\mu_g)\mu_{h^*}}{\mu_f + 2\mu_g + \mu_{h^*}} < 1 - \frac{1}{n} \frac{(\mu_f + 2\mu_g)\mu_{h^*}}{\mu_f + 2\mu_g + \mu_{h^*}}$.

As shown in Section 5.2, in large scale applications one can expect much more than twice larger steps and so we can expect a much faster algorithm by using large steps than by using the steps proposed in [29].

5 Numerical Experiments

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

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:

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

The problem takes place on a 3D image of the brains of size $40 \times 48 \times 34$. The optimization variable x is a real vector with one entry in each voxel, that is $n = 65,280$. Matrix M is the discretized 3D gradient. This is a sparse matrix of size $195,840 \times 65,280$ with 2 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 [21]. 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 Cython² and duplicated each dual variable two times. Note that as $h = \alpha(1 - r) \|\cdot\|_{2,1}$ is not separable, we need to compute 12 dual components of \bar{y}_{k+1} for each primal variable $x_{k+1}^{(i)}$ updated and then use only 6 of them to update $z_{k+1}^{(j)}$ for $j \in J(i_{k+1})$. This procedure is explained in detail in Section 3.3. We chose σ_j such that $\rho(\sum_{j \in J(i)} \sigma_j M_{j,i}^* M_{j,i})$ is of the same order of magnitude as β_i and τ_i equal to 0.95 times its upper bound in Assumption 2.1. We compared Algorithm 2 against:

- Vü-Condat’s algorithm [51, 16],
- Chambolle-Pock’s algorithm [12],
- FISTA [2] with an inexact resolution of the proximal operator of TV and a momentum factor ensuring convergence [10],
- L-BFGS [54] 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.

Note that Chambolle-Pock needs to compute the singular values decomposition of A (which explains the flat shape of the performance curve when the algorithm starts). FISTA and Vü-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 behaviour as Algorithm 2 except for $\alpha = 0.1$, $r = 0.9$ where it suffers from the non-smoothness 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:

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

²The code is available on <http://perso.telecom-paristech.fr/~ofercoq/Software.html>

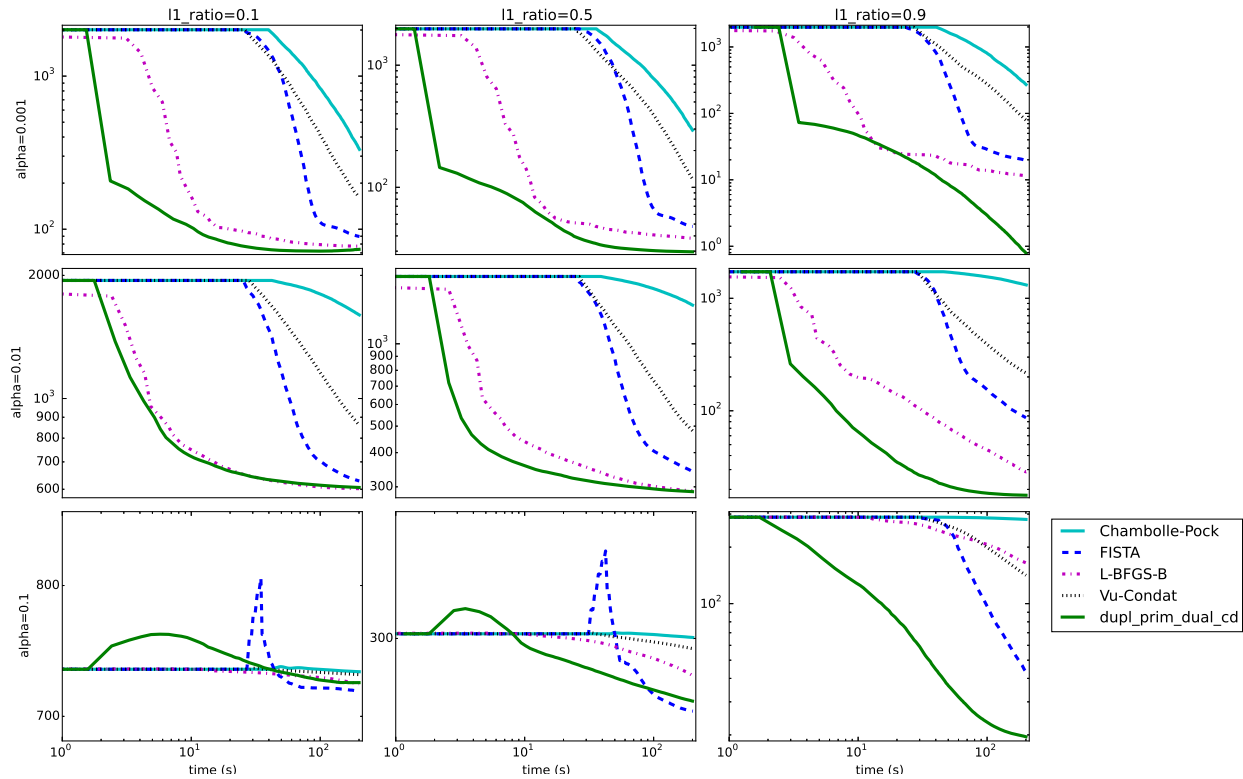


Figure 1: Comparison of algorithms for $TV+L_1$ -regularized regression at various regularization parameters. For each problem, we compute the dual function at the last iterate (this amounts to solving a Lasso problem). Then we compare the primal objective curves to this reference value and we plot them in logarithmic scale. Note 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 non-differentiable and non-separable.

As is common practice for this problem, we solve instead the Dual Support Vector Machine problem:

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

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 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 experiments³, we consider:

- the RCV1 dataset [30] 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 $C_i = \frac{1}{n}$ for all 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 [26]: the data is a mix of 14740 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 a too small 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 $f(x) = \frac{1}{2} \|AD(b)x\|_2^2 - e^T x$, $g(x) = \sum_{i=1}^n I_{[0, C_i]}(x_i)$, $h(y) = I_{\{0\}}(y)$ (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:

- SCDA [43]: note that SDCA simply forgets $I_{\{0\}}(y)$ in order to be able to apply the classical coordinate descent method and thus will not converge to an optimal solution.
- RCD [34]: at each iteration, the algorithm selects two coordinates randomly and performs a coordinate descent step according to these two variables. Updating two variables at a times allows us to satisfy the linear constraint at each iteration.
- Primal-dual coordinate descent (PD-CD) with small steps using the step size $\tau < \frac{1}{\beta(f)/2 + \sigma\rho(K^*K)}$ as in [29].
- Algorithm 2 with $\mathcal{J}(i) = \{i\} \times J(i)$ for all i (PC-CD).
- Algorithm 1, i.e. Algorithm 2 with $\mathcal{J}(i) = \cup_{1 \leq j \leq n} \{j\} \times J(j)$ for all i in order to maintain a single Lagrange multiplier (PD-CD without duplication).

We can see on 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 of [34] does work but is not competitive in terms of rate of convergence. Also, as expected, using small steps [29] 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 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 [14] but it needed 175s to solve the (medium-size) RCV1 problem.

³Code available on <https://github.com/ofercq/lightning>

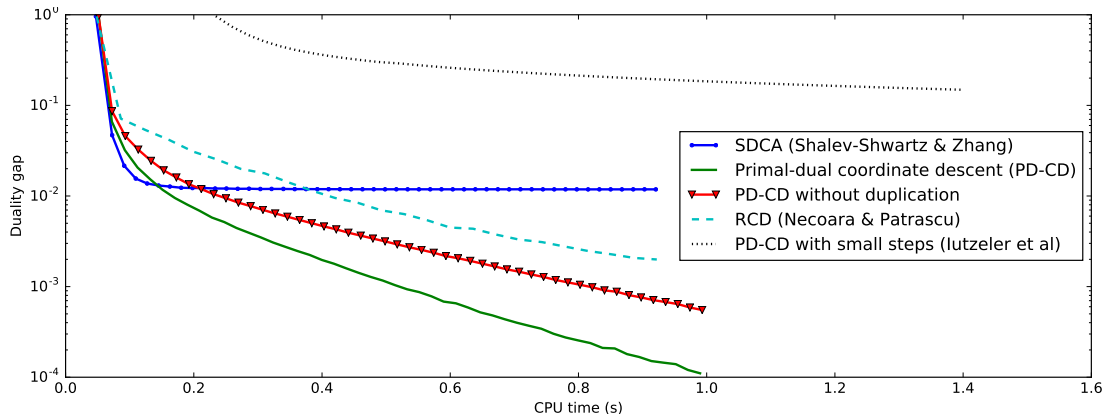


Figure 2: Comparison of dual algorithms for the resolution of linear SVM on the RCV1 dataset. We report the value of the duality gap after a post-processing to recover feasible primal and dual variables. Primal variables are recovered as suggested in [43] and the intercept is recovered by exact minimization 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 [29] needs first to compute the Lipschitz constant of the gradient.

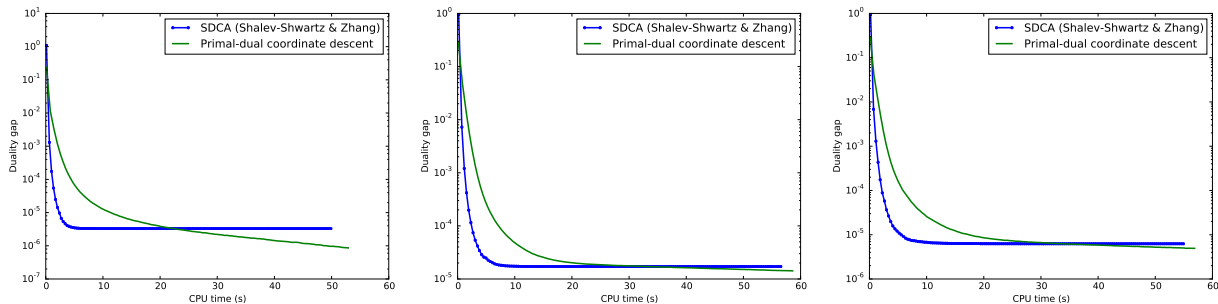


Figure 3: Comparison of dual algorithms for the resolution of linear SVM on the KDD cup 2009 dataset for the appetency, churn and up-selling tasks (one plot for each). We did the same post-processing as in Fig. 2. We stopped each algorithm after 300 passes through the data. We can see here also that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

6 Conclusion

In this work, we combined features of two seemingly incompatible versions of coordinate descent: one based on Fejér monotonicity [15], which allows non-separable non-smooth functions, and one based on the decrease of the function value [38], 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 Vũ-Condat’s method, one should be able to replace β_i by $\beta_i/2$ in the step size condition. A more prospective research, motivated by [33], consists in studying the impact of the non-smooth functions on the range of step sizes ensuring convergence.

Acknowledgement

We are grateful to Elvis Dohmatob for letting us use his benchmarking tool [19].

References

- [1] H. H. BAUSCHKE, J. M. BORWEIN, AND P. L. COMBETTES, *Bregman monotone optimization algorithms*, SIAM Journal on control and optimization, 42 (2003), pp. 596–636.
- [2] A. BECK AND M. TEOULLE, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM Journal on Imaging Sciences, 2 (2009), pp. 183–202.
- [3] D. P. BERTSEKAS, *Incremental proximal methods for large scale convex optimization*, Mathematical programming, 129 (2011), pp. 163–195.
- [4] D. P. BERTSEKAS AND J. N. TSITSIKLIS, *Parallel and distributed computation: numerical methods*, Prentice-Hall, Inc., 1989.
- [5] P. BIANCHI AND O. FERCOQ, *Using big steps in coordinate descent primal-dual algorithms*, in Proc. of the Conference on Decision and Control, 2016.
- [6] P. BIANCHI, W. HACHEM, AND F. IUTZELER, *A stochastic coordinate descent primal-dual algorithm and applications to large-scale composite optimization*, arXiv preprint arXiv:1407.0898, (2014).
- [7] S. BOYD, N. PARIKH, E. CHU, B. PELEATO, AND J. ECKSTEIN, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Foundations and Trends® in Machine Learning, 3 (2011), pp. 1–122.
- [8] V. CEVHER, S. BECKER, AND M. SCHMIDT, *Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics*, Signal Processing Magazine, IEEE, 31 (2014), pp. 32–43.
- [9] A. CHAMBOLLE, V. CASELLES, D. CREMERS, M. NOVAGA, AND T. POCK, *An introduction to total variation for image analysis*, Theoretical foundations and numerical methods for sparse recovery, 9 (2010), pp. 263–340.
- [10] A. CHAMBOLLE AND C. DOSSAL, *On the convergence of the iterates of “FISTA”*, HAL report, (2014).
- [11] A. CHAMBOLLE, M. J. EHRHARDT, P. RICHTÁRIK, AND C.-B. SCHÖNLIEB, *Stochastic primal-dual hybrid gradient algorithm with arbitrary sampling and imaging application*, arXiv preprint arXiv:1706.04957, (2017).
- [12] A. CHAMBOLLE AND T. POCK, *A first-order primal-dual algorithm for convex problems with applications to imaging*, Journal of Mathematical Imaging and Vision, 40 (2011), pp. 120–145.

- [13] A. CHAMBOLLE AND T. POCK, *On the ergodic convergence rates of a first-order primal–dual algorithm*, Mathematical Programming, (2015), pp. 1–35.
- [14] C.-C. CHANG AND C.-J. LIN, *LIBSVM: A library for support vector machines*, ACM Transactions on Intelligent Systems and Technology (TIST), 2 (2011), p. 27.
- [15] P. L. COMBETTES AND J.-C. PESQUET, *Stochastic quasi-Fejér block-coordinate fixed point iterations with random sweeping*, SIAM Journal on Optimization, 25 (2015), pp. 1221–1248.
- [16] L. CONDAT, *A primal–dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms*, Journal of Optimization Theory and Applications, 158 (2013), pp. 460–479.
- [17] D. DAVIS AND W. YIN, *A three-operator splitting scheme and its optimization applications*, arXiv preprint arXiv:1504.01032, (2015).
- [18] D. DAVIS AND W. YIN, *Faster convergence rates of relaxed peaceman-rachford and admm under regularity assumptions*, Mathematics of Operations Research, (2017).
- [19] E. DOHMATOB, A. GRAMFORT, B. THIRION, AND G. VAROQUAUX, *Benchmarking solvers for tv-l1 least-squares and logistic regression in brain imaging*, in Pattern Recognition in Neuroimaging (PRNI), IEEE, 2014.
- [20] O. FERCOQ AND P. RICHTÁRIK, *Accelerated, parallel and proximal coordinate descent*, SIAM Journal on Optimization, 25 (2015), pp. 1997–2023.
- [21] J. FRIEDMAN, T. HASTIE, H. HÖFLING, AND R. TIBSHIRANI, *Pathwise coordinate optimization*, Ann. Appl. Stat., 1 (2007), pp. 302–332.
- [22] D. GABAY, *Chapter ix applications of the method of multipliers to variational inequalities*, Studies in mathematics and its applications, 15 (1983), pp. 299–331.
- [23] D. GABAY AND B. MERCIER, *A dual algorithm for the solution of nonlinear variational problems via finite element approximation*, Computers & Mathematics with Applications, 2 (1976), pp. 17–40.
- [24] X. GAO, Y. XU, AND S. ZHANG, *Randomized primal-dual proximal block coordinate updates*, arXiv preprint arXiv:1605.05969, (2016).
- [25] 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*, Revue française d’automatique, informatique, recherche opérationnelle. Analyse numérique, 9 (1975), pp. 41–76.
- [26] 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.*, in KDD Cup, 2009, pp. 1–22.
- [27] B. HE AND X. YUAN, *Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective*, SIAM Journal on Imaging Sciences, 5 (2012), pp. 119–149.
- [28] R. A. HORN AND C. R. JOHNSON, *Matrix analysis*, Cambridge university press, 2012.
- [29] F. IUTZELER, P. BIANCHI, P. CIBLAT, AND W. HACHEM, *Asynchronous distributed optimization using a randomized alternating direction method of multipliers*, in Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on, IEEE, 2013, pp. 3671–3676.
- [30] D. D. LEWIS, Y. YANG, T. G. ROSE, AND F. LI, *Rcv1: A new benchmark collection for text categorization research*, Journal of Machine Learning Research, 5 (2004), pp. 361–397.
- [31] Q. LIN, Z. LU, AND L. XIAO, *An accelerated proximal coordinate gradient method*, in Advances in Neural Information Processing Systems, 2014, pp. 3059–3067.

- [32] Z. Q. LUO AND P. TSENG, *A coordinate gradient descent method for nonsmooth separable minimization*, Journal of optimization theory and applications, 72 (2002).
- [33] J. MAIRAL, *Incremental majorization-minimization optimization with application to large-scale machine learning*, SIAM Journal on Optimization, 25 (2015), pp. 829–855.
- [34] I. NECOARA AND A. PATRASCU, *A random coordinate descent algorithm for optimization problems with composite objective function and linear coupled constraints*, tech. report, Politehnica University of Bucharest, 2012.
- [35] Y. NESTEROV, *Efficiency of coordinate descent methods on huge-scale optimization problems*, SIAM Journal on Optimization, 22 (2012), pp. 341–362.
- [36] Y. NESTEROV, *Subgradient methods for huge-scale optimization problems*, Mathematical Programming, 146 (2014), pp. 275–297.
- [37] J.-C. PESQUET AND A. REPETTI, *A class of randomized primal-dual algorithms for distributed optimization*, Journal of Nonlinear Convex Analysis, 16 (2015).
- [38] P. RICHTÁRIK AND M. TAKÁČ, *Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function*, Mathematical Programming, 144 (2014), pp. 1–38.
- [39] P. RICHTÁRIK AND M. TAKÁČ, *Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function*, Mathematical Programming, 144 (2014), pp. 1–38.
- [40] P. RICHTÁRIK AND M. TAKÁČ, *Parallel coordinate descent methods for big data optimization*, Mathematical Programming, (2015), pp. 1–52.
- [41] P. RICHTÁRIK AND M. TAKÁČ, *Efficient serial and parallel coordinate descent method for huge-scale truss topology design*, in Operations Research Proceedings, Springer, 2012, pp. 27–32.
- [42] 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.
- [43] S. SHALEV-SHWARTZ AND T. ZHANG, *Stochastic dual coordinate ascent methods for regularized loss minimization*, Journal of Machine Learning Research, 14 (2013), pp. 567–599.
- [44] T. SUZUKI, *Stochastic dual coordinate ascent with alternating direction method of multipliers*, in Proceedings of the 31st International Conference on Machine Learning (ICML-14), 2014, pp. 736–744.
- [45] Q. TRAN-DINH AND V. CEVHER, *A primal-dual algorithmic framework for constrained convex minimization*, arXiv preprint arXiv:1406.5403, (2014).
- [46] Q. TRAN-DINH, O. FERCOQ, AND V. CEVHER, *A smooth primal-dual optimization framework for nonsmooth composite convex minimization*, arXiv preprint:1507.06243, (2016).
- [47] P. TSENG, *On accelerated proximal gradient methods for convex-concave optimization*, Submitted to SIAM Journal on Optimization, (2008).
- [48] P. TSENG AND C. O. L. MANGASARIAN, *Convergence of a block coordinate descent method for non-differentiable minimization*, J. Optim Theory Appl, (2001), pp. 475–494.
- [49] P. TSENG AND S. YUN, *A coordinate gradient descent method for nonsmooth separable minimization*, Mathematical Programming, 117 (2009), pp. 387–423.
- [50] 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.

- [51] B. C. VÛ, *A splitting algorithm for dual monotone inclusions involving cocoercive operators*, Advances in Computational Mathematics, 38 (2013), pp. 667–681.
- [52] J. WARGA, *Minimizing certain convex functions*, Journal of the Society for Industrial & Applied Mathematics, 11 (1963), pp. 588–593.
- [53] Y. ZHANG AND L. XIAO, *Stochastic primal-dual coordinate method for regularized empirical risk minimization*, arXiv preprint arXiv:1409.3257, (2014).
- [54] C. ZHU, R. H. BYRD, P. LU, AND J. NOCEDAL, *Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization*, ACM Transactions on Mathematical Software (TOMS), 23 (1997), pp. 550–560.