Separable Approximations and Decomposition Methods for the Augmented Lagrangian

Rachael Tappenden Peter Richtárik Burak Büke [*]

August 30, 2013

Abstract

In this paper we study decomposition methods based on separable approximations for minimizing the augmented Lagrangian. In particular, we study and compare the Diagonal Quadratic Approximation Method (DQAM) of Mulvey and Ruszczyński [13] and the Parallel Coordinate Descent Method (PCDM) of Richtárik and Takáč [23]. We show that the two methods are equivalent for feasibility problems up to the selection of a single step-size parameter. Furthermore, we prove an improved complexity bound for PCDM under strong convexity, and show that this bound is at least $8(L'/\bar{L})(\omega-1)^2$ times better than the best known bound for DQAM, where ω is the degree of partial separability and L' and $\bar{L}$ are the maximum and average of the block Lipschitz constants of the gradient of the quadratic penalty appearing in the augmented Lagrangian.

1 Introduction

With the rise and ubiquity of digital and data technology, practitioners in nearly all industries need to solve optimization problems of increasingly larger sizes. As a consequence, new tools and methods are required to solve these big data problems, and to do so efficiently.

In this work, we are concerned with convex optimization problems with an objective function that is separable into blocks of variables and where these blocks are linked by a subset of constraints which nevertheless make the problem nonseparable. Nonseparability is a source of difficulty in solving these very large optimization problems. This structure is particularly relevant in stochastic optimization problems where each block relates to a certain scenario and involves only variables related to that particular scenario. The objective function expressed as an expectation is separable in these blocks and the linking constraints (called non-anticipativity constraints) encode the natural requirement that decisions be based only on information available at the time of decision making. Applications that can be modeled as large scale stochastic optimization problems include multicommodity network flow problems, financial planning problems and airline routing.

A classical approach to solving such problems is to use the augmented Lagrangian by relaxing the linking constraints. The augmented Lagrangian idea was first introduced independently by

[*] All authors: James Clerk Maxwell Building, School of Mathematics, The University of Edinburgh, United Kingdom. The work of all three authors was supported by the EPSRC grant EP/I017127/1 (Mathematics for Vast Digital Resources). The work of PR and RT was also partially supported by the Centre for Numerical Algorithms and Intelligent Software (funded by EPSRC grant EP/G036136/1 and the Scottish Funding Council).

Hestenes [7] and Powell [19] and convergence of the associated augmented Lagrangian method was established later by Rockafellar [25, 26]. Advantages of this approach include the simplicity and stability of the multiplier iterations, the possibility of starting from an arbitrary multiplier, and the fact that there is no master problem to solve. However, the augmented Lagrangian is nonseparable, so the problem is still difficult to solve.

The nonseparability of the augmented Lagrangian has motivated the development of decomposition techniques. In an early work, Stephanopoulos and Westerberg [34] suggest decomposing the augmented Lagrangian using linear approximations and Watanabe et al. [38] use a transformation method to deal with the nonseparable cross products. The progressive hedging algorithm of Rockafellar and Wets [27] also aims to tackle the nonseparability of the augmented Lagrangian. In a more recent line of work, Ruszczyński [28, 29] and Mulvey and Ruszczyński [13, 14] propose and analyze a diagonal quadratic approximation (DQA) to the augmented Langrangian and an associated diaginal quadratic approximation method (DQAM). By approximating the original problem by one that is separable into blocks, these techniques make a significant difference in terms of solvability because the problem is broken down into a number of problems of a more manageable size. Decomposition techniques have become even more attractive with the advances in parallel computing: since the decomposed subproblems can be solved independently, parallelism is possible and this leads to acceleration.

A recent development in the area of decomposition techniques is the Expected Separable Overapproximation (ESO) of Richtárik and Takáč and the associated parallel coordinate descent method (PCDM) presented in [23] (this is discussed in detail in Section 5).

(Block) coordinate descent methods, early variants of which can be traced back to a 1870 paper of Schwarz [30] and beyond, have recently become very popular due to their low per-iteration cost and good scalability properties. While convergence results were established several decades ago, iteration complexity bounds were not studied until recently [37]. Randomized coordinate and block coordinate descent methods were proposed and analyzed in several settings, such as for smooth convex minimization problems [17, 22, 24], L_1-regularized problems [31], composite problems [11, 22, 36], nonsmooth convex problems [6], nonconvex problems [12, 18] and problems with separable constraints [15, 16]. Parallel coordinate descent methods were developed and analyzed in [4, 23, 35, 5, 32], primal-dual methods in [33, 35] and inexact methods in [36]. The methods are used in a number of applications, including linear classification [8, 3, 35], compressed sensing [10], truss topology design [21], solving linear systems of equations [9] and group lasso problems [20].

1.1 Augmented Lagrangian

Our work is motivated by the need to solve huge scale instances of constrained convex optimization problems of the form

$$\min_{x^{(1)},\dots,x^{(n)}} \quad \sum_{i=1}^{n} g_i(x^{(i)}) \tag{1a}$$

$$\text{subject to} \quad \sum_{i=1}^{n} A_i x^{(i)} = b \tag{1b}$$

$$x^{(i)} \in X_i, \quad i = 1, 2, \dots, n, \tag{1c}$$

where for $i = 1, 2, \dots, n$ we assume that $X_i \subseteq \mathbf{R}^{N_i}$ are convex and closed sets, $g_i : \mathbf{R}^{N_i} \to \mathbf{R} \cup \{+\infty\}$ are convex and closed extended real-valued functions and $A_i \in \mathbf{R}^{m \times N_i}$.

While the objective function (1a) and the constraints (1c) are separable in the decision vectors $x^{(1)},\dots,x^{(n)}$, the linear constraint (1b) links them together, which makes the problem difficult to solve. Moreover, we are interested in the case when n is very large (millions, billions and more), which introduces further computational challenges.

It will be useful to think of the decision vectors $\{x^{(i)}\}$ as "blocks" of a single decision vector $x \in \mathbf{R}^N$, with $N = \sum_i N_i$. This can be achieved as follows. We first partition the $N \times N$ identity matrix I columnwise into n submatrices $U_i \in \mathbf{R}^{N\times N_i}$, $i = 1,2,\dots,n$, so that $I = [U_1,\dots,U_n]$, and then set $x = \sum_i U_i x^{(i)}$. That is, x is the vector composed by stacking the vectors $x^{(i)}$ on top of each other. It is easy to see that $x^{(i)} = U_i^T x \in \mathbf{R}^{N_i}$. Moreover, if we let

$$A \overset{\text{def}}{=} \sum_{i=1}^{n} A_i U_i^T \in \mathbf{R}^{m\times N},$$

then (1b) can be written compactly as $Ax = b$. Note also that

$$A_i = AU_i, \qquad i = 1,2,\dots,n. \tag{2}$$

If we now write $g(x) \overset{\text{def}}{=} \sum_i g_i(x^{(i)})$ and $X \overset{\text{def}}{=} \sum_i U_i X_i \subseteq \mathbf{R}^N$, then problem (1a)–(1c) takes the following form:

$$\min_{x\in\mathbf{R}^N} \quad g(x) \tag{3a}$$

$$\text{subject to} \quad Ax = b \tag{3b}$$

$$x \in X. \tag{3c}$$

A typical approach to overcoming the issue of nonseparability of the linking constraint (3b) is to drop it and instead consider the *augmented Lagrangian*,

$$F_\pi(x) \quad \overset{\text{def}}{=} \quad g(x) + \langle \pi, b - Ax\rangle + \tfrac{r}{2}\|b - Ax\|^2,$$

where $\pi \in \mathbf{R}^m$ is a vector of Lagrange multipliers, $r > 0$ is a penalty parameter and $\|u\| = \langle u,u\rangle^{1/2} = (\sum_j u_j^2)^{1/2}$ is the standard Euclidean norm. Now, the Method of Multipliers [2, 7] can be employed to solve problem (1) as described below (Algorithm 1).

Algorithm 1 (Method of Multipliers)

```
Initialization: $\pi_0 \in \mathbf{R}^m$ and iteration counter $k = 0$
while the stopping condition has not been met do
    Step 1: Fix the multiplier $\pi_k$ and solve
        $z_k \leftarrow \min_{x\in X} F_{\pi_k}(x).$    (4a)
    Step 2: Update the multiplier
        $\pi_{k+1} \leftarrow \pi_k + r(b - Az_k),$    (4b)
        and update the iteration counter $k \leftarrow k + 1$.
end while
```

2 The Problem and Our Contributions

The focus of this paper is on the optimization problem (4a). Hence, we need not be concerned about the dependence of F on π and will henceforth refer to the objective function, dropping the constant term $\langle \pi, b\rangle$, as $F(x)$. Ignoring the constant term $\langle \pi, b\rangle$, problem (4a) is a *convex composite* optimization problem, i.e., a problem of the form

$$\min_{x\in\mathbf{R}^N} \{F(x) \overset{\text{def}}{=} f(x) + \Psi(x)\}, \tag{5}$$

where f is a smooth convex function and Ψ is a separable (possibly nonsmooth) convex function. Indeed, we may set

$$f(x) \overset{\text{def}}{=} \frac{r}{2}\|b - Ax\|^2 = \frac{r}{2}\left\| b - \sum_{i=1}^n A_i x^{(i)} \right\|^2, \tag{6}$$

and

$$\Psi(x) \overset{\text{def}}{=} \begin{cases} g(x) - \langle \pi, Ax\rangle, & x \in X, \\ +\infty, & \text{otherwise.} \end{cases}$$

The main purpose of this work is to draw links between two existing decomposition methods for solving (5), one old and one new, both based on separable approximations to the objective function. In particular, we consider DQAM of Mulvey and Ruszczyński [13, 14, 29] and PCDM of Richtárik and Takáč [23], respectively. Our main contributions (not in order of significance) include:

1. **Two measures of separability.** We show that the parameter "number of neighbours", used in the analysis of DQAM [29], and the degree of partial separability, used in the analysis of PCDM [23], coincide up to an additive constant in the case of quadratic f.

2. **Two generalizations of DQAM.** We provide a simplified derivation of the diagonal quadratic approximation, which enables us to propose two generalizations of DQAM (Section 4.2) to non-quadratic functions f, based on

   (i) a finite difference separable approximation to the augmented Lagrangian (Algorithm 3), and

   (ii) a quadratic approximation with the Hessian matrix replaced by an approximation of its block diagonal (Algorithm 4).

   We do not study the complexity of these algorithms in this paper.

3. **Equivalence of PCDM and DQAM for smooth problems.** We identify a situation in which the second of our generalizations of DQAM (Algorithm 4) coincides with a "fully parallel" variant of PCDM (Algorithm 6) for an appropriate selection of parameters of the method (see Section 6.4, Theorem 8). This happens for problems with arbitrary smooth f and $\Psi \equiv 0$.

4. **Improved complexity of PCDM under strong convexity.** We derive an improved complexity result for PCDM in the case when F is strongly convex (Section 7, Theorem 11). The result is much better than that in [23] in situations where the strong convexity constant of F is much larger than the sum of the strong convexity constants of the constituent functions f and Ψ.

5. **Versatility of PCDM.** PCDM enjoys complexity guarantees even in the case when F is merely convex, as opposed to it being strongly convex. Moreover, PCDM is flexible in that it allows for an *arbitrary* number of block updates per iteration, whereas DQAM needs to update *all blocks.*

6. **Complexity in the strongly convex case.** We study the newly developed complexity guarantees for (fully parallel variant of) PCDM (Algorithm 6) and the existing convergence rates for DQAM and show that even though DQAM is specifically designed to approximate the augmented Lagrangian, PCDM has much better theoretical guarantees (Section 7.3). In particular, if F is strongly convex, both DQAM and PCDM converge linearly; that is, $F(x_{k+1}) \leq qF(x_k)$, where q depends on the method. However, we show that q is much better (i.e., smaller) for PCDM than for DQAM, which then leads to vast speedups in terms of iteration complexity. In particular, we show that the theoretical bound for the number of iterations required to find an ϵ-approximate solution is at least

$$\frac{16(\omega-1)^3}{\omega} \times \frac{L'}{\bar{L}} \qquad (\geq 8\tfrac{L'}{\bar{L}}(\omega-1)^2 \text{ for } \omega \geq 2) \tag{7}$$

times larger for DQAM than for (fully parallel) PCDM. Here, ω is the degree of partial separability[1] of f (defined in Section 3), and L' and $\bar{L}$ are the maximum and average of the constants $L_i = r\|A_i^T A_i\|$, $i = 1, 2, \ldots, n$, respectively. Note that the speedup factor (7) is larger than 1000 for $\omega = 10$ even in the case when $L' = \bar{L}$. In practice, however, L' will typically be larger than $\bar{L}$, often much larger.

The form of the speedup factor (7) comes from the fact that DQAM depends on $(\omega-1)^3$ and L' (while PCDM depends on ω and $\bar{L}$), which adversely affects its theoretical complexity rate. Let us comment that Mulvey and Ruszczyński [13] remarked that the dependence of DQAM on ω is in practice much better than cubic, although this was not previously established theoretically. We thus answer their conjecture in the affirmative, albeit for a (as we shall see, not so very) different method. To the best of our knowledge, no improved results were available in the literature up to this point.

7. **Optimal number of block updates per iteration.** We show that under a simple parallel computing model it is optimal for PCDM to update as many block in a single iteration as there are parallel processors (Section 7.4, Theorem 14). As a consequence, the DQAM approach of updating *all* blocks in a single iteration is less than optimal.

8. **Computations.** We also provide preliminary numerical results that show the practical advantages of PCDM.

3 Two Measures of Separability

In this section we provide a link between the measures of separability of f utilized in the analysis of DQAM [13] and PCDM [23]. In the first case, the quantity is defined specifically for a quadratic objective; in the second case the definition is general. As we shall see, both quantities coincide in the quadratic case. As the complexity of the two methods depends on these quantities, our

[1] The multiplicative improvement factor (7) is only valid for $\omega \geq 2$ as DQAM was not analyzed in the case $\omega = 1$.

observation allows us to compare the convergence rates. Both measures of separability are to be understood with respect to the fixed block structure introduced before.

We first define a separability measure introduced for the convex quadratic $f(x) = \frac{r}{2}\|b - Ax\|^2$ by Ruszczyński [29, Section 3] (and called the "number of neighbors" therein).

Let A_{ji} be the j-th row of matrix A_i. Let m_i be the number of nonzero rows in A_i and for each i define an $m \times m_i$ matrix E^i as follows: $E^i_{jl} = 1$ if A_{ji} is the l-th consecutive nonzero row of the matrix A_i, and 0 otherwise. Note that E^i is a matrix containing zeros and m_i ones, one in each column. Further, for any $i \in \{1, 2, \dots, n\}$ and $u \in \{1, 2, \dots, m_i\}$ define

$$V(i,u) \overset{\text{def}}{=} \{(i', u') \ : \ i \in \{1, 2, \dots, n\},\ u' \in \{1, 2, \dots, m_{i'}\},\ k \neq i,\ \langle E^i_u, E^{i'}_{u'} \rangle \neq 0\}, \tag{8}$$

where E^i_u is the u-th column of matrix E^i.

Definition 1 (Ruszczyński separability). *The Ruszczyński degree of separability of the function f defined in* (6) *is*

$$\omega_R = \max\{|V(i,u)| \ : \ i = 1, 2, \dots, n,\ u = 1, 2, \dots, m_i\}. \tag{9}$$

We now define the measure of separability used by Richtárik and Takáč [23] in the analysis of PCDM.

Definition 2 (Partial separability). *A smooth convex function $f : \mathbf{R}^N \to \mathbf{R}$ is partially separable of degree ω if there exists a collection $\mathcal{J}$ of subsets of $\{1, 2, \dots, n\}$ such that*

$$f(x) = \sum_{J \in \mathcal{J}} f_J(x) \qquad \text{and} \qquad \max_{J \in \mathcal{J}} |J| \leq \omega, \tag{10}$$

where for each J, f_J is a smooth convex function that depends on $x^{(i)}$ for $i \in J$ only.

Our first result says that in the case of convex quadratics, the two measures of separability defined above coincide. This will allow us to provide a direct comparison of the complexity results of PCDM and DQAM.

Theorem 3. *For convex quadratic function f given by* (6) *we have $\omega = \omega_R + 1$.*

Proof. First, we can write

$$f(x) = \frac{r}{2} \sum_{j=1}^{m} \left(b_j - \sum_{i=1}^{n} A_{ji} x^{(i)} \right)^2, \tag{11}$$

where b_j is the j-th entry of b. Note that all summands in the decomposition are convex and smooth. Moreover, summand j depends on $x^{(i)}$ if and only if $A_{ji} \neq 0$. If we now let

$$\omega_j = |\{i \ : \ A_{ji} \neq 0\}|, \qquad j = 1, 2, \dots, m, \tag{12}$$

then we conclude that f is partially separable of degree

$$\omega = \max_{j \in \{1, 2, \dots, m\}} \omega_j. \tag{13}$$

In the rest of the proof we proceed in two steps.

(i) Let us fix $i \in \{1,2,\ldots,n\}$, $u \in \{1,2,\ldots,m_i\}$ and let $j = j(i,u)$ be such row index for which $E^i_{ju} = 1$. Note that, since E^i is a 0-1 matrix with exactly one entry of each column equal to 1, we have $E^i_{j'u} = 0$ for all $j' \neq j$. This means that for any $i' \in \{1,2,\ldots,n\}$ and $u' \in \{1,2,\ldots,m_{i'}\}$,

$$\langle E^i_u, E^{i'}_{u'}\rangle \neq 0 \quad \Leftrightarrow \quad E^{i'}_{ju'} = 1. \tag{14}$$

Likewise, $E^{i'}$ is has at most entry equal to 1 in each row. Moreover, the j-th row of $E^{i'}$ contains 1 precisely when $A_{ji'} \neq 0$. This means that

$$|\{u' \;:\; E^{i'}_{ju'} = 1\}| = \begin{cases} 1 & \text{if} \quad A_{ji'} \neq 0, \\ 0 & \text{if} \quad A_{ji'} = 0. \end{cases} \tag{15}$$

We now have

$$\begin{aligned} |V(i,u)| &\overset{(8)}{=} |\{(i',u') \;:\; i' \neq i,\; \langle E^i_u, E^{i'}_{u'}\rangle \neq 0\}| \\ &\overset{(14)}{=} |\{(i',u') \;:\; i' \neq i,\; E^{i'}_{ju'} = 1\}| \\ &= \sum_{i' \neq i} |\{u' \;:\; E^{i'}_{ju'} = 1\}| \\ &\overset{(15)+(12)}{=} \omega_j - 1. \end{aligned} \tag{16}$$

(ii) Building on the result from part (i), we can now write

$$\begin{aligned} \omega_R &\overset{(9)}{=} \max\{|V(i,u)| \;:\; i \in \{1,2,\ldots,n\},\; u \in \{1,2,\ldots,m_i\}\} \\ &\overset{(16)}{=} \max\{\omega_{j(i,u)} - 1 \;:\; i \in \{1,2,\ldots,n\},\; u \in \{1,2,\ldots,m_i\} \\ &= \max_{j \in \{1,2,\ldots,m\}} \omega_j - 1 \\ &\overset{(13)}{=} \omega - 1. \end{aligned}$$

In the third identity above we used the simple observation that every row $j \in \{1,2,\ldots,m\}$ for which $\omega_j \neq 0$ can be written as $j = j(i,u)$ for any i for which $A_{ji} \neq 0$, and some u (which depends on i).

□

Let us remark that besides (11), we could have decomposed f also as

$$f(x) = \frac{r}{2}\left(\|b\|^2 - 2\sum_{i=1}^n \langle b, A_i x^{(i)}\rangle + \sum_{i=1}^n \sum_{j=1}^n \langle A_i x^{(i)}, A_j x^{(j)}\rangle\right), \tag{17}$$

with each summand depending on at most 2 blocks of x. However, we *cannot* conclude that f is partially separable of degree 2 because the terms are *not* all convex, which is required in the definition of partial separability.

4 Diagonal Quadratic Approximation Method

In this section we present the Diagonal Quadratic Approximation Method (DQAM) that was introduced and analysed in a series of papers by Mulvey and Ruszczyński [13, 14], Ruszczyński [29] and Berger, Mulvey and Ruszczyński [1]. As explained in Section 1.1, the augmented Lagrangian is nonseparable because of the cross products $\langle A_i h^{(i)}, A_j h^{(j)}\rangle$ appearing in $f(x+h)$. The DQAM provides a separable approximation of $f(x+h)$ by ignoring these cross terms; this approximation is referred to as the diagonal quadratic approximation (DQA). This makes Step 1 of the method of multipliers ((4a) in Algorithm 1) significantly easier to solve, and amenable to parallel processing.

First, notice that we can write

$$\begin{aligned}
f(x+h) &= \tfrac{r}{2}\|b - A(x+h)\|^2 \\
&= \tfrac{r}{2}\|b\|^2 - r\langle b, A(x+h)\rangle + \tfrac{r}{2}\left(\|Ax\|^2 + 2\langle Ax, Ah\rangle + \|Ah\|^2\right) \\
&= f(x) + \langle f'(x), h\rangle + \tfrac{r}{2}\|Ah\|^2 \\
&= f(x) + \langle f'(x), h\rangle + \tfrac{r}{2}\sum_{i=1}^n \|A_i h^{(i)}\|^2 + \tfrac{r}{2}(\|Ah\|^2 - \sum_{i=1}^n \|A_i h^{(i)}\|^2) \\
&= f(x) + \sum_{i=1}^n \langle (f'(x))^{(i)}, h^{(i)}\rangle + \tfrac{r}{2}\sum_{i=1}^n \|A_i h^{(i)}\|^2 + \tfrac{r}{2}\sum_{i\neq j}\langle A_i h^{(i)}, A_j h^{(j)}\rangle. \qquad (18)
\end{aligned}$$

Now observe that it is only the last term in (18), composed of products $\langle A_i h^{(i)}, A_j h^{(j)}\rangle$ for $i \neq j$, which is not separable. Ignoring these terms, we get a separable approximation of $f(x+h)$ in h,

$$f(x+h) \approx f^{\text{DQA}}(x+h) \overset{\text{def}}{=} f(x) + \langle f'(x), h\rangle + \frac{r}{2}\sum_{i=1}^n \|A_i h^{(i)}\|^2, \qquad (19)$$

which in turn leads to a separable approximation of $F(x+h)$ in h:

$$F(x+h) \overset{(5)}{=} f(x+h) + \Psi(x+h) \overset{(19)}{\approx} f^{\text{DQA}}(x+h) + \Psi(x+h). \qquad (20)$$

Mulvey and Ruszczyński [13] propose a slightly less transparent construction of the same approximation. For a fixed x, they approximate $f(y)$ via replacing the cross-products $\langle A_i y^{(i)}, A_j y^{(j)}\rangle$, for $i \neq j$, by

$$\langle A_i y^{(i)}, A_j x^{(j)}\rangle + \langle A_i x^{(i)}, A_j y^{(j)}\rangle - \langle A_i x^{(i)}, A_j x^{(j)}\rangle. \qquad (21)$$

Clearly, this is equivalent to what we do above, which can be verified by substituting $y = x + h$ into (21).

4.1 The algorithm

We now present the DQA method (Algorithm 2). The algorithm replaces Step 1 of the Method of Multipliers (Algorithm 1). In what follows, $\theta \in (0,1)$ is a user defined parameter.

Algorithm 2 (DQAM: Diagonal Quadratic Approximation Method)

```
for $k = 0, 1, 2, \dots$ do
    Step 1a: Solve for $h_k$
        $h_k \leftarrow \arg \min_{h \in \mathbf{R}^N} \left\{ f^{\mathrm{DQA}}(x_k + h) + \Psi(x_k + h) \right\}$    (22a)
    Step 1b: Determine intermediate vector $y_k$
        $y_k \leftarrow x_k + h_k$    (22b)
    Step 1c: Form the new iterate $x_{k+1}$
        $x_{k+1} \leftarrow (1 - \theta)x_k + \theta y_k$    (22c)
end for
```

Let us now comment on the individual steps of Algorithm 2. Step 1a is easy to execute because the function that is being minimized in (22a) is separable in h, and hence the problem decomposes into n independent *lower-dimensional* problems:

$$h_k^{(i)} = \arg \min_{h^{(i)} \in \mathbf{R}^{N_i}} \left\{ \langle (f'(x_k))^{(i)}, h^{(i)} \rangle + \frac{r}{2} \|A_i h^{(i)}\|^2 + \Psi_i(x_k^{(i)} + h^{(i)}) \right\}, \quad i = 1, 2, \dots, n.$$

Moreover, the problems are *independent*, and hence the updates $h_k^{(1)}, \cdots h_k^{(n)}$ can be computed *in parallel.* In (22b) an intermediate vector y_k is formed, and then in (22c) a convex combination of the current iterate x_k and the intermediate vector y_k is taken to produce the new iterate x_{k+1}. Step (22c) is needed because DQAM uses a local approximation, so if the new point $x_k + h_k$ is far from x_k, the approximation error may be too big and a reduction in the objective function value is not guaranteed. This would lead to serious stability and convergence problems in general, and hence, Step 1c is employed as a correction step for regularizing the method.

4.2 Two generalizations

DQAM was originally designed and analyzed for convex quadratics. Here we propose two generalizations of the method to non-quadratic convex functions f. Our generalizations are based on the following simple result.

Proposition 4. *If $f(x) = \frac{r}{2}\|b - Ax\|^2$, then for all $x, h \in \mathbf{R}^N$,*

$$f^{DQA}(x + h) = f(x) + \sum_{i=1}^{n} \left[f(x + U_i h^{(i)}) - f(x) \right] \tag{23}$$

and

$$f^{DQA}(x + h) \quad = \quad f(x) + \sum_{i=1}^{n} \left[\langle (f'(x))^{(i)}, h^{(i)} \rangle + \tfrac{1}{2} \langle C_i(x) h^{(i)}, h^{(i)} \rangle \right], \tag{24}$$

where $C_i(x) = U_i^T f''(x) U_i$.

Proof. First note that

$$\begin{aligned}\sum_{i=1}^n \left[f(x+U_ih^{(i)})-f(x)\right] &= \sum_{i=1}^n \left[\tfrac{r}{2}\|b-Ax-A_ih^{(i)}\|^2-\tfrac{r}{2}\|b-Ax\|^2\right]\\ &= \sum_{i=1}^n \left[r\langle Ax-b,A_ih^{(i)}\rangle+\tfrac{r}{2}\|A_ih^{(i)}\|^2\right]\\ &= \langle f'(x),h\rangle+\tfrac{r}{2}\sum_{i=1}^n\|A_ih^{(i)}\|^2,\end{aligned}$$

which, in view of (19), establishes (23). Finally, (24) follows from (19) and the fact that

$$\tfrac{r}{2}\|A_ih^{(i)}\|^2=\tfrac{1}{2}\langle U_i^Tf''(x)U_ih^{(i)},h^{(i)}\rangle,$$

which in turn follows from the identities $f''(x)=rA^TA$ and $A_i=AU_i$. □

Our two generalized methods are obtained by replacing $f^{\mathrm{DQA}}(x+h)$ in Step 1 of Algorithm 2 by one of the two approximations (23) and (24) (in the second case we allow for $C_i(x)$ to be an arbitrary positive semidefinite matrix and not necessarily $U_i^Tf''(x)U_i$), leading to Algorithm 3 and Algorithm 4, respectively.

Algorithm 3 (Generalization of DQAM: Finite Differences Approximation)

```
for $k = 0, 1, 2, \dots$ do
    Step 1a: Solve for $h_k$
        $$h_k \leftarrow \arg\min_{h\in\mathbf{R}^N}\left\{f(x_k)+\sum_{i=1}^n\left[f(x_k+U_ih^{(i)})-f(x_k)\right]+\Psi(x_k+h)\right\} \qquad (25a)$$
    Step 1b: Determine intermediate vector $y_k$
        $$y_k \leftarrow x_k+h_k \qquad (25b)$$
    Step 1c: Form the new iterate $x_{k+1}$
        $$x_{k+1} \leftarrow (1-\theta)x_k+\theta y_k \qquad (25c)$$
end for
```

Algorithm 3 is based on a finite difference approximation, and is applicable to (possibly) nonsmooth functions. Algorithm 4 is based on a separable quadratic approximation. To the best of our knowledge, these algorithms have not been previously proposed, with the exception of the case when f is a convex quadratic when both methods coincide with DQAM.

Algorithm 4 (Generalization of DQAM: Separable Quadratic Approximation)

```
for $k = 0, 1, 2, \dots$ do
    Step 1a: Solve for $h_k$
        $$h_k \leftarrow \arg\min_{h \in \mathbf{R}^N} \left\{ f(x_k) + \langle f'(x_k), h \rangle + \tfrac{1}{2} \sum_{i=1}^{n} \langle C_i(x_k) h^{(i)}, h^{(i)} \rangle + \Psi(x_k + h) \right\} \qquad (26a)$$
    Step 1b: Determine intermediate vector $y_k$
        $$y_k \leftarrow x_k + h_k \qquad (26b)$$
    Step 1c: Form the new iterate $x_{k+1}$
        $$x_{k+1} \leftarrow (1 - \theta) x_k + \theta y_k \qquad (26c)$$
end for
```

In this paper we do not analyze any of these methods. Instead, we propose that DQAM be replaced by PCDM, described in the next section.

5 Parallel Coordinate Descent Method

As discussed in the introduction, we propose that instead of implementing Step 1 of the Method of Multipliers (Algorithm 1) using DQAM, a parallel coordinate descent method (PCDM) be used instead. This section is devoted to describing the method, developed by Richtárik and Takáč [23].

5.1 Block samplings

As we shall see, unlike DQAM where all blocks are updated at each iteration, PCDM allows for an (almost) arbitrary random subset of blocks to be updated at each iteration. The purpose of this section is to formalize this.

In particular, at iteration k only blocks $i \in S_k \subseteq \{1, 2, \dots, n\}$ are updated, where $\{S_k\}$, $k \geq 0$, are iid random sets having the following two properties:

$$\mathbf{P}(i \in S_k) = \mathbf{P}(j \in S_k) \qquad \text{for all} \qquad i, j \in \{1, 2, \dots, n\}, \tag{27}$$

$$\mathbf{P}(i \in S_k) > 0 \qquad \text{for all} \qquad i \in \{1, 2, \dots, n\}. \tag{28}$$

It is easy to see that, necessarily, $\mathbf{P}(i \in S_k) = \frac{\mathbf{E}[|S_k|]}{n}$. Following [23], for simplicity we refer to an arbitrary random set-valued mapping with values in the power set $2^{\{1,2,\dots,n\}}$ by the name *block sampling*, or simply *sampling*. A sampling S_k is called *uniform* if it satisfies (27) and *proper* if it satisfies (28).

In [23], PCDM was analyzed for all proper uniform samplings. However, better complexity results were obtained for so called *doubly uniform* samplings, which belong to the family of uniform samplings. For brevity purposes, in this paper we concentrate on a subclass of doubly uniform samplings called τ-*nice* samplings, which we now define.

Definition 5 (τ-nice sampling)**.** *Let τ be an integer between 1 and n. A sampling $\hat{S}$ is called τ-nice if for all $S \subseteq \{1, 2, \dots, n\}$,*

$$\mathbf{P}(\hat{S} = S) = \begin{cases} 0, & |S| \neq \tau, \\ \frac{1}{\binom{n}{\tau}}, & \textit{otherwise.} \end{cases}$$

A natural candidate for τ is the number of available processors/threads as then updates to the τ blocks of x_k can be computed in parallel. As we shall later see, this is also the optimal choice from the complexity point of view (Theorem 14).

5.2 Expected Separable Overapproximation (ESO)

Fixing positive scalars $w_1, \dots, w_n$ (we write $w = (w_1, \dots, w_n)$), let us define a separable norm on $\mathbf{R}^N$ by

$$\|x\|_w \overset{\text{def}}{=} \left(\sum_{i=1}^{n} w_i \|x^{(i)}\|_{(i)}^2 \right)^{1/2}, \quad x \in \mathbf{R}^N, \tag{29}$$

where for each $i = 1, 2, \dots, n$ we fix a positive definite matrix $B_i \in \mathbf{R}^{N_i \times N_i}$ and set

$$\|t\|_{(i)} \overset{\text{def}}{=} \langle B_i t, t \rangle^{1/2}, \quad t \in \mathbf{R}^{N_i}. \tag{30}$$

We can now define the concept of expected separable overapproximation.

Definition 6 (Expected Separable Overapproximation (ESO) [23])**.** *Let β and $w_1, \dots, w_n$ be positive constants and $\hat{S}$ be a proper uniform sampling. We say that $f : \mathbf{R}^N \to \mathbf{R}$ admits a (β, w)-ESO with respect to $\hat{S}$ (and, for simplicity, we write $(f, \hat{S}) \sim ESO(\beta, w)$) if for all $x, h \in \mathbf{R}^N$,*

$$\mathbf{E}\left[f\left(x + \textstyle\sum_{i \in \hat{S}} U_i h^{(i)}\right)\right] \leq f(x) + \frac{\mathbf{E}[|\hat{S}|]}{n} \left(\langle f'(x), h \rangle + \frac{\beta}{2} \|h\|_w^2 \right). \tag{31}$$

In Section 5.3 we describe how the ESO is used to design a parallel coordinate descent method for solving problem (5). The issue of how the parameters w and β giving rise to an ESO can be determined/computed will be discussed in Section 5.4.

5.3 The algorithm

Unlike with DQAM, were f is replaced by f^{DQA} and Ψ is kept intact, PCDM replaces *both* f and Ψ. This is because in PCDM we compute an approximation to

$$\mathbf{E}\left[F(x + \textstyle\sum_{i \in \hat{S}} U_i h^{(i)})\right] = \mathbf{E}\left[f(x + \textstyle\sum_{i \in \hat{S}} U_i h^{(i)}) + \Psi(x + \textstyle\sum_{i \in \hat{S}} U_i h^{(i)})\right], \tag{32}$$

which (unless $|\hat{S}| = n$) affects Ψ as well. It can be verified (see [23, Section 3]) that due to separability of Ψ the following identity holds:

$$\mathbf{E}\left[\Psi(x + \textstyle\sum_{i \in \hat{S}} U_i h^{(i)})\right] = \left(1 - \frac{\mathbf{E}[|\hat{S}|]}{n}\right) \Psi(x) + \frac{\mathbf{E}[|\hat{S}|]}{n} \Psi(x + h). \tag{33}$$

Substituting (33) and (31) into (32), we obtain

$$\mathbf{E}\left[F(x+\textstyle\sum_{i\in\hat{S}}U_ih^{(i)})\right] \leq F^{\mathrm{ESO}}(x+h) \overset{\mathrm{def}}{=} \left(1-\frac{\mathbf{E}[|\hat{S}|]}{n}\right)F(x)+\frac{\mathbf{E}[|\hat{S}|]}{n}H_{\beta,w}(x+h), \tag{34}$$

where

$$H_{\beta,w}(x+h) \overset{\mathrm{def}}{=} f(x)+\langle f'(x),h\rangle+\frac{\beta}{2}\|h\|_w^2+\Psi(x+h), \tag{35}$$

which is separable in h:

$$H_{\beta,w}(x+h) \overset{(29)+(30)}{=} f(x)+\sum_{i=1}^{n}\left\{\langle (f'(x))^{(i)},h^{(i)}\rangle+\frac{\beta w_i}{2}\langle B_ih^{(i)},h^{(i)}\rangle+\Psi_i(x^{(i)}+h^{(i)})\right\}. \tag{36}$$

We are now ready to present the parallel coordinate descent method (Algorithm 5).

Algorithm 5 (PCDM: Parallel Coordinate Descent Method)

```
Initialization: $x_0 \in \mathbf{R}^N$, ESO parameters $(\beta, w)$
for $k = 0, 1, 2, \ldots$ do
    Step 1a: Solve
        $h_k \leftarrow \arg\min_{h\in\mathbf{R}^N} F^{\mathrm{ESO}}(x_k+h)$    (37a)
    Step 1b: Update $x_k$
        $x_{k+1} \leftarrow x_k + \sum_{i\in S_k} U_i h_k^{(i)}$    (37b)
end for
```

Given an iterate x_k, in (37a) we compute

$$h_k = h(x_k) \overset{\mathrm{def}}{=} \arg\min_{h\in\mathbf{R}^N} F^{\mathrm{ESO}}(x_k+h) \overset{(34)}{=} \arg\min_{h\in\mathbf{R}^N} H_{\beta,w}(x_k+h). \tag{38}$$

Further, note that (37b) is equivalent to writing

$$x_{k+1}^{(i)} = \begin{cases} x_k^{(i)}, & i \notin S_k, \\ x_k^{(i)}+h_k^{(i)}, & i \in S_k. \end{cases}$$

That is, only blocks belonging to the random set S_k are updated. This means that in (37a) we need not compute all blocks of h_k. In view of (36) and (38), this is possible, and hence (37a) can be replaced by

$$h_k^{(i)} \leftarrow \arg\min_{h^{(i)}\in\mathbf{R}^{N_i}}\left\{\langle (f'(x_k))^{(i)},h^{(i)}\rangle+\frac{\beta w_i}{2}\langle B_ih^{(i)},h^{(i)}\rangle+\Psi_i(x_k^{(i)}+h^{(i)})\right\}, \quad i\in S_k. \tag{39}$$

5.4 ESO for partially separable smooth convex functions

In order for PCDM to be implementable, one needs first to compute the parameters $w_1, \dots, w_n$ (defining the norm $\|\cdot\|_w$) and $\beta > 0$ for which $(f, \hat{S}) \sim ESO(\beta, w)$, i.e., for which (31) holds. Clearly, the parameters β and w depend on f and $\hat{S}$.

In what follows we will assume that the gradient of f is block Lipschitz. That is, there exist positive constants $L_1, \dots, L_n$ such that for all $x \in \mathbf{R}^N$, $i \in \{1, 2, \dots, n\}$ and $h^{(i)} \in \mathbf{R}^{N_i}$,

$$\|(f'(x + U_i t))^{(i)} - (f'(x))^{(i)}\|^*_{(i)} \leq L_i \|t\|_{(i)}, \tag{40}$$

where $\|s\|^*_{(i)} \overset{\text{def}}{=} \max\{\langle s, x\rangle \ : \ \|x\|_w = 1\} = \langle B_i^{-1} s, s\rangle^{1/2}$ is the conjugate norm to $\|\cdot\|_w$.

Theorem 7 (Theorem 14 in [23])**.** *Assume f is convex, partially separable of degree ω, and has block Lipschitz gradient with constants $L_1, L_2, \dots, L_n > 0$. Further, assume that $\hat{S}$ is a τ-nice sampling, where $\tau \in \{1, 2, \dots, n\}$. Then*

$$(f, \hat{S}) \sim ESO(\beta, w),$$

where

$$\beta = 1 + \frac{(\omega - 1)(\tau - 1)}{\max\{1, n - 1\}}, \qquad w_i = L_i, \qquad i = 1, 2, \dots, n. \tag{41}$$

In Section 7 we study the complexity of PCDM in the case covered by the above theorem (and under a further strong convexity assumption).

Consider now the special case of convex quadratic f given by (6). If the matrices $A_i^T A_i$, $i = 1, 2, \dots, n$, are all positive definite, we can choose $B_i = r A_i^T A_i$, $i = 1, 2, \dots, n$, in which case we will have $L_i = 1$ for all i. Otherwise we can choose B_i to be the $N_i \times N_i$ identity matrix, and then

$$L_i = r\|A_i^T A_i\| \overset{\text{def}}{=} r \max_{\|h^{(i)}\| \leq 1} \|A_i^T A_i h^{(i)}\|, \tag{42}$$

where both norms in the definition are the standard Euclidean norms in $\mathbf{R}^{N_i}$.

5.5 Fully parallel coordinate descent method

PCDM used with an n-nice sampling $\hat{S}$ resembles DQAM in two ways: i) it updates *all* blocks during each iteration, ii) it is not randomized. Indeed,

$$\sum_{i \in \hat{S}} U_i h^{(i)} = \sum_{i=1}^{n} U_i h^{(i)} = h, \tag{43}$$

and hence

$$\begin{aligned} F(x + h) &\overset{(43)}{=} \mathbf{E}\left[F(x + \textstyle\sum_{i \in \hat{S}} U_i h^{(i)})\right] \overset{(34)}{\leq} F^{\text{ESO}}(x + h) \\ &\overset{(34)+(35)}{=} f(x) + \langle f'(x), h\rangle + \tfrac{\beta}{2}\|h\|_w^2 + \Psi(x + h). \end{aligned}$$

In particular, in the setting of Theorem 7 we have $\beta = \omega$ and $w = L = (L_1, \dots, L_n)$, and Algorithm 5 specializes to Algorithm 6.

Algorithm 6 (Fully Parallel Coordinate Descent Method)

```
Initialization: $x_0 \in \mathbf{R}^N$
for $k = 0, 1, 2, \dots$ do
    Step 1a: Solve
        $h_k \leftarrow \arg\min_{h \in \mathbf{R}^N} \left\{ f(x_k) + \langle f'(x_k), h \rangle + \frac{\omega}{2} \sum_{i=1}^{n} \langle L_i B_i h^{(i)}, h^{(i)} \rangle + \Psi(x_k + h) \right\}$    (44a)
    Step 1b: Update
        $x_{k+1} \leftarrow x_k + h_k$    (44b)
end for
```

6 Links Between DQAM and PCDM

In this section we discuss and compare DQAM and PCDM. We highlight some of the main differences between the two methods, and describe a special case where the methods coincide.

6.1 Fully parallel vs partially parallel updating

One of the main differences between DQAM and PCDM is the number of blocks that must be updated at each iteration. At each iteration of DQAM, *all n* blocks must be updated. This highlights the fact that DQAM uses a *fully parallel* update scheme. On the other hand, PCDM is more flexible as it is able to update τ blocks at each iteration where $1 \leq \tau \leq n$. This is beneficial because in practice there are usually fewer processors than the number of blocks. So, PCDM can act as a *serial* method if $\tau = 1$, a *fully parallel* method if $\tau = n$, or it can be optimized to the number of processors p (so $\tau = p$). The advantages of updating $\tau = p$ blocks at each iteration of PCDM is established theoretically in Section 7.

Because DQAM updates all n blocks at each iteration, it is a Jacobi type method, whereas PCDM can be interpreted as a Jacobi type method when $\tau = n$, a Gauss-Seidel type method when $\tau = 1$, or a hybrid Jacobi-Gauss-Seidel method when $1 < \tau < n$.

6.2 Flexibility of PCDM

PCDM can be applied to a general convex composite function. Specifically, f is only assumed to be smooth and convex. Further, the algorithm is guaranteed to converge when applied to a general smooth convex function, and can be equipped with iteration complexity bounds (see [23]). On the other hand, the convergence results for DQAM have been only derived under the assumption that f is quadratic and strongly convex; there are no convergence guarantees for a function f with any other structure. Complexity estimates for both methods are discussed in detail in Section 7.

Notice that DQAM has been tailored specifically for an augmented Lagrangian objective function so it is reasonable that the function f is assumed to be quadratic and strongly convex in this context. However, this assumption restricts the range of problems that can be solved using DQA, while PCDM can be applied to a much wider class of problems.

6.3 Approximation type and algorithm philosophy

In DQAM, a local two-sided approximation to the cross products is employed. The error associated with the approximation is of the order $o(\|h\|_2^2)$, which explains that, if the update h_k is too large, then the model loses accuracy. This justifies the need for a correction step (22c) so as to ensure that x_{k+1} is not too far from x_k. This ensures a reduction in the objective value and ultimately, algorithm convergence. The need for a correction scheme within DQAM is also apparent from the finite differences formulation presented in Algorithm 3. Consider the summation in (25a), and for simplicity assume that $\Psi \equiv 0$. Then the block update $h_k^{(i)}$ is that which minimizes the function value difference in the i-th block coordinate direction, independently of all the other blocks $j \neq i$. Clearly, this will *not* guarantee that $F(x_k + h_k) \leq F(x_k)$ because the function F is not block separable. A simple 2D quadratic example showing that this approach is doomed to fail was described in [35].

In contrast to the DQAM scheme, PCDM employs a one-sided *global* expected separable overapproximation of the augmented Lagrangian function (5), which guarantees to produce a new random iterate x_{k+1} that, on average, decreases the objective function. That is, x_{k+1} satisfies $\mathbf{E}[F(x_{k+1}) \mid x_k] \leq F(x_k)$. It turns out that this is sufficient to obtain a high probability complexity result and therefore there is no need for a correction step in PCDM. In fact, as we shall see in Section 6.4, a "correction step" is already embedded in the approximation in the form of the ESO parameter β.

Note that, besides DQAM, there are many other algorithms that follow a "step-then-correct" strategy. One example are trust region methods, where a solution to some subproblem is found, the "goodness" of the solution is measured, and then the size of the trust region is adjusted to reflect the "goodness". A second example is the conditional gradient algorithm, which builds a linear approximation to the objective function, finds the minimizer of the linearized problem (the "step") and then "corrects" by taking a convex combination of the previous point and the step to reduce the objective value. This correction step is implicitly built-in for PCDM, in the choice of the constant β.

6.4 A special case in which the methods coincide

So far we have highlighted some of the differences between DQAM and PCDM. However, in this section we present a special case where the two methods coincide.

Theorem 8. *Assume f is partially separable of degree ω, and has block Lipschitz gradient with constants $L_1, L_2, \ldots, L_n > 0$. Further, assume $\Psi \equiv 0$. Then Algorithm 4 (generalization of DQAM) coincides with Algorithm 6 (fully parallel PCDM) under the following choice of parameters:*

$$C_i(x_k) \equiv L_i B_i \quad (i = 1, 2, \ldots, n), \qquad \theta = \tfrac{1}{\omega}. \tag{45}$$

Proof. In Algorithm 4 we have $x_{k+1} = (1-\theta)x_k + \theta(x_k + h_k)$, where

$$h_k = \arg\min_{h \in \mathbf{R}^N} \{\langle f'(x_k), h\rangle + \tfrac{1}{2}\sum_{i=1}^{n} \langle C_i(x_k) h^{(i)}, h^{(i)}\rangle\}. \tag{46}$$

Due to separability of the objective function in (46) and the choice of parameters (45), we see that $h_k^{(i)} = -\frac{1}{L_i} B_i^{-1}(f'(x_k))^{(i)}$, $i = 1, 2, \ldots, n$, and hence

$$x_{k+1}^{(i)} = (1-\theta)x_k^{(i)} + \theta(x_k^{(i)} + h_k^{(i)}) = x_k^{(i)} - \tfrac{1}{\omega L_i} B_i^{-1}(f'(x_k))^{(i)}. \tag{47}$$

In Algorithm 6 we have $x_{k+1} = x_k + h_k$, where

$$h_k = \arg\min_{h\in\mathbf{R}^N} \left\{ \langle f'(x_k), h\rangle + \frac{\omega}{2}\sum_{i=1}^{n} \langle L_i B_i h^{(i)}, h^{(i)}\rangle \right\}. \tag{48}$$

Using separability of the objective function in (48), we again obtain the same formula (47) for x_{k+1}, establishing the equivalence of the two methods. □

A few remarks:

- In the context of the original problem (1), the case covered by the above theorem corresponds to a feasibility problem ($\Psi \equiv 0$ means that $g \equiv 0$).
- DQAM was analyzed in [29] only for the parameter θ in the interval $(0, \frac{1}{2(\omega-1)})$. For $\omega > 1$ this leads to *smaller steps* than the PCDM default choice $\theta = \frac{1}{\omega}$, which then translates to slower convergence for DQAM.

7 Complexity of DQAM and PCDM under Strong Convexity

In this section we study and compare the convergence rates of DQAM and PCDM under the assumption of strong convexity of the objective function. We limit ourselves to this case as complexity estimates for DQAM are not available otherwise. Both DQAM and PCDM benefit from linear convergence, but the rate is much better for PCDM than for DQA.

Strong convexity. We assume that F is strongly convex with respect to the norm $\|\cdot\|_w$ for some vector of positive weights $w = (w_1, \dots, w_n)$ specified in the results, with (strong) convexity parameter $\mu_F > 0$. A function $\phi : \mathbf{R}^N \to \mathbf{R} \cup \{+\infty\}$ is strongly convex with respect to the norm $\|\cdot\|_w$ with convexity parameter $\mu_\phi = \mu_\phi(w) \geq 0$ if for all $x, y \in \operatorname{dom}\phi$,

$$\phi(y) \geq \phi(x) + \langle \phi'(x), y - x\rangle + \frac{\mu_\phi}{2}\|y - x\|_w^2, \tag{49}$$

where $\phi'(x)$ is any subgradient of ϕ at x. The case with $\mu_\phi(w) = 0$ reduces to convexity. It will be useful to note that for any $t > 0$,

$$\mu_\phi(tw) = \frac{\mu_\phi(w)}{t}. \tag{50}$$

Strong convexity of F may come from f or Ψ or both and we will write μ_f (resp. μ_Ψ) for the strong convexity parameter of f (resp. Ψ). It is easy to see that

$$\mu_F \geq \mu_f + \mu_\Psi. \tag{51}$$

Note that the strong convexity constant of F can be *arbitrarily larger* than the sum of the strong convexity constants of the functions f and Ψ. Indeed, consider the following simple 2D example ($N = n = 2$): $f(x) = \frac{\mu}{2}(x^{(1)})^2$, $\Psi(x) = \frac{\mu}{2}(x^{(2)})^2$, where $\mu > 0$. Let $\|x\|_w$ be the standard Euclidean norm (i.e., $B_i = 1$ and $w_i = 1$ for $i = 1, 2$). Clearly, neither f nor Ψ is strongly convex ($\mu_f = \mu_\Psi = 0$). However, F is strongly convex with constant $\mu_F = \mu$.

In the rest of the section we will repeatedly use the following simple result.

Lemma 9. *Let $\xi_0 > \epsilon > 0$ and $\gamma \in (0,1)$. If $k \geq \frac{1}{\gamma}\log\left(\frac{\xi_0}{\epsilon}\right)$, then $(1-\gamma)^k\xi_0 \leq \epsilon$.*

Proof. $(1-\gamma)^k\xi_0 = (1 - \frac{1}{1/\gamma})^{(1/\gamma)(\gamma k)}\xi_0 \leq e^{-\gamma k}\xi_0 \leq e^{-\log(\xi_0/\epsilon)}\xi_0 = \epsilon.$ □

7.1 PCDM

We now derive a new improved complexity result for PCDM. In [23, Theorem 20] the authors prove an iteration complexity bound based on the assumption that $\mu_f + \mu_\Psi > 0$. Here we obtain a new and tighter complexity result under the weaker assumption $\mu_F > 0$. As discussed above, μ_F can be substantially bigger than $\mu_f + \mu_\Psi$, which implies that our complexity bound can be much better.

The following auxiliary result is an improvement on Lemma 17(ii) in [23] and will be used in the proof of our main complexity result.

Lemma 10. *If $\mu_F(w) > 0$ and $\beta \geq \mu_f(w)$, then for all $x \in \operatorname{dom} F$*

$$H_{\beta,w}(x + h(x)) - F^* \leq \frac{\beta - \mu_f(w)}{\mu_F(w) + \beta - \mu_f(w)}(F(x) - F^*). \tag{52}$$

Proof. Let $\mu_F = \mu_F(w)$, $\mu_f = \mu_f(w)$ and $\mu_\Omega = \mu_\Omega(w)$. By Lemma 16 in [23], we have

$$H_{\beta,w}(x + h(x)) \leq \min_{y \in \mathbf{R}^N} \left\{ F(y) + \frac{\beta - \mu_f}{2}\|y - x\|_w^2 \right\}. \tag{53}$$

Using this, we can further write

$$\begin{aligned} H_{\beta,w}(x + h(x)) &\overset{(53)}{\leq} \min_{y = \lambda x^* + (1-\lambda)x,\ \lambda \in [0,1]} \left\{ F(y) + \frac{\beta - \mu_f}{2}\|y - x\|_w^2 \right\} \\ &= \min_{\lambda \in [0,1]} \left\{ F(\lambda x^* + (1-\lambda)x) + \frac{(\beta - \mu_f)\lambda^2}{2}\|x - x^*\|_w^2 \right\} \\ &\leq \min_{\lambda \in [0,1]} \left\{ \lambda F^* + (1-\lambda)F(x) - \frac{\mu_F \lambda(1-\lambda) - (\beta - \mu_f)\lambda^2}{2}\|x - x^*\|_w^2 \right\}, \end{aligned} \tag{54}$$

where in the last step we have used strong convexity of F. Notice that $\lambda^* \overset{\text{def}}{=} \mu_F/(\mu_F + \beta - \mu_f) \in (0, 1]$ and that $\mu_F(1 - \lambda^*) - (\beta - \mu_f)\lambda^* = 0$. It now only remains to substitute λ^* into (54) and subtract F^* from the resulting inequality. □

We now present our main complexity result. It gives a bound on the number of iterations required by PCDM (Algorithm 5) to obtain an ϵ solution with high probability. The result is generic in the sense that it applies to any smooth convex function and proper uniform sampling as long as the parameters β and w giving rise to an ESO are known.

Theorem 11. *Assume that $F = f + \Psi$ is strongly convex with respect to the norm $\|\cdot\|_w$ ($\mu_F(w) > 0$) and let $S_0, S_1, \dots$ be iid proper uniform samplings satisfying*

$$(f, S_0) \sim ESO(\beta, w).$$

Choose an initial point $x_0 \in \mathbf{R}^N$, target confidence level $\rho \in (0, 1)$, target accuracy level $0 < \epsilon < F(x_0) - F^*$ and iteration counter

$$K \geq \frac{n}{\mathbf{E}[|S_0|]} \frac{\beta + \mu_F(w) - \mu_f(w)}{\mu_F(w)} \log\left(\frac{F(x_0) - F^*}{\epsilon\rho} \right). \tag{55}$$

If $\{x_k\}$, $k \geq 0$, are the random points generated by PCDM (Algorithm 5) as applied to problem (5)*, then*

$$\mathbf{P}(F(x_K) - F^* \leq \epsilon) \geq 1 - \rho.$$

Proof. Let $\alpha = \frac{\mathbf{E}[|S_0|]}{n}$ and $\xi_k = F(x_k) - F^*$. Then for all $k \geq 0$,

$$\mathbf{E}[\xi_{k+1} \mid x_k] \overset{(34)}{\leq} (1-\alpha)\xi_k + \alpha(H_{\beta,w}(x_k + h(x_k)) - F^*) \overset{\text{(Lemma 10)}}{\leq} \Big(1 - \underbrace{\tfrac{\alpha\mu_F(w)}{\mu_F(w)+\beta-\mu_f(w)}}_{\overset{\text{def}}{=}\gamma}\Big)\xi_k. \tag{56}$$

Note that Lemma 10 is applicable as the assumption $\beta \geq \mu_f(w)$ is satisfied due to the fact that $(f, \hat{S}) \sim ESO(\beta, w)$ (see [23, Section 4]). Further, note that $\gamma > 0$ since $\alpha > 0$ and $\mu_F(w) > 0$. Moreover, $\gamma \leq 1$ since $\alpha \leq 1$ and $\beta \geq \mu_f(w)$. By taking expectation in x_k through (56), we obtain $\mathbf{E}[\xi_k] \leq (1-\gamma)^k\xi_0$. Applying Markov inequality, Lemma 9 and (55), we obtain

$$\mathbf{P}(\xi_K > \epsilon) \leq \frac{\mathbf{E}[\xi_K]}{\epsilon} \leq \frac{(1-\gamma)^K\xi_0}{\epsilon} \leq \rho,$$

establishing the result. □

In order to compare the complexity of PCDM with that of DQAM, which is a fully parallel method, we now derive a specialized complexity result for the fully parallel variant of PCDM (Algorithm 6). The method is no longer stochastic in this situation, i.e., the sequence of vectors $\{x_k\}$, $k \geq 0$, is deterministic. Hence, we give a standard complexity result as opposed to a high probability one. Finally, we make use of the fact that for partially separable functions f, the parameters β and w are known.

Theorem 12. *Assume $f : \mathbf{R}^N \to \mathbf{R}$ is partially separable of degree ω, and has block Lipschitz gradient with constants $L_1, L_2, \dots, L_n > 0$. Further assume that $F = f + \Psi$ is strongly convex with $\mu_F(L) > 0$, where $L = (L_1, \dots, L_n)$. Finally, let $\{x_k\}_{k\geq 0}$ be the sequence generated by fully parallel PCDM (Algorithm 6). Then for all $k \geq 0$,*

$$F(x_{k+1}) - F^* \leq q^{\scriptscriptstyle PCDM}(F(x_k) - F^*), \tag{57}$$

where

$$q^{\scriptscriptstyle PCDM} = 1 - \frac{\mu_F(L)}{\omega + \mu_F(L) - \mu_f(L)}. \tag{58}$$

Moreover, if we let $\epsilon < F(x_0) - F^*$ and

$$k \geq \frac{1}{1 - q^{\scriptscriptstyle PCDM}} \log\left(\frac{F(x_0) - F^*}{\epsilon}\right), \tag{59}$$

then $F(x_k) - F^* \leq \epsilon$.

Proof. Let $\hat{S}$ be the fully parallel sampling, i.e., the n-nice sampling. Applying Theorem 7, we see that $(f, \hat{S}) \sim ESO(\beta, w)$, with $\beta = \omega$ and $w = L$. Following the first part of the proof of Theorem 11, we have $\alpha = 1$ and $\xi_{k+1} \leq (1-\gamma)\xi_k$, where $\gamma = \mu_F(L)/(\mu_F(L) + \omega - \mu_f(L))$, establishing (57). The second statement follows directly by applying Lemma 9. □

7.2 DQAM

We now present a complexity result for DQAM, established in [29].

Theorem 13 (Theorem 2 in [29])**.** *Let $f(x) = \frac{r}{2}\|b - Ax\|^2$ be partially separable of degree $\omega > 1$. Assume that F $(= f + \Psi)$ is strongly convex with $\mu_F(e) > 0$, where $e \in \mathbf{R}^n$ is the vector of all ones. Further assume that the sets X_i, $i = 1, \dots, n$, are bounded. Let $\{x_k\}$, $k \geq 0$, be the sequence generated by DQAM (Algorithm 2) with $\theta = \frac{1}{2(\omega-1)}$. Then for all $k \geq 0$,*

$$F(x_{k+1}) - F^* \leq q^{DQAM}\big(F(x_k) - F^*\big),$$

where

$$q^{DQAM} = 1 - \frac{\mu_F(e)}{16L'(\omega-1)^3 + 4(\omega-1)\mu_F(e)}, \tag{60}$$

and $L' \overset{def}{=} \max_{1\leq i\leq n} r\|A_i\|^2$. Moreover, if we let $\epsilon < F(x_0) - F^*$ and

$$k \geq \frac{1}{1 - q^{DQAM}} \log\left(\frac{F(x_0) - F^*}{\epsilon}\right), \tag{61}$$

then $F(x_k) - F^* \leq \epsilon$.

Ruszczyński analyzed DQAM for a range of parameters θ: $\theta \in (0, 1/(\omega-1))$ [29, Theorem 1; $\mu = 0$]. However, the choice $\theta = 1/(2(\omega-1))$ is optimal [29, Eq (5.11)], and the above theorem presents Ruszczyński's result for this optimal choice of the stepsize parameter. A table translating the notation used in this paper and [29] is included in Appendix B.

7.3 Comparison of the Linear Rates of DQAM and PCDM

We now compare the convergence rates q^{DQAM} and q^{PCDM} defined in (60) and (58), respectively, and the resulting iteration complexity guarantees. We will argue that q^{PCDM} can be much better (i.e., smaller) than q^{DQAM}, leading to vastly improved iteration complexity bounds. However, as we shall see, in practice the fully parallel PCDM method and DQAM behave similarly, with PCDM being about twice as fast as DQAM.

Before we start with the comparison, recall from (42) that the gradient of $f(x) = \frac{r}{2}\|b - Ax\|^2$ (i.e., f covered by Theorem 13) is block Lipschitz with constants $L_i = r\|A_i^TA_i\|$, $i = 1, 2, \dots, n$. Hence, $L' = \max_i L_i$, which draws a link between the quantities L_i, $i = 1, 2, \dots, n$, appearing in Theorem 12 and L' appearing in Theorem 13.

- **Identical Lipschitz constants.** Assume now that $L_i = L'$ for all $i = 1, 2, \dots, n$ and let $L = (L_1, \dots, L_n)$, as in Theorem 12. Using (50) we observe that

  $$\mu_\phi(L) = \mu_\phi(L'e) = \frac{1}{L'}\mu_\phi(e), \tag{62}$$

  whence

  $$q^{\mathrm{PCDM}} \overset{(58)+(62)}{=} 1 - \frac{\mu_F(e)}{L'\omega + \mu_F(e) - \mu_f(e)}. \tag{63}$$

We can now directly compare q^{PCDM} and q^{DQAM} by comparing (63) and (60). Clearly[2],

$$16L'(\omega-1)^3 \geq L'\omega \qquad \text{and} \qquad 4(\omega-1)\mu_F(e) \geq \mu_F(e) - \mu_f(e), \tag{64}$$

and hence $q^{\text{PCDM}} \leq q^{\text{DQAM}}$. However, both inequalities in (64) can be very loose, which means that q^{PCDM} can be much better than q^{DQAM}. For instance, in the case when $\mu_F(e) = \mu_f(e)$, we have

$$\frac{1-q^{\text{PCDM}}}{1-q^{\text{DQAM}}} = \frac{16L'(\omega-1)^3 + 4(\omega-1)\mu_F(e)}{L'\omega} \geq \frac{16(\omega-1)^3}{\omega}. \tag{65}$$

In view of (59) and (61), this means that the number of DQAM iterations needed to obtain an ϵ-solution is larger than that for PCDM *by at least the multiplicative factor* $16(\omega-1)^3/\omega$. For instance, the theoretical iteration complexity of DQAM is more than 1000 times worse than that of PCDM for $\omega = 10$.

- **Varying Lipschitz Constants.** If the constants $L_1, \dots, L_n$ are not all equal, it is somewhat difficult to compare the complexity rates as we cannot directly compare the strong convexity constants $\mu_\phi(L)$ and $\mu_\phi(e)$ (for $\phi = F$ and $\phi = f$). What we can do, however, is to at least make sure that the "scaling" is identical in both. Here is what we mean by that. Recall that $\mu_\phi(w)$ is the strong convexity constant of ϕ wrt a *weighted norm* $\|x\|_w$ defined by (29). As we have remarked in (50), if we scale the weights by a positive factor $t > 0$, the corresponding strong convexity constant scales by $1/t$. Hence, $\mu_\phi(L)$ and $\mu_\phi(e)$ cannot be considered comparable unless $\sum_i L_i = \sum_i e_i = n$. Of course, even if this was the case, it is possible that the strong convexity constants might be very different. However, in this case there is at least no reason to suspect a-priori that one might be larger than the other, and hence they are comparable in that sense.

  If we let $\bar{L} = \frac{1}{n}\sum_i L_i$ and $w_i = L_i/\bar{L}$ for $i = 1, 2, \dots, n$, then $\sum_i w_i = n$, and hence, as explained above,

  $$\mu_\phi(w) \approx \mu_\phi(e). \tag{66}$$

  Furthermore, since $w = L/\bar{L}$, we have

  $$\mu_\phi(L) = \mu_\phi\left(\bar{L}w\right) \overset{(50)}{=} \frac{1}{\bar{L}}\mu_\phi(w) \approx \frac{1}{\bar{L}}\mu_\phi(e).$$

  The above is an analogue of (62) and we can therefore now continue our comparison in the same way as we did for the case with identical Lipschitz constants. In particular, if $\mu_F(e) = \mu_f(e)$ (for simplicity), then as above we can argue that

  $$\frac{1-q^{\text{PCDM}}}{1-q^{\text{DQAM}}} = \frac{16L'(\omega-1)^3 + 4(\omega-1)\mu_F(e)}{\bar{L}\omega} \geq \frac{16(\omega-1)^3}{\omega}\frac{L'}{\bar{L}}. \tag{67}$$

  Therefore, PCDM has an even more dramatic theoretical advantage compared to DQAM in the case when the maximum Lipschitz constant L' is much larger than the average $\bar{L}$.

[2] This holds as long as $\omega > 1$, which is the case covered by Theorem 13 and hence assumed here.

7.4 Optimal number of block updates

In this section we propose a simplified model of parallel computing and in it study the performance of a family of parallel coordinate descent methods parameterized by a single parameter: the number of blocks being updated in a single iteration.

In particular, consider the family of PCDMs where S_k is a τ-nice sampling and $\tau \in \{1, 2, \dots, n\}$. Now assume we have $p \in \{1, 2, \dots, n\}$ processors/threads available, each able to compute and apply to the current iterate the update $h^{(i)}(x_k)$ for a single block i, in a unit of time. PCDM, as analyzed, is a synchronous method. That is, a new parallel iteration can only start once the previous one is finished, and hence updating τ blocks will take $\lceil \frac{\tau}{p} \rceil$ amount of time. On the other hand, the iteration complexity of PCDM is better for higher τ. Indeed, by Theorem 7, f satisfies an ESO with respect to $\hat{S}$ with parameters $w = L = (L_1, \dots, L_n)$ and $\beta = \beta(\tau) = 1 + \frac{(\omega-1)(\tau-1)}{n-1}$, where ω is degree of partial separability of f (we assume $n > 1$). If, moreover, $\mu_F(L) = \mu_f(L)$, which is often the case as Ψ is often not strongly convex, then Theorem 11 says that PCDM needs $\frac{n}{\tau}\beta(\tau)c$ iterations, where c is a constant independent of τ, to solve (5) with high probability. Hence, the total amount of time needed for PCDM to solve the problem is equal to

$$T(\tau) = \lceil \tfrac{\tau}{p} \rceil \tfrac{n}{\tau} \beta(\tau) c.$$

We can now ask the following natural question: what $\tau \in \{1, 2, \dots, n\}$ minimizes $T(\tau)$? We now show that the answer is $\tau = p$.

Theorem 14. *Assume $f : \mathbf{R}^N \to \mathbf{R}$ is convex, partially separable of degree ω, and has block Lipschitz gradient with constants $L_1, L_2, \dots, L_n > 0$, where $n > 1$. Further assume $\mu_F(L) = \mu_f(L) > 0$ and consider the family of parallel coordinate descent methods with τ-nice sampling, where $\tau \in \{1, 2, \dots, n\}$, applied to problem* (5)*. Under the parallel computing model with $p \in \{1, 2, \dots, n\}$ processors described above, the method with $\tau = p$ is optimal.*

Proof. We only need to show that

$$p = \arg\min\{T(\tau) \ : \ \tau = 1, 2, \dots, n\}.$$

It is easy to see that $\frac{n}{\tau}\beta(\tau)$ is decreasing in τ. Since $\lceil \frac{\tau}{p} \rceil$ is constant for $kp + 1 \le \tau \le kp$, it suffices to consider $\tau = kp$ for $k = 1, 2, \dots$ only. Finally, $T(kp) = \frac{n}{p}\beta(kp)c$ is increasing in k since $\beta(\cdot)$ is increasing, and we conclude that $k = 1$ and hence $\tau = p$ is optimal. □

8 Numerical Results

In this section we present two numerical experiments that support the findings of this paper. In both experiments we choose $f(x) = \frac{1}{2}\|b - Ax\|^2$ and $\Psi \equiv 0$.

The first experiment considers the performance of DQAM and the fully parallel variant of PCDM in the above setting where we know that the two methods coincide up to he selection of the stepsize parameters ω and θ (recall Section 6.4). Here we focus on comparing the effects of using the DQAM stepsize $\theta = 1/(2(\omega - 1))$ versus the larger PCDM stepsize $\theta = 1/\omega$.

The second experiment compares DQAM, fully parallel variant of PCDM (i.e., PCDM used with n-nice sampling) and PCDM used with τ-nice sampling, in the situation when the number of available processors is τ, while varying ω (degree of partial separability of f) and τ.

8.1 Impact of the different stepsizes of DQAM and PCDM

Suppose that A has primal block angular structure

$$A = \begin{bmatrix} C \\ D \end{bmatrix} = \begin{bmatrix} C_1 & & \\ & \ddots & \\ & & C_n \\ D_1 & \dots & D_n \end{bmatrix},$$

where C_i, D_i are matrices of appropriate sizes. Notice that when $D = 0$, the problem is partially separable of degree $\omega = 1$ (i.e., it is fully separable) with respect to the natural block structure (i.e., blocks corresponding to the column submatrices $[C_i; 0; D_i]$). If D is completely dense, the problem is nonseparable ($\omega = n$). In general, the degree of separability of f is equal to the number of matrices D_i that contain at least one nonzero entry.

In this (small scale) experiment we set $n = 100$ and let $C_1, \dots, C_{100}$ be 10% dense matrices of size 150×100. Subsequently, A is a $15,001 \times 10,000$ sparse matrix. The degree of separability of f varies, and is controlled by setting a subset of the matrices $D_1, \dots, D_n$ to zero.

Twenty five random pairs (A, b) were generated for each $\omega \in \{2, 4, 8, 16, 32\}$, and DQAM and fully parallel variant of DQAM were applied to each problem instance. A stopping condition of $f(x) \leq 10^{-4} b^T b$ was employed; the results of this experiment are presented in Figure 8.1. All data points are averages over 25 runs.

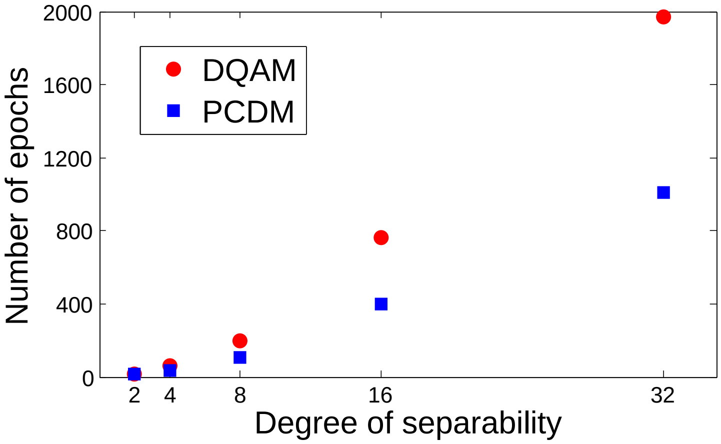

Figure 1: This plot shows the number of epochs (a full sweep through the data, i.e., all $i = 1, \dots, n$ blocks of x are updated in one epoch) needed to solve the problem as a function of the degree of separability ω.

Notice that when $\omega = 2$, DQAM and PCDM require the same number of epochs to solve the problem. This is because $\theta = 1/(2(\omega - 1)) = 1/2 = 1/\omega$. Then as ω grows, PCDM performs far better than DQAM, requiring almost 50% fewer epochs than DQAM.

8.2 Comparison of full vs partial parallelization

Recall that unlike DQAM, PCDM is able to update τ blocks at each iteration, for any τ in the set $\{1, 2, \ldots, n\}$, demonstrating useful flexibility of the algorithm. By PCDM(τ) we denote the variant of PCDM in which τ blocks are updated at each iteration, using a τ-nice sampling. In this experiment we investigate the performance of DQAM, PCDM(n) (which in the plots we refer to simply as PCDM) and PCDM(τ), for a selection of parameters τ (the number of processors), and ω (the degree of partial separability).

Let us call the time taken for all τ processors to update a single block, one "time unit". Then, after one time unit of PCDM(τ), new gradient information is available to be utilized during the next time unit, which is much earlier than if all n blocks need to be updated in each iteration. On the other hand, for DQAM and PCDM, one iteration corresponds to all n blocks of x being updated. Subsequently, if there are τ processors available, one iteration of DQAM or PCDM (one epoch) corresponds to $\lceil \frac{n}{\tau} \rceil$ time units. However, PCDM(τ) will need to perform more iterations than both DQAM and PCDM. When both of these factors are taken into account, we have shown in Theorem ??? that PCDM(τ) is optimal in terms of overall complexity if there are τ processors.

The purpose of this experiment is to investigate this phenomenon numerically. Further, let A be a $2 \cdot 10^4 \times 10^4$ sparse matrix, with at most ω nonzero entries per row. Let the stopping condition be $f(x) \leq 10^{-4} b^T b$. The experiment was run for three instances: $\omega = 20, 60, 100$, and for each ω and varying τ, the average number of time units required by DQAM, PCDM and PCDM(τ) were recorded. The results are shown in Figure 2.

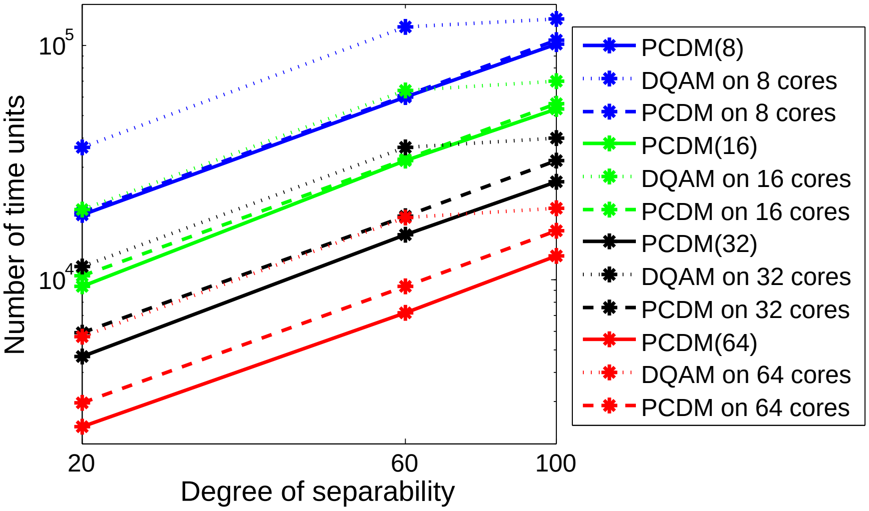

Figure 2: For each fixed $\tau \in \{8, 16, 32, 64\}$, PCDM($\tau$) (solid line) is better than PCDM (dashed line), and noth are far better than DQAM (dotted line).

The colors in Figure 2 correspond to different values of τ. The solid lines correspond to PCDM(τ), while the dotted line (respectively dashed line) corresponds to DQAM (respectively PCDM) run with τ processors available. As ω increases, all algorithms require a higher number of time units. Further, as the number of available processors increases, the number of time units decreases. More importantly, for any fixed τ, PCDM(τ), requires far fewer time units than PCDM, and both require many fewer time units than DQAM. (Notice the log scale.) This demonstrates the practical advantage of 'optimizing' PCDM(τ) to the number of available processors, as described in Section 7.4.

We have also recorded the average cpu time, and the resulting curves are visually indistinguishable from those in Figure 2; only the scale of the vertical axis changes.

References

[1] Arno J. Berger, John M. Mulvey, and Andrzej Ruszczyński. An extension of the DQA algorithm to convex stochastic programs. *SIAM Journal on Optimization*, 4(4):735–753, November 1994.

[2] Dimitri Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods.* Athena Scientific, 1996.

[3] Mathieu Blondel, Kazuhiro Seki, and Kuniaki Uehara. Block coordinate descent algorithms for large-scale sparse multiclass classification. *Machine Learning*, 2013.

[4] Joseph K. Bradley, Aapo Kyrola, Danny Bickson, and Carlos Guestrin. Parallel coordinate descent for L1-regularized loss minimization. In *28th International Conference on Machine Learning*, 2011.

[5] Olivier Fercoq. Parallel coordinate descent for the Adaboost problem. Technical report, July 2013.

[6] Olivier Fercoq and Peter Richtárik. Smoothed parallel coordinate descent method. Technical report, 2013.

[7] Magnus R. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4:303–320, 1969.

[8] Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S Sathiya Keerthi, and S Sundararajan. A dual coordinate descent method for large-scale linear svm. In *ICML 2008*, pages 408–415, 2008.

[9] Yin Tat Lee and Aaron Sidford. Effcient accelerated coordinate descent methods and faster algorithms for solving linear systems. *arXiv:1305:1922v1*, 2013.

[10] Yingying Li and Stanley Osher. Coordinate descent optimization for l_1 minimization with application to compressed sensing; a greedy algorithm. *Inverse Problems and Imaging*, 3:487–503, August 2009.

[11] Zhaosong Lu and Lin Xiao. On the complexity analysis of randomized block-coordinate descent methods. Technical report, May 2013. arXiv:1305.4723.

[12] Zhaosong Lu and Lin Xiao. Randomized block coordinate non-monotone gradient method for a class of nonlinear programming. Technical report, June 2013. arXiv:1306.5918.

[13] John M. Mulvey and Andrzej Ruszczyński. A diagonal quadratic approximation method for large scale linear programs. *Operations Research Letters*, 12:205–215, 1992.

[14] John M. Mulvey and Andrzej Ruszczyński. A new scenario decomposition method for large scale stochastic optimization. *Operations Research*, 43(3):477–490, 1995.

[15] Ion Necoara, Yurii Nesterov, and Francois Glineur. Efficiency of randomized coordinate descent methods on optimization problems with linearly coupled constraints. Technical report, June 2012.

[16] Ion Necoara and Andrei Patrascu. A random coordinate descent algorithm for optimization problems with composite objective function and linear coupled constraints. Technical report, University Politehnica Bucharest, 2012. arXiv:1302.3074.

[17] Yurii Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimimization*, 22(2):341–362, 2012.

[18] Andrei Patrascu and Ion Necoara. Efficient random coordinate descent algorithms for large-scale structured nonconvex optimization. Technical report, University Politehnica Bucharest, May 2013.

[19] Michael J. D. Powell. A method for nonlinear constraints in minimization problems. In Roger Fletcher, editor, *Optimization*, pages 283–298. Academic Press, 1972.

[20] Zhiwei (Tony) Qin, Katya Scheinberg, and Donald Goldfarb. Efficient block-coordinate descent algorithms for the group lasso. Technical report, Department of Industrial Engineering and Operations Research, Columbia University, 2010.

[21] Peter Richtárik and Martin Takáč. Efficient serial and parallel coordinate descent methods for huge-scale truss topology design. In *Operations Research Proceedings 2011*, pages 27–32. Springer, 2012.

[22] Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming, Ser. A*, 2012.

[23] Peter Richtárik and Martin Takáč. Parallel coordinate descent methods for big data optimization. Technical report, November 2012. arXiv:1212.0873.

[24] Peter Richtárik and Martin Takáč. Efficiency of randomized coordinate descent methods on minimization problems with a composite objective function. In *4th Workshop on Signal Processing with Adaptive Sparse Structured Representations*, June 2011.

[25] R. Tyrell Rockafellar. The multiplier method of Hestenes and Powell applied to convex programming. *Journal of Optimization Theory and Applications*, 12:555–562, 1973.

[26] R. Tyrell Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of Operations Research*, 1:97–116, 1976.

[27] R. Tyrell Rockafellar and Roger J.-B. Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, 16:1–23, 1991.

[28] Andrzej Ruszczyński. An augmented Lagrangian method for block diagonal linear programming problems. *Operations Research Letters*, 8:287–294, 1989.

[29] Andrzej Ruszczyński. On convergence of an augmented Lagrangian decomposition method for sparse convex optimization. *Mathematics of Operations Reseach*, 20(3):634–656, 1995.

[30] Hermann Schwarz. Über einen Grenzübergang durch alternierendes Verfahren. *Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich*, 15:272–286, 1870.

[31] Shai Shalev-Shwartz and Ambuj Tewari. Stochastic methods for l_1 regularized loss minimization. In *26th International Conference on Machine Learning*, 2009.

[32] Shai Shalev-Shwartz and Tong Zhang. Accelerated mini-batch stochastic dual coordinate ascent. Technical report, May 2013. arXiv:1305.2581.

[33] Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14:567–599, 2013.

[34] George Stephanopoulos and Arthur W. Westerberg. The use of Hestenes' method of multipliers to resolve dual gaps in engineering system optimization. *Journal of Optimization Theory and Applications*, 15:285–309, 1975.

[35] Martin Takáč, Avleen Bijral, Peter Richtárik, and Nathan Srebro. Mini-batch primal and dual methods for SVMs. In *30th International Conference on Machine Learning*, 2013.

[36] Rachael Tappenden, Peter Richtárik, and Jacek Gondzio. Inexact coordinate descent: complexity and preconditioning. Technical report, April 2013. arXiv:1304.5530.

[37] Paul Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109:475–494, June 2001.

[38] N. Watanabe, Y. Nishimura, and M. Matsubara. Decomposition in large system optimization using the method of multipliers. *Journal of Optimization Theory and Applications*, 25:181–193, 1978.

A Notation Dictionary

For the reader interested in comparing our work with the paper [29] directly, we have included a brief dictionary translating some of the key notation (Table 1).

Table 1: Notation dictionary.

Ruszczyński [29]	This paper
L	n
N	$\omega - 1$
x_i	$x^{(i)}$
$\tilde{x}$	x
x	y
$x - \tilde{x}$	$h = y - x$
τ	θ
ρ	r
$\rho\alpha^2$	L'
γ	$\mu_F(e)/2$
$\frac{1}{2}r\Vert b - \sum_{i=1}^n A_i x_i\Vert_2^2$	$f(x)$
$f_i(x_i) - \langle A_i^T \pi, x_i\rangle$	$\Psi_i(x^{(i)})\ \ (= g_i(x_i) - \langle A_i^T \pi, x_i\rangle)$
$\Lambda(x)$	$F(x) = f(x) + \Psi(x)$
$\Lambda_i(x_i, \tilde{x})$	$f(x + U_i h^{(i)}) + \Psi_i(y^{(i)})$
$\tilde{\Lambda}(x, \tilde{x})$	$f(x) + \sum_{i=1}^n [f(x + U_i h^{(i)}) - f(x)] + \Psi(x + h)$