A BLOCK COORDINATE DESCENT METHOD OF MULTIPLIERS: CONVERGENCE ANALYSIS AND APPLICATIONS

Mingyi Hong[1], *Tsung-Hui Chang*[2], *Xiangfeng Wang*[3], *Meisam Razaviyayn*[1], *Shiqian Ma*[4], *Zhi-Quan Luo*[1]

[1] University of Minnesota, Minneapolis, USA
[2] National Taiwan University of Science and Technology, Taiwan, R.O.C.
[3] Nanjing University, Nanjing, P. R. China
[4] The Chinese University of Hong Kong, Hong Kong, P. R. China

ABSTRACT

In this paper, we consider a nonsmooth convex problem with linear coupling constraints. Problems of this form arise in many modern large-scale signal processing applications including the provision of smart grid networks. In this work, we propose a new class of algorithms called the *block coordinate descent method of multipliers* (BCDMM) to solve this family of problems. The BCDMM is a primal-dual type of algorithm. It optimizes an (approximate) augmented Lagrangian of the original problem one block variable per iteration, followed by a gradient update for the dual variable. We show that under certain regularity conditions, and when the order for which the block variables are either updated in a deterministic or a random fashion, the BCDMM converges to the set of optimal solutions. The effectiveness of the algorithm is illustrated using large-scale basis pursuit and smart grid problems.

1. INTRODUCTION

Consider the problem of minimizing a convex function $f(x)$ subject to linear equality constraints:

$$\begin{array}{ll} \min & f(x) := g\left(x_1, \cdots, x_K\right) + \sum_{k=1}^K h_k(x_k) \\ \text{subject to} & E_1x_1 + E_2x_2 + \cdots + E_Kx_K = q, \\ & x_k \in X_k, \quad k = 1, 2, ..., K, \end{array} \tag{1.1}$$

where $g(\cdot)$ is a smooth convex function; h_k is a nonsmooth convex function; $x = (x_1^T, ..., x_K^T)^T \in \Re^n$ is a partition of the optimization variable x, $x_k \in \Re^{n_k}$; $X = \prod_{k=1}^K X_k$ is the feasible set for x; $q \in \Re^m$ is a vector. Let $E := (E_1, \cdots, E_K)$ and $h(x) := \sum_{k=1}^K h_k(x_k)$.

Many problems arise in modern large-scale signal processing, machine learning and smart grid systems can be formulated into the form (1.1). A well-known example is the basis pursuit (BP) problem, which solves the following nonsmooth problem [1]

$$\min_x \|x\|_1 \quad \text{s.t.} \quad Ex = q,\ x \in X. \tag{1.2}$$

One important application of this model is in compressive sensing, where a sparse signal x needs to be recovered using a small number of observations q (i.e., $m \ll n$) [1]. Let us partition x by $x = [x_1^T, \cdots, x_K^T]^T$ where $x_k \in \Re^{n_k}$, and partition E accordingly. Then the BP problem can be written in the form of (1.1)

$$\min_x \sum_{k=1}^K \|x_k\|_1 \quad \text{s.t.} \quad \sum_{k=1}^K E_k x_k = q,\ x_k \in X_k,\ \forall\, k. \tag{1.3}$$

The second example has to do with the design of the smart grid system. Consider a power grid system in which a utility company bids the electricity from the power market and serves a neighborhood with K customers. The total cost for the utility includes the bidding cost in a wholesale day-ahead market and a real-time market. In the envisioned smart grid system, the utility will have the ability to control the power consumption of some appliances (e.g., controlling the charging rate of electrical vehicles) in a way to minimize its total cost. This problem, known as the demand response (DR) control problem, is central to the success of the smart grid system [2, 3, 4].

Let p_ℓ, $\ell = 1, \cdots, L$, denote the bids in a day-ahead market for a period L. Let $\mathbf{\Psi}_k\mathbf{x}_k$ denote the load profile of a customer $k = 1, \cdots, K$, where $\mathbf{x}_k \in \Re^{n_k}$ are some control variables for the equipments of customer k, and $\mathbf{\Psi}_k \in \Re R^{L\times n_k}$ contains the information related to the appliance load model [5]. The retailer aims at minimizing the bidding cost as well as the cost incurred by power imbalance in the next day [2, 3, 4]

$$\begin{aligned} \min_{\{\mathbf{x}_k\},\mathbf{p}} \quad & C_p\left[\Big(\sum_{k=1}^K \mathbf{\Psi}_k\mathbf{x}_k - \mathbf{p}\Big)^+\right] + C_s\left[\Big(\mathbf{p} - \sum_{k=1}^K \mathbf{\Psi}_k\mathbf{x}_k\Big)^+\right] + C_d(\mathbf{p}) \\ \text{s.t.} \quad & \mathbf{x}_k \in X_k,\ k = 1, \cdots, K,\ \mathbf{x} \geq 0,\ \mathbf{p} \geq 0 \end{aligned} \tag{1.4}$$

where $C_p(\cdot)$ and $C_s(\cdot)$ are increasing functions which model the cost incurred by insufficient and excessive power bids, respectively; $C_d(\cdot)$ represents the bidding cost function; $(x)^+ := \max\{x, 0\}$; X_k is some compact set; see [3]. Upon introducing a new variable $\mathbf{z} = \Big(\sum_{k=1}^K \mathbf{\Psi}_k\mathbf{x}_k - \mathbf{p}\Big)^+$, the above problem can be equivalently transformed into the form of (1.1):

$$\begin{aligned} \min_{\{\mathbf{x}_k\},\mathbf{p},\mathbf{z}} \quad & C_p(\mathbf{z}) + C_s\big(\mathbf{z} + \mathbf{p} - \sum_{k=1}^K \mathbf{\Psi}_k\mathbf{x}_k\big) + C_d(\mathbf{p}) \\ \text{s.t.} \quad & \sum_{k=1}^K \mathbf{\Psi}_k\mathbf{x}_k - \mathbf{p} - \mathbf{z} \leq 0,\ \mathbf{z} \geq 0,\ \mathbf{p} \geq 0,\ \mathbf{x}_k \in X_k,\ \forall\, k. \end{aligned} \tag{1.5}$$

The third example arises in optimizing the cognitive radio network (CRN). Consider a network with K secondary users (SUs) and a single secondary base station (SBS) operating on M parallel frequency tones. The SUs are interested in transmitting their messages to the SBS. Let s_k^m denote user k's transmit power on mth channel;

Research of M. Hong and Z.-Q. Luo is supported in part by the National Science Foundation, grant number CCF-1216858, and by a research gift from Huawei Technologies Inc. Research of S. Ma was supported in part by the Hong Kong Research Grants Council (RGC) Early Career Scheme (ECS) (Project ID: CUHK 439513). T.-H. Chang is supported by National Science Council, Taiwan (R.O.C.), under grant NSC 102-2221-E-011-005-MY3

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

let h_k^m denote the channel between user k and the SBS on tone m; let P_k denote SU k's total power budget. Also suppose that there are L primary users (PU) in the system, and let $g_{k\ell}^m$ denote the channel between the kth SU to the ℓth PU. The secondary network aims at maximizing the system throughput, subject to the constraint that certain interference temperature (IT) constraints measured at the receivers of the PUs are not violated [6, 7]:

$$\begin{aligned} \max_{\{s_k^m\}} \quad & \sum_{m=1}^{M} \log\left(1+\sum_{k=1}^{K}|h_k^m|^2 s_k^m\right) \\ \text{s.t.} \quad & s_k^m \geq 0,\ \sum_{m=1}^{M} s_k^m \leq P_k,\ \sum_{k=1}^{K}|g_{k\ell}^m|^2 s_k^m \leq I_\ell^m,\ \forall\, \ell,\ k,\ m \end{aligned} \tag{1.6}$$

where $I_\ell^m \geq 0$ denote the IT constraint for PU ℓ on tone m. Clearly this problem is also in the form of (1.1).

2. LITERATURE REVIEW

In the absence of the linear coupling constraints, a well known technique for solving (1.1) is to use the so-called block coordinate descent (BCD) method whereby, at every iteration, the following subproblem is solved for a single block of variables, while the remaining blocks are held fixed:

$$\min_{x_k \in \mathcal{X}_k} \ g(x_1^r, \ldots, x_{k-1}^r, x_k, x_{k+1}^{r-1}, \ldots, x_K^{r-1}) + h_k(x_k). \tag{2.7}$$

Since each step involves solving a simple subproblem of small size, the BCD method can be quite effective for solving large-scale problems; see e.g., [8, 9, 10, 11] for various applications in statistics, signal processing and machine learning. The existing theoretical analysis of the BCD method requires the uniqueness of the minimizer for each subproblem (2.7), or the quasi convexity of f [12, 13, 14, 15]. When problem (2.7) is not easily solvable, a popular approach is to solve an approximated version of problem (2.7), yielding the block coordinate gradient decent (BCGD) algorithm, or the block coordinate proximal gradient (BCPG) algorithm in the presence of nonsmooth function h [16, 17, 10, 18, 19].

When the linear coupling constraint is present, it is well known that the BCD-type algorithm may fail to find any (local) optimal solution [20]. A popular algorithm for solving this type of problem is the so-called alternating direction method of multipliers (ADMM) [21, 22]. In the ADMM method, instead of maintaining feasibility all the time, the constraint $Ex = q$ is dualized using the Lagrange multiplier y and a quadratic penalty term is added. The resulting *augmented Lagrangian function* is of the form:

$$L(x; y) = f(x) + \langle y, q - Ex\rangle + \frac{\rho}{2}\|q - Ex\|^2, \tag{2.8}$$

where $\rho > 0$ is a constant, and $\langle\cdot,\cdot\rangle$ denotes the inner product operator. The ADMM updates the primal block variables $x_1, \ldots, x_K$ in a block coordinate manner to minimize $L(x; y)$, which often leads to simple subproblems with closed form solutions. These updates are followed by a gradient ascent update of the dual variable y.

Unfortunately, neither BCD nor ADMM can be used to solve problem (1.1). In fact, due to its multi-block structure as well as the variable coupling in *both* the objective and the constraints, this problem cannot be handled by the existing methods for big data including SpaRSA [23], FPC-BB [24], FISTA [25] and ALM [26]. The main contribution of this paper is to propose and analyze a block coordinate descent method of multipliers (BCDMM) and its randomized version that can solve problem (1.1) efficiently. The proposed algorithm is flexible, because the primal per-block problem can be solved *inexactly*, and this allows one to perform simple gradient or proximal gradient step for difficult subproblems.

3. THE BCDMM ALGORITHM

In its basic form, the BCDMM algorithm optimizes certain upper bound of the augmented Lagrangian (2.8) one block variable at a time, followed by a gradient type update for the dual variable. In particular, at iteration $r+1$, the block variable k is updated by solving the following subproblem

$$\begin{aligned} \min_{x_k \in X_k} \quad & u_k\left(x_k; x_1^{r+1}, \cdots, x_{k-1}^{r+1}, x_k^r, \cdots, x_K^r\right) \\ & + \langle y^{r+1}, q - E_k x_k\rangle + h_k(x_k) \end{aligned} \tag{3.9}$$

where $u_k(\cdot; x_1^{r+1}, \cdots, x_{k-1}^{r+1}, x_k^r, \cdots, x_K^r)$ is certain upper bound of $g(x) + \frac{\rho}{2}\|q - Ex\|^2$ at a given iterate. To simplify notations, let us define a new set of auxiliary variables

$$w_k^r = [x_1^r, \cdots, x_{k-1}^r, x_k^{r-1}, x_{k+1}^{r-1}, \cdots, x_K^{r-1}],\ k = 1, \cdots, K,$$
$$w_{K+1}^r = x^r, \quad w_1^r = x^{r-1}.$$

The BCDMM algorithm is described in the following table.

The BCDMM Algorithm

At each iteration $r \geq 1$:

$$\begin{cases} y^{r+1} = y^r + \alpha^r(q - Ex^r) = y^r + \alpha^r\left(q - \sum_{k=1}^{K} E_k x_k^r\right), \\ x_k^{r+1} = \arg\min_{x_k \in X_k} u_k(x_k; w_k^{r+1}) - \langle y^{r+1}, E_k x_k\rangle + h_k(x_k),\ \forall\, k \end{cases}$$

where $\alpha^r > 0$ is the step size for the dual update.

In this paper, we also consider a randomized version of the BCDMM algorithm, in each iteration of which a single block of primal or dual variable is randomly picked to update.

The R-BCDMM Algorithm

Select a vector $\{p_k > 0\}_{k=0}^K$ such that $\sum_{k=0}^K p_k = 1$; At iteration $t \geq 1$, pick $k \in \{0, \cdots, K\}$ with probability p_k and

If $k = 0$

$$y^{t+1} = y^t + \alpha^t(q - Ex^t),$$

$$x_k^{t+1} = x_k^t,\ k = 1, \cdots, K.$$

Else If $k \in \{1, \cdots, K\}$

$$x_k^{t+1} = \text{argmin}_{x_k \in X_k}\, u_k(x_k; x^t) - \langle y^r, E_k x_k\rangle + h_k(x_k),$$

$$x_j^{t+1} = x_j^t,\ \forall\, j \neq k, \quad y^{t+1} = y^t.$$

End

where $\alpha^t > 0$ is the step size for the dual update.

As explained in [11, 27], the randomized version of the BCD-type of algorithm is useful under many practical scenarios, for example when not all data is available at all times. Note that here we have used the index "t" to differentiate the iteration of R-BCDMM with that of the BCDMM. The reason is that in R-BCDMM, at each iteration only a single block variable (primal or dual) is updated, while in BCDMM all primal and dual variables are updated once.

4. CONVERGENCE ANALYSIS

4.1. Main Assumptions

Suppose f is a closed proper convex function in $\Re R^n$. Let $\mathrm{dom}\ f$ denote the effective domain of f and let $\mathrm{int}(\mathrm{dom}\ f)$ denote the interior of $\mathrm{dom}\ f$. Let x_{-k} (and similarly E_{-k}) denote the vector x with x_k removed. We make the following standing assumptions regarding the problem (1.1):

Assumption A.

(a) Problem (1.1) is convex, its global minimum is attained and so is its dual optimal value. The intersection $X \cap \mathrm{int}(\mathrm{dom}\ f) \cap \{x \mid Ex = q\}$ is nonempty.

(b) The function $g(x)$ can be decomposed as $g(x) = \ell(Ax) + \langle x, b\rangle$, where $\ell(\cdot)$ is a strictly convex and continuously differentiable function on $\mathrm{int}(\mathrm{dom}\ \ell)$, and A is some given matrix (not necessarily full column rank).

Each nonsmooth function h_k, if present, takes the form

$$h_k(x_k) = \lambda_k \|x_k\|_1 + \sum_J w_J \|x_{k,J}\|_2,$$

where $x_k = (\cdots, x_{k,J}, \cdots)$ is a partition of x_k with J being the partition index; $\lambda_k \geq 0$ and $w_J \geq 0$ are some constants.

(c) The feasible sets X_k, $k = 1, \cdots, K$ are compact polyhedral sets, and are given by $X_k := \{x_k \mid C_k x_k \geq c_k\}$, for some matrix $C_k \in \Re^{m_k \times n_k}$ and $c_k \in \Re^{m_k}$.

Next we make the following assumptions regarding the approximation function $u_k(\cdot;\cdot)$ in (3.9).

Assumption B.

(a) $u_k(x_k; x) = g(x) + \frac{\rho}{2}\|Ex - q\|^2, \quad \forall\, x \in X,\ \forall\, k.$

(b) $u_k(v_k; x) \geq g(v_k, x_{-k}) + \frac{\rho}{2}\|E_k v_k - q + E_{-k} x_{-k}\|^2,\ \forall\, v_k \in X_k,\ \forall\, x \in X,\ \forall\, k.$

(c) $\nabla u_k(x_k; x) = \nabla_k \left(g(x) + \frac{\rho}{2}\|Ex - q\|^2\right), \forall\, k,\ \forall\, x \in X.$

(d) For any given x, $u_k(v_k; x)$ is continuous in v_k and x; Moreover, it is strongly convex in v_k:

$$u_k(v_k; x) \geq u_k(\hat{v}_k; x) + \langle \nabla u_k(\hat{v}_k; x), v_k - \hat{v}_k\rangle + \frac{\gamma_k}{2}\|v_k - \hat{v}_k\|^2,\ \forall\, v_k,\ \hat{v}_k \in X_k,\ \forall\, x \in X.$$

where γ_k is independent of the choice of x.

(e) For given x, $u_k(v_k; x)$ has Lipschitz continuous gradient:

$$\|\nabla u_k(v_k; x) - \nabla u_k(\hat{v}_k; x)\| \leq L_k \|v_k - \hat{v}_k\|, \quad \forall\, \hat{v}_k,\ v_k \in X_k,\ \forall\, k,\ \forall\, x \in X, \tag{4.10}$$

where $L_k > 0$ is some constant.

Below we give a few remarks about the assumptions made above.

Remark 4.1 *The form of $g(\cdot)$ assumed in Assumption A(b) is fairly general. For example it includes the cases like $g(\cdot) = \sum_{k=1}^K \ell_k(A_k x_k)$, or $g(\cdot) = \ell(\sum_{k=1}^K A_k x_k)$, or the combination of these two, where $\ell_k(\cdot)$'s are strictly convex functions and A_k's are matrices not necessarily with full rank. Moreover, since the matrix A is not required to have full rank, $g(x)$ (hence $f(x)$) is not necessarily strongly convex with respect to x. Note that all three examples mentioned in Section 1 satisfy Assumption A(b). Moreover, this assumption requires that the nonsmooth function $h_k(\cdot)$ is in the form of mixed ℓ_1 and ℓ_2 norm.*

Remark 4.2 *Assumption B indicates that for any x, each $u_k(\cdot; x)$ is a locally tight upper bound for $g(x) + \frac{\rho}{2}\|q - Ex\|^2$ (the latter function itself satisfies Assumption B trivially). In many practical applications especially for nonsmooth problems, optimizing such functions often leads to much simpler subproblems than working directly with the original function; see e.g., [8, 26, 28, 29]. As an example, suppose the augmented Lagrangian is given by:*

$$L(x; y) = \sum_{k=1}^K \|x_k\|_2 + \langle y, q - Ax\rangle + \rho\|Ax - q\|^2.$$

Then at $(r+1)$-th iteration, the subproblem for x_k is given by

$$x_k^{r+1} = \arg\min_{x_k \in X_k} \|x_k\|_2 + \langle y^{r+1}, q - A_k x_k\rangle + \rho\|A_k x_k - d_k\|^2,$$

and this problem does not have closed form solution. A well-known strategy is to perform a proximal gradient step [30], that is, to solve the following approximated problem instead

$$\min_{x_k \in X_k} \|x_k\|_2 + \langle y^{r+1}, q - A_k x_k\rangle + \langle 2\rho A_k^T(A_k x_k^r - d_k), x_k\rangle + \frac{\tau}{2}\|x_k - x_k^r\|^2 \tag{4.11}$$

for some constant $d_k = q - \sum_{j<k} A_j x_j^{r+1} - \sum_{j>k} A_j x_j^r$. This problem readily admits a closed form solution; see e.g. [31, 17]. Moreover, when choosing $\tau \geq \|A_k^T A_k\|$, the strongly convex function $\langle 2\rho A_k^T(A_k x_k^r - d_k), x_k\rangle + \frac{\tau}{2}\|x_k - x_k^r\|^2$ is an approximation function that satisfies Assumption B (up to some constant).

Remark 4.3 *The strong convexity assumption for the approximation function $u_k(\cdot;\cdot)$ in B(d) is quite mild, see the examples given in the previous remark. This assumption ensures the iterates of (randomized) BCDMM are well defined.*

Now we are ready to present the main convergence result for the BCDMM and R-BCDMM.

Theorem 4.1 *Suppose Assumptions A and B hold. Suppose that the sequence of stepsizes $\{\alpha^r\}_r$ satisfies*

$$\sum_{r=1}^\infty \alpha^r = \infty, \quad \lim_r \alpha^r = 0. \tag{4.12}$$

Then we have the following:

1. *For the BCDMM, the sequence of constraint violations $\{\|Ex^r - q\|\}$ converges to zero. Further, every limit point of $\{x^r, y^r\}$ is a primal-dual optimal solution for problem* (1.1).
2. *For the R-BCDMM, the sequence of the constraint violation $\{\|Ex^t - q\|\}$ converges to zero with probability 1 (w.p.1). Further, every limit point of $\{x^t, y^t\}$ is a primal-dual optimal solution for problem* (1.1) *w.p.1.*

This result shows that by properly choosing the stepsizes, the convergence of (R-)BCDMM can be guaranteed, regardless of the number of primal blocks. Due to space limitations, we refer the readers to [32] for detailed proofs.

5. SIMULATION RESULTS

In this section, we present numerical results to demonstrate the effectiveness of BCDMM for the BP and the DR problem.

Let us first consider the BP problem (1.3), and fix each block variable x_k to be a *scalar*. Then the primal subproblem for BCDMM at r-th iteration for k-th variable is given by

$$\min_{x_k} \frac{1}{\rho\|e_k\|^2}|x_k| + \frac{1}{2}\left(x_k + \frac{e_k^T c_k^r}{\|e_k\|^2}\right)^2 \tag{5.13}$$

Table 1. Average #MVM performance for different algorithms.

	BCDMM	PALM	DALM	FISTA
#MVM	**226**	948	840	768

Table 2. Relative error performance of BCDMM for large-scale problem.

# of iterations	Exp. 1	Exp. 2
1	1	1
5	0.35	0.35
10	0.0012	0.16
15	7e-6	2e-3
20	N/A	1e-5
25	N/A	8e-7

where e_k is the k-th column of E, $c_k^r = w_{-k}^r + y^{r+1}/\rho - q$. This problem can be solved in closed-form by the soft-thresholding operator. We randomly generate the matrix $E \in \Re^{m\times n}$ and the true solutions $\bar{x}$ with each of their nonzero component following standard Gaussian distribution. We let E be a dense matrix, and $\bar{x}$ be a sparse vector, with each component having probability $p \in (0,1)$ to be nonzero (see [33] for details). We normalize the columns of E to have norm 1. The stepsize in BCDMM is given as follows: $\rho = 10 \times m/\|q\|_1$, $\alpha^r = \rho\frac{10+1}{\sqrt{r+10}}$. The BCDMM algorithm is compared with a number of well-known algorithms for BP such as DALM, PALM, FISTA, etc., see [33] for detailed review and implementation of these algorithms.

We first consider a relatively small problem with $n = 10000$ and $m = 3000$ and $p = 0.06$. The stopping criteria for all the algorithms is that either the iteration counter is larger than 1000, or the relative error $\|x^r - \bar{x}\|/\|\bar{x}\| \le 10^{-10}$. In Table 5, we show the averaged performance (over 100 problem realizations) for different algorithms. For a fair comparison of the computational cost, the algorithms are compared according to the number of matrix-vector multiplications, denoted by #MVM, which includes both Ex and $E^T y$. Clearly the BCDMM approach exhibits superior performance over all other algorithms.

It is worth mentioning that except for BCDMM, all the rest of the algorithms suffer from pitfalls that prevent them from solving really large problems. For example the PALM require the knowledge of $\rho(E^TE)$ (the largest eigenvalue of E^TE), the version of DALM with convergence guarantee requires the inversion of EE^T [34], both of which are difficult operations when E is large (say when n and m are larger than 10^6). The FISTA algorithm either needs $\rho(E^TE)$, or is required to perform backtrack line search within each iteration [25], both of which are again difficult to implement for large size problems. In contrast, each step of the BCDMM algorithm is simple and has closed-form solution, which makes it easily scalable for large problems. We have also tested the BCDMM on two large experiments [1]: experiment 1 with $m = 10^6$, $n = 10^3$ and $\|\bar{x}\|_0 = 28$; experiment 2 with $n = 10^6$, $m = 2 \times 10^3$ and $\|\bar{x}\|_0 = 82$. It takes 7 GB and 14 GB of memory space to store the data of these problems, respectively. For both problems, the BCDMM performs quite well: for the first (resp. the second) experiment it takes around 15 iterations and 60 seconds (resp. 25 iterations and 200 seconds) to reduce the relative error to about 10^{-6}.

Let us now test BCDMM on the DR problem described in (1.5). Suppose that there are up to 3000 users in the system with each user having 4 controllable appliances; also assume that each day is divided into 96 time periods. That is, $m = 96$ and $n_k = 96 \times 4$. The load model is generated according to [5]. The interested readers are referred to [3] for detailed modeling on the construction of the

[1] We use a PC with 128 GB RAM and 24 Intel Xeon 2.67 GHz cores.

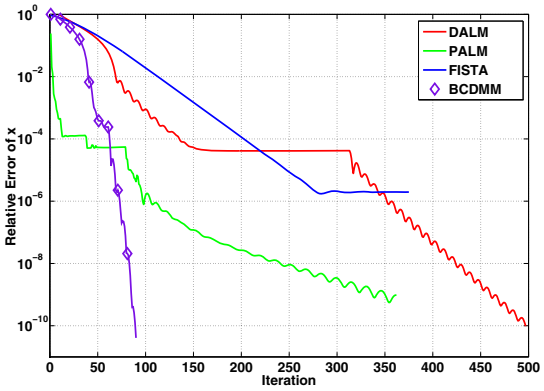

Fig. 1. Relative error performance for all algorithms on a small-size problem. $n = 10000$, $m = 3000$, $p = 0.06$. The relative error is given by $= \|\bar{x} - x^r\|/\|\bar{x}\|$.

Algorithm	K=50	K=100	K=500	K=1000	K=3000
BCDMM	0.4860	0.8099	3.3964	4.648	14.827
SG	0.9519	1.5630	9.4835	16.595	60.896
Unscheduled	1.0404	1.7940	7.5749	14.389	45.900

Table 3. Total Cost Performance of Different Approaches (10^3 unit price).

matrices $\{\mathbf{\Psi}_k\}_{k=1}^K$. For simplicity, we assume that the day-ahead bidding is completed, with power supply $\mathbf{p}$ determined by an average of 5 random generation of all the uncontrolled consumptions of the users. This reduces problem (1.5) to having only $\{\mathbf{x}_k\}_{k=1}^K$ and $\mathbf{z}$ as optimization variables. Additionally, we let $C_p(\cdot)$ and $C_s(\cdot)$ be the quadratic costs.

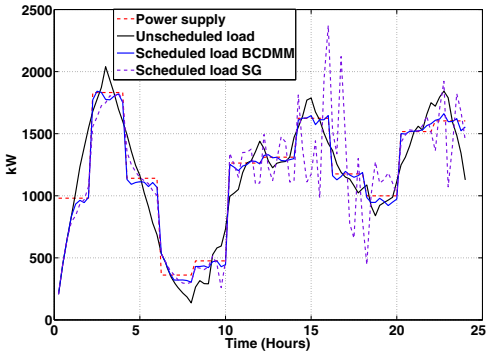

Fig. 2. The unscheduled consumption, power supply and the scheduled consumption by BCDMM and subgradient algorithm.

We compare our proposed algorithm with the dual subgradient (SG) algorithm [3][2]. We let both algorithms run 200 iterations. Note that each iteration SG is computationally more expensive, as it involves in solving a linear program [3], while each iteration of the BCDMM is again in closed-form. In Table 3, we compare the total costs incurred by the BCDMM and SG with that of unscheduled loads. Clearly the BCDMM is able to achieve about 50% of cost reduction, while the SG algorithm fails to converge within 200 iterations, thus results in significantly larger costs. In Fig. 2, we show the unscheduled consumption, power supply and the scheduled consumption for BCDMM and SG. The BCDMM can track the supply curve well, while the SG fails to do so within 200 iterations.

[2] Note that here the dual SG is applied to the DR with quadratic costs, which is different than [3], where it is applied to a problem with *linear* costs.

6. REFERENCES

[1] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[2] M. Alizadeh, X. Li, Z. Wang, A. Scaglione, and R. Melton, "Demand side management in the smart grid: Information processing for the power switch," *IEEE Signal Processing Magazine*, vol. 59, no. 5, pp. 55–67, 2012.

[3] T.-H. Chang, M. Alizadeh, and A. Scaglione, "Coordinated home energy management for real-time power balancing," in *Proc. IEEE PES General Meeting*, July 2012, pp. 1 –8.

[4] N. Li, L. Chen, and S. H. Low, "Optimal demand response based on utility maximization in power networks," in *IEEE PES General Meeting*, 2011, pp. 1–8.

[5] J. V. Paatero and P. D. Lund, "A model for generating household electricity load profiles," *International Journal on Energy Research*, vol. 20, pp. 273–290, 2006.

[6] FCC, "In the matter of facilitating opportunisties for flexible, efficient and reliable spectrum use emplying Coginitive Raido technologies," Dec 2003, ET Docket No. 03-108.

[7] M. Hong and A. Garcia, "Equilibrium pricing of interference in cognitive radio networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 12, pp. 6058 –6072, 2011.

[8] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1126–1153.

[9] A. Saha and A. Tewari, "On the nonaymptotic convergence of cyclic coordinate descent method," *SIAM Journal on Optimization*, vol. 23, no. 1, pp. 576–601, 2013.

[10] S. Shalev-Shwartz and A. Tewari, "Stochastic methods for ℓ_1 regularized loss minimization," *Journal of Machine Learning Research*, vol. 12, pp. 1865–1892, 2011.

[11] Y. Nesterov, "Efficiency of coordiate descent methods on huge-scale optimization problems," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 341–362, 2012.

[12] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *Journal of Optimization Theory and Applications*, vol. 103, no. 9, pp. 475–494, 2001.

[13] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*, Athena Scientific, Belmont, MA, 1996.

[14] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods, 2nd ed*, Athena Scientific, Belmont, MA, 1997.

[15] L. Grippo and M. Sciandrone, "On the convergence of the block nonlinear Gauss-Seidel method under convex constraints," *Operations Research Letters*, vol. 26, pp. 127–136, 2000.

[16] P. Tseng and S. Yun, "A coordinate gradient descent method for nonsmooth separable minimization," *Mathematical Programming*, vol. 117, pp. 387–423, 2009.

[17] H. Zhang, J. Jiang, and Z.-Q. Luo, "On the linear convergence of a proximal gradient method for a class of nonsmooth convex minimization problems," *Journal of the Operations Research Society of China*, vol. 1, no. 2, pp. 163–186, 2013.

[18] A. Beck and L. Tetruashvili, "On the convergence of block coordinate descent type methods," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2037–2060, 2013.

[19] M. Hong, X. Wang, M. Razaviyayn, and Z.-Q. Luo, "Iteration complexity analysis of block coordinate descent methods," 2013, Preprint, available online arXiv:1310.6957.

[20] M. Solodov, "On the convergence of constrained parallel variable distribution algorithms," *SIAM Journal on Optimization*, vol. 8, no. 1, pp. 187–196, 1998.

[21] 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*, vol. 3, no. 1, pp. 1–122, 2011.

[22] J. Eckstein and D. P. Bertsekas, "On the douglas-rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, no. 1, pp. 293–318, 1992.

[23] S.J. Wright, R.D. Nowak, and M.A.T. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2479 –2493, 2009.

[24] E. Hale, W. Yin, and Y. Zhang, "Fixed-point continuation for ℓ_1-minimization: Methodology and convergence," *SIAM Journal on Optimization*, vol. 19, no. 3, pp. 1107–1130, 2008.

[25] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imgaging Science*, vol. 2, no. 1, pp. 183–202, 2009.

[26] D. Goldfarb, S. Ma, and K. Scheinberg, "Fast alternating linearization methods for minimizing the sum of two convex functions," *Mathematical Programming*, vol. 141, no. 1-2, pp. 349–382, 2012.

[27] P. Richtarik and M. Takac, "Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function," *Mathematical Programming*, pp. 1–38, 2012.

[28] X. Wang and X. Yuan, "The linearized alternating direction method of multipliers for dantzig selector," *SIAM Journal on Scientific Computing*, vol. 34, no. 5, pp. 2792–2811, 2012.

[29] X. Zhang, M. Burger, and S. Osher, "A unified primal-dual algorithm framework based on bregman iteration," *Journal of Scientific Computing*, vol. 46, no. 1, pp. 20–46, 2011.

[30] P. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, Springer Optimization and Its Applications, pp. 185–212. Springer New York, 2011.

[31] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[32] M. Hong, X. Wang T.-H. Chang, M. Razaviyayn, S. Ma, and Z.-Q. Luo, "A block successive upper bound minimization method of multipliers for linearly constrained convex optimization," 2013, Preprint, available online arXiv:1401.7079.

[33] A. Y. Yang, S.S. Sastry, A. Ganesh, and Y. Ma, "Fast l1-minimization algorithms and an application in robust face recognition: A review," in *IEEE International Conference on Image Processing (ICIP)*, 2010, pp. 1849–1852, online at http://www.eecs.berkeley.edu/ yang/software/l1benchmark/.

[34] J. Yang and Y. Zhang, "Alternating direction algorithms for l1-problems in compressive sensing," *SIAM Journal on Scientific Computing*, pp. 250–278, 2011.