European Journal of Operational Research 231 (2013) 274–281

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Continuous Optimization

On the convergence of inexact block coordinate descent methods for constrained optimization

A. Cassioli [1], D. Di Lorenzo, M. Sciandrone *

Università degli Studi di Firenze, Dipartimento di Ingegneria dell'Informazione, Via di S. Marta 3, 50145 Firenze, Italy

ARTICLE INFO

Article history:
Received 15 November 2012
Accepted 30 May 2013
Available online 10 June 2013

Keywords:
Nonlinear programming
Block coordinate descent methods
Inexact decomposition methods
Gradient projection
Frank–Wolfe

ABSTRACT

We consider the problem of minimizing a smooth function over a feasible set defined as the Cartesian product of convex compact sets. We assume that the dimension of each factor set is huge, so we are interested in studying inexact block coordinate descent methods (possibly combined with column generation strategies). We define a general decomposition framework where different line search based methods can be embedded, and we state global convergence results. Specific decomposition methods based on gradient projection and Frank–Wolfe algorithms are derived from the proposed framework. The numerical results of computational experiments performed on network assignment problems are reported.

1. Introduction

Let us consider the problem

$$\min_{x} \quad f(x), \qquad \text{s.t.} \quad x \in \mathcal{F} = \mathcal{F}_1 \times \mathcal{F}_2 \times \cdots \times \mathcal{F}_L \subset \mathbb{R}^n, \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function, $\mathcal{F}_h \subseteq \mathbb{R}^{n_h}$, $h \in \{1,\ldots,L\}$, are compact convex sets, and $n_1 + \cdots + n_h + \cdots + n_L = n$. Accordingly to the structure of the feasible set we partition the vector of variables as

$$x = (x_{(1)},\ldots,x_{(h)},\ldots,x_{(L)})^T,$$

where $x_{(h)} \in \mathbb{R}^{n_h}$, $h \in \{1,\ldots,L\}$, is the hth block component.

We assume that the dimension n_h of each block h is extremely large, so we are interested in studying decomposition-based methods.

In some cases, related to the structure of the objective function and/or to the structure of the factor sets $\mathcal{F}_h$, it could be convenient to sequentially operate on the block components $x_{(h)}$. With this in mind, in this work we focus on the class of block coordinate descent methods. In this context, the Gauss–Seidel algorithm is one of the most popular method: at each iteration k, given the current point x^k, the block components $x_{(h)}$ are sequentially updated by exactly solving the corresponding subproblems, that is, by setting for $h \in \{1,\ldots,L\}$

$$x_{(h)}^{k+1} \in \arg\min_{x_{(h)} \in \mathcal{F}_h} f\left(x_{(1)}^{k+1},\ldots,x_{(h-1)}^{k+1},x_{(h)},x_{(h+1)}^{k},\ldots,x_{(L)}^{k}\right). \tag{2}$$

The literature on the convergence of exact decomposition algorithms (as the above Gauss–Seidel algorithm) is wide (see, e.g., [3,9,13,15]). Note that the computation of the exact solutions of the generated subproblems may be expensive whenever these latter do not have a particular structure. This has motivated several studies on inexact decomposition methods [14,21,4,17–19,23,20].

We observe that in some applications (for instance, like those concerning network equilibrium problems [7]) it is convenient to work on a lower-dimensional subset $\mathcal{F}_h(x^k)$ of $\mathcal{F}_h$, that in general depends on the current feasible iterate x^k. This will allow to solve problems in which the number of variables in each block is so huge that it is not reasonable to completely enumerate them a priori. In these cases it is suitable to adopt a column generation strategy, for which only the variables needed to reach optimality are iteratively added. Another case that can greatly benefit from this strategy is when a restriction of the feasible set can give some kind of computational advantages, as for instance the possibility to compute efficiently the objective function.

On these bases we can define, for instance, a Gauss–Seidel decomposition method operating on the restrictions of the factor sets $\mathcal{F}_h$. Formally, the updating rule (2) could be modified as follows:

$$x_{(h)}^{k+1} \in \arg\min_{x_{(h)} \in \mathcal{F}_h(x^k)} f\left(x_{(1)}^{k+1},\ldots,x_{(h)},\ldots,x_{(L)}^{k}\right). \tag{3}$$

* Corresponding author. Tel.: +39 0554796460.

E-mail addresses: cassioli@lix.polytechnique.fr (A. Cassioli), dilorenzo@dsi.unifi.it (D. Di Lorenzo), sciandro@dsi.unifi.it (M. Sciandrone).

[1] Present address: LIX, École Polytechnique, Route de Saclay, 91128 Palaiseau, France.

0377-2217/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2013.05.049

A key issue regards the properties of the sets $\mathcal{F}_h(x^k)$ (replacing $\mathcal{F}_h$) needed to ensure convergence to the above scheme based, as the standard Gauss–Seidel method, on exact minimizations. Furthermore, for the reasons already explained, we are interested in designing decomposition methods that do not necessarily require the computation of the exact solutions of the generated subproblems.

We will define a general decomposition framework with guaranteed theoretical convergence properties, such that:

(i) it is possible to consider restrictions of the factor sets $\mathcal{F}_h$ in order to manage the huge number of variables n_h;
(ii) the generated subproblems can be inexactly solved whenever the computation of their exact solutions is too expensive;
(iii) there is some degree of freedom in the selection of the blocks.

The above points distinguish the present work that of some cited papers. Convergent inexact block-descent methods for smooth problems are proposed in [4,18]. This latter work, as well as [17], presents also convergence rates results of the defined algorithms. The very recent paper [19] provides a unified convergence analysis for a general class of inexact block-descent methods even for nonsmooth problems. Various types of updating rules can be embedded in the presented general framework, including the cyclic updating rule, the Gauss–Southwell update rule or the overlapping essentially cyclic update rule. However, the general inexact decomposition schemes proposed in [4,18,19] do not admit the possibility of operating on restrictions of the factor sets, thus precluding the possibility of employing column generation strategies.

Inexact decomposition algorithms for specific (linearly constrained) problems are presented in [14,21,23,20], but, differently from our general framework, they require specific assumptions on the factor sets and on the objective function.

Concerning point (i), we state some properties to ensure that the restrictions of the factor sets yield a suitable representation of these latter in a lower dimensional subspace. Regarding points (ii) and (iii), we introduce general conditions on the block selection, and on the iterative mappings operating on the restrictions of the factor sets to guarantee global convergence properties. As a result, we present a general inexact decomposition framework whose key elements are the definition of lower dimensional restrictions of the factor sets, and the employment of line search based iterative minimization mappings defined on the introduced restrictions of the factor sets. Starting from the decomposition framework, specific convergent algorithms, based on gradient projection and Frank–Wolfe directions are designed. Summarizing, the main aim of the paper is to develop a unifying global convergence theory for inexact block coordinate descent methods possibly applied in connection with column generation strategies.

The paper is organized as follows. The decomposition framework is presented in Section 2, as well as its convergence properties and the required assumptions. In Section 3 we describe two known line search mappings, the standard Armijo-type line search and a derivative-free line search, and we recall their theoretical properties. Two inexact decomposition methods, based on gradient projection and Frank–Wolfe directions, are defined in Section 4. In Section 5 we analyze a wide class of problems that fulfill the assumptions required for the convergence of the proposed framework. Finally, in Section 6 we report the numerical results of computational experiments performed on network assignment problems.

Notation. We suppose that the vector $x \in \mathbb{R}^n$ is partitioned into component vectors $x_{(h)} \in \mathbb{R}^{n_h}$. Note that we use bracketed subscripts to denote a subvector. An additional subscript is used to identify a variable of a specific block, e.g. we denote by $x_{(h),\ i}$ the variable i of block h. The partial gradient of f with respect to $x_{(h)}$, evaluated at x, is indicated by $\nabla_{(h)} f(x) \in \mathbb{R}^{n_h}$. A *critical point* for problem (1) is a point $\bar{x} \in \mathcal{F}$ such that $\nabla f(\bar{x})^T(x - \bar{x}) \geqslant 0$ for every $x \in \mathcal{F}$, where $\nabla f(x) \in \mathbb{R}^n$ denotes the gradient of f at x. Finally, we indicate by $\|\cdot\|$ the Euclidean norm (on the appropriate space).

2. An inexact decomposition-based framework

In this section we propose an inexact decomposition framework in which at each iteration a block of variables is chosen and the corresponding sub-problem is inexactly solved. The key issues of the proposed decomposition framework are the introduction of suitable low dimensional restrictions of the sets whose Cartesian product yields the feasible set, and the employment of block descent methods to inexactly solve the generated subproblems. First we formally define the restriction of a set $\mathcal{F}_h$.

Definition 1. Given a point $y \in \mathcal{F}$, for $h \in \{1,\ldots,L\}$, we denote by $\mathcal{F}_h(y)$ the **restriction** of $\mathcal{F}_h$ at y, a closed convex set such that $\mathcal{F}_h(y) \subseteq \mathcal{F}_h$.

The proposed algorithm, named IDA, is depicted in Algorithm 1.

```
Algorithm 1: Inexact Decomposition Algorithm (IDA)
Input: $x^0 \in \mathcal{F}$
$k \leftarrow 0$;
while stopping criterion is not fulfilled do
    choose $h^k \in \{1,\ldots,L\}$;
    define $\mathcal{F}_{h^k}(x^k)$;
    define a feasible descent direction $d^k$ (depending on $\mathcal{F}_{h^k}(x^k)$) such
    that $d^k_{(h)} = 0_{(h)}$ for any $h \neq h^k$;
    compute $\alpha^k$ by means of a suitable line search along $d^k$;
    $x^{k+1} \leftarrow x^k + \alpha^k d^k$;
    $k \leftarrow k + 1$;
end
```

In order to perform the convergence analysis of Algorithm IDA, we need to introduce suitable assumptions on the sets $\mathcal{F}_h(\cdot)$, on the block selection, on the search direction d^k, and on the line search procedure that computes the stepsize α^k.

The first assumption is on the properties of the sets $\mathcal{F}_h(\cdot)$ for $h \in \{1,\ldots,L\}$.

Assumption 1.

(i) The number of possible restricted sets $\mathcal{F}_h(x)$, with x varying in $\mathcal{F}$, is finite.
(ii) Let $K \subseteq \mathbb{N}$ be an infinite subsequence such that, for all $k \in K$, $x^k \in \mathcal{F}$ and

$$\mathcal{F}_h(x^k) = \mathcal{F}_h^\star \quad \forall k \in K.$$

Assume that $x^k \to \bar{x}$ for $k \in K$ and $k \to \infty$. If

$$\nabla_{(h)} f(\bar{x})^T (x_{(h)} - \bar{x}_{(h)}) \geqslant 0 \quad \forall x_{(h)} \in \mathcal{F}_h^\star,$$

then it holds

$$\nabla_{(h)} f(\bar{x})^T (x_{(h)} - \bar{x}_{(h)}) \geqslant 0 \quad \forall x_{(h)} \in \mathcal{F}_h.$$

We observe that (i) of Assumption 1 is needed to ensure that along an infinite sequence of iterates it is possible to extract a subsequence for which the same restriction of the factor set is selected. Condition (ii) requires that the restriction $\mathcal{F}_h(x^k)$ captures, in the limit, the geometry of the set $\mathcal{F}_h$ in terms of optimality conditions. Examples of restrictions satisfying Assumption 1 are defined in Section 5 with reference to a general class of linearly constrained problems (see Proposition 8).

A second assumption requires that each block component $x_{(l)}$ for $l \in \{1,\ldots,L\}$ is periodically considered at Step 3 within a prefixed maximum number of iterations.

Assumption 2. There exists an integer $M > 0$ such that, for all $k \geqslant 0$ and for all $l \in \{1,\ldots,L\}$, we can find an index $l(k)$, with $0 \leqslant l(k) \leqslant M$, such that at Step 1 we have $h^{k+l(k)} = l$.

The next assumption needs to be satisfied by the line search mapping, and by the choice of the search direction.

Assumption 3. At every iteration k, the line search procedure computes a value of α^k such that

$$f(x^k + \alpha^k d^k) \leqslant f(x^k).$$

Furthermore, if $\{x^k\}$ is a sequence of feasible points convergent to a point $\bar{x}$ and

$$\lim_{k\to\infty}(f(x^k) - f(x^k + \alpha^k d^k)) = 0, \tag{4}$$

then we have

$$\lim_{k\to\infty} \nabla f(x^k)^T d^k = 0, \quad \lim_{k\to\infty}\|\alpha^k d^k\| = 0. \tag{5}$$

Line search mappings are described and analyzed in the appendix. We remark that the condition (5) stated in Assumption 3 requires, as usual in the context of decomposition methods, that the distance $\|x^{k+1} - x^k\|$ tends to zero. Indeed, at each iteration, a single block of variables is updated, so that, to attain convergence it may be necessary that "consecutive" points $x^k, x^{k+1}, x^{k+2}, \ldots,$ tend to the same limit point.

This requirement can be satisfied by an iteration of the form

$$x^{k+1} = x^k + \alpha^k d^k,$$

where α^k is determined by an Armijo-type line search (see Section 3.1) provided, for instance, that d^k is obtained by a gradient projection. In other cases, for instance, whenever d^k is a Frank–Wolfe-type direction, the adoption of an Armijo-type line search is not sufficient to guarantee that the distance between successive points tends to zero. This motivates the adoption of a line search (see Section 3.2) based on an acceptance condition different from that of Armijo's rule.

Finally, we state the following assumption on the search direction.

Assumption 4. Let $\{x^k\}_K$ be a subsequence of feasible points convergent to a point $x^\star$, and such that

$$\mathcal{F}_{h^k}(x^k) = \mathcal{F}_h^* \quad \forall k \in K. \tag{6}$$

We have that

$$\lim_{k\in K, k\to\infty} \nabla f(x^k)^T d^k = 0, \tag{7}$$

implies that

$$\nabla_{(h)} f(x^\star)^T \left(x_{(h)} - x^\star_{(h)}\right) \geqslant 0 \quad \forall x_{(h)} \in \mathcal{F}^\star_h. \tag{8}$$

We are ready to state global convergence properties of IDA. Note that the following theoretical result is a generalized convergence theorem that follows directly from the stated assumptions, and can be used to design algorithmic strategies for specific classes of problems whose convergence can be proved by satisfying the required assumptions.

Proposition 1. *Let $\{x^k\}$ be the sequence generated by* IDA. *Suppose that Assumptions 1–4 are satisfied. Then $\{x^k\}$ admits limit points and each limit point is a critical point for problem (1).*

Proof. The sequence $\{x^k\}$ belongs to the feasible compact set, so $\{x^k\}$ admits limit points. Let $x^\star$ be a limit point of $\{x^k\}$, i.e., there exists an infinite subset $K \subseteq \mathbb{N}$ such that

$$\lim_{k\in K, k\to\infty} x^k = x^\star. \tag{9}$$

By Assumption 3 it holds $f(x^{k+1}) \leqslant f(x^k)$ so that, as f is bounded below, we can write

$$\lim_{k\to\infty} f(x^{k+1}) - f(x^k) = 0. \tag{10}$$

Using (10) and (5) of Assumption 3 we obtain

$$\lim_{k\in K, k\to\infty} \nabla f(x^k)^T d^k = 0. \tag{11}$$

From (5) of Assumption 3 it follows:

$$\lim_{k\in K, k\to\infty} \|\alpha^k d^k\| = \lim_{k\in K, k\to\infty} \|x^{k+1} - x^k\| = 0. \tag{12}$$

Then, by induction, we can write for any $l \in \mathbb{N}$,

$$\lim_{k\in K, k\to\infty} x^{k+l} = x^\star, \tag{13}$$

$$\lim_{k\in K, k\to\infty} \nabla f(x^{k+l})^T d^{k+l} = 0. \tag{14}$$

From (i) of Assumption 1 it follows that, for every $j \in \{1,\ldots,L\}$, we can find an infinite subset $K_j \subseteq K$ and a set $\mathcal{F}^\star_j$ such that

$$\mathcal{F}_j(x^k) = \mathcal{F}^\star_j \quad \forall k \in K_j.$$

Recalling Assumption 2 we have that there exists an index $j(k)$, with $0 \leqslant j(k) \leqslant M$, such that $h^{k+j(k)} = j$, so that, using (13) and (14) we can write

$$\lim_{k\in K_j, k\to\infty} x^{k+j(k)} = x^\star, \tag{15}$$

$$\lim_{k\in K_j, k\to\infty} \nabla f(x^{k+j(k)})^T d^{k+j(k)} = 0. \tag{16}$$

From (15) and (16) and Assumption 4 we have

$$\nabla_{(j)} f(x^\star)^T \left(x_{(j)} - x^\star_{(j)}\right) \geqslant 0 \quad \forall x_{(j)} \in \mathcal{F}^\star_j. \tag{17}$$

Taking into account (17) and recalling (ii) of Assumption 1 we have

$$\nabla_{(j)} f(x^\star)^T \left(x_{(j)} - x^\star_{(j)}\right) \geqslant 0 \quad \forall x_{(j)} \in \mathcal{F}_j.$$

This equation holds for every $j \in \{1,\ldots,L\}$, and hence the proposition is proved. □

3. Line search mappings

In this section we describe the well-known Armijo-type line search algorithm, and a line search previously introduced in the context of decomposition methods for unconstrained optimization [8]. We state the convergence properties of the described line search mappings.

3.1. The Armijo line search

Let $d^k \in \mathbb{R}^n$ be a feasible direction at $x^k \in \mathcal{F}$. Let β^k be the maximum feasible step length along d^k. Taking, for instance, $\hat{x}^k \in \mathcal{F}$ and $d^k = \hat{x}^k - x^k \neq 0$, it follows $\beta^k \geqslant 1$.

Assumption 5. Assume that $\{d^k\}$ is a sequence of feasible search directions such that

(a) for all k we have $\|d^k\| \leqslant M$ for a given number $M > 0$;
(b) for all k we have $\nabla f(x^k)^T d^k < 0$.

An Armijo-type line search algorithm and its properties are described below (see, e.g., [3]).

```
Algorithm 2: Armijo-type line search ALS
  Input: $x^k, d^k, \beta^k$
  Output: $\alpha^k$
select $\lambda > 0$, $\delta \in (0,1)$, $\gamma \in (0,1/2)$;
$\alpha^k \leftarrow \min\{\beta^k, \lambda\}$;
while $f(x^k + \alpha^k d^k) > f(x^k) + \gamma\alpha^k \nabla f(x^k)^T d^k$ do
    $\alpha^k \leftarrow \delta\alpha^k$;
end
return $\alpha^k$
```

Proposition 2. *Let $\{x^k\}$ be a sequence of points belonging to the feasible set $\mathcal{F}$, and let $\{d^k\}$ be a sequence of search directions satisfying Assumption 5. Then:*

(i) *Algorithm* ALS *determines, in a finite number of iterations, a scalar α^k such that*

$$f(x^k + \alpha^k d^k) \leqslant f(x^k) + \gamma\alpha^k \nabla f(x^k)^T d^k; \tag{18}$$

(ii) *if $\{x^k\}$ converges to $\bar{x}$ and*

$$\lim_{k\to\infty}(f(x^k) - f(x^k + \alpha^k d^k)) = 0, \tag{19}$$

then we have

$$\lim_{k\to\infty} \beta^k \nabla f(x^k)^T d^k = 0. \tag{20}$$

3.2. A quadratic line search

The properties of an Armijo-type line search stated in Proposition 2 do not guarantee, without further assumptions on the search direction d^k, that the distance between successive points tends to zero (which is a usual requirement, as discussed in Section 2, of decomposition methods). Such a property can be satisfied by the line search algorithm described below and based on an acceptance condition

$$f(x^k + \alpha^k d^k) \leqslant f(x^k) - \gamma(\alpha^k\|d^k\|)^2,$$

replacing the Armijo's condition

$$f(x^k + \alpha^k d^k) \leqslant f(x^k) + \gamma\alpha^k \nabla f(x^k)^T d^k.$$

```
Algorithm 3: Quadratic Line Search QLS
  Input: $x^k, d^k, \beta^k$
  Output: $\alpha^k$
select $\lambda > 0$, $\delta \in (0,1)$, $\gamma > 0$;
$\alpha^k \leftarrow \min\{\beta^k, \lambda\}$;
while $f(x^k + \alpha^k d^k) > f(x^k) - \gamma(\alpha^k)^2\|d^k\|^2$ do
    $\alpha^k \leftarrow \delta\alpha^k$;
end
return $\alpha^k$
```

The properties of Algorithm QLS are stated in the next proposition (whose proof can be derived, with minor modifications, from the one of Proposition 4.2 in [8]).

Proposition 3. *Let $\{x^k\}$ be a sequence of points belonging to the feasible set $\mathcal{F}$, and let $\{d^k\}$ be a sequence of search directions satisfying Assumption 5. Then:*

(i) *Algorithm* QLS *determines, in a finite number of iterations, a scalar α^k such that*

$$f(x^k + \alpha^k d^k) \leqslant f(x^k) - \gamma(\alpha^k\|d^k\|)^2; \tag{21}$$

(ii) *if $\{x^k\}$ converges to $\bar{x}$ and*

$$\lim_{k\to\infty}(f(x^k) - f(x^k + \alpha^k d^k)) = 0, \tag{22}$$

then we have

$$\lim_{k\to\infty} \alpha^k\|d^k\| = 0, \quad \lim_{k\to\infty} \beta^k \nabla f(x^k)^T d^k = 0. \tag{23}$$

4. Convergent block-descent methods

In this section we present two decomposition algorithms derived from the IDA framework. The two algorithms are based on gradient projection and Frank–Wolfe directions respectively, and on the adoption of suitable line searches. In both algorithms we implicitly assume that, if $\nabla f(x^k)^T d^k = 0$, then we consider a null step along the search direction, that is $\alpha^k = 0$.

4.1. Gradient projection-based IDA

The gradient projection-based algorithm is formally defined by Algorithm 4. We denote by $P_{\mathcal{F}_{h^k}(x^k)}\left[x^k_{(h)} - \nabla_{(h)} f(x^k)\right]$ the projection of the point $x^k_{(h)} - \nabla_{(h)} f(x^k)$ onto the set $\mathcal{F}_{h^k}(x^k)$.

```
Algorithm 4: Gradient Projection-IDA (GP-IDA)
  Input: $x^0 \in \mathcal{F}$
$k \leftarrow 0$;
while stopping criterion is not fulfilled do
    choose $h^k \in \{1, \ldots, L\}$;
    define $\mathcal{F}_{h^k}(x^k)$;
    foreach $i \in \{1, \ldots, L\} : i \neq h^k$ do
        $d^k_{(i)} = 0_{(i)}$;
    end
    $d^k_{(h^k)} \leftarrow P_{\mathcal{F}_{h^k}(x^k)}[x^k_{(h)} - \nabla_{(h)} f(x^k)] - x^k_{(h^k)}$;
    compute $\alpha^k$ by means of the Armijo line search along $d^k$;
    $x^{k+1} \leftarrow x^k + \alpha^k d^k$;
    $k \leftarrow k + 1$;
end
```

The global convergence of Algorithm 4 is stated in Proposition 4.

Proposition 4. *Let $\{x^k\}$ be the sequence generated by* GP-IDA. *Suppose that Assumptions 1 and 2 are satisfied. Then $\{x^k\}$ admits limit points and each limit point is a critical point for problem (1).*

Proof. We will show that the search direction d^k and the stepsize α^k are such that Assumptions 3, 4 hold, so that the thesis follows from Proposition 1. Hence, we assume that there exists an infinite subset K such that

$$\lim_{k\in K, k\to\infty} x^k = x^\star, \tag{24}$$

$$\lim_{k\in K, k\to\infty} (f(x^k + \alpha^k d^k) - f(x^k)) = 0. \tag{25}$$

From the properties of the projection mapping we get

$$\nabla f(x^k)^T d^k = \nabla_{(h^k)} f(x^k)^T d^k_{(h^k)} \leqslant -\|\hat{x}^k_{(h^k)} - x^k_{(h^k)}\|^2 \leqslant 0, \tag{26}$$

where

$$\hat{x}^k_{(h^k)} = P_{\mathcal{F}_{h^k}(x^k)}\left[x^k_{(h^k)} - \nabla_{(h^k)} f(x^k)\right]. \tag{27}$$

For all k we have either

$$\nabla f(x^k)^T d^k < 0 \quad \text{and} \quad \alpha^k > 0,$$

where α^k is determined by means of the Armijo line search along the search direction d^k, or

$$\nabla f(x^k)^T d^k = 0 \quad \text{and} \quad \alpha^k = 0. \tag{28}$$

Note that, due to the convexity of $\mathcal{F}_{h^k}(x^k)$, the maximum feasible step length β^k along d^k is greater than or equal to 1. Furthermore, as the closed convex set $\mathcal{F}_{h^k}(x^k)$ belongs, by assumption, to the compact set $\mathcal{F}$, we have that the search direction d^k is bounded.

Using (25) and assertion (ii) of Proposition 2, taking into account (28), we obtain

$$\lim_{k\in K, k\to\infty} \nabla f(x^k)^T d^k = 0. \tag{29}$$

From (29) and (26) it follows

$$\lim_{k\in K, k\to\infty} \|\hat{x}^k_{(h^k)} - x^k_{(h^k)}\| = \lim_{k\to\infty} \|d^k\| = 0, \tag{30}$$

and hence, as α^k is bounded above, we can write

$$\lim_{k\in K, k\to\infty} \|x^{k+1} - x^k\| = \lim_{k\in K, k\to\infty} \|\alpha^k d^k\| = 0. \tag{31}$$

Thus, (29) and (31) imply that Assumption 3 holds. Now let $K_h \subset K$ be an infinite subset such that

$$\mathcal{F}_{h^k}(x^k) = \mathcal{F}^\star_h \quad \forall k \in K_h.$$

From (30), recalling the continuity of the projection mapping, we obtain

$$x^\star_{(h)} = P_{\mathcal{F}^\star_h}\left[x^\star_{(h)} - \nabla_{(h)} f(x^\star)\right], \tag{32}$$

which implies that

$$\nabla_{(h)} f(x^\star)^T \left(x_{(h)} - x^\star_{(h)}\right) \geqslant 0 \quad \forall x_{(h)} \in \mathcal{F}^\star_h. \tag{33}$$

Thus, Assumption 4 is satisfied, and this concludes the proof. □

Note that the convergence analysis of GP-IDA cannot be derived from results stated in [4], since GP-IDA involves restrictions $\mathcal{F}_h(x^k)$ of the prefixed subsets $\mathcal{F}_h$.

4.2. Frank–Wolfe-based IDA

The decomposition method based on the Frank–Wolfe direction is described in Algorithm 5.

```
Algorithm 5: Frank-Wolfe-IDA (FW-IDA)
Input: $x^0 \in \mathcal{F}$
$k \leftarrow 0$;
while stopping criterion is not fulfilled do
    choose $h^k \in \{1, \dots, L\}$;
    define $\mathcal{F}_{h^k}(x^k)$;
    foreach $i \in \{1, \dots, L\} : i \neq h^k$ do
        $d^k_{(i)} = 0_{(i)}$;
    end
    let $\hat{x}^k_{(h^k)} \in \arg\min_{x_{(h^k)} \in \mathcal{F}_{h^k}(x^k)} \nabla_{(h^k)} f(x^k)^T x_{(h^k)}$;
    $d^k_{(h^k)} \leftarrow \hat{x}^k_{(h^k)} - x^k_{(h^k)}$;
    compute $\alpha^k$ by means of the quadratic line search along $d^k$;
    $x^{k+1} \leftarrow x^k + \alpha^k d^k$;
    $k \leftarrow k + 1$;
end
```

The global convergence of Algorithm 5 is established in Proposition 5.

Proposition 5. *Let $\{x^k\}$ be the sequence generated by* FW-IDA. *Suppose that* Assumptions 1 and 2 *are satisfied. Then $\{x^k\}$ admits limit points and each limit point is a critical point for problem* (1).

Proof. We will show that the search direction d^k and the stepsize α^k are such that Assumptions 3 and 4 hold, so that the thesis follows from Proposition 1. Hence, we assume that there exists an infinite subset K such that

$$\lim_{k\in K, k\to\infty} x^k = x^\star, \tag{34}$$

$$\lim_{k\in K, k\to\infty} (f(x^k + \alpha^k d^k) - f(x^k)) = 0. \tag{35}$$

For all k we have either

$$\nabla f(x^k)^T d^k < 0 \quad \text{and} \quad \alpha^k > 0,$$

where α^k is determined by means of the quadratic line search along the search direction d^k, or

$$\nabla f(x^k)^T d^k = 0 \quad \text{and} \quad \alpha^k = 0. \tag{36}$$

Note that, due to the convexity of $\mathcal{F}_{h^k}(x^k)$, the maximum feasible step length β^k along d^k is greater than or equal to 1. Furthermore, as the closed convex set $\mathcal{F}_{h^k}(x^k)$ belongs, by assumption, to the compact set $\mathcal{F}$, we have that the search direction d^k is bounded. Using (35) and Proposition 3 (reported in the appendix), and taking into account (36) we obtain

$$\lim_{\substack{k\in K\\ k\to\infty}} \nabla f(x^k)^T d^k = 0, \quad \lim_{\substack{k\in K\\ k\to\infty}} \|\alpha^k d^k\| = 0, \tag{37}$$

which implies Assumption 3.

Let $K_h \subset K$ be an infinite subset such that

$$\mathcal{F}_{h^k}(x^k) = \mathcal{F}^\star_h \quad \forall k \in K_h.$$

The direction d^k is such that, for every $x_{(h)} \in \mathcal{F}^\star_h$

$$\nabla f(x^k)^T d^k \leqslant \nabla_{(h)} f(x^k)^T \left(x_{(h)} - x^k_{(h)}\right). \tag{38}$$

From (34), (37) and (38), recalling the continuity of the gradient, we obtain

$$\nabla_{(h)} f(x^\star)^T \left(x_{(h)} - x^\star_{(h)}\right) \geqslant 0 \quad \forall x_{(h)} \in \mathcal{F}^\star_h. \tag{39}$$

Thus, Assumption 4 is satisfied, and this concludes the proof. □

We remark that global convergence results of inexact block coordinate descent methods (in the general case of nonconvex objective functions) based on Frank–Wolfe iterations were not known. A randomized block-coordinate variant of the classic Frank–Wolfe algorithm for convex optimization with block-separable constraints has been presented in [12], where convergence rates results in the duality gap have been established.

5. Problems with a single equality constraint and box constraints

In this section we will show how a wide class of problems has a structure that allows restrictions on the factor sets which satisfy Assumption 1.

Let us consider problem (1) with

$$\mathcal{F}_h = \left\{x_{(h)} \in \mathbb{R}^{n_h} : \ a^T_{(h)} x_{(h)} = b_{(h)}, \ l_{(h)} \leqslant x_{(h)} \leqslant u_{(h)}\right\}, \tag{40}$$

where we assume that $a_{(h),\,i} \neq 0$ for $i \in \{1, \dots, n_h\}$. Problem (1) with factor sets $\mathcal{F}_h$ defined by (40) includes, for instance, Network Equilibrium (NE) problems [7], training problems of Support Vector Machine (SVM) [22,6], which represent an important tool both for classification and regression problems, portfolio selection problems [10], optimal control problems [2].

We first state the optimality conditions in a compact form. To this aim, given a point $x \in \mathcal{F}$, we define the index sets

$$
\begin{aligned}
L_h(x) &= \{i :\ x_{(h),i} = l_{(h),i}\},\\
L_h^-(x) &= \{i \in L_h(x) :\ a_{(h),i} < 0\},\\
L_h^+(x) &= \{i \in L_h(x) :\ a_{(h),i} > 0\},
\end{aligned}
$$

$$
\begin{aligned}
U_h(x) &= \{i :\ x_{(h),i} = u_{(h),i}\},\\
U_h^-(x) &= \{i \in U_h(x) :\ a_{(h),i} < 0\},\\
U_h^+(x) &= \{i \in U_h(x) :\ a_{(h),i} > 0\}.
\end{aligned}
$$

We introduce the following index sets

$$
\begin{aligned}
R_h(x) &= L_h^+(x) \cup U_h^-(x) \cup \{i :\ l_{(h),i} < x_{(h),i} < u_{(h),i}\},\\
S_h(x) &= L_h^-(x) \cup U_h^+(x) \cup \{i :\ l_{(h),i} < x_{(h),i} < u_{(h),i}\}.
\end{aligned}
$$

We recall the following results [14].

Proposition 6. *Let $\{x^k\}$ be a sequence of feasible points for problem (1), with factor sets $\mathcal{F}_h$ of the form (40), convergent to a point $\bar{x}$, and let $h \in \{1,\dots,L\}$. Then for sufficiently large values of k we have*

$$R_h(\bar{x}) \subseteq R_h(x^k) \quad S_h(\bar{x}) \subseteq S_h(x^k).$$

Proposition 7. *Let $\bar{x}$ be a feasible point for problem (1), with factor sets $\mathcal{F}_h$ of the form (40), and let $h \in \{1,\dots,L\}$. Then*

$$\nabla_{(h)} f(\bar{x})^T (x_{(h)} - \bar{x}_{(h)}) \geqslant 0 \quad \forall x_{(h)} \in \mathcal{F}_h,$$

if and only if

$$\max_{i \in R_h(\bar{x})} \left\{ -\frac{\nabla_{(h),i} f(\bar{x})}{a_{(h),i}} \right\} \leqslant \min_{j \in S_h(\bar{x})} \left\{ -\frac{\nabla_{(h),j} f(\bar{x})}{a_{(h),j}} \right\}. \tag{41}$$

Given a point $x^k \in \mathcal{F}$, let

$$
\begin{aligned}
I_h(x^k) &= \left\{ i \in \{1,\dots,n_h\} :\ i \in \arg\max_{i \in R_h(x^k)} -\frac{\nabla_{(h),i} f(x^k)}{a_{(h),i}} \right\},\\
J_h(x^k) &= \left\{ j \in \{1,\dots,n_h\} :\ j \in \arg\min_{j \in S_h(x^k)} -\frac{\nabla_{(h),j} f(x^k)}{a_{(h),j}} \right\}.
\end{aligned} \tag{42}
$$

Given $i_h^k \in I_h(x^k)$ and $j_h^k \in J_h(x^k)$, we define the *working set*

$$W_h(x^k) = i_h^k \cup j_h^k \cup (R_h(x^k) \cap S_h(x^k)), \tag{43}$$

and we introduce the restriction $\mathcal{F}_h(x^k)$ of the factor set $\mathcal{F}_h$ as follows

$$\mathcal{F}_h(x^k) = \{x_{(h)} \in \mathcal{F}_h,\ x_i = x_i^k\ \forall i \notin W_h(x^k)\}. \tag{44}$$

Note that the working set contains the pair $\left(i_h^k, j_h^k\right)$ that most violates the optimality conditions, and the indexes corresponding to all the variables with inactive bounds. The presence of the *maximal violating pair* allows us to satisfy requirement (ii) of Assumption 1, while the inclusion of all the variables with inactive bounds is needed to ensure that also requirement (i) is satisfied. Now we will show that Assumption 1 holds on the restriction $\mathcal{F}_h(x^k)$.

Proposition 8. *For any point $x \in \mathcal{F}$, let $\mathcal{F}_h(x)$ be the restriction of the feasible set $\mathcal{F}_h$ defined as in (44) (replacing x^k with x). Then Assumption 1 holds.*

Proof.

(i) Let $x \in \mathcal{F}$. Note that an index $i \in \{1,\dots,n_h\}$ does not belong to $W_h(x)$ only if either $x_{(h),\, i} = l_{(h),i}$ or $x_{(h),\, i} = u_{(h),\, i}$. Then we have

$$|\cup_{x \in \mathcal{F}} \{\mathcal{F}_h(x)\}| \leqslant 3^n.$$

(ii) Now let K be an infinite index subset such that $\mathcal{F}_h(x^k) = \mathcal{F}_h^\star$ for all $k \in K$. We can extract a further infinite subset, relabelled by K, such that

$$I_h(x^k) = I_h^\star \quad J_h(x^k) = J_h^\star \quad \forall k \in K,$$

where the index sets $I_h(x^k)$, $J_h(x^k)$ are defined in (42). If

$$\nabla_{(h)} f(\bar{x})^T (x_{(h)} - \bar{x}_{(h)}) \geqslant 0 \quad \forall x_{(h)} \in \mathcal{F}_h^\star,$$

then, using the optimality conditions stated in Proposition 7, we can write

$$-\frac{\nabla_{(h),i^\star} f(\bar{x})}{a_{(h),i^\star}} \leqslant -\frac{\nabla_{(h),j^\star} f(\bar{x})}{a_{(h),j^\star}} \quad i^\star \in I^\star,\ j^\star \in J^\star. \tag{45}$$

Let us assume by contradiction that

$$\nabla_{(h)} f(\bar{x})^T (\hat{x}_{(h)} - \bar{x}_{(h)}) < 0 \quad \text{for some } \hat{x}_{(h)} \in \mathcal{F}_h. \tag{46}$$

Then, there exists a pair $(\hat{i},\hat{j})$ of indexes such that

$$-\frac{\nabla_{(h),\hat{i}} f(\bar{x})}{a_{(h),\hat{i}}} > -\frac{\nabla_{(h),\hat{i}} f(\bar{x})}{a_{(h),\hat{i}}} \quad \hat{i} \in R_h(\bar{x}),\ \hat{j} \in S_h(\bar{x}). \tag{47}$$

Using the definition of $i^\star$ and $j^\star$, and recalling Proposition 6, we obtain

$$-\frac{\nabla_{(h),\hat{i}} f(\bar{x})}{a_{(h),\hat{i}}} \leqslant -\frac{\nabla_{(h),i^\star} f(\bar{x})}{a_{(h),\hat{i}^\star}} \leqslant -\frac{\nabla_{(h),j^\star} f(\bar{x})}{a_{(h),j^\star}} \leqslant -\frac{\nabla_{(h),\hat{j}} f(\bar{x})}{a_{(h),\hat{j}}},$$

which contradicts (47). □

Finally, we remark that there exist algorithms (see, e.g., [11,16]) that perform projections onto feasible sets of the form (40) in linear time. Then, the GP-IDA algorithm, described in Section 4, can be effectively applied to classes of problems whose feasible set has the structure considered in this section.

6. Numerical results on network assignment problems

In this section we report the results obtained by the proposed inexact decomposition methods on network assignment problems, which is a class of convex large-scale optimization problems.

Network assignment problems are a widely studied subject in many research fields, as for instance transportation and data transmission. The aim of a network equilibrium model is to predict the link flows of a network whose arc costs depend on the origin/destination routes chosen by its users (travellers or data package). Denoting by L the number of Origin/Destination (OD) pairs, and by x the vector of path flows, under suitable assumptions (see, e.g., [1,7] for the technical details), network equilibrium model leads to an optimization problem of the form

$$
\begin{aligned}
\min\ & f(x),\\
\text{s.t.}\ & e_{(h)}^T x_{(h)} = b_h \quad \forall h \in \{1,\dots,L\},\\
& x_{(h)} \geqslant 0 \quad \forall h \in \{1,\dots,L\},
\end{aligned} \tag{48}
$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a convex continuously differentiable function; n_h is the number of paths of the hth OD pair, $e_{(h)} \in \mathbb{R}^{n_h}$ is a vector whose components are all ones, b_h is the demand of the h-th OD pair, with $h \in \{1,\dots,L\}$, $n = n_1 + n_2 + \dots + n_L$, and the partial derivative of f with respect to $x_{(h),\, i}$ is the cost of path i of the hth OD pair.

The convex problem (48) has a very simple structure, as its feasible set $\mathcal{F}$ is the Cartesian product of simplices. However, problem (48) can be considered a "virtual" formulation: indeed, in any real application it is not reasonable to completely enumerate, a priori, all the paths, since this would be too expensive. This motivates the need of introducing restrictions of the factor sets formally defined in the preceding section. More specifically, given the current feasible point x^k, for each $h \in \{1,\dots,L\}$, the restricted set of variables contains the variables whose current value is strictly positive and the one corresponding to the cheaper path to the destination, that is, the variable $x_{(h),\pi_h(x^k)}$, being

Table 1
Data set information.

Name	# Nodes	# Links	# OD-pairs
Barcelona	1020	2522	7922
Winnipeg	1067	2975	4344
Berlin Central	12,981	28,376	49,688
Chicago Sketch	933	2950	93,135

Table 2
Barcelona road network: CPU time (seconds) required to attain $r_{gap}(x^k) \leqslant \epsilon$.

Algorithm	$\epsilon = 10^{-4}$	$\epsilon = 10^{-5}$	$\epsilon = 10^{-6}$
GP-IDA	1.81	3.46	16.19
GP-EDA	107.05	243.54	291.94
GP	245.45	732.25	818.66
FW-IDA	4.09	27.75	$*$
OBA	3.76	5.00	5.44

Table 3
Winnipeg road network: CPU time (seconds) required to attain $r_{gap}(x^k) \leqslant \epsilon$.

Algorithm	$\epsilon = 10^{-4}$	$\epsilon = 10^{-5}$	$\epsilon = 10^{-6}$
GP-IDA	3.22	18.79	57.11
GP-EDA	154.03	478.03	1596.67
GP	64.51	126.09	293.06
FW-IDA	21.67	147.16	$*$
OBA	9.46	11.21	12.25

Table 4
Berlin Central road network: CPU time (seconds) required to attain $r_{gap}(x^k) \leqslant \epsilon$.

Algorithm	$\epsilon = 10^{-4}$	$\epsilon = 10^{-5}$	$\epsilon = 10^{-6}$
GP-IDA	32.78	106.57	175.14
GP-EDA	397.08	1014.59	2828.35
GP	154.60	2029.64	10082.73
FW-IDA	114.96	1360.90	$*$
OBA	343.83	374.60	394.90

Table 5
Chicago Sketch road network: CPU time (seconds) required to attain $r_{gap}(x^k) \leqslant \epsilon$.

Algorithm	$\epsilon = 10^{-4}$	$\epsilon = 10^{-5}$	$\epsilon = 10^{-6}$
GP-IDA	25.90	384.12	611.04
GP-EDA	1076.44	3343.16	8969.27
GP	9283.45	$*$	$*$
FW-IDA	34.72	258.70	$*$
OBA	24.86	25.67	33.06

$$\pi_h(x^k) \in \arg \min_{i \in \{1,\ldots,n_h\}} \frac{\partial f(x^k)}{\partial x_{(h),i}}. \tag{49}$$

In the decomposition schemes of GP-IDA and FW-IDA we consider a number of block components equal to the number of origins, so that each given block component groups the variables of the OD pairs associated to the given origin. The line search parameters δ and γ have been set equal to the values 0.5 and 10^{-4} respectively. We have performed numerical experiments on the freely available data sets from the repository of Hillel Bar-Gera.[2] Some additional information about the problem sizes are reported in Table 1.

For all tests we report the so-called *relative gap* (see for instance [1] for more details), a widely used quality function defined for any $x \in \mathcal{F}$ as

$$r_{gap}(x) = 1 - \frac{\sum_{h=1}^{L} \pi_h(x) b_h}{\nabla f(x)^T x} = \frac{\nabla f(x)^T x - \nabla f(x)^T \hat{x}}{\nabla f(x)^T x}, \tag{50}$$

where

$$\hat{x} \in \arg \min_{y \in \mathcal{F}} \nabla f(x)^T y.$$

Note that $r_{gap}(x) \geqslant 0$ and $r_{gap}(x) = 0$ if and only if x is a solution of the network equilibrium problem.

The performances of GP-IDA and FW-IDA have been compared with those of OBA (Origin-Based Assignment), a well established specialized algorithm for the Traffic Assignment Problem (see [1]), for which an executable is freely available.[3] Furthermore, in order to measure the computational improvements brought by the inexact decomposition, we implemented the GP-EDA algorithm, where the subproblems are solved exactly, and the GP algorithm, which represents the standard projected gradient algorithm without any decomposition.

All algorithms, except OBA, have been coded in C++ using the Boost Graph Library[4] implementation of the Dijkstra algorithm for the shortest path computation, as well as the graph representation. All tests have been performed on a Intel Core i7 2.93 gigahertz standard desktop machine with 3 gigabytes of RAM. For each test problem and for each algorithm we report in Tables 2–5 the CPU time required to satisfy the stopping criterion, i.e., for attaining a value of the relative gap (see (50)) less than or equal to the tolerance ϵ, which has been fixed to different values. The symbol $*$ indicates that the algorithm was not able to satisfy the stopping criterion within 5×10^3 seconds for Winnipeg, 10^4 seconds for Barcelona and 2×10^4 seconds for Berlin Central and Chicago Sketch.

From the results reported in Tables 2–5, we can observe that GP-IDA algorithm is competitive with OBA code, and that the Frank–Wolfe based-method, as well-known, is poorly effective for the considered class of problems. We may note that the performances of GP-IDA are clearly better than those of OBA code on the biggest network considered, that is, Berlin Central network. For the other networks, OBA code outperforms GP-IDA when a relatively high accuracy (say a relative gap lower than 10^{-5}) is required (the difference is particularly significant in the Chicago Sketch network). However, as pointed out in [5], in practical cases, requiring a relative gap lower than 10^{-4} does not change by any significant amount the link flow values of the solution. On the whole, as OBA code is highly specialized for the particular problem data and structure, the results of the numerical experiments point out the validity of the proposed inexact decomposition approach. We also notice that the results show the effectiveness of the inexact decomposition strategy compared with an exact decomposition method (GP-EDA) and with a method (GP) not using a decomposition technique.

Acknowledgements

The authors are grateful to the three reviewers for the careful reading of the paper and for some useful suggestions.

References

[1] H. Bar-Gera, Origin-based algorithm for the traffic assignment problem, Transportation Science 36 (2002) 398–417.

[2] R.O. Barr, E.G. Gilbert, Some efficient algorithms for a class of optimization problems arising in optimal control, IEEE Transactions on Automatic Control 14 (1969) 640–652.

[3] D.P. Bertsekas, J.N. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[2] http://www.bgu.ac.il/bargera/tntp/.

[3] http://www.openchannelsoftware.org/projects/Origin-Based_Assignment/.

[4] http://www.boost.org/doc/libs/1_46_1/libs/graph/doc/index.html.

[4] S. Bonettini, Inexact block coordinate descent methods with application to non-negative matrix factorization, IMA Journal of Numerical Analysis 31 (2011) 1431–1452.

[5] D. Boyce, B. Ralevic-Dekic, H. Bar-Gera, Convergence of traffic assignments: how much is enough?, ASCE Journal of Transportation Engineering 130 (1) (2004) 49–55

[6] C.C. Chang, C.W. Hsu, C.-J. Lin, The analysis of decomposition methods for support vector machines, IEEE Transactions on Neural Networks 11 (2000) 1003–1008.

[7] M. Florian, D.W. Hearn, Network equilibrium models, in: M.O. Ball, T.L. Magnanti, C.L. Monma, G.L. Nemhauser (Eds.), Hanbooks in Operations Research and Management Science, Network Routing, vol. 8, Elsevier Science B.V., Amsterdam, 1995, pp. 485–550.

[8] L. Grippo, M. Sciandrone, Globally convergent block-coordinate techniques for unconstrained optimization, Optimization Methods and Software 10 (4) (1999) 587–637.

[9] L. Grippo, M. Sciandrone, On the convergence of the block nonlinear gauss-seidel method under convex constraints, Operations Research Letters 26 (3) (2000) 127–136.

[10] C. Kao, L.F. Lee, M.M. Pitt, Simulated maximum likelihood estimation of the linear expenditure system with binding non-negativity constraints, Annals of Economics and Finance 2 (2001) 203–223.

[11] K. Kiwiel, On linear-time algorithms for the continuous quadratic knapsack problem, Journal of Optimization Theory and Applications 134 (3) (2007) 549–554.

[12] S. Lacoste-Julien, M. Jaggi, M. Schmidt, P. Pletscher, Block-coordinate Frank–Wolfe optimization for structural SVMs, ICML 2013: International Conference on Machine Learning (2013) 53–61.

[13] C.-J. Lin, On the convergence of the decomposition method for support vector machines, IEEE Transactions on Neural Networks 12 (6) (2001) 1288–1298.

[14] C. J Lin, S. Lucidi, L. Palagi, A. Risi, M. Sciandrone, A decomposition algorithm model for singly linearly constrained problems subject to lower and upper bounds, Journal of Optimization Theory and Applications 141 (2009) 107–126.

[15] Z.Q. Luo, P. Tseng, On the convergence of the coordinate descent method for convex differentiable minimization, Journal of Optimization Theory and Applications 72 (1992) 7–35.

[16] C. Michelot, A finite algorithm for finding the projection of a point onto the canonical simplex of R^n, Journal of Optimization Theory and Applications 50 (1986) 195–200.

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

[18] M. Patriksson, Decomposition methods for differentiable optimization problems over Cartesian product sets, Computational Optimization and Applications 9 (1) (1998) 5–42.

[19] M. Razaviyayn, M. Hong, Z.Q. Luo, A unified convergence analysis of block successive minimization methods for nonsmooth optimization, SIAM Journal on Optimization (2013) (in press).

[20] T. Serafini, L. Zanni, On the working set selection in gradient projection-based decomposition techniques for support vector machines, Optimization Methods and Software 20 (2005) 273–307.

[21] P. Tseng, S. Yun, A block-coordinate gradient descent method for linearly constrained nonsmooth separable optimization, Journal of Optimization Theory and Applications 140 (2009) 513–535.

[22] V. Vapnik, Statistical Learning Theory, Wiley, 1998.

[23] S. Yun, K.C. Toh, A coordinate gradient descent method for l1-regularized convex minimization, Computational Optimization and Applications 48 (2) (2011) 273–307.