Comput Optim Appl (2007) 38: 217–234
DOI 10.1007/s10589-007-9044-x

A convergent decomposition algorithm for support vector machines

S. Lucidi · L. Palagi · A. Risi · M. Sciandrone

Published online: 30 May 2007

Abstract In this work we consider nonlinear minimization problems with a single linear equality constraint and box constraints. In particular we are interested in solving problems where the number of variables is so huge that traditional optimization methods cannot be directly applied. Many interesting real world problems lead to the solution of large scale constrained problems with this structure. For example, the special subclass of problems with convex quadratic objective function plays a fundamental role in the training of Support Vector Machine, which is a technique for machine learning problems. For this particular subclass of convex quadratic problem, some convergent decomposition methods, based on the solution of a sequence of smaller subproblems, have been proposed. In this paper we define a new globally convergent decomposition algorithm that differs from the previous methods in the rule for the choice of the subproblem variables and in the presence of a proximal point modification in the objective function of the subproblems. In particular, the new rule for sequentially selecting the subproblems appears to be suited to tackle

S. Lucidi · L. Palagi
Dipartimento di Informatica e Sistemistica "Antonio Ruberti", Università di Roma "La Sapienza", Via Buonarroti 12, 00185 Roma, Italy

S. Lucidi
e-mail: lucidi@dis.uniroma1.it

L. Palagi
e-mail: palagi@dis.uniroma1.it

A. Risi
Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti",
Consiglio Nazionale delle Ricerche, Viale Manzoni 30, 00185 Roma, Italy
e-mail: risi@iasi.cnr.it

M. Sciandrone (✉)
Dipartimento di Sistemi e Informatica, Università di Firenze, Via di S.ta Marta 3, 50139, Firenze, Italy
e-mail: sciandro@dsi.unifi.it

large scale problems, while the introduction of the proximal point term allows us to ensure the global convergence of the algorithm for the general case of nonconvex objective function. Furthermore, we report some preliminary numerical results on support vector classification problems with up to 100 thousands variables.

Keywords Large scale optimization · Decomposition methods · Proximal point modification · Support vector machine

1 Introduction

Let us consider the problem

$$\begin{aligned} \min\ & f(x) \\ & a'x = b, \quad l \le x \le u, \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function and $a, l, u \in \mathbb{R}^n$, with $l < u$, $b \in \mathbb{R}$. We allow the possibility that some of the variables are unbounded by permitting both $l_i = -\infty$ and $u_i = \infty$ for some $i \in \{1, \ldots, n\}$. Moreover, we assume, without loss of generality that $a_i \neq 0$ for all $i = 1, \ldots, n$, thought our approach can be extended with minor modifications to include the case where $a_i = 0$ for some i.

Problems with structure (1) arise directly or as subproblems in several applications. Among these, there are portfolio selection problems, optimal control, image processing, optimal allocation, maximum-likelihood estimation, knapsack problems (see e.g. [2, 9, 14, 16, 17, 24, 26, 27, 33] and references therein). Moreover, a continuous formulation of a classical problem in graph theory, namely the maximum clique problem [6, 25], leads to a problem of the form (1) with indefinite quadratic function. Recently there has been a growing interest in Support Vector Machine (SVM) [32] which is a promising technique for solving a variety of machine learning and function estimation problems. The SVM technique leads to solve a large dimensional problem of the form (1) with the distinguish features:

- f is a convex quadratic function with dense Hessian matrix;
- $a_i \in \{-1, 1\}$, $b = 0$;
- $-\infty < l_i < u_i < \infty$.

Several approaches are developed in the literature for SVM that is typically a huge application (see e.g. [11, 15, 22, 23, 29] and references therein).

In the sequel we do not assume any convexity or any special structure of the objective function, hence we are interested in finding stationary points of problem (1). The main difficulty in computing a stationary point of the problem (1) (whose feasible set has a very simple structure) is mainly related to the dimension n of the problem. In particular, when n is extremely large and the problem is not sparse, traditional optimization methods cannot be directly employed. Then, we focus the attention on large dimensional problems and we are interested in studying convergent block decomposition methods, which involve the solution of subproblems of smaller dimension in place of the original problem.

The most popular convergent decomposition methods, such as the Successive Overrelaxation algorithm, the Jacobi and the Gauss–Seidel algorithms are applica-

ble only when the feasible set is the Cartesian product of subsets defined in smaller subspaces [5]. Since the feasible set of problem (1) contains an equality constraint, such decomposition methods cannot be employed. Moreover, in [30] a set of counterexamples concerning unconstrained problems has been reported, where the Gauss–Seidel method may not even converge towards stationary points. This evidences the difficulty of ensuring convergence properties of decomposition methods, even in the simplest case of unconstrained optimization problems.

In a general decomposition framework, at each iteration k, the vector of variables x is partitioned into two subvectors $(x_W, x_{\overline{W}})$, where $W \subset \{1, \ldots, n\}$ identifies the variables of the subproblem to be solved and it is called *working set*, and $\overline{W} = \{1, \ldots, n\} \setminus W$ (for notational convenience the dependence of W and $\overline{W}$ on k is omitted). Then, starting from the current feasible vector $x^k = (x_W^k, x_{\overline{W}}^k)$, the subvector x_W^{k+1} is computed as the solution of the following subproblem

$$\begin{aligned} \min_{x_W} \; & f\left(x_W, x_{\overline{W}}^k\right) \\ & a'_W x_W = b - a'_{\overline{W}} x_{\overline{W}}^k, \quad l_W \le x_W \le u_W. \end{aligned}$$

The subvector $x_{\overline{W}}^{k+1}$ is unchanged, i.e., $x_{\overline{W}}^{k+1} = x_{\overline{W}}^k$, and the new iterate is given by $x^{k+1} = (x_W^{k+1}, x_{\overline{W}}^{k+1})$. In general, the cardinality q of the working set W, i.e. the dimension of the subproblem to be solved at each iteration, is chosen according to the available computational capability or to the problem structure. The rule of selection of the indices in the working set W at each iteration plays a crucial role in proving the convergence properties of the sequence $\{x^k\}$ generated by the decomposition method.

Up to our knowledge, decomposition methods with theoretical convergence properties have been proposed with reference to problem (1) in the special case of SVM's learning problem (see [18–20]), namely in the case of minimization of a convex quadratic function on a bounded convex set. In this context, the most popular algorithm is the SVM$^{\text{light}}$ algorithm [15], whose selection rule of the working sets requires, at each iteration, the application of a specific ordering procedure of a vector of dimension n connected to the violation of the Karush–Kuhn–Tucker conditions. For any even size q of the working set, the asymptotic convergence of the SVM$^{\text{light}}$ algorithm has been proved under the assumption that the quadratic objective function satisfies some strict block convexity assumption [18]. In the special case of $q = 2$, the convergence of the decomposition algorithm (SMO [29]) is guaranteed only requiring that the quadratic function is convex [19].

The convergence analysis of SVM$^{\text{light}}$ algorithm, developed in [18], highlights that the strict block convexity hypothesis on the objective function permits to ensure that the distance between successive points of the generated sequence tends to zero, i.e. that

$$\lim_{k \to \infty} \|x^{k+1} - x^k\| = 0. \tag{2}$$

As pointed out in [12, 13], this property is an important requirement in the context of a decomposition strategy, and proximal point techniques can be employed to attain (2) even in the case of nonconvex objective function (see also [11] for a different use of a proximal-point modification within an interior-point method for SVM).

In this paper, we define a decomposition algorithm model (DAM) for problem of type (1), which differs from the existing methods in the rule for the choice of the working sets and in the introduction of a proximal point modification in the objective function of the subproblems. In particular, we introduce a general condition to be satisfied by the sequence of working sets, and we show that it is possible to satisfy it by using a prefixed number of working sets in a cyclic order. The main benefit of this choice stays in the fact that the selection of the variables in the working set does not require to apply any specific ordering procedure, that means that no additional computational effort is required to identify the subproblem to be solved. On the other hand, the proximal point modification introduced in the objective functions of the subproblems allows us to ensure property (2) without further assumptions. Thus, we prove the asymptotic convergence of the decomposition algorithm DAM without requiring the convexity of the objective function, and assuming only the existence of limit points of the generated sequence.

It is worthwhile to remark that, since the selection condition on the working set does not require on-line computation, it could also be possible to define a parallel version of the decomposition algorithm DAM. Parallelism may further speed up the computation and this is particularly desirable in the case of large and dense problems (see e.g. [5, 10, 28] for parallel decomposition approaches). However, the definition of a parallel version of DAM will be object of further study.

The paper is organized as follows. In Sect. 2, we introduce some basic notation and preliminary technical results. In Sect. 3, we define the decomposition algorithm model DAM and we present the convergence analysis. In Sect. 4 we report the results of the computational experiments performed on support vector classification problems with up to 100 thousands variables.

2 Notation and preliminary results

In this section we state some results that will be used in the convergence analysis of the decomposition algorithm defined in the next section.

First we introduce some basic notation and definitions (see e.g. [3]). Throughout the paper, we denote by $\mathcal{F}$ the feasible set of problem (1), namely

$$\mathcal{F} = \{x \in \mathbb{R}^n \colon a'x = b,\ l \le x \le u\}.$$

For every feasible point x, we denote the sets of indices of active (lower and upper) bounds as follows:

$$L(x) = \{i \colon x_i = l_i\}, \qquad U(x) = \{i \colon x_i = u_i\}.$$

Further, at a feasible point x, the set of the feasible directions is the cone

$$D(x) = \left\{d \in \mathbb{R}^n \colon a'd = 0,\ d_i \ge 0,\ \forall i \in L(x),\ \text{and } d_i \le 0,\ \forall i \in U(x)\right\}.$$

Since the feasible set of problem (1) is convex, if a point $x^* \in \mathcal{F}$ is a minimum point of f over $\mathcal{F}$ then it satisfies the following necessary optimality condition:

$$\nabla f(x^*)'d \ge 0 \quad \text{for every } d \in D(x^*). \tag{3}$$

A point $x^* \in \mathcal{F}$ satisfying condition (3) is said to be a *critical* (or *stationary*) point of problem (1). Since the constraints defining the feasible set $\mathcal{F}$ are linear, condition (3) is equivalent to the Karush–Kuhn–Tucker (KKT) conditions that, by simple manipulation, state the existence of a scalar λ^* such that

$$(\nabla f(x^*))_i + \lambda^* a_i \begin{cases} \geq 0 & \text{if } i \in L(x^*), \\ \leq 0 & \text{if } i \in U(x^*), \\ = 0 & \text{if } i \notin L(x^*) \cup U(x^*). \end{cases}$$

The sets L and U can be split in L^+, L^-, and U^+, U^- respectively, where

$$L^-(x) = \{i \in L(x)\colon\ a_i < 0\}, \qquad L^+(x) = \{i \in L(x)\colon\ a_i > 0\},$$
$$U^-(x) = \{i \in U(x)\colon\ a_i < 0\}, \qquad U^+(x) = \{i \in U(x)\colon\ a_i > 0\},$$

and we can immediately state the following proposition on the equivalence between critical points and KKT points.

Proposition 1 KKT conditions *A feasible point x^* is a critical point of problem (1) if and only if there exists a scalar λ^* such that*

$$\begin{aligned} \lambda^* &\geq -\frac{(\nabla f(x^*))_i}{a_i} \quad \forall i \in L^+(x^*) \cup U^-(x^*), \\ \lambda^* &\leq -\frac{(\nabla f(x^*))_i}{a_i} \quad \forall i \in L^-(x^*) \cup U^+(x^*), \\ \lambda^* &= -\frac{(\nabla f(x^*))_i}{a_i} \quad \forall i \notin L(x^*) \cup U(x^*). \end{aligned} \tag{4}$$

Exploiting the particular structure of $\mathcal{F}$ we can give equivalent versions of the KKT conditions that are useful in the definition of decompositions algorithms. In particular, at a feasible point x, we introduce the following index sets:

$$\begin{aligned} R(x) &= L^+(x) \cup U^-(x) \cup \{i\colon\ l_i < x_i < u_i\}, \\ S(x) &= L^-(x) \cup U^+(x) \cup \{i\colon\ l_i < x_i < u_i\}. \end{aligned} \tag{5}$$

Then we can state the following results whose proofs are reported in the Appendix.

Proposition 2 *A feasible point x^* is a KKT point of problem (1) if and only if there exists no pair of indices i and j, with $i \in R(x^*)$ and $j \in S(x^*)$, such that*

$$-\frac{(\nabla f(x^*))_i}{a_i} > -\frac{(\nabla f(x^*))_j}{a_j}. \tag{6}$$

Proposition 3 *Let $\hat{x}$ be a feasible point.*

(i) *If $\hat{x}$ is not a KKT point, then the following strict inequality holds*

$$\max_{h \in R(\hat{x})} -\frac{(\nabla f(\hat{x}))_h}{a_h} > \min_{h \in S(\hat{x})} -\frac{(\nabla f(\hat{x}))_h}{a_h}; \tag{7}$$

(ii) *For each pair* $i \in R(\hat{x})$ *and* $j \in S(\hat{x})$, *the direction* $d^{i,j} \in \mathbb{R}^n$ *such that*

$$d_i^{i,j} = \frac{1}{a_i}, \qquad d_j^{i,j} = -\frac{1}{a_j}, \qquad d_h^{i,j} = 0 \quad \text{for } h \neq i, j \tag{8}$$

is a feasible direction at $\hat{x}$. *Furthermore, if the pair* $i \in R(\hat{x})$ *and* $j \in S(\hat{x})$ *satisfies*

$$-\frac{(\nabla f(\hat{x}))_i}{a_i} > -\frac{(\nabla f(\hat{x}))_j}{a_j}, \tag{9}$$

then $d^{i,j}$ *is also a descent direction at* $\hat{x}$, *that is*

$$d^{i,j} \in D(\hat{x}) \quad \text{and} \quad \nabla f(\hat{x})' d^{i,j} < 0.$$

The first part of the next proposition essentially establishes that, given any convergent sequence of feasible points $\{x^k\}$, the set of feasible directions at the limit point $\bar{x} \in \mathcal{F}$ is contained in the sets of the feasible directions at points x^k sufficiently close to $\bar{x}$. This result can be deduced by [21, Proposition 1]; however, for sake of completeness, we have reported in Appendix the proof adapted to the specific problem.

Proposition 4 *Let* $\{x^k\}$ *be a sequence of feasible points converging to a point* $\bar{x} \in \mathcal{F}$. *Then, for sufficiently large values of* k *we have*:

(i) $D(\bar{x}) \subseteq D(x^k)$;
(ii) $R(\bar{x}) \subseteq R(x^k)$ *and* $S(\bar{x}) \subseteq S(x^k)$.

3 A decomposition algorithm model (DAM)

The basic strategy of a decomposition method is to perform, at each iteration, the minimization of the objective function on the feasible set with respect only to a subset of variables, holding fixed the remaining ones.

In order to describe a decomposition framework, given a vector $x \in \mathbb{R}^n$, and an index set $W \subseteq \{1, \ldots, n\}$, we adopt the (already introduced) notation $x_W \in \mathbb{R}^{|W|}$ to indicate the subvector of x made up of the component x_i with $i \in W$. More in particular, at each iteration k, given the working set $W \subset \{1, \ldots, n\}$ and starting from the current feasible vector $x^k = (x_W^k, x_{\overline{W}}^k)$, the new iterate $x^{k+1} = (x_W^{k+1}, x_{\overline{W}}^{k+1})$ is obtained by computing x_W^{k+1} as a stationary point such that $f(x_W^{k+1}, x_{\overline{W}}^k) \leq f(x_W^k, x_{\overline{W}}^k)$ of the following problem:

$$\begin{aligned} \min_{x_W} \; & f\left(x_W, x_{\overline{W}}^k\right) \\ & a_W' x_W = b - a_{\overline{W}}' x_{\overline{W}}^k, \quad l_W \leq x_W \leq u_W, \end{aligned} \tag{10}$$

and by setting $x_{\overline{W}}^{k+1} = x_{\overline{W}}^k$, being $\overline{W} = \{1, \ldots, n\} \setminus W$ (for notational convenience we have omitted here and in some cases later the dependence of W and $\overline{W}$ on the iteration counter k). We note that by construction the new iterate x^{k+1} is feasible and it results $f(x^{k+1}) \leq f(x^k)$.

An important issue in the design of a convergent decomposition method is the rule to select the indices in working set W at each iteration. Next we discuss this issue and we state a general condition on the working set selection. Later we present and motivate a proximal point modification of the objective function of the subproblem (10).

Working set selection condition First of all we observe that, due to the presence of the linear equality constraint, any feasible direction at a feasible point must have at least two nonzero components. Hence the smallest number of variables that can be changed at each iteration is two, so that the cardinality q of the working set must be at least two.

As said in the introduction, decomposition methods have been proposed for problem of type (1) arising in the field of Support Vector Machine, where the objective function $f(x)$ is a quadratic convex function. Among them, the most popular one is the SVM$^{\text{light}}$ algorithm [15] which uses a selection rule for the working set related to the violation of the KKT conditions. To be more precise, given an even integer $q \geq 2$, $q/2$ indices $\{i^1, \ldots, i^{q/2}\}$ are sequentially selected in $R(x^k)$ so that

$$-\frac{\nabla f(x^k)_{i^1}}{a_{i^1}} \geq -\frac{\nabla f(x^k)_{i^2}}{a_{i^2}} \geq \cdots \geq -\frac{\nabla f(x^k)_{i^{q/2}}}{a_{i^{q/2}}},$$

and $q/2$ indices $\{j^1, \ldots, j^{q/2}\}$ are sequentially selected in $S(x^k)$ so that

$$-\frac{\nabla f(x^k)_{j^1}}{a_{j^1}} \leq -\frac{\nabla f(x^k)_{j^2}}{a_{j^2}} \leq \cdots \leq -\frac{\nabla f(x^k)_{j^{q/2}}}{a_{j^{q/2}}};$$

the working set is defined by letting $W^k = \{i^1, \ldots, i^{q/2}, j^1, \ldots, j^{q/2}\}$. Hence, in the case of SVM$^{\text{light}}$ algorithm, the definition of the working set W^k requires, at each iteration, the evaluation of the whole (scaled) gradient and its partial ordering. This may be computationally disadvantageous when the dimension n is large, and we want to avoid it. To this aim, we propose a different approach for the selection of the working set W^k by introducing a general condition to be met by the sequence of index sets $\{W^k\}$. This condition takes inspiration from an earlier convergence proof [18]. Actually the proof is by contradiction on the existence of a limit point $\bar{x}$ such that

$$\nabla f(\bar{x})^T d^{i,j} < 0, \quad \text{for some pair } (i, j) \in R(\bar{x}) \times S(\bar{x}).$$

The contradiction is obtained by showing that the pair (i, j) is eventually selected. On this basis, we define the following condition on the working sets $\{W^k\} \subseteq \{1, \ldots, n\}$.

Working set selection (WSS) condition The sequence of index sets $\{W^k\} \subseteq \{1, \ldots, n\}$ is such that, for all $k \geq 0$ and for each pair of indices $i, j \in \{1, \ldots, n\}$ such that

$$i \in R(x^k), \ j \in S(x^k) \quad \text{and} \quad -\frac{(\nabla f(x^k))_i}{a_i} > -\frac{(\nabla f(x^k))_j}{a_j},$$

there exists an integer $m_{i,j}(k)$, with $k \leq m_{i,j}(k) \leq M + k$ and $M \geq 0$, satisfying

$$(i, j) \in W^{m_{i,j}(k)}.$$

WSS condition above essentially requires that, starting from the current iteration k, each pair of indices (i, j), that identifies, according to (ii) of Proposition 3, a feasible and descent direction at x^k, is inserted in the working set within a maximum number M of successive iterations.

We observe that a possible simple way to satisfy the WSS condition on the working sets consists in using prefixed index sets in cyclic order as working sets. In particular, given the index sets $V^0, \ldots, V^M \subset \{1, \ldots, n\}$ of cardinality $q^0, \ldots, q^M$ and such that each pair $i, j \in \{1, \ldots, n\}$ is contained (at least) in a set V^l, the working set W^k can be defined as:

$$W^k = V^{k \bmod M} \quad \text{for } k = 0, 1, \ldots$$

where $k \bmod M$ denotes the remainder of k `modulo` M. As an example, for $n = 6$ and letting $q = 3$ for all the sets, we get that the sets

$$V^0 = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, \qquad V^1 = \begin{pmatrix} 1 \\ 4 \\ 5 \end{pmatrix}, \qquad V^2 = \begin{pmatrix} 1 \\ 5 \\ 6 \end{pmatrix},$$

$$V^3 = \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix}, \qquad V^4 = \begin{pmatrix} 2 \\ 3 \\ 5 \end{pmatrix}, \qquad V^5 = \begin{pmatrix} 3 \\ 4 \\ 6 \end{pmatrix}$$

present the above property with $M = 5$. The use of prefixed index sets as working sets permits to select the subproblems variables without applying any specific procedure. In this way, it is not required a computational effort to individuate the subproblem variables at any iteration; moreover, since it does not require any on-line computation, it appears well-suited for the definition of a parallel decomposition algorithm.

Proximal point modification The need of introducing a proximal point modification in the objective function of subproblem (10) has been briefly discussed in the introduction. Actually, adopting a decomposition strategy, optimality conditions with respect to the variables associated to the selected working sets are satisfied in different successive points, that are solutions of the corresponding subproblems. Therefore, in order to ensure convergence of the produced sequence, it may be necessary to enforce that the distance between successive points tends to zero, i.e.,

$$\lim_{k \to \infty} \|x^{k+1} - x^k\| = 0. \tag{11}$$

In general, without suitable convexity assumptions on the objective function, the above property could not be guaranteed. Indeed, the asymptotic convergence of SVM$^{\text{light}}$ algorithm has been established in [18], under the assumption that the quadratic objective function is strictly convex with respect to any block component of cardinality less or equal than q.

In order to ensure property (11), at each iteration k, we consider the following modified subproblem

$$\begin{aligned} \min_{x_W} \quad & f\left(x_W, x^k_{\overline{W}}\right) + \tau \left\| x_W - x^k_W \right\|^2 \\ & a'_W x_W = b - a'_{\overline{W}} x^k_{\overline{W}}, \quad l_W \le x_W \le u_W, \end{aligned} \tag{12}$$

where the objective function contains the additional *proximal point term* $\tau\|x_W - x_W^k\|^2$, being $\tau > 0$ (see e.g. [1, 4, 13, 31]). Then, the subvector x_W^{k+1} is determined by computing any stationary point (not necessarily a global minimum) of problem (12) such that $f(x_W^{k+1}, x_{\overline{W}}^k) + \tau\|x_W^{k+1} - x_W^k\|^2 \leq f(x_W^k, x_{\overline{W}}^k)$ (which is immediately attained by using some descent method for the solution of problem (12)). The subvector $x_{\overline{W}}^{k+1}$ is not modified, i.e., $x_{\overline{W}}^{k+1} = x_{\overline{W}}^k$. We note that in the case of quadratic objective function, as in SVM classification problems, the introduction of the proximal point term preserves the quadratic structure of the objective function. Moreover, if the quadratic function is convex, the quadratic subproblems become strictly convex.

We are now ready to introduce the decomposition algorithm model DAM. The convergence results of DAM will be stated under the assumption that the generated sequence admits limit points.

Decomposition Algorithm Model (DAM)
Data. A feasible point x^0, $\tau > 0$.
Initialization. Set $k = 0$.
While (stopping criterion not satisfied)

1. Select the working set W^k;
2. Set $W = W^k$. Find a stationary point x_W^* of problem (12) s.t.

$$f\left(x_W^*, x_{\overline{W}}^k\right) + \tau\left\|x_W^* - x_W^k\right\|^2 \leq f\left(x_W^k, x_{\overline{W}}^k\right);$$

3. Set

$$x_i^{k+1} = \begin{cases} x_i^* & \text{if } i \in W \\ x_i^k & \text{otherwise;} \end{cases}$$

4. Set $k = k + 1$.

end while
Return $x^* = x^k$

Let us introduce the notation D_{W^k} for the set of directions $d \in \mathbb{R}^n$ such that $d_i \neq 0$ only if $i \in W^k$, namely

$$D_{W^k} = \left\{d \in \mathbb{R}^n\colon\ d_i \neq 0 \Longrightarrow i \in W^k\right\}.$$

In the next lemma we show that the point x^{k+1} produced at the kth iteration satisfies the optimality condition with respect to the variables associated to the working set W^k.

Lemma 1 *Let $\{x^k\}$ be the sequence generated by Algorithm DAM. Then we have*:

$$\nabla f(x^{k+1})'d + 2\tau(x^{k+1} - x^k)'d \geq 0 \quad \forall d \in D_{W^k} \cap D(x^{k+1}) \tag{13}$$

where $D(x^{k+1})$ is the set of feasible directions at x^{k+1}.

Proof For simplicity let $W = W^k$, and consider any $d \in D_W \cap D(x^{k+1})$. Let $d = (d_W, d_{\overline{W}})'$ be the partition of the direction d corresponding to index sets W and $\overline{W}$.

Namely, d_W is the subvector of d with elements in W and, since $d \in D_W$, we have that $d_{\overline{W}} = 0$. Recalling that $x_W^{k+1} = x_W^*$ and $x_{\overline{W}}^{k+1} = x_{\overline{W}}^k$, it is immediate to verify that the subvector d_W is a feasible direction for the subproblem (12) at x_W^*. Since the feasible set of problem (12) is convex, the necessary optimality condition can be written as:

$$\nabla_W f\left(x_W^*, x_{\overline{W}}^k\right)' d_W + 2\tau\left(x_W^* - x_W^k\right)' d_W \geq 0$$

where $\nabla_W f$ denotes the subvector of ∇f with components in W. Recalling again that $x_W^{k+1} = x_W^*$, $x_{\overline{W}}^{k+1} = x_{\overline{W}}^k$ and $d_{\overline{W}} = 0$, we get

$$\nabla f(x^{k+1})' d + 2\tau(x^{k+1} - x^k)' d = \nabla_W f\left(x_W^*, x_{\overline{W}}^k\right)' d_W + 2\tau\left(x_W^* - x_W^k\right)' d_W \geq 0,$$

and hence the result. □

Before stating the main convergence result, we show that, thanks to the proximal point modification, (11) holds for DAM. In particular the following proposition holds.

Proposition 5 *Assume that DAM does not terminate and let $\{x^k\}$ be the sequence generated by it. If $\{x^k\}$ admits limit points, then we have*

$$\lim_{k\to\infty} \|x^{k+1} - x^k\| = 0.$$

Proof Let $\bar{x}$ be any limit point of $\{x^k\}$, i.e., there exists a subsequence $\{x^k\}_K$ such that $\lim_{k\to\infty, k\in K} x^k = \bar{x}$. From the instructions of the algorithm, we have for all k

$$f(x^{k+1}) + \tau\|x^{k+1} - x^k\|^2 \leq f(x^k), \tag{14}$$

so that the sequence $\{f(x^k)\}$ is decreasing. Since $\{x^k\}_K$ converges to $\bar{x}$ and f is continuous, we have that $\{f(x^k)\}_K$ converges to $f(\bar{x})$, and this implies that the entire sequence $\{f(x^k)\}$ converges to $f(\bar{x})$. Then, the convergence of the sequence $\{f(x^k)\}$ to a finite value and (14) imply that $\|x^{k+1} - x^k\| \to 0$. □

Remark 1 It is worthwhile to remark that the same result of proposition above can be proved also with $\tau = 0$ under some convexity assumption on f. In particular, if the function f is strictly convex with respect to any subset $I \subseteq \{1, \ldots, n\}$ such that $|I| \leq q$, the same assertion of the proposition can be proved by similar reasonings used in the proof of [5, Proposition 3.9].

The asymptotic convergence of the decomposition algorithm DAM is proved in the following proposition.

Proposition 6 *Assume that DAM does not terminate and that the sequence of working sets $\{W^k\}$ satisfies the WSS condition. Let $\{x^k\}$ be the sequence generated by DAM. Then, every limit point of $\{x^k\}$ is a KKT point of problem* (1).

Proof Let $\bar{x}$ be any limit point of a subsequence of $\{x^k\}$, i.e., there exists an infinite subset $K \subseteq \{0, 1, \ldots\}$ such that $x^k \to \bar{x}$ for $k \in K$, $k \to \infty$. Since the feasible set $\mathcal{F}$ is closed and $x^k \in \mathcal{F}$ for all k, the point $\bar{x}$ is feasible.

By contradiction, let us assume that $\bar{x}$ is not a KKT point of problem (1). By Proposition 3 there exists at least a pair $(i, j) \in R(\bar{x}) \times S(\bar{x})$, and a direction $d^{i,j} \in D(\bar{x})$ defined as in (8) such that:

$$\nabla f(\bar{x})' d^{i,j} < 0. \tag{15}$$

By Proposition 4, we have that $i \in R(x^k)$ and $j \in S(x^k)$ for k sufficiently large. Furthermore, from (15), using the definition of $d^{i,j}$ and recalling the continuity of the gradient, we have that $-\frac{(\nabla f(x^k))_i}{a_i} > -\frac{(\nabla f(x^k))_j}{a_j}$. Then, the WSS condition on the working sets implies that for $k \in K$ sufficiently large, there exists an index $m_{i,j}(k)$, with $m_{i,j}(k) - k \leq M$, such that the pair (i, j) is inserted in the working set at iteration $m_{i,j}(k)$, i.e.

$$(i, j) \in W^{m_{i,j}(k)}. \tag{16}$$

We can write

$$\begin{aligned} \left\| x^{m_{i,j}(k)+1} - x^k \right\| \leq & \left\| x^{m_{i,j}(k)+1} - x^{m_{i,j}(k)} \right\| + \left\| x^{m_{i,j}(k)} - x^{m_{i,j}(k)-1} \right\| \\ & + \cdots + \left\| x^{k+1} - x^k \right\|. \end{aligned} \tag{17}$$

Since $x^k \to \bar{x}$, by Proposition 5 we have

$$\lim_{k \to \infty} \left\| x^{k+1} - x^k \right\| = 0. \tag{18}$$

Then, recalling that $m_{i,j}(k) - k \leq M$, from (17) and (18) we get

$$\lim_{k \to \infty, k \in K} x^{m_{i,j}(k)+1} = \bar{x}. \tag{19}$$

As $x^{m_{i,j}(k)+1} \to \bar{x}$, by Proposition 4 we get that, for $k \in K$ sufficiently large, the set $D(\bar{x})$ of feasible directions at $\bar{x}$ is contained in the set $D(x^{m_{i,j}(k)+1})$ of feasible directions at $x^{m_{i,j}(k)+1}$. In particular, since $d^{i,j} \in D(\bar{x})$, we have that

$$d^{i,j} \in D(x^{m_{i,j}(k)+1}) \quad \text{for } k \in K, \text{ and } k \text{ sufficiently large.} \tag{20}$$

By (16) and (20) we know in particular that

$$d^{i,j} \in D_{W^{m_{i,j}(k)}} \cap D(x^{m_{i,j}(k)+1}) \quad \text{for } k \in K, \text{ and } k \text{ sufficiently large.}$$

It follows that an infinite subset $K_1 \subseteq K$ exists such that $d^{i,j}$ satisfies the assumption of Lemma 1 and hence we can write, for every $k \in K_1$, the optimality condition

$$\nabla f\left(x^{m_{i,j}(k)+1}\right)' d^{i,j} + 2\tau \left(x^{m_{i,j}(k)+1} - x^{m_{i,j}(k)}\right)' d^{i,j} \geq 0 \quad \text{for all } k \in K_1.$$

Taking limits for $k \to \infty$, $k \in K_1$, recalling (18,19), and the continuity of ∇f, we obtain

$$\nabla f(\bar{x})' d^{i,j} \geq 0,$$

and this contradicts (15). □

4 Computational experiments on support vector classification problems

Given a training set of input-target pairs (u^i, y^i), $i = 1, \ldots, n$, with $u^i \in \mathbb{R}^m$, and $y^i \in \{-1, 1\}$, the SVM classification technique requires the solution of the following convex quadratic programming problem

$$\begin{aligned} &\min f(x) = \tfrac{1}{2} x^{\mathrm{T}} Q x - e^{\mathrm{T}} x \\ &\text{s.t.} \quad y^{\mathrm{T}} x = 0, \quad 0 \le x \le Ce, \end{aligned} \tag{21}$$

where $x \in \mathbb{R}^n$, Q is a $n \times n$ positive definite matrix, $e \in \mathbb{R}^n$ is the vector of all ones, $y \in \{-1, 1\}^n$ and C is a positive scalar. The generic element q_{ij} of the matrix Q is given by $y^i y^j K(u^i, u^j)$, where $K(u, z) = \phi(u)'\phi(z)$ is the kernel function related to the nonlinear function ϕ that maps the data from the input space into the feature space. The most widely used kernels are the following:

- linear: $K(u, z) = u'z$;
- polynomial: $K(u, z) = (\gamma u'z + r)^d$, with $\gamma > 0$;
- Gaussian: $K(u, z) = \exp(-\gamma \|u^2 - z^2\|)$, with $\gamma > 0$;

where γ, r, d are kernel parameters.

We present an easily implementable version of Algorithm DAM for the quadratic programming programs (21), where we set the dimension q of the working set equal to two. We observe that in this case, similar to [19], the convergence of Algorithm DAM holds with $\tau = 0$. Thus, in correspondence to a given working set W, the subproblem (12) takes the form

$$\begin{aligned} \min_{x_W} \; & f\left(x_W, x^k_{\overline{W}}\right) = \tfrac{1}{2} x'_W Q_{WW} x_W - \left(e - Q_{W\overline{W}} x^k_{\overline{W}}\right)' x_W, \\ & y'_W x_W = -y'_{\overline{W}} x^k_{\overline{W}}, \quad 0 \le x_W \le C e_W. \end{aligned} \tag{22}$$

Moreover, as we set the dimension q of the working set equal to two, the exact solution of subproblem (22) can be determined analytically (see, e.g., [8]). As regards the definition of the sequence of working sets satisfying the WSS condition, we consider in a cyclic order the pairs

$$(1, 2),\ (1, 3),\ \ldots,\ (1, n),\ (2, 3),\ \ldots,\ (2, n),\ \ldots,\ (n-1, n), \tag{23}$$

and we select as working set the first pair $(i, j) \in R(x^k) \times S(x^k)$ such that

$$-\frac{(\nabla f(x^k))_i}{a_i} > -\frac{(\nabla f(x^k))_j}{a_j}.$$

The performance of this cyclic version of DAM has been compared with that of LIBSVM [7], which is a widely used decomposition algorithm for SVM classification (and regression) implementing an efficient version of SVM$^{\text{light}}$ algorithm [15]. In particular the working set selection used in LIBSVM is the same of SVM$^{\text{light}}$ (see the subparagraph of Sect. 3) with dimension q of the working set equal to two.

In order to make some fair computational comparison between LIBSVM and this version of algorithm DAM, this latter has been implemented using the available code (written in C++) of LIBSVM, where the choice of the working set has been modified.

The stopping criterion adopted is based on a test on the decrease of the objective function value obtained in the last P iterations, and on a test on the number of iterations performed. More specifically, both LIBSVM and DAM are stopped whenever either

$$\max_{0\le t\le P} \Delta f_{k-t} \le 10^{-9},$$

where $\Delta f_{k-t} = f(x^{k-t-1}) - f(x^{k-t})$ with $P = 30$, or the number of iterations performed is equal to 20000.

For experimentation, three test problems described below were used.

Problem P1 (Mushroom problem) [www.ics.uci.edu/mlearn/MLRepository.html]. The task is that of distinguishing edible and poisonous mushrooms. The data set used consists of 8000 pairs, the dimension of the input space is $m = 125$.

Problem P2 (Random problem [11]) Starting from two linearly separable sets of points, a nonseparable data set was constructed by changing the classification of a certain number (equal to 1% of the overall points) of randomly chosen observations. The data set consists of 10000 pairs, the dimension of the input space is $m = 10$.

Problem P3 (kddcup problem) [kdd.ics.uci.edu/databases/kddcup99/kddcup99.html]. The task is to build a network intrusion detector, which must distinguish between "bad" connections and "good" normal connections. The data set used consists of 100000 pairs, the dimension of the input space is $m = 39$.

The experiments were performed setting in (21), the parameter C equal to 100, and using both the polynomial (with $\gamma = 1$, $r = 1$ and $d = 1, 2, 3, 4$) and the Gaussian (with $\gamma = 1$) kernels. The algorithms ran on a 1.84 GHz AMD Athlon with 256 megabytes of RAM.

The results are shown in Tables 1–3, where we report the number of iterations (n_i), the attained function value ($f^\star$), the training set accuracy (%), the required cpu time (cpu) and the cpu time devoted to the working set selection (cpu_w), both expressed in seconds. The results concerning the polynomial kernel correspond to the values of the parameter d for which the best training set accuracy provided by one of the two algorithms was highest, that is $d = 4$ for Problem P1, $d = 1$ for Problem P2, and $d = 2$ for Problem P3.

By comparing the two algorithms in terms of obtained solutions (namely the values of $f^\star$ and %) it appears that the behaviour of LIBSVM is better than the one of DAM in the case of Gaussian kernel, while, in the case of polynomial kernel, DAM clearly outperforms LIBSVM on Problems P2 and P3 and it is comparable with LIBSVM on Problem P1. We note that on Problem P2 with the polynomial kernel, the two algorithms provide close objective function values, but quite different training accuracy. Further experiments were performed on Problem P2 by increasing the number of iterations up to 50000, but the difference in terms of training accuracy remains relevant (for instance, by running 50000 iterations we obtained an accuracy of 63.59% for LIBSVM and of 90.97% for DAM).

As regards the comparison in terms of cpu time, we can observe that the behaviour of DAM is better than that of LIBSVM in all the problems with the polynomial kernel.

Table 1 Comparisons on Problem P1

Algorithm	Polynomial					Gaussian				
	n_i	$f^\star$	%	cpu	cpu_w	n_i	$f^\star$	%	cpu	cpu_w
LIBSVM	14769	−0.00013	100	47	1.5	20000	−1070.14	100	1133	2.6
DAM	20000	−0.00012	100	37	0.1	20000	−1015.92	100	583	0.1

Table 2 Comparisons on Problem P2

Algorithm	Polynomial					Gaussian				
	n_i	$f^\star$	%	cpu	cpu_w	n_i	$f^\star$	%	cpu	cpu_w
LIBSVM	20000	−483.6	63.64	34	2.8	20000	−4889.25	100	196	2.8
DAM	20000	−482.2	95.12	29	0.1	20000	−3982.84	99.19	89	0.1

Table 3 Comparisons on Problem P3

Algorithm	Polynomial					Gaussian				
	n_i	$f^\star$	%	cpu	cpu_w	n_i	$f^\star$	%	cpu	cpu_w
LIBSVM	20000	−0.3	75.75	267	36	20000	−980.28	99.99	2683	38.7
DAM	20000	−6.17	95.82	229	4	20000	−172.29	97.18	2818	0.3

In Problems P1 and P2 with Gaussian kernel, DAM clearly outperforms LIBSVM, while the behaviour of this latter is better than the one of DAM in Problem P3. In all the experiments the computational effort of DAM to select the working sets is quite lower than that of LIBSVM. We remark that both the algorithms implement the same technique to avoid kernel evaluations as much as possible. In particular, they dynamically cache only the most recently used columns of the matrix Q (the cache memory size was set to the default value of 40 MB). Thus, although in principle the computational cost per iteration of the two algorithms is different only for the effort in the working set selection, a very high difference in terms of cpu time may occur as consequence of the caching technique. For instance, the difference in the solution of Problem P1 with Gaussian kernel (where the algorithms perform the same number of iterations) is mainly due to the fact that, as result of the caching technique, LIBSVM performs 39989 kernel columns evaluations, versus 20989 performed by DAM. We observe that the caching effect is relevant for the cyclic selection (23), since the first column of Q is naturally cached, while LIBSVM may select quite different pairs in the beginning. This becomes an advantage for the proposed method when the runs are stopped quite early.

According to the very limited experimentation, it would seem that the simple version of Algorithm DAM (where the dimension q of the working set has been set equal to two) could represent a valid alternative to LIBSVM. Moreover, the potential computational advantages of the approach characterizing Algorithm DAM could be exploited much more with larger dimensions of the working sets. Indeed, as q increases, the computational saving in the selection of the working set may become

more and more relevant. However, values of q greater than two imply the adoption of an iterative algorithm for computing the solutions of the generated subproblems. Therefore, the study of efficient iterative solvers for convex quadratic programs deserves particular attention and will be the object of future work.

5 Conclusion

The contribution of the paper is the definition of a decomposition algorithm (DAM), whose global convergence can be proved under mild assumptions. Algorithm DAM is based on a general condition on the working sets that can be satisfied, in particular, by using prefixed index sets in cyclic order. In this case, no on-line computation is required to select the subproblem variables to be optimized, so that a parallel version of the algorithm could be naturally designed. This will be object of further study. It is also worth to mention that in this paper it is assumed that a stationary point of the subproblem (12) is determined exactly, which can be done efficiently in the case of quadratic programming problems of dimension $q = 2$. Therefore, an important point to be investigated is the definition of convergent decomposition methods based on larger working sets and on the computation of inexact solutions of the subproblem (12).

Appendix

Proof of Proposition 2 First we assume that the feasible point x^* is a KKT point of problem (1). If one of the sets $R(x^*)$, $S(x^*)$ is empty, then the assertion of the proposition is obviously true. If both the sets $R(x^*)$ and $S(x^*)$ are not empty, then Proposition 1 implies the existence of a multiplier λ^* such that the pair (x^*, λ^*) satisfies conditions (4) which can be written as follows:

$$\max_{i \in L^+(x^*) \cup U^-(x^*)} \left\{ -\frac{(\nabla f(x^*))_i}{a_i} \right\} \leq \lambda^* \leq \min_{i \in L^-(x^*) \cup U^+(x^*)} \left\{ -\frac{(\nabla f(x^*))_i}{a_i} \right\},$$

$$\lambda^* = -\frac{(\nabla f(x^*))_i}{a_i} \quad \forall i \notin L(x^*) \cup U(x^*).$$

Then recalling the definition of the sets $R(x^*)$ and $S(x^*)$, we can write:

$$\max_{h \in R(x^*)} -\frac{(\nabla f(x^*))_h}{a_h} \leq \min_{h \in S(x^*)} -\frac{(\nabla f(x^*))_h}{a_h},$$

which implies that there exists no pair of indices i and j, with $i \in R(x^*)$ and $j \in S(x^*)$, satisfying (6).

Assume now that there exists no pair of indices i and j, with $i \in R(x^*)$ and $j \in S(x^*)$, satisfying (6). First we consider the case that one of the sets $R(x^*)$, $S(x^*)$ is empty. Suppose, without loss of generality, that $R(x^*) = \emptyset$ which implies that

$\{i\colon l_i < x_i^* < u_i\} = \emptyset$. Hence we have that $S(x^*) = L^-(x^*) \cup U^+(x^*) = \{1, \ldots, n\}$. Therefore conditions (4) are satisfied by choosing any λ^* such that

$$\lambda^* \leq \min_{1 \leq i \leq n} -\frac{(\nabla f(x^*))_i}{a_i}.$$

In case that both the sets $R(x^*)$ and $S(x^*)$ are not empty, by assumption we have that

$$\max_{h \in R(x^*)} -\frac{(\nabla f(x^*))_h}{a_h} \leq \min_{h \in S(x^*)} -\frac{(\nabla f(x^*))_h}{a_h}.$$

Therefore we can define a multiplier λ^* such that

$$\max_{h \in R(x^*)} -\frac{(\nabla f(x^*))_h}{a_h} \leq \lambda^* \leq \min_{h \in S(x^*)} -\frac{(\nabla f(x^*))_h}{a_h}, \tag{24}$$

so that the first and second sets of inequalities of (4) are satisfied. Then the definition of the sets $R(x^*)$, $S(x^*)$ and the choice of the multiplier λ^* (given by (24)) imply that

$$\max_{\{i\colon l_i < x_i < u_i\}} -\frac{(\nabla f(x^*))_i}{a_i} \leq \lambda^* \leq \min_{\{i\colon l_i < x_i < u_i\}} -\frac{(\nabla f(x^*))_i}{a_i},$$

so that the set of equalities of (4) is verified. □

Proof of Proposition 3 If $\hat{x}$ is not a KKT point then Proposition 2 implies that both $R(\hat{x})$ and $S(\hat{x})$ are not empty and ensures that for at least one pair $i \in R(\hat{x})$ and $j \in S(\hat{x})$ we have that

$$-\frac{(\nabla f(\hat{x}))_i}{a_i} > -\frac{(\nabla f(\hat{x}))_j}{a_j}.$$

Hence we get easily

$$\max_{h \in R(\hat{x})} -\frac{(\nabla f(\hat{x}))_h}{a_h} \geq -\frac{(\nabla f(\hat{x}))_i}{a_i} > -\frac{(\nabla f(\hat{x}))_j}{a_j} \geq \min_{h \in S(\hat{x})} -\frac{(\nabla f(\hat{x}))_h}{a_h}$$

and this proves point (i) of the proposition.

Let us prove point (ii). We show that the defined direction $d^{i,j} \in D(\hat{x})$, namely that

$$a' d^{i,j} = 0 \quad \text{and} \quad d_i^{i,j} \geq 0 \quad \forall i \in L(\hat{x}), \quad \text{and} \quad d_j^{i,j} \leq 0 \quad \forall j \in U(\hat{x}).$$

Indeed, the definition of $d^{i,j}$ yields that $a' d^{i,j} = a_i d_i^{i,j} + a_j d_j^{i,j} = 0$. Moreover, we have $i \in R(\hat{x})$, so that, if $i \in L(x)$, then, by (5), we must have $i \in L^+(\hat{x})$, and hence $d_i^{i,j} = 1/a_i > 0$. Analogously, since $j \in S(\hat{x})$, if $j \in U(\hat{x})$ then $j \in U^+(\hat{x})$ and hence $d_j^{i,j} = -1/a_j < 0$. The same conclusion can be drawn for the other two cases.

Furthermore, if a pair $i \in R(\hat{x})$ and $j \in S(\hat{x})$ exists such that (9) holds, then we can write

$$\nabla f(\hat{x})' d^{i,j} = \frac{(\nabla f(\hat{x}))_i}{a_i} - \frac{(\nabla f(\hat{x}))_j}{a_j} < 0.$$

□

Proof of Proposition 4 For each feasible x, the set of the feasible directions at x is the cone $D(x) = \mathcal{N} \cap \mathcal{T}(x)$ where $\mathcal{N} = \{d \in R^n\colon\ a'd = 0\}$ and

$$\mathcal{T}(x) = \big\{d \in R^n\colon\ d_i \geq 0,\ \forall i \in L(x),\ \text{and } d_i \leq 0,\ \forall i \in U(x)\big\}.$$

In order to prove assertion (i), it is sufficient to show that $\mathcal{T}(\bar{x}) \subseteq \mathcal{T}(x^k)$ for sufficiently large values of k. Hence, we prove that, for sufficiently large values of k,

$$L(x^k) \subseteq L(\bar{x}), \qquad U(x^k) \subseteq U(\bar{x}). \tag{25}$$

Assume by contradiction that (25) does not hold and without loss of generality assume that $L(x^k) \not\subseteq L(\bar{x})$. Hence for each k belonging to an infinite subset $K \subseteq \{0, 1, \ldots\}$ an integer j^k exists, such that $j^k \in L(x^k)$ and $j^k \notin L(\bar{x})$. Since j^k belongs to a finite set, we can extract a subset $K_1 \subseteq K$ such that $j^k = \bar{j}$ for each $k \in K_1$. Then we have

$$x_{\bar{j}}^k = l_{\bar{j}} \quad \text{for all } k \in K_1. \tag{26}$$

Taking limits in (26) for $k \to \infty, k \in K_1$, we obtain that $\bar{x}_{\bar{j}} = l_{\bar{j}}$ and this contradicts the fact that $\bar{j} \notin L(\bar{x})$.

Now let us prove assertion (ii). The proof is by contradiction. Assume that an integer $\bar{j}$ exists, such that $\bar{j} \in R(\bar{x})$ and $\bar{j} \notin R(x^k)$ for each $k \geq \bar{k}$. We note that the index sets defined in (5) can be also rewritten in the form:

$$R(x) = \big\{i\colon\ (x_i < u_i \text{ and } a_i > 0) \text{ or } (x_i > l_i \text{ and } a_i < 0)\big\},$$
$$S(x) = \big\{i\colon\ (x_i < u_i \text{ and } a_i < 0) \text{ or } (x_i > l_i \text{ and } a_i > 0)\big\}.$$

We can assume without loss of generality that $a_{\bar{j}} > 0$ so that, by definition of $R(\bar{x})$, we get $\bar{x}_{\bar{j}} < u_{\bar{j}}$. By assumption $\bar{j} \notin R(x^k)$, that implies that $x_{\bar{j}}^k = u_{\bar{j}}$ for $k \geq \bar{k}$. Since $x^k \to \bar{x}$ for $k \to \infty$, this implies $\bar{x}_{\bar{j}} = u_{\bar{j}}$ which leads to a contradiction. □

References

1. Auslender, A.: Asymptotic properties of the Fenchel dual functional and applications to decomposition problems. J. Optim. Theory Appl. **73**, 427–449 (1992)
2. Barr, R.O., Gilbert, E.G.: Some efficient algorithms for a class of abstract optimization problems arising in optimal control. IEEE Trans. Autom. Control **14**, 640–652 (1969)
3. Bertsekas, D.P.: Nonlinear Programming, 2nd edn. Athena Scientific, Belmont (1999)
4. Bertsekas, D., Tseng, P.: Partial proximal minimization algorithm for convex programming. SIAM J. Optim. **4**, 551–572 (1994)
5. Bertsekas, D.P., Tsitsiklis, J.N.: Parallel and Distributed Computation. Prentice-Hall, Englewood Cliffs (1989)
6. Bomze, I.M.: Evolution towards the Maximum clique. J. Glob. Optim. **10**, 143–164 (1997)
7. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines. Software, available at http://www.csie.ntu.edu.tw/~cjlin/libsvm (2001)
8. Cristianini, N., Shawe-Taylor, J.: An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge (2000)
9. Einbu, J.M.: Optimal allocation of continuous resources to several activities with a concave return function—some theoretical results. Math. Oper. Res. **3**, 82–88 (1978)
10. Ferris, M.C., Mangasarian, O.L.: Parallel variable distribution. SIAM J. Optim. **4**, 1–21 (1994)

11. Ferris, M.C., Munson, T.S.: Interior-point methods for massive support vector machines. SIAM J. Optim. **13**, 783–804 (2003)
12. Grippo, L., Sciandrone, M.: Globally convergent block-coordinate techniques for unconstrained optimization. Optim. Methods Softw. **10**(4), 587–637 (1999)
13. Grippo, L., Sciandrone, M.: On the convergence of the block nonlinear Gauss–Seidel method under convex constraints. Oper. Res. Lett. **26**(3), 127–136 (2000)
14. Hearn, D.W., Lawphongpanich, S., Ventura, J.A.: Restricted simplicial decomposition: computation and extensions. Math. Program. Study **31**, 99–118 (1987)
15. Joachims, T.: Making large scale SVM learning practical. In: Schölkopf, C.B.B., Smola, A. (eds.) Advances in Kernel Methods—Support Vector Learning. MIT, Cambridge (1998)
16. Kao, C., Lee, L.-F., Pitt, M.M.: Simulated Maximum Likelihood Estimation of the linear expenditure system with binding non-negativity constraints. Ann. Econ. Finance **2**, 203–223 (2001)
17. Kiwiel, K.C.: A dual method for certain positive semidefinite quadratic problems. SIAM J. Sci. Stat. Comput. **10**, 175–186 (1989)
18. Lin, C.-J.: On the convergence of the decomposition method for support vector machines. IEEE Trans. Neural Netw. **12**, 1288–1298 (2001)
19. Lin, C.-J.: Asymptotic convergence of an SMO algorithm without any assumptions. IEEE Trans. Neural Netw. **13**, 248–250 (2002)
20. Lin, C.-J.: A formal analysis of stopping criteria of decomposition methods for support vector machines. IEEE Trans. Neural Netw. **13**, 1045–1052 (2002)
21. Lucidi, S., Sciandrone, M., Tseng, P.: Objective-derivative-free methods for constrained optimization. Math. Program. **92**(1), 37–59 (2002)
22. Mangasarian, O.L.: Generalized support vector machines. In: Smola, A., Bartlett, P., Schölkopf, B., Schurmans, D. (eds.) Advances in Large Margin Classifiers, pp. 135–146. MIT, Cambridge (2000)
23. Mangasarian, O.L., Musicant, D.R.: Successive overrelaxation for support vector machines. IEEE Trans. Neural Netw. **10**, 1032–1037 (1999)
24. Melman, A., Rabinowitz, G.: An efficient method for a class of continuous knapsack problems. SIAM Rev. **42**, 440–448 (2000)
25. Motzkin, T.S., Strauß, E.G.: Maxima for graphs and a new proof of a theorem of Turan. Can. J. Math. **17**, 533–540 (1965)
26. Nielsen, S.S., Zenios, S.A.: Massively parallel algorithms for singly constrained convex programming. ORSA J. Comput. **4**, 166–181 (1992)
27. Pang, J.S.: A new and efficient algorithm for a class of portfolio selection problem. Oper. Res. **28**, 754–767 (1980)
28. Patriksson, M.: Decomposition methods for differentiable optimization problems over Cartesian product sets. Comput. Optim. Appl. **9**, 5–42 (1998)
29. Platt, J.: Sequential minimal optimization: a fast algorithm for training support vector machines. In: Schölkopf, C.B.B., Smola, A. (eds.) Advances in Kernel Methods—Support Vector Learning, pp. 185–208. MIT, Cambridge (1998)
30. Powell, M.J.D.: On search directions for minimization algorithms. Math. Program. **4**, 193–201 (1973)
31. Tseng, P.: Decomposition algorithms for convex differentiable minimization. J. Optim. Theory Appl. **70**, 109–135 (1991)
32. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York (1995)
33. Ziemba, W.T., Parkan, C., Brooks-Hill, R.: Calculation of investment portfolios with risk free borrowing and lending. Manag. Sci. **21**, 209–222 (1974)