Global Convergence of SMO Algorithm for Support Vector Regression

Author(s): Norikazu Takahashi, Jun Guo and Tetsuo Nishi

Journal: IEEE Transactions on Neural Networks
Volume: 19
Number: 6
Pages: 971–982
Month: June
Year: 2008
Published Version: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4479862

©2008 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Global Convergence of SMO Algorithm for Support Vector Regression

Norikazu Takahashi, *Member, IEEE,* Jun Guo, and Tetsuo Nishi, *Fellow, IEEE*

***Abstract*— Global convergence of the sequential minimal optimization (SMO) algorithm for support vector regression (SVR) is studied in this paper. Given l training samples, SVR is formulated as a convex quadratic programming problem with l pairs of variables. We prove that if two pairs of variables violating the optimality condition are chosen for update in each step and subproblems are solved in a certain way then the SMO algorithm always stops within a finite number of iterations after finding an optimal solution. Also, efficient implementation techniques for the SMO algorithm are presented and compared experimentally with other SMO algorithms.**

***Index Terms*— support vector regression, sequential minimal optimization, convergence, quadratic programming**

I. Introduction

TRAINING of a support vector machine (SVM) for pattern classification is formulated as a quadratic programming (QP) problem in a number of variables equal to the number of training samples [1]. Recently several decomposition methods for solving QP problems arising in the training of SVMs have been proposed [2]–[6]. Sequential minimal optimization (SMO) algorithm proposed by Platt [3] and SVM$^{\mathrm{light}}$ proposed by Joachims [4] are well-known and widely used decomposition methods. The basic strategy commonly used in these methods is to repeat two operations: 1) choosing a fixed number of variables and 2) solving the QP subproblem with respect to the selected variables, until an optimal solution is found. Compared with the direct application of traditional QP solvers, decomposition methods not only use less amount of memory but also require shorter computation time. In addition, some properties on global convergence of the method have recently been clarified [7]–[15].

This paper considers SMO algorithms for support vector regression (SVR). Given l training samples, SVR is formulated as a QP problem with $2l$ variables, say α_1, α_2, ..., α_l, $\hat{\alpha}_1$, $\hat{\alpha}_2$, ..., $\hat{\alpha}_n$, where two variables α_i and $\hat{\alpha}_i$ are closely related to each other.

A simple approach to this problem is to consider the $2l$ variables: α_1, α_2, ..., α_l, $\hat{\alpha}_1$, $\hat{\alpha}_2$, ..., $\hat{\alpha}_n$, are independent from each other and to apply decomposition methods for pattern classification directly. For example, a software library called LIBSVM [16], [17] selects two variables from the $2l$ variables and solves the QP subproblem with respect to the selected variables in each step. This is thought of as a special case of the generalized SMO algorithm [13]. An advantages of this approach is that the global convergence is guaranteed by some existing results concerning the global convergence of the decomposition methods [7]–[15].

On the other hand, many researchers have tried to develop SVR-oriented SMO algorithms by making use of the close relationship between α_i and $\hat{\alpha}_i$. Smola and Schölkopf [18] proposed an SMO algorithm for SVR by extending the basic idea of Platt's SMO. This algorithm selects two pairs of variables (or four variables): α_i, $\hat{\alpha}_i$, α_j and $\hat{\alpha}_j$, in each step according to the strategy similar to Platt's SMO, and solves the QP subproblem with respect to the selected variables analytically by considering four cases depending on the values of two pairs of variables. Shevade *et al.* [19] pointed out that the method for updating the bias in Smola and Schölkopf's algorithm is inefficient, and made some improvements. Also, they proposed a new method for selecting variables based on the degree of violation of the optimality condition. Flake and Lawrence [20] showed that the QP problems with $2l$ variables can be transformed into non-smooth optimization problems with l variables $\beta_i \triangleq \alpha_i - \hat{\alpha}_i$, $i = 1, 2, \ldots, l$, and proposed an SMO algorithm for solving these non-smooth optimization problems. An advantage of Flake and Lawrence's algorithm is that the procedure for solving subproblems can be expressed in a very simple form.

While the global convergence property of SMO algorithms for pattern classification has been clarified by several researchers [12]–[15], this is not the case at all for the above mentioned SMO algorithms for SVR. The most important difference is that SMO algorithms for SVR choose four variables in each step while those for pattern classification choose two. For this reason, the existing results cannot be directly applied to the convergence analysis of SMO algorithms for SVR.

In this paper, we study the global convergence of a general SMO algorithms based on Flake and Lawrence's formulation. By using the same approach as in [13] and [14], we prove that the algorithm reaches an optimal solution within a finite number of iterations if two conditions are satisfied. One is that the algorithm must choose two variables violating the optimality condition in each step. The other is that the values of the selected variables must be updated by using not an exact solution of the subproblem but an approximate solution in a certain situation. One may think this is strange, but, as far as the approach based on [13] and [14] is concerned, the global convergence cannot be proved without this condition.

This work was presented in part at the 2006 International Joint Conference on Neural Networks, Vancouver, BC, Canada, July 16–21, 2006. This work was partly supported by the Okawa Foundation research grant, the Ministry of Education, Culture, Sports, Science and Technology, Grant-in-Aid for JSPS Research Fellows, 18-9473, and the 21st Century COE Program "Reconstruction of Social Infrastructure Related to Information Science and Electrical Engineering".

N. Takahashi and J. Guo are with the Department of Computer Science and Communication Engineering, Kyushu University, Fukuoka, 819-0395 Japan (norikazu@csce.kyushu-u.ac.jp, guojun@kairo.csce.kyushu-u.ac.jp).

T. Nishi is with the Faculty of Science and Engineering, Waseda University, Tokyo, 162-0072 Japan (nishi-t@waseda.jp).

This paper is organized as follows. Section II describes the basic principle of SVR and Flake and Lawrence's formulation as a non-smooth optimization problem. In Section III, the optimality condition for the problem, the notion of violating pair, and a general form of the SMO algorithm are presented. How to solve subproblems is addressed in Section IV. In Section V, we give a rigorous proof for the global convergence of the SMO algorithm. Implementation issues will be discussed in Section VI. Finally, Section VII provides our conclusion.

II. Support Vector Regression

Consider a set of training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{l}$, where $\boldsymbol{x}_i \in \mathbb{R}^d$ is the i-th sample value of the input vector, $y_i \in \mathbb{R}$ is the corresponding value of the model output and l is the number of training samples. The goal of regression is to provide an estimate of the dependence of y on $\boldsymbol{x}$, which can be expressed as

$$y = f(\boldsymbol{x}) + \nu$$

where $f(\cdot)$ is an unknown nonlinear function and ν is a noise independent of $\boldsymbol{x}$.

Basic concepts of SVMs can be applied to regression problems by introducing a certain loss function such as quadratic function, Laplace function, Huber function and ε-insensitive function. Among them, ε-insensitive function is one of the most widely used loss functions because it can produce sparse support vectors. Throughout this paper, we will focus our attention on SVR with the ε-insensitive function defined as

$$|x|_\varepsilon = \begin{cases} 0 & \text{if } |x| < \varepsilon \\ |x| - \varepsilon & \text{otherwise} \end{cases}$$

In SVR, the unknown function $f(\boldsymbol{x})$ is assumed to be expressed in the following form:

$$f_{\mathrm{SVR}}(\boldsymbol{x}) = \boldsymbol{w}^T \phi(\boldsymbol{x}) + b$$

where $\boldsymbol{w}$ is the coefficient vector, b is the bias, and $\phi(\cdot)$ is the pre-specified nonlinear mapping. Values of the coefficient vector and the bias are determined so that the functional defined by

$$C \sum_{i=1}^{l} |y_i - f_{\mathrm{SVR}}(\boldsymbol{x}_i)|_\varepsilon + \|\boldsymbol{w}\|^2$$

is minimized, where C is a positive constant, $|\cdot|_\varepsilon$ is the ε-insensitive loss function defined above, and $||\cdot||$ denotes the Euclidean norm. The dual form of this problem leads to the following convex QP problem with $2l$ variables.

Problem 1: Find $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_l, \hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_l]^T \in \mathbb{R}^{2l}$ that minimizes

$$Q(\boldsymbol{\alpha}) = -\sum_{i=1}^{l} y_i(\alpha_i - \hat{\alpha}_i) + \varepsilon \sum_{i=1}^{l} (\alpha_i + \hat{\alpha}_i) + \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} (\alpha_i - \hat{\alpha}_i)(\alpha_j - \hat{\alpha}_j) K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

subject to the constraints:

$$\sum_{i=1}^{l} (\alpha_i - \hat{\alpha}_i) = 0 \tag{1}$$

$$0 \le \alpha_i \le C, \quad i = 1, 2, \ldots, l \tag{2}$$

$$0 \le \hat{\alpha}_i \le C, \quad i = 1, 2, \ldots, l \tag{3}$$

where $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j)$ and ε is a user-specified positive constant.

Let $\mathcal{F}_1 \subset \mathbb{R}^{2l}$ be the feasible region of Problem 1. $K(\cdot, \cdot)$ in the objective function $Q(\boldsymbol{\alpha})$ is called a kernel function. Throughout this paper, we assume that the kernel function $K(\cdot, \cdot)$ satisfies Mercer's condition. It is thus always guaranteed that the $l \times l$ matrix $\boldsymbol{K} = [k_{ij}]$ with $k_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is positive semi-definite.

Let $\boldsymbol{\alpha}^* = [\alpha_1^*, \alpha_2^*, \ldots, \alpha_l^*, \hat{\alpha}_1^*, \hat{\alpha}_2^*, \ldots, \hat{\alpha}_l^*]^T$ be any optimal solution of Problem 1. Then it follows from the Karush-Kuhn-Tucker (KKT) conditions for Problem 1 that the condition

$$\alpha_i^* \hat{\alpha}_i^* = 0, \quad i = 1, 2, \ldots, l \tag{4}$$

holds [18], [19]. This means that the set of optimal solutions of Problem 1 does not change even though the condition

$$\alpha_i \hat{\alpha}_i = 0, \quad i = 1, 2, \ldots, l \tag{5}$$

is added to the constraints (1)–(3). Let us now introduce new variables $\beta_i = \alpha_i - \hat{\alpha}_i$ for $i = 1, 2, \ldots, l$. Then, since $\alpha_i + \hat{\alpha}_i$ can be expressed as $|\beta_i|$ under the conditions (2), (3) and (5), Problem 1 can be reformulated in terms of $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_l]^T$ as follows [20]:

Problem 2: Find $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_l]^T \in \mathbb{R}^l$ that minimizes

$$W(\boldsymbol{\beta}) = -\sum_{i=1}^{l} y_i \beta_i + \varepsilon \sum_{i=1}^{l} |\beta_i| + \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \beta_i \beta_j k_{ij} \tag{6}$$

subject to the constraints:

$$\sum_{i=1}^{l} \beta_i = 0$$

$$-C \le \beta_i \le C, \quad i = 1, 2, \ldots, l$$

Note that the objective function $W(\boldsymbol{\beta})$ is neither quadratic nor differentiable any more. However, it is still convex and therefore Problem 2 has no local minimum. In the following, the feasible region of Problem 2 is denoted by $\mathcal{F}_2 \subset \mathbb{R}^l$. It is apparent that $\mathcal{F}_2$ is not empty.

Let $\boldsymbol{\beta}^* = [\beta_1^*, \beta_2^*, \ldots, \beta_l^*]^T$ be any optimal solution of Problem 2. Then the approximating function $f_{\mathrm{SVR}}(\boldsymbol{x})$ derived by SVR can be expressed as

$$f_{\mathrm{SVR}}(\boldsymbol{x}) = \sum_{i=1}^{l} \beta_i^* K(\boldsymbol{x}_i, \boldsymbol{x}) + b$$

where the value of b is determined so that

$$f_{\mathrm{SVR}}(\boldsymbol{x}_m) = \sum_{i=1}^{l} \beta_i^* K(\boldsymbol{x}_i, \boldsymbol{x}_m) + b = y_m - \varepsilon$$

holds for any m such that $0 < \beta_m^* < C$ or

$$f_{\mathrm{SVR}}(\boldsymbol{x}_n) = \sum_{i=1}^{l} \beta_i^* K(\boldsymbol{x}_i, \boldsymbol{x}_n) + b = y_n + \varepsilon$$

holds for any n such that $-C < \beta_n^* < 0$ [20].

III. SMO Algorithm for SVR

In this section, we present a general form of SMO algorithms for solving Problem 2. Before doing so, we derive the optimality condition for Problem 2. We also introduce the notion of violating pairs.

A. Optimality Condition

As shown in [19], it follows from the KKT condition for Problem 1 that $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_l, \hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_l] \in \mathcal{F}_1$ is an optimal solution if and only if there exists a constant λ such that the following conditions are satisfied for $i = 1, 2, \ldots, l$.

1) If $0 < \alpha_i < C$ and $\hat{\alpha}_i = 0$ then $y_i - \varepsilon - \sum_{j=1}^{l}(\alpha_j - \hat{\alpha}_j)k_{ij} = \lambda$.
2) If $\alpha_i = 0$ and $0 < \hat{\alpha}_i < C$ then $y_i + \varepsilon - \sum_{j=1}^{l}(\alpha_j - \hat{\alpha}_j)k_{ij} = \lambda$.
3) If $\alpha_i = C$ and $\hat{\alpha}_i = 0$ then $y_i - \varepsilon - \sum_{j=1}^{l}(\alpha_j - \hat{\alpha}_j)k_{ij} \geq \lambda$.
4) If $\alpha_i = 0$ and $\hat{\alpha}_i = C$ then $y_i + \varepsilon - \sum_{j=1}^{l}(\alpha_j - \hat{\alpha}_j)k_{ij} \leq \lambda$.
5) If $\alpha_i = \hat{\alpha}_i = 0$ then $y_i - \varepsilon - \sum_{j=1}^{l}(\alpha_j - \hat{\alpha}_j)k_{ij} \leq \lambda$ and $y_i + \varepsilon - \sum_{j=1}^{l}(\alpha_j - \hat{\alpha}_j)k_{ij} \geq \lambda$.

Substituting $\beta_i = \alpha_i - \hat{\alpha}_i$ into these conditions, we can derive the optimality conditions for Problem 2 as follows:

$$\begin{cases} h_i^+(\boldsymbol{\beta}) \leq -\lambda, & \text{if } \beta_i = C \\ h_i^+(\boldsymbol{\beta}) = -\lambda, & \text{if } 0 < \beta_i < C \\ h_i^-(\boldsymbol{\beta}) \leq -\lambda \leq h_i^+(\boldsymbol{\beta}), & \text{if } \beta_i = 0 \\ h_i^-(\boldsymbol{\beta}) = -\lambda, & \text{if } -C < \beta_i < 0 \\ h_i^-(\boldsymbol{\beta}) \geq -\lambda, & \text{if } \beta_i = -C \end{cases} \tag{7}$$

where

$$h_i^{\pm}(\boldsymbol{\beta}) = -y_i \pm \varepsilon + \sum_{j=1}^{l} k_{ij}\beta_j. \tag{8}$$

Since (7) is rewritten as

$$\begin{aligned} \max_{\beta_i = C} h_i^+(\boldsymbol{\beta}) &\leq -\lambda \\ \max_{0 < \beta_i < C} h_i^+(\boldsymbol{\beta}) \leq -\lambda &\leq \min_{0 < \beta_i < C} h_i^+(\boldsymbol{\beta}) \\ \max_{\beta_i = 0} h_i^-(\boldsymbol{\beta}) \leq -\lambda &\leq \min_{\beta_i = 0} h_i^+(\boldsymbol{\beta}) \\ \max_{-C < \beta_i < 0} h_i^-(\boldsymbol{\beta}) \leq -\lambda &\leq \min_{-C < \beta_i < 0} h_i^-(\boldsymbol{\beta}) \\ -\lambda &\leq \min_{\beta_i = -C} h_i^-(\boldsymbol{\beta}) \end{aligned}$$

we can express the optimality condition by a single inequality as follows:

$$\begin{aligned} \max\left\{ \max_{0 < \beta_i \leq C} h_i^+(\boldsymbol{\beta}), \max_{-C < \beta_i \leq 0} h_i^-(\boldsymbol{\beta}) \right\} \\ \leq \min\left\{ \min_{-C \leq \beta_i < 0} h_i^-(\boldsymbol{\beta}), \min_{0 \leq \beta_i < C} h_i^+(\boldsymbol{\beta}) \right\}. \end{aligned} \tag{9}$$

Let $D_i^+W(\boldsymbol{\beta})$ and $D_i^-W(\boldsymbol{\beta})$ denote the right-hand and left-hand partial derivatives of $W(\boldsymbol{\beta})$ with respect to β_i, that is,

$$\begin{aligned} D_i^+W(\boldsymbol{\beta}) &= \lim_{\delta \to 0+} \frac{W(\boldsymbol{\beta} + \delta \boldsymbol{e}_i) - W(\boldsymbol{\beta})}{\delta} \\ D_i^-W(\boldsymbol{\beta}) &= \lim_{\delta \to 0-} \frac{W(\boldsymbol{\beta} + \delta \boldsymbol{e}_i) - W(\boldsymbol{\beta})}{\delta} \end{aligned}$$

where $\boldsymbol{e}_i$ is the vector whose i-th element is $+1$ and others are zero. Then it is easily seen that in $\mathcal{F}_2$ these partial derivatives can be expressed in terms of $h_i^+(\boldsymbol{\beta})$ and $h_i^-(\boldsymbol{\beta})$ defined in (8) as follows:

$$\begin{aligned} D_i^+W(\boldsymbol{\beta}) &= \begin{cases} h_i^+(\boldsymbol{\beta}), & \beta_i \geq 0 \\ h_i^-(\boldsymbol{\beta}), & \beta_i < 0 \end{cases} \\ D_i^-W(\boldsymbol{\beta}) &= \begin{cases} h_i^+(\boldsymbol{\beta}), & \beta_i > 0 \\ h_i^-(\boldsymbol{\beta}), & \beta_i \leq 0 \end{cases} \end{aligned}$$

Therefore, the optimality condition (9) can be rewritten in a simpler form as

$$\max_{-C < \beta_i \leq C} D_i^-W(\boldsymbol{\beta}) \leq \min_{-C \leq \beta_i < C} D_i^+W(\boldsymbol{\beta}). \tag{10}$$

Since the SMO algorithm generally does not provide an exact optimal solution in a finite number of iterations, we will consider hereafter the relaxed optimality condition:

$$\max_{-C < \beta_i \leq C} D_i^-W(\boldsymbol{\beta}) \leq \min_{-C \leq \beta_i < C} D_i^+W(\boldsymbol{\beta}) + \tau \tag{11}$$

instead of (10), where τ is a positive tolerance parameter. Eq.(11) is called the τ-optimality condition.

B. Violating Pair

With the strict optimality condition (10), we can define the violating pair as follows; a pair of indices (i, j) is said to be a violating pair at $\boldsymbol{\beta} \in \mathcal{F}_2$ if the following conditions hold:

$$-C < \beta_i \leq C, \;\; -C \leq \beta_j < C, \;\; D_i^-W(\boldsymbol{\beta}) > D_j^+W(\boldsymbol{\beta})$$

It is obvious that the optimality condition (10) holds at $\boldsymbol{\beta}$ if and only if there exists no violating pair at $\boldsymbol{\beta}$. Similarly, we can define the violating pair corresponding to (11). A pair of indices (i, j) satisfying

$$-C < \beta_i \leq C, \;\; -C \leq \beta_j < C, \;\; D_i^-W(\boldsymbol{\beta}) > D_j^+W(\boldsymbol{\beta}) + \tau \tag{12}$$

is called a τ-violating pair at $\boldsymbol{\beta} \in \mathcal{F}_2$. The τ-optimality condition (11) holds at $\boldsymbol{\beta}$ if and only if there exists no τ-violating pair at $\boldsymbol{\beta}$. It is very important to note here that two statements "(i, j) is a τ-violating pair" and "(j, i) is a τ-violating pair" must be distinguished in this paper.

C. SMO Algorithm

A general SMO algorithm for solving Problem 2 is described as follows:

Algorithm 1: Given training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{l}$, kernel function $K(\cdot, \cdot)$ and positive constants C, ε and τ, execute the following procedure.

Step 1: Set $n := 0$ and choose an initial point $\boldsymbol{\beta}^{(0)} \in \mathcal{F}_2$.

Step 2: If $\boldsymbol{\beta} = \boldsymbol{\beta}^{(n)}$ satisfies the τ-optimality condition (11) then stop.

Step 3: Choose a τ-violating pair $(i^{(n)}, j^{(n)})$ at $\boldsymbol{\beta}^{(n)}$ and solve Problem 2 under the additional constraints $\beta_k = \beta_k^{(n)}, \forall k \notin \{i^{(n)}, j^{(n)}\}$. Let $\boldsymbol{\beta}^{(n+1)}$ be the point thus obtained. Set $n := n + 1$ and go to Step 2.

It is apparent that the objective function $W(\boldsymbol{\beta})$ is bounded from below in the feasible region $\mathcal{F}_2$ and the sequence

$\{W(\boldsymbol{\beta}^{(n)})\}_{n=0}^{\infty}$ obtained by Algorithm 1 is monotone decreasing, that is, $W(\boldsymbol{\beta}^{(n+1)}) \leq W(\boldsymbol{\beta}^{(n)})$ holds for all n. Therefore $W(\boldsymbol{\beta}^{(n)})$ necessarily converges to a certain value. On the other hand, however, it is not clear whether or not Algorithm 1 always stops within a finite number of iterations. To make clear the condition which guarantees the termination of Algorithm 1 is one of the main goal in this paper.

Since the optimization problems arising in Step 3, which will be called subproblems in the following, have only two variables, they can be solved very easily. In the next section, we will describe in detail how these subproblems can be solved.

IV. How to Solve Subproblems

As a general form of subproblems arising in Step 3 of Algorithm 1, we will consider in this section the following problem: Given a point $\tilde{\boldsymbol{\beta}} = [\tilde{\beta}_1, \tilde{\beta}_2, \ldots, \tilde{\beta}_l]^T \in \mathcal{F}_2$ and two indices $i, j \in \{1, 2, \ldots, l\}$ such that (i, j) is a τ-violating pair at $\tilde{\boldsymbol{\beta}}$, find $\boldsymbol{\beta}$ that minimizes $W(\boldsymbol{\beta})$ under the constraints $\boldsymbol{\beta} \in \mathcal{F}_2$ and $\beta_k = \tilde{\beta}_k, \forall k \neq i, j$. This problem has essentially only two variables β_i and β_j and they must satisfy $\beta_i + \beta_j = \tilde{\beta}_i + \tilde{\beta}_j$. Therefore it is convenient for us to parameterize $\boldsymbol{\beta}$ as $\boldsymbol{\beta}(t) = [\beta_1(t), \beta_2(t), \ldots, \beta_l(t)]^T$ where

$$\beta_i(t) = \tilde{\beta}_i + t, \quad \beta_j(t) = \tilde{\beta}_j - t, \quad \beta_k(t) = \tilde{\beta}_k, \ \ \forall k \neq i, j.$$

Then the problem can be reformulated as follows:

Problem 3: Find t that minimizes

$$\begin{aligned}\psi(t) &= W(\boldsymbol{\beta}(t)) \\ &= (-y_i + \sum_{k=1}^{l} \tilde{\beta}_k k_{ik} + y_j - \sum_{k=1}^{l} \tilde{\beta}_k k_{jk})t \\ &\quad + \frac{1}{2}\eta t^2 + \varepsilon|\tilde{\beta}_i + t| + \varepsilon|\tilde{\beta}_j - t| + W_c\end{aligned}$$

subject to $-C \leq \beta_i(t) \leq C$ and $-C \leq \beta_j(t) \leq C$, where $\eta = k_{ii} + k_{jj} - 2k_{ij} \geq 0$ and W_c represents a collection of constant terms.

We now introduce some notations for later discussions. First, according to the notation used in [13], we express (β_i, β_j) as β_{ij} for simplicity. We also use $\beta_{ij}^{(n)}$ and $\beta_{ij}(t)$ in the same sense. Let S be the square region defined by $S = [-C, C] \times [-C, C] \subset \mathbb{R}^2$. Let S_1, S_2, S_3 and S_4 be four square regions defined as follows:

$$\begin{aligned}S_1 &= \{\beta_{ij} \in S \,:\, 0 < \beta_i \leq C,\ 0 \leq \beta_j < C\} \\ S_2 &= \{\beta_{ij} \in S \,:\, -C < \beta_i \leq 0,\ 0 \leq \beta_j < C\} \\ S_3 &= \{\beta_{ij} \in S \,:\, -C < \beta_i \leq 0,\ -C \leq \beta_j < 0\} \\ S_4 &= \{\beta_{ij} \in S \,:\, 0 < \beta_i \leq C,\ -C \leq \beta_j < 0\}\end{aligned}$$

The interior and boundary of each region S_i are denoted by intS_i and ∂S_i, respectively. Similarly, the interior and boundary of the region S are denoted by int$S$ and ∂S, respectively. Vertices of four square regions S_1, S_2, S_3 and S_4 are denoted by $V_1 = (C, 0)$, $V_2 = (C, C)$, $V_3 = (0, C)$, $V_4 = (-C, C)$, $V_5 = (-C, 0)$, $V_6 = (-C, -C)$, $V_7 = (0, -C)$ and $V_8 = (C, -C)$. Also, the sides excluding vertices of square regions, are denoted by $E_1, E_2, \ldots, E_{12}$ as

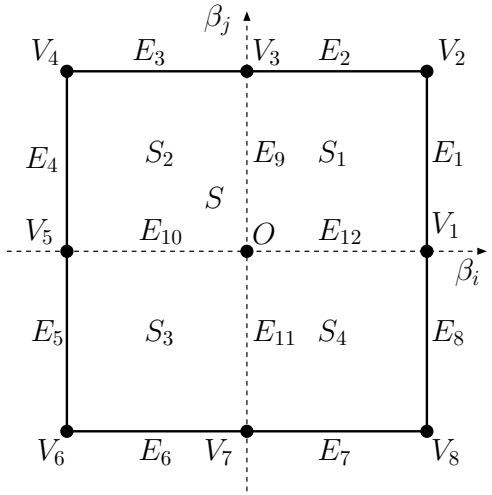

Fig. 1. Square regions $S, S_1, \ldots, S_4$, vertices $V_1, V_2, \ldots, V_8$, and sides $E_1, E_2, \ldots, E_{12}$.

$$\begin{aligned}E_1 &= \{\beta_{ij} \in S \,:\, \beta_i = C,\ 0 < \beta_j < C\}, \\ E_2 &= \{\beta_{ij} \in S \,:\, 0 < \beta_i < C,\ \beta_j = C\},\end{aligned}$$

and so on (see Fig.1).

Figure 2 shows the diagonal line on which the point $\beta_{ij}(t)$ moves with t, and the graph of the objective function $\psi(t)$ versus t. It is obvious that the objective function $\psi(t)$ is convex and hence there is no local minimum. However, since $\psi(t)$ is not differentiable at $t = -\tilde{\beta}_i$ and $t = \tilde{\beta}_j$, we need to consider two types of derivatives: the right-hand and left-hand derivatives defined by

$$\begin{aligned}\psi'_+(t) &= \lim_{\delta \to 0+} \frac{\psi(t+\delta) - \psi(t)}{\delta} \\ \psi'_-(t) &= \lim_{\delta \to 0-} \frac{\psi(t+\delta) - \psi(t)}{\delta}\end{aligned}$$

at these two points. With these notations, we can say, for example, that $t = t^*$ is an optimal solution of Problem 3 if $\beta_{ij}(t^*)$ is an interior point of S and $\psi'_-(t^*) \leq 0 \leq \psi'_+(t^*)$ holds. Here we should note that $\psi'_-(t) = \psi'_+(t)$ holds for any $t \notin \{-\tilde{\beta}_i, \tilde{\beta}_j\}$.

Lemma 1: The following equations hold for the derivatives of $\psi(t)$ and $W(\boldsymbol{\beta})$.

$$\psi'_+(t) = D_i^+ W(\boldsymbol{\beta}(t)) - D_j^- W(\boldsymbol{\beta}(t)) \tag{13}$$

$$\psi'_-(t) = D_i^- W(\boldsymbol{\beta}(t)) - D_j^+ W(\boldsymbol{\beta}(t)) \tag{14}$$

Proof: We will prove only (13) because (14) can be done in the same way. Since $\mathrm{d}\beta_i(t)/\mathrm{d}t = 1$ and $\mathrm{d}\beta_j(t)/\mathrm{d}t = -1$, the right-hand derivative of $\psi(t)$ at $t = 0$ can be expressed as follows:

$$\begin{aligned}\psi'_+(0) &= D_i^+ W(\tilde{\boldsymbol{\beta}}) \cdot \frac{\mathrm{d}\beta_i(t)}{\mathrm{d}t} + D_j^- W(\tilde{\boldsymbol{\beta}}) \cdot \frac{\mathrm{d}\beta_j(t)}{\mathrm{d}t} \\ &= D_i^+ W(\tilde{\boldsymbol{\beta}}) - D_j^- W(\tilde{\boldsymbol{\beta}})\end{aligned} \tag{15}$$

Furthermore, since $\psi'_+(t) = \psi'_+(0)\big|_{\tilde{\boldsymbol{\beta}} = \boldsymbol{\beta}(t)}$ holds, by substituting $\tilde{\boldsymbol{\beta}} = \boldsymbol{\beta}(t)$ into (15) we can derive (13). ■

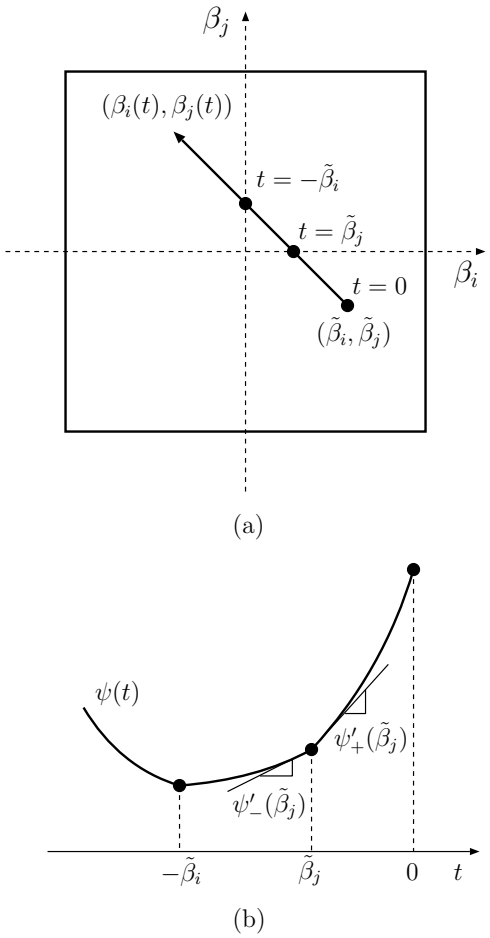

Fig. 2. (a) Diagonal line represented by $(\beta_i(t), \beta_j(t))$. (b) Objective function $\psi(t)$ and its right-hand and left-hand derivatives.

Lemma 2: The pair of indices (i, j) is a τ-violating pair at $\boldsymbol{\beta}(t)$ if and only if

$$-C < \beta_i(t) \le C, \quad -C \le \beta_j(t) < C, \quad \psi'_-(t) > \tau\,.$$

Proof: The conditions can be immediately derived by substituting (14) into (12). ■

Lemma 3: Let t^* be an optimal solution of Problem 3. Then $t^* < 0$.

Proof: Since (i, j) is a τ-violating pair at $\boldsymbol{\beta}(0) = \tilde{\boldsymbol{\beta}}$, it follows from Lemma 2 that $\tilde{\beta}_i > -C$, $\tilde{\beta}_j < C$ and $\psi'_-(0) > \tau > 0$. Since $\psi(t)$ is convex, t^* must be negative. ■

Lemma 4: t^* is an optimal solution of Problem 3 if and only if one of the following two conditions hold.

1) $\beta_{ij}(t^*) \in \text{int}S$ and $\psi'_-(t^*) \le 0 \le \psi'_+(t^*)$
2) $\beta_{ij}(t^*) \in \left(\cup_{k=3}^5 \{V_k\}\right) \cup \left(\cup_{k=2}^5 E_k\right)$ and $\psi'_+(t^*) \ge 0$

Proof: It is apparent from the convexity of $\psi(t)$ and Lemma 3. ■

We easily see from these lemmas that an optimal solution t^* can be found by decreasing the value of t from 0 until $\beta_{ij}(t)$ satisfies one of two conditions given in Lemma 4. This search process is formally stated as follows.

Algorithm 2: For Problem 3, find t^* by executing the following procedure.

Step 1: Set $m := 0$ and $t_m := 0$.

Step 2: If $\eta \neq 0$ then set $t_{\min} := -\psi'_-(t_m)/\eta$. Otherwise, set $t_{\min} := -\infty$.

Step 3: Let S_k be the region to which $\beta_{ij}(t_m)$ belongs. If $\beta_{ij}(t_{\min}) \in S_k$ then set $t^* := t_{\min}$ and stop. Otherwise, set $\hat{t}_{\min} := \inf\{t : \beta_{ij}(t) \in S_k\}$.

Step 4: If either i) $\beta_{ij}(\hat{t}_{\min}) \in \partial S$ or ii) $\beta_{ij}(\hat{t}_{\min}) \in \text{int}S$ and $\psi'_-(\hat{t}_{\min}) \le 0$ holds, set $t^* := \hat{t}_{\min}$ and stop. Otherwise set $m := m + 1$ and $t_m := \hat{t}_{\min}$ and go to Step 2.

One can easily see that t^* found by Algorithm 2 is always an optimal solution of Problem 3. However, we will hereafter not consider Algorithm 2 but Algorithm 3 given below, which is obtained by replacing the condition $\psi'_-(\hat{t}_{\min}) \le 0$ in Step 4 of Algorithm 2 with $\psi'_-(\hat{t}_{\min}) \le \tau$ where τ is the positive tolerance parameter introduced in (11). The reason why we do not consider Algorithm 2 will be explained later.

Algorithm 3: For Problem 3, find t^{**} by executing the following procedure.

Step 1: Set $m := 0$ and $t_m := 0$.

Step 2: If $\eta \neq 0$ then set $t_{\min} := -\psi'_-(t_m)/\eta$. Otherwise, set $t_{\min} := -\infty$.

Step 3: Let S_k be the region to which $\beta_{ij}(t_m)$ belongs. If $\beta_{ij}(t_{\min}) \in S_k$ then set $t^{**} := t_{\min}$ and stop. Otherwise, set $\hat{t}_{\min} := \inf\{t : \beta_{ij}(t) \in S_k\}$.

Step 4: If either i) $\beta_{ij}(\hat{t}_{\min}) \in \partial S$ or ii) $\beta_{ij}(\hat{t}_{\min}) \in \text{int}S$ and $\psi'_-(\hat{t}_{\min}) \le \tau$ holds, set $t^{**} := \hat{t}_{\min}$ and stop. Otherwise set $m := m + 1$ and $t_m := \hat{t}_{\min}$ and go to Step 2.

Difference in the behavior of Algorithms 2 and 3 is illustrated in Fig.3 where it is assumed that $\psi(t)$ is minimized at $t = t_A$, the point $(\beta_1(t_A), \beta_2(t_A))$ belongs to S, $\psi(t)$ is not smooth at $t = t_B \in (t_A, 0)$, and the derivatives of $\psi(t)$ at $t = t_B$ satisfy $0 < \psi'_-(t_B) \le \tau < \psi'_+(t_B)$. In this case, Algorithm 2 returns the optimal solution t_A while Algorithm 3 returns the approximate solution t_B.

The following lemma shows some important properties of Algorithm 3.

Lemma 5: Let t^{**} be the solution obtained by Algorithm 3. Then the following statements hold true.

1) $\psi(t^{**}) < \psi(0)$.
2) The right-hand derivative of $\psi(t)$ at $t = t^{**}$ satisfies
$$\psi'_+(t^{**}) \ge 0$$
in any case. The left-hand derivative satisfies
$$\psi'_-(t^{**}) \le \tau \tag{16}$$
if $\beta_{ij}(t^{**}) \in \text{int}S$. In particular,
$$\psi'_-(t^{**}) = \psi'_+(t^{**}) = 0$$
holds if $\beta_{ij}(t^{**}) \in \cup_{k=1}^4 \text{int}S_k$.
3) The pair (i, j) is not a τ-violating pair at $\boldsymbol{\beta}(t^{**})$.
4) The following inequality holds:
$$\psi(0) - \psi(t^{**}) \ge \frac{\tau}{2\sqrt{2}} \|\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}(t^{**})\| \tag{17}$$
where $\|\cdot\|$ is the Euclidean norm.

Proof: It is obvious from Lemma 3 and Algorithm 3 that the first and second statements hold true. Substituting (14) into (16), we have

$$D_i^- W(\boldsymbol{\beta}(t^{**})) \le D_j^+ W(\boldsymbol{\beta}(t^{**})) + \tau$$

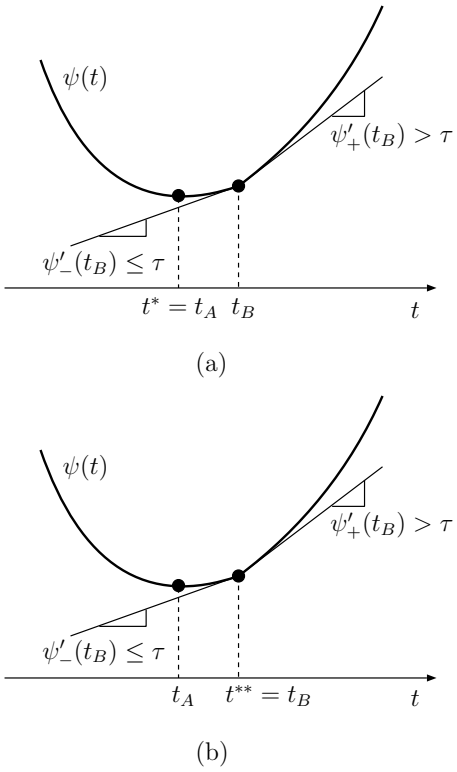

Fig. 3. Comparison between (a) Algorithm 2 and (b) Algorithm 3.

which implies that (i, j) is not a τ-violating pair. Thus the third statement holds true. The fourth statement can be proved in the same way as [13, Lemma 1]. Let $\bar{m}$ be the value of m when Algorithm 3 stops. Then there are three possible cases: $\bar{m} = 0$, $\bar{m} = 1$ and $\bar{m} = 2$. We will consider here only the last one, because it is the most general among three. In this case, $\psi(t)$ can be expressed as

$$\psi(t) = \frac{1}{2}\eta(t - t_2)^2 + \psi'_-(t_2)(t - t_2) + \psi(t_2)$$

in the interval $t^{**} \le t \le t_2$, where we easily see from Step 4 of Algorithm 3 that

$$\psi'_-(t_2) > \tau\,. \tag{18}$$

Let us show now that

$$\psi(t_2) - \psi(t^{**}) \ge \frac{\tau}{2}(t_2 - t^{**}) \tag{19}$$

holds for any $\eta \ge 0$. Suppose first that $\eta = 0$. Since $\psi(t)$ is linear in this case, we have from (18) that $\psi(t_2) - \psi(t^{**}) > \tau(t_2 - t^{**})$. Thus (19) apparently holds. Suppose next that $\eta > 0$. If the constraint $\beta_{ij}(t) \in S$ is ignored, the function $\psi(t)$ is minimized at $t = t_Q \triangleq t_2 - \psi'_-(t_2)/\eta < 0$. Since $t_Q \le t^{**} \le t_2$ and $\psi(t)$ is convex, we have

$$\frac{t_2 - t^{**}}{t_2 - t_Q} \cdot \psi(t_Q) + \frac{t^{**} - t_Q}{t_2 - t_Q} \cdot \psi(t_2) \ge \psi(t^{**})\,.$$

Since $\psi(t_Q) = \psi(t_2) - (\psi'_-(t_2))^2/(2\eta)$, the above inequality can be transformed as follows:

$$\psi(t_2) - \psi(t^{**}) \ge \frac{t_2 - t^{**}}{t_2 - t_Q} \cdot \frac{(\psi'_-(t_2))^2}{2\eta}$$

Substituting $t_Q = t_2 - \psi'_-(t_2)/\eta$ and taking (18) into account, we derive (19). Therefore (19) holds for any $\eta \ge 0$. Applying the above argument to the intervals $t_2 \le t \le t_1$ and $t_1 \le t \le 0$, we have

$$\psi(t_1) - \psi(t_2) \ge \frac{\tau}{2}(t_1 - t_2) \tag{20}$$

$$\psi(0) - \psi(t_1) \ge \frac{\tau}{2}(-t_1) \tag{21}$$

From (19)-(21), we obtain

$$\psi(0) - \psi(t^{**}) \ge \frac{\tau}{2}(-t^{**}) = \frac{\tau}{2\sqrt{2}}\|\bar{\boldsymbol{\beta}} - \boldsymbol{\beta}(t^{**})\|$$

which completes the proof. ■

We should note that the inequality like (17) cannot be derived for Algorithm 2. Inequality (17) plays an important role in our convergence proof for Algorithm 1, which will be presented in the next section. This is the main reason why we consider Algorithm 3 instead of Algorithm 2.

V. Global Convergence of Algorithm 1

In this section, we prove the convergence of Algorithm 1 under the assumption that the subproblems in Step 3 are solved approximately by Algorithm 3. Although the subproblems are not solved exactly, it is apparently true that $\boldsymbol{\beta}^{(n)}$ belongs to the feasible region $\mathcal{F}_2$ of Problem 2 for all n and that $W(\boldsymbol{\beta}^{(n)})$ is monotone decreasing with respect to n. This implies that the sequence $\{W(\boldsymbol{\beta}^{(n)})\}_{n=0}^{\infty}$ converges to a certain value since $W(\boldsymbol{\beta})$ is bounded from below in $\mathcal{F}_2$.

The following theorem is the main result of this paper.

Theorem 1: Algorithm 1 stops in a finite number of iterations for any $\tau > 0$ if subproblems in Step 3 are solved approximately by Algorithm 3.

For any execution of Algorithm 1, we define the set of integers $L(p, q)$ as

$$L(p, q) = \{n : (i^{(n)}, j^{(n)}) = (p, q)\}$$

where (p, q) is any pair of indices. Let I_∞ be the set of pairs (p, q) such that $|L(p, q)| = \infty$ where $|L(p, q)|$ represents the cardinality of the set $L(p, q)$. Obviously an execution of the Algorithm 1 stops within a finite number of iterations if and only if $I_\infty = \emptyset$.

The first step to prove Theorem 1 is to show that the sequence $\{\boldsymbol{\beta}^{(n)}\}_{n=1}^{\infty}$ is convergent.

Lemma 6: The sequence $\{\boldsymbol{\beta}^{(n)}\}_{n=0}^{\infty}$ generated by Algorithm 1 converges to a point in $\mathcal{F}_2$ as $n \to \infty$.

Proof: Since $\{W(\boldsymbol{\beta}^{(n)})\}_{n=0}^{\infty}$ is a decreasing sequence that is bounded from below, it converges to some value as $n \to \infty$. It follows from (17) that

$$\frac{2\sqrt{2}}{\tau}\left(W(\boldsymbol{\beta}^{(n)}) - W(\boldsymbol{\beta}^{(n+1)})\right) \ge \|\boldsymbol{\beta}^{(n)} - \boldsymbol{\beta}^{(n+1)}\|\,.$$

By repeated application of the triangle inequality we get

$$\frac{2\sqrt{2}}{\tau}\left(W(\boldsymbol{\beta}^{(n)}) - W(\boldsymbol{\beta}^{(n+k)})\right) \ge \|\boldsymbol{\beta}^{(n)} - \boldsymbol{\beta}^{(n+k)}\|$$

which means $\{\boldsymbol{\beta}^{(n)}\}_{n=0}^{\infty}$ is a Cauchy sequence. Since $\mathcal{F}_2$ is closed, $\{\boldsymbol{\beta}^{(n)}\}_{n=0}^{\infty}$ converges to some point in $\mathcal{F}_2$. ■

The limit point of the sequence $\{\boldsymbol{\beta}^{(n)}\}_{n=0}^{\infty}$ will be denoted by $\bar{\boldsymbol{\beta}}$ in the following.

Lemma 7: Let (p,q) be any pair belonging to I_∞. There exists an n_0 such that $\beta_{pq}^{(n+1)}$ belongs to $\cup_{k=1}^4 \partial S_k$ for all $n \in L(p,q) \geq n_0$.

Proof: We will only show in the following that

$$\beta_{pq}^{(n+1)} \in \text{int}S_2 \tag{22}$$

does not hold infinitely many times. The proofs for the regions S_1, S_3 and S_4 are omitted because they are similar to the one given here. Assume that (22) holds for infinitely many n's in $L(p,q)$. Then, at least one of the following two cases occurs infinitely many times.

1) $\beta_{pq}^{(n)} \in S_2$ and $\beta_{pq}^{(n+1)} \in \text{int}S_2$
2) $\beta_{pq}^{(n)} \in S_1 \cup S_3 \cup S_4$ and $\beta_{pq}^{(n+1)} \in \text{int}S_2$

We first consider Case 1). Let $M_1(p,q) \subseteq L(p,q)$ be the set of integers n such that $\beta_{pq}^{(n)} \in S_2$ and $\beta_{pq}^{(n+1)} \in \text{int}S_2$. Since (p,q) is a τ-violating pair at $\boldsymbol{\beta}^{(n)}$ for all $n \in M_1(p,q)$,

$$D_p^- W(\boldsymbol{\beta}^{(n)}) - D_p^+ W(\boldsymbol{\beta}^{(n)}) = h_p^-(\boldsymbol{\beta}^{(n)}) - h_q^+(\boldsymbol{\beta}^{(n)}) > \tau$$

holds for all $n \in M_1(p,q)$. By taking the limit, we have

$$h_p^-(\bar{\boldsymbol{\beta}}) - h_q^+(\bar{\boldsymbol{\beta}}) \geq \tau\,. \tag{23}$$

On the other hand, since $\beta_{pq}^{(n+1)} \in \text{int}S_2$ for all $n \in M_1(p,q)$, it follows from Part 2) of Lemma 5 that

$$D_p^- W(\boldsymbol{\beta}^{(n+1)}) - D_p^+ W(\boldsymbol{\beta}^{(n+1)}) = h_p^-(\boldsymbol{\beta}^{(n+1)}) - h_q^+(\boldsymbol{\beta}^{(n+1)}) = 0$$

holds for all $n \in M_1(p,q)$. Thus we have

$$h_p^-(\bar{\boldsymbol{\beta}}) - h_q^+(\bar{\boldsymbol{\beta}}) = 0$$

which contradicts (23). We next consider Case 2). In this case, $\bar{\beta}_{pq}$ belongs to $\partial S_2 \cap (\partial S_1 \cup \partial S_3 \cup \partial S_4) = \{V_3\} \cup E_9 \cup \{O\} \cup E_{10} \cup \{V_5\}$. Let $M_2(p,q) \subseteq L(p,q)$ be the set of integers n such that $\beta_{pq}^{(n)} \in S_1 \cup S_3 \cup S_4$ and $\beta_{pq}^{(n+1)} \in \text{int}S_2$. Since $\beta_{pq}^{(n+1)} \in \text{int}S_2$ for all $n \in M_2(p,q)$,

$$D_p^- W(\boldsymbol{\beta}^{(n+1)}) - D_p^+ W(\boldsymbol{\beta}^{(n+1)}) = h_p^-(\boldsymbol{\beta}^{(n+1)}) - h_q^+(\boldsymbol{\beta}^{(n+1)}) = 0$$

holds for infinitely many n's. By taking the limit, we have

$$h_p^-(\bar{\boldsymbol{\beta}}) - h_q^+(\bar{\boldsymbol{\beta}}) = 0 \tag{24}$$

Let θ_n be the positive number satisfying

$$(1-\theta_n)\beta_{pq}^{(n)} + \theta_n \beta_{pq}^{(n+1)} \in \partial S_2$$

and let us define $\boldsymbol{\gamma}^{(n)}$ as

$$\boldsymbol{\gamma}^{(n)} = (1-\theta_n)\boldsymbol{\beta}^{(n)} + \theta_n \boldsymbol{\beta}^{(n+1)}$$

Since $\gamma_{pq}^{(n)} \in \partial S_2 \cap (\partial S_1 \cup \partial S_3 \cup \partial S_4)$ converges to $\bar{\beta}_{pq}$, we see from (24) that there exists an n_1 such that

$$h_p^-(\boldsymbol{\gamma}^{(n)}) - h_q^+(\boldsymbol{\gamma}^{(n)}) \leq \tau,\ \forall n \in M_2(p,q) \geq n_1\,.$$

This equation together with the fact that $\beta_{pq}^{(n+1)} \in \text{int}S_2$ contradicts Step 4 of Algorithm 3. ■

Lemma 8: If $(p,q) \in I_\infty$ then $\bar{\beta}_{pq}$ belongs to $\cup_{k=1}^4 \partial S_k$.

Proof: We prove the thesis by contradiction. Assume without loss of generality that $\bar{\beta}_{pq} \in \text{int}S_1$. Since $\text{int}S_1$ is an open set, there exists an n_1 such that $\beta_{pq}^{(n)} \in \text{int}S_1,\ \forall n \geq n_1$. However, this contradicts Lemma 7. ■

Lemma 9: There exists an n_0 such that $\beta_{i^{(n)}j^{(n)}}^{(n)} \in \cup_{k=1}^4 \partial S_k$ and $\beta_{i^{(n)}j^{(n)}}^{(n+1)} \in \cup_{k=1}^4 \partial S_k$ for all $n \geq n_0$.

Proof: It is easily seen from Lemma 7 and the definition of I_∞ that there exists an n_1 such that $(i^{(n)}, j^{(n)}) \in I_\infty$ and $\beta_{i^{(n)}j^{(n)}}^{(n+1)} \in \cup_{k=1}^4 \partial S_k$ hold for all $n \geq n_1$. Therefore for each $n \geq n_1$ there are two possibilities: i) $\beta_{i^{(n)}j^{(n)}}^{(n)} \in \cup_{k=1}^4 \partial S_k$ and ii) $\beta_{i^{(n)}j^{(n)}}^{(n)} \in \cup_{k=1}^4 \text{int}S_k$. Let $B^{(n)}$ denote the number of components of $\boldsymbol{\beta}^{(n)}$ satisfying $\beta_i^{(n)} \in \{-C, 0, C\}$. In Case i) $B^{(n+1)}$ is equal to $B^{(n)}$ or greater than $B^{(n)}$ by one. On the other hand, in Case ii) $B^{(n+1)}$ is greater than $B^{(n)}$ by at least one. Therefore Case ii) cannot occur infinitely many times after n reaches n_1 because otherwise $B^{(n)}$ goes to infinity, which of course contradicts the fact that the number of components of $\boldsymbol{\beta}^{(n)}$ is finite. ■

From the above discussions we see that for any pair $(p,q) \in I_\infty$ one of the following situations occurs.

a) $\beta_{pq}^{(n)} \in E_1$ and $\beta_{pq}^{(n+1)} \in E_2$ hold for infinitely many n's in $L(p,q)$. In this case, $\bar{\beta}_{pq} = V_2 = (C,C)$ and the following inequality holds.
$$h_p^+(\bar{\boldsymbol{\beta}}) \geq h_q^+(\bar{\boldsymbol{\beta}}) + \tau \tag{25}$$
b) $\beta_{pq}^{(n)} \in E_{12}$ and $\beta_{pq}^{(n+1)} \in E_9$ hold for infinitely many n's in $L(p,q)$. In this case, $\bar{\beta}_{pq} = O = (0,0)$ and the inequality (25) holds.
c) $\beta_{pq}^{(n)} \in E_9$ and $\beta_{pq}^{(n+1)} \in E_3$ hold for infinitely many n's in $L(p,q)$. In this case, $\bar{\beta}_{pq} = V_3 = (0,C)$ and the following inequality holds.
$$h_p^-(\bar{\boldsymbol{\beta}}) \geq h_q^+(\bar{\boldsymbol{\beta}}) + \tau \tag{26}$$
d) $\beta_{pq}^{(n)} \in E_{10}$ and $\beta_{pq}^{(n+1)} \in E_4$ hold for infinitely many n's in $L(p,q)$. In this case, $\bar{\beta}_{pq} = V_5 = (-C,0)$ and the inequality (26) holds.
e) $\beta_{pq}^{(n)} \in E_{11}$ and $\beta_{pq}^{(n+1)} \in E_{10}$ hold for infinitely many n's in $L(p,q)$. In this case, $\bar{\beta}_{pq} = O = (0,0)$ and the following inequality holds.
$$h_p^-(\bar{\boldsymbol{\beta}}) \geq h_q^-(\bar{\boldsymbol{\beta}}) + \tau \tag{27}$$
f) $\beta_{pq}^{(n)} \in E_6$ and $\beta_{pq}^{(n+1)} \in E_5$ hold for infinitely many n's in $L(p,q)$. In this case, $\bar{\beta}_{pq} = V_6 = (-C,-C)$ and the inequality (27) holds.
g) $\beta_{pq}^{(n)} \in E_8$ and $\beta_{pq}^{(n+1)} \in E_{12}$ hold for infinitely many n's in $L(p,q)$. In this case, $\bar{\beta}_{pq} = V_1 = (C,0)$ and the following inequality holds.
$$h_p^+(\bar{\boldsymbol{\beta}}) \geq h_q^-(\bar{\boldsymbol{\beta}}) + \tau \tag{28}$$
h) $\beta_{pq}^{(n)} \in E_7$ and $\beta_{pq}^{(n+1)} \in E_{11}$ hold for infinitely many n's in $L(p,q)$. In this case, $\bar{\beta}_{pq} = V_7 = (0,-C)$ and the inequality (28) holds.

The eight possible movements from $\beta_{pq}^{(n)}$ to $\beta_{pq}^{(n+1)}$ for sufficiently large $n \in L(p,q)$ are shown in Fig.4.

We are now ready for giving the proof of Theorem 1.

Proof of Theorem 1: Suppose that Algorithm 1 does not stop. Then I_∞ has at least one pair. Let (p,q) be any pair in

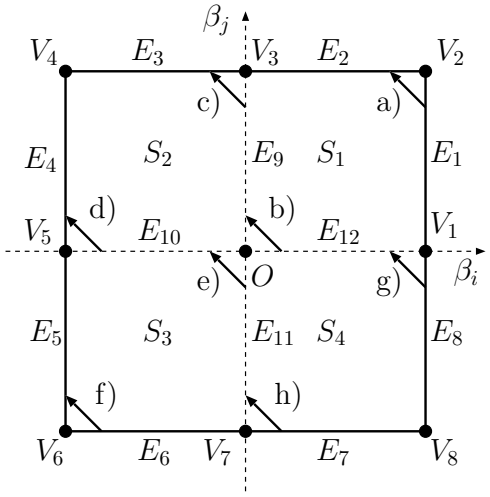

Fig. 4. Eight possible movements from $\beta_{pq}^{(n)}$ to $\beta_{pq}^{(n+1)}$.

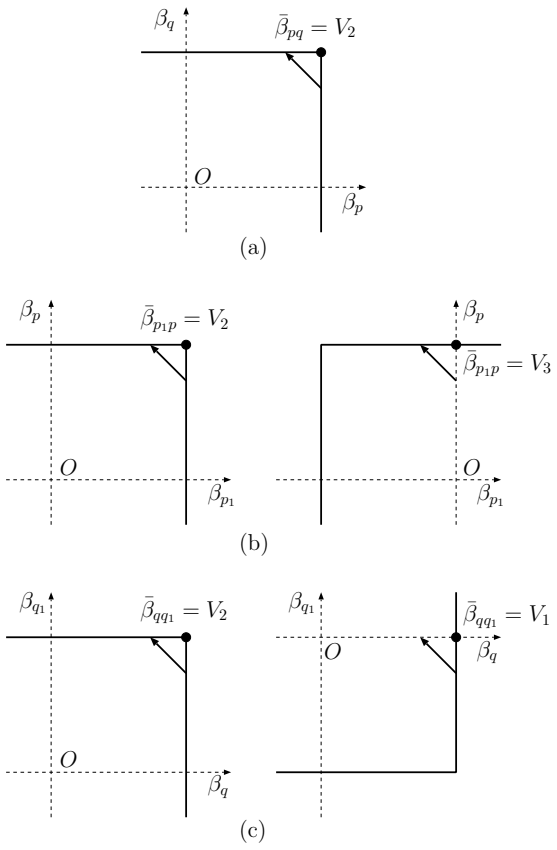

Fig. 5. Possible movements of (a) $\beta_{pq}^{(n)}$, (b) $\beta_{p_1p}^{(n)}$ and (c) $\beta_{qq_1}^{(n)}$ for sufficiently large n.

I_∞. In case a) described above, since $\beta_{pq}^{(n)}$ moves from E_1 to E_2 infinitely many times, it must return from E_2 to E_1 infinitely many times. In order for this to occur, there must exist an index p_1 such that $\beta_{p_1p}^{(n)}$ moves either i) from E_1 to E_2 or ii) from E_9 to E_3 infinitely many times, and an index q_1 such that $\beta_{qq_1}^{(n)}$ moves either i) from E_1 to E_2 or ii) from E_8 to E_{12} infinitely many times (see Fig.5). Therefore (p_1, p) belongs to I_∞ and one of the following conditions holds:

$$\bar{\beta}_{p_1p} = (C, C), \quad h_{p_1}^+(\bar{\boldsymbol{\beta}}) \geq h_p^+(\bar{\boldsymbol{\beta}}) + \tau \tag{29}$$

$$\bar{\beta}_{p_1p} = (0, C), \quad h_{p_1}^-(\bar{\boldsymbol{\beta}}) \geq h_p^+(\bar{\boldsymbol{\beta}}) + \tau \tag{30}$$

Similarly (q, q_1) belongs to I_∞ and one of the following conditions holds:

$$\bar{\beta}_{qq_1} = (C, C), \quad h_q^+(\bar{\boldsymbol{\beta}}) \geq h_{q_1}^+(\bar{\boldsymbol{\beta}}) + \tau \tag{31}$$

$$\bar{\beta}_{qq_1} = (C, 0), \quad h_q^+(\bar{\boldsymbol{\beta}}) \geq h_{q_1}^-(\bar{\boldsymbol{\beta}}) + \tau \tag{32}$$

We easily see from (29)–(32) that p, q, p_1 and q_1 are different from each other. For example, in the case where $\bar{\beta}_{p_1p} = (C, C)$ and $\bar{\beta}_{qq_1} = (C, C)$, we have from (25), (29) and (31)

$$h_{q_1}^+(\bar{\boldsymbol{\beta}}) < h_q^+(\bar{\boldsymbol{\beta}}) < h_p^+(\bar{\boldsymbol{\beta}}) < h_{p_1}^+(\bar{\boldsymbol{\beta}})$$

Thus p, q, p_1 and q_1 are different from each other. In the case where $\bar{\beta}_{p_1p} = (0, C)$ and $\bar{\beta}_{qq_1} = (C, 0)$, it is obvious that $p_1 \neq p, q$ and $q_1 \neq p, q$. Moreover, since

$$h_{q_1}^-(\bar{\boldsymbol{\beta}}) < h_q^+(\bar{\boldsymbol{\beta}}) < h_p^+(\bar{\boldsymbol{\beta}}) < h_{p_1}^-(\bar{\boldsymbol{\beta}})$$

holds from (25), (30) and (32), we can conclude that $p_1 \neq q_1$.

In each case b) to h), there must be two indices p_1 and q_1 such that $(p_1, p) \in I_\infty$, $(q, q_1) \in I_\infty$, and p, q, p_1 and q_1 are different from each other, as we have seen in case a). Relationship between (p, q) and (p_1, p) for cases a) – h) is summarized in Table I, and relationship between (p, q) and (q, q_1) for cases a) – h) is summarized in Table II.

In the following, we will show from Table I that I_∞ must contain infinitely many pairs. We will not consider Table II, but the same discussion can be made by using Table II. Let us first suppose that one of the cases a), c), e) and g) occurs for $(p, q) \in I_\infty$. Then, by applying the relationship shown in Table I repeatedly, we see that there must exist an infinite sequence of indices $p_0(= p), p_1, p_2, \ldots$ such that

$$(p_{i+1}, p_i) \in I_\infty, \quad i = 0, 1, 2, \ldots \tag{33}$$

$$\bar{\beta}_{p_i} \in \{0, C\}, \quad i = 0, 1, 2, \ldots \tag{34}$$

$$h_{p_0}^{s_0}(\bar{\boldsymbol{\beta}}) < h_{p_1}^{s_1}(\bar{\boldsymbol{\beta}}) < h_{p_2}^{s_2}(\bar{\boldsymbol{\beta}}) < \cdots \tag{35}$$

where s_i represents "+" if $\bar{\beta}_{p_i} = C$ and "−" if $\bar{\beta}_{p_i} = 0$. Eq.(35) implies that all indices $p, p_1, p_2, \ldots$ are different from each other. Therefore, I_∞ must contain infinitely many pairs. Let us next suppose that one of the cases b), d), f) and h) occurs for $(p, q) \in I_\infty$. Then there must exist an infinite sequence of indices $p_0(= p), p_1, p_2, \ldots$ such that

$$(p_{i+1}, p_i) \in I_\infty, \quad i = 0, 1, 2, \ldots \tag{36}$$

$$\bar{\beta}_{p_i} \in \{0, -C\}, \quad i = 0, 1, 2, \ldots \tag{37}$$

$$h_{p_0}^{s_0}(\bar{\boldsymbol{\beta}}) < h_{p_1}^{s_1}(\bar{\boldsymbol{\beta}}) < h_{p_2}^{s_2}(\bar{\boldsymbol{\beta}}) < \cdots \tag{38}$$

where s_i represents "+" if $\bar{\beta}_{p_i} = 0$ and "−" if $\bar{\beta}_{p_i} = -C$. Eq.(38) implies that all indices $p, p_1, p_2, \ldots$ are different each other. Therefore, I_∞ must contain infinitely many pairs.

Since these results apparently contradict the fact that the number of different pairs contained in I_∞ is finite, we can conclude that I_∞ is empty, that is, Algorithm 1 always stops within a finite number of iterations. ■

TABLE I
RELATION BETWEEN (p, q) AND (p_1, p).

$\bar{\beta}_{pq}$	$\bar{\beta}_{p_1 p}$	$h_p^{\pm}(\bar{\boldsymbol{\beta}})$, $h_{p_1}^{\pm}(\bar{\boldsymbol{\beta}})$
a) (C, C)	a) (C, C)	$h_p^+(\bar{\boldsymbol{\beta}}) < h_{p_1}^+(\bar{\boldsymbol{\beta}})$
	c) $(0, C)$	$h_p^+(\bar{\boldsymbol{\beta}}) < h_{p_1}^-(\bar{\boldsymbol{\beta}})$
b) $(0, 0)$	b) $(0, 0)$	$h_p^+(\bar{\boldsymbol{\beta}}) < h_{p_1}^+(\bar{\boldsymbol{\beta}})$
	d) $(-C, 0)$	$h_p^+(\bar{\boldsymbol{\beta}}) < h_{p_1}^-(\bar{\boldsymbol{\beta}})$
c) $(0, C)$	e) $(0, 0)$	$h_p^-(\bar{\boldsymbol{\beta}}) < h_{p_1}^-(\bar{\boldsymbol{\beta}})$
	g) $(C, 0)$	$h_p^-(\bar{\boldsymbol{\beta}}) < h_{p_1}^+(\bar{\boldsymbol{\beta}})$
d) $(-C, 0)$	f) $(-C, -C)$	$h_p^-(\bar{\boldsymbol{\beta}}) < h_{p_1}^-(\bar{\boldsymbol{\beta}})$
	h) $(0, -C)$	$h_p^-(\bar{\boldsymbol{\beta}}) < h_{p_1}^+(\bar{\boldsymbol{\beta}})$
e) $(0, 0)$	e) $(0, 0)$	$h_p^-(\bar{\boldsymbol{\beta}}) < h_{p_1}^-(\bar{\boldsymbol{\beta}})$
	g) $(C, 0)$	$h_p^-(\bar{\boldsymbol{\beta}}) < h_{p_1}^+(\bar{\boldsymbol{\beta}})$
f) $(-C, -C)$	f) $(-C, -C)$	$h_p^-(\bar{\boldsymbol{\beta}}) < h_{p_1}^-(\bar{\boldsymbol{\beta}})$
	h) $(0, -C)$	$h_p^-(\bar{\boldsymbol{\beta}}) < h_{p_1}^+(\bar{\boldsymbol{\beta}})$
g) $(C, 0)$	a) (C, C)	$h_p^+(\bar{\boldsymbol{\beta}}) < h_{p_1}^+(\bar{\boldsymbol{\beta}})$
	c) $(0, C)$	$h_p^+(\bar{\boldsymbol{\beta}}) < h_{p_1}^-(\bar{\boldsymbol{\beta}})$
h) $(0, -C)$	b) $(0, 0)$	$h_p^+(\bar{\boldsymbol{\beta}}) < h_{p_1}^+(\bar{\boldsymbol{\beta}})$
	d) $(-C, 0)$	$h_p^+(\bar{\boldsymbol{\beta}}) < h_{p_1}^-(\bar{\boldsymbol{\beta}})$

TABLE II
RELATION BETWEEN (p, q) AND (q, q_1).

$\bar{\beta}_{pq}$	$\bar{\beta}_{qq_1}$	$h_q^{\pm}(\bar{\boldsymbol{\beta}})$, $h_{q_1}^{\pm}(\bar{\boldsymbol{\beta}})$
a) (C, C)	a) (C, C)	$h_{q_1}^+(\bar{\boldsymbol{\beta}}) < h_q^+(\bar{\boldsymbol{\beta}})$
	g) $(C, 0)$	$h_{q_1}^-(\bar{\boldsymbol{\beta}}) < h_q^+(\bar{\boldsymbol{\beta}})$
b) $(0, 0)$	b) $(0, 0)$	$h_{q_1}^+(\bar{\boldsymbol{\beta}}) < h_q^+(\bar{\boldsymbol{\beta}})$
	h) $(0, -C)$	$h_{q_1}^-(\bar{\boldsymbol{\beta}}) < h_q^+(\bar{\boldsymbol{\beta}})$
c) $(0, C)$	a) (C, C)	$h_{q_1}^+(\bar{\boldsymbol{\beta}}) < h_q^+(\bar{\boldsymbol{\beta}})$
	g) $(C, 0)$	$h_{q_1}^-(\bar{\boldsymbol{\beta}}) < h_q^+(\bar{\boldsymbol{\beta}})$
d) $(-C, 0)$	b) $(0, 0)$	$h_{q_1}^+(\bar{\boldsymbol{\beta}}) < h_q^+(\bar{\boldsymbol{\beta}})$
	h) $(0, -C)$	$h_{q_1}^-(\bar{\boldsymbol{\beta}}) < h_q^+(\bar{\boldsymbol{\beta}})$
e) $(0, 0)$	c) $(0, C)$	$h_{q_1}^+(\bar{\boldsymbol{\beta}}) < h_q^-(\bar{\boldsymbol{\beta}})$
	e) $(0, 0)$	$h_{q_1}^-(\bar{\boldsymbol{\beta}}) < h_q^-(\bar{\boldsymbol{\beta}})$
f) $(-C, -C)$	d) $(-C, 0)$	$h_{q_1}^+(\bar{\boldsymbol{\beta}}) < h_q^-(\bar{\boldsymbol{\beta}})$
	f) $(-C, -C)$	$h_{q_1}^-(\bar{\boldsymbol{\beta}}) < h_q^-(\bar{\boldsymbol{\beta}})$
g) $(C, 0)$	c) $(0, C)$	$h_{q_1}^+(\bar{\boldsymbol{\beta}}) < h_q^-(\bar{\boldsymbol{\beta}})$
	e) $(0, 0)$	$h_{q_1}^-(\bar{\boldsymbol{\beta}}) < h_q^-(\bar{\boldsymbol{\beta}})$
h) $(0, -C)$	d) $(-C, 0)$	$h_{q_1}^+(\bar{\boldsymbol{\beta}}) < h_q^-(\bar{\boldsymbol{\beta}})$
	f) $(-C, -C)$	$h_{q_1}^-(\bar{\boldsymbol{\beta}}) < h_q^-(\bar{\boldsymbol{\beta}})$

VI. IMPLEMENTATION AND EXPERIMENTS

A. Implementation of SMO Algorithm

The authors have recently studied in detail how to implement the SMO algorithm discussed in the previous section [21]. In the following, we will briefly review their results.

Convergence rate of the SMO algorithm strongly depends on how to choose a violating pair in each iteration. The method used in [21] is to choose $i^{(n)}$ and $j^{(n)}$ so that the following two conditions are satisfied.

$$i^{(n)} \in \{i : D_i^- W(\boldsymbol{\beta}^{(n)}) \geq D_k^- W(\boldsymbol{\beta}^{(n)}),\ \forall k\} \tag{39}$$

$$j^{(n)} \in \{j : D_j^+ W(\boldsymbol{\beta}^{(n)}) \leq D_k^+ W(\boldsymbol{\beta}^{(n)}),\ \forall k\} \tag{40}$$

Since $\phi'_{-}(t)$ is expressed as (14), the objective function $W(\boldsymbol{\beta})$ is expected to decrease most rapidly when $i^{(n)}$ and $j^{(n)}$ satisfy (39) and (40), respectively. In this sense, this method is based on the steepest descent. It is apparent that any pair $(i^{(n)}, j^{(n)})$ satisfying (39) and (40) is a τ-violating pair because there exists at least one τ-violating pair (i, j) at $\boldsymbol{\beta}^{(n)}$, which satisfies $D_i^- W(\boldsymbol{\beta}^{(n)}) > D_j^+ W(\boldsymbol{\beta}^{(n)}) + \tau$, as far as $\boldsymbol{\beta}^{(n)}$ does not satisfy the τ-optimality condition.

The method expressed by (39) and (40) is identical to the one proposed by Shevada *et al.* [19]. On the other hand, Smola and Schölkopf [18] and Flake and Lawrence [20] used methods based on Platt's SMO algorithm [3].

How to implement Algorithm 3 is another important problem. Of course we can encode the algorithm exactly same as described in Section IV, but by using the technique introduced by Flake and Lawrence [20] we can obtain a code which is more compact and faster.

The pseudo-code given in [21] is shown in Fig.6 where $\mathrm{sgn}(u)$ takes 1 if $u \geq 0$ and -1 otherwise, and $\mathrm{step}(u)$ takes 1 if $u \geq 0$ and 0 otherwise. The first part (lines 4–20) is a generalization of the code given in [20]. On the other hand, the second part (lines 21–39) was newly added by the authors, because the case where $\eta = 0$ was not considered in [20].

It is important to note that the pseudo-code shown in Fig.6 is equivalent to Algorithm 3 only if $\tau < \epsilon$. However, since the value of τ is usually set to a sufficiently small positive number comparing with ϵ (see [19] for example), this assumption will not be violated in practice.

B. Experiments

In order to examine the efficiency of the authors' SMO algorithm [21], we compare the following six algorithms in terms of CPU time and the number of iterations for several benchmark data sets.

1) The algorithm used in LIBSVM version 2.71 or earlier [16], [17]. Two variables which most violate the KKT condition for Problem 1 are selected from $2l$ variables. The relationship between α_i and $\hat{\alpha}_i$ is not considered explicitly. Subproblems are solved exactly.
2) Smola and Schölkopf's SMO algorithm [18]. Two pairs of variables α_i, $\hat{\alpha}_i$, α_j and $\hat{\alpha}_j$ are selected according to the strategy similar to Platt's SMO algorithm. Subproblems are solved exactly.
3) Shevada *et al.*'s SMO algorithm [19]. Two pairs of variables α_i, $\hat{\alpha}_i$, α_j and $\hat{\alpha}_j$ which most violate the KKT condition for Problem 1 are selected. Subproblems are solved exactly.
4) Flake and Lawrence's SMO algorithm [20]. Two variables β_i and β_j are selected according to the strategy similar to Platt's SMO algorithm. Subproblems are solved exactly.
5) The authors' SMO algorithm [21].

```
1.  s := β̄_i + β̄_j;
2.  η := k_ii + k_jj − 2k_ij;
3.  v := −y_i + Σ_{n=1}^{l} k_in β̄_n + y_j − Σ_{n=1}^{l} k_jn β̄_n;
4.  if (η > 0) {
5.     β_i := β̄_i + (τ − v)/η;
6.     β_j := s − β_i;
7.     if (β_i · β_j > 0) {
8.        m := β_i · step(β_i) − β_j · (1 − step(β_i));
9.        Δ := τ/η;
10.       β_i := β_i − min{m, Δ};
11.    elseif (β_i · β_j < 0) {
12.       Δ := (2ε + τ · sgn(β_i))/η;
13.       β_i := β_i − min{|β_i|, |β_j|, Δ} · sgn(β_i);
14.    }
15.    if (s ≥ 0 and β_i < s − C)
16.       β_i := s − C;
17.    elseif (s < 0 and β_i < −C)
18.       β_i := −C;
19.    β_j := s − β_i;
20. }
21. else {
22.    if (β̄_i ≤ 0 and β̄_j ≥ 0)
23.       β_i := s · step(s) − C;
24.    elseif ((β̄_i > 0 and β̄_j ≥ 0) or (β̄_i ≤ 0 and β̄_j < 0)) {
25.       if (|s| ≥ C or v > τ + 2ε)
26.          β_i := s · step(s) − C;
27.       else
28.          β_i := s · (1 − step(s));
29.    }
30.    else {
31.       if (v > τ + 2ε)
32.          β_i := s · step(s) − C;
33.       elseif (v > τ)
34.          β_i := s · (1 − step(s));
35.       else
36.          β_i := s · step(s);
37.    }
38.    β_j := s − β_i;
39. }
```

Fig. 6. Pseudo code of Algorithm 3

6) The same algorithm as 5) except that subproblems are solved exactly by Algorithm 2 which is encoded in a similar way as the pseudo code shown in Fig.6.

Although some new variable selection methods have recently been proposed for pattern classification problems [22]–[24], we do not consider these methods in our experiments because it is not clear whether they can be applied to Problem 2 directly. However, it is of course important to verify the applicability of these methods to Problem 2. This will be another research topic for future.

All the above algorithms were implemented in MATLAB and run on a PC with Intel Xeon dual processor 2.8GHz and 2GB of RAM. As the kernel function, we used the RBF kernel defined by $K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\|\boldsymbol{x} - \boldsymbol{y}\|^2/(2\sigma^2))$. Benchmark data sets are from UCI Machine Learning Repository [25], StatLib [26] and [20], and listed in Table III[1] where the dimension of the input patterns (d), the total number of samples ($l_{\rm total}$), the number of samples used for training (l), and the values of parameters ε, τ, C and σ are specified. The values of $\boldsymbol{x}_i$ and y_i in training samples are normalized before training so that $\boldsymbol{x}_i \in [-1, 1]^d$ and $y_i \in [-1, 1]$ hold for all i. The values of hyperparameters ε, τ and C are fixed to 0.1, 0.01 and 10, respectively, for all data sets. On the other hand, the value of the kernel parameter σ was selected from $\{0.01, 0.05, 0.1, 0.5\}$ so that the error rate for test samples is minimized.

[1]All data sets in Table III are also available at [16].

TABLE III
BASIC INFORMATION ON BENCHMARK DATA SETS AND PARAMETER SETTING

Data	d	$l_{\rm total}$	l	ε	τ	C	σ
abalone	8	4177	1000	0.1	0.01	10	0.5
bodyfat	14	252	100	0.1	0.01	10	0.1
housing	13	506	200	0.1	0.01	10	0.1
mg	6	1385	600	0.1	0.01	10	0.5
mpg	7	392	200	0.1	0.01	10	0.5
pyrim	27	74	50	0.1	0.01	10	0.01
space-ga	6	3107	1000	0.1	0.01	10	0.1
trizaines	60	186	100	0.1	0.01	10	0.05

For each benchmark data sets, we generate ten sets of l training samples which are randomly selected from the whole training samples, and apply the six different algorithms to those ten sets of training samples. Experimental results are summarized in Table IV where the number of iterations, CPU time and the number of jumps are presented. Each value is obtained by averaging the results for ten sets. The number of jumps represents how many transitions between regions S_1, S_2, S_3 and S_4 in Fig.1 occurred in solving QP subproblems. Although global convergence is theoretically guaranteed only for Methods 1 and 5, τ-optimal solutions were always reached in our experiments.

We first compare the six methods simply in terms of CPU time. It is seen from Table IV that Method 1, the algorithm used in LIBSVM version 2.71 or earlier, is the fastest for six benchmark data sets. Methods 3, 5 and 6 are slower than Method 1 but the difference is not so significant. Methods 2 and 4 are the slowest. In particular, for *mg*, they are about four time slower than others. In order to explain these results, we focus our attention on the number of jumps. Table IV shows that those methods which select variables most violating the optimality condition (Methods 3, 5 and 6) require a much smaller number of jumps than those methods which select variables in a way similar to Platt's SMO algorithms (Methods 2 and 4). Since all variables are set to zero initially, this means that the former group of methods select variables more appropriately than the latter one. Table IV also shows that the number of jumps is zero or extremely small compared to the number of iterations for Methods 3, 5 and 6. These experimental results strongly support a conclusion from [17, Theorem 2] which says that no jump occurs at the final stage. It is obvious that the advantage of updating two variables β_i and β_j in Methods 5 and 6 (or two pairs of variables α_i, $\hat{\alpha}_i$, α_j and $\hat{\alpha}_j$ in Methods 2, 3 and 4) becomes small as the number of jumps decreases. In particular, if no jump occurs, these

TABLE IV
EXPERIMENTAL RESULTS

Data	Method	Iteration	CPU Time (sec)	Jumps
abalone	1	27566.0	29.1142	
	2	34146.2	38.9869	119.3
	3	27480.6	29.0673	28.8
	4	34088.8	38.3673	117.7
	5	27219.7	29.0467	28.7
	6	27480.6	29.1021	28.8
bodyfat	1	66.1	0.3465	
	2	402.8	0.5306	37.6
	3	66.1	0.3491	0
	4	411.9	0.5318	38.4
	5	66.1	0.3488	0
	6	66.1	0.3493	0
housing	1	140.8	1.2289	
	2	935.2	2.0986	131.2
	3	140.8	1.2492	0
	4	906.9	2.0513	125.6
	5	140.8	1.2364	0
	6	140.8	1.2380	0
mg	1	18779.5	16.5560	
	2	48012.3	38.2765	518.9
	3	18649.8	16.5473	29.2
	4	48758.4	38.3542	532.1
	5	18741.1	16.5728	29.3
	6	18649.8	16.5095	29.2
mpg	1	1541.5	1.1145	
	2	4876.8	4.0746	401.8
	3	1544.7	1.1583	5.3
	4	4962.7	4.1538	410.3
	5	1555.0	1.1685	5.4
	6	1544.7	1.1567	5.3
pyrim	1	64.5	0.0972	
	2	181.9	0.1453	14.9
	3	64.5	0.1046	0
	4	186.7	0.1484	15.0
	5	64.5	0.1012	0
	6	64.5	0.1037	0
space-ga	1	265.8	6.1267	
	2	1364.5	9.1324	263.7
	3	265.8	6.1358	0
	4	1327.8	9.0638	257.1
	5	265.8	6.1338	0
	6	265.8	6.1325	0
trizaines	1	219.6	0.3621	
	2	644.8	0.8346	20.1
	3	219.6	0.3687	0
	4	684.8	0.8665	21.6
	5	219.6	0.3649	0
	6	219.6	0.3652	0

methods will be slower than Method 1 because the procedure for solving QP subproblems is unnecessarily complicated. This is a possible reason why Method 1 is faster than Methods 3, 5 and 6.

We next evaluate the effectiveness of the pseudo code of Algorithm 3 shown in Fig. 6. This evaluation can be done by comparing Methods 3 and 6 because these two methods differ only in the procedure for finding exact solutions of QP subproblems. It is seen from Table IV that Method 6 is faster than Method 3 for most data sets but the difference is very small. Therefore the implementation technique shown in Fig. 6 did not have the intended effect for the benchmark data sets used in our experiments.

We finally consider the influence of using Algorithm 3 for solving QP subproblems instead of Algorithm 2 on the performance of the SMO algorithm by comparing Methods 5 and 6. It is seen from Table IV that there is no significant difference between these two methods. This is because the behavior of Algorithms 2 and 3 differ only when the jump occurs and the number of jumps is very small for all benchmark data sets.

VII. CONCLUSION

We have proved that if a τ-violating pair is chosen for update in each step and subproblems are solved by Algorithm 3 then the SMO algorithm stops within a finite number of iterations after finding a τ-optimal solution. On the other hand, however, it is still not clear whether or not the SMO algorithm converges to a τ-optimal solution when subproblems are solved exactly. This may be a future problem to be attacked. Another future problem is to study global convergence of more general decomposition algorithms for SVR where more than two variables are updated in each step. This issue is also mentioned in [8].

REFERENCES

[1] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[2] E. Osuna, R. Freund, and F. Girosi, "An improved training algorithm for support vector machines," in *Proceedings of the 1997 IEEE Workshop on Neural Networks for Signal Processing*, 1997, pp. 511–519.

[3] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods: Support Vector Machines*, B. Schölkopf, C. Burges, and A. Smola, Eds. Cambridge, MA: MIT Press, 1998.

[4] T. Joachims, "Making large-scale support vector machine learning practical," in *Advances in Kernel Methods: Support Vector Machines*, B. Schölkopf, C. Burges, and A. Smola, Eds. Cambridge, MA: MIT Press, 1998.

[5] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, "Improvements to Platt's SMO algorithm for SVM classifier design," *Neural Computation*, vol. 13, pp. 637–649, 2001.

[6] C.-W. Hsu and C.-J. Lin, "A simple decomposition method for support vector machines," *Machine Learning*, vol. 46, pp. 291–314, 2002.

[7] C.-C. Chang, C.-W. Hsu, and C.-J. Lin, "The analysis of decomposition methods for support vector machines," *IEEE Trans. Neural Networks*, vol. 11, no. 4, pp. 1003–1008, 2000.

[8] C.-J. Lin, "On the convergence of the decomposition method for support vector machines," *IEEE Trans. Neural Networks*, vol. 12, pp. 1288–1298, Nov. 2001.

[9] ——, "A formal analysis of stopping criteria of decomposition methods for support vector machines," *IEEE Trans. Neural Networks*, vol. 13, pp. 1045–1052, Sept. 2002.

[10] N. List and H. U. Simon, "A general convergence theorem for the decomposition method," in *Proceedigns of the 17th Annual Conference on Learning Theory*, 2004, pp. 363–377.

[11] N. Takahashi and T. Nishi, "Global convergence of decomposition learning methods for support vector machines," *IEEE Trans. Neural Networks*, vol. 17, no. 6, pp. 1362–1369, Nov. 2006.

[12] C.-J. Lin, "Asymptotic convergence of an SMO algorithm without any assumption," *IEEE Trans. Neural Networks*, vol. 13, pp. 248–250, Jan. 2002.

[13] S. S. Keerthi and E. G. Gilbert, "Convergence of a generalized SMO algorithm for SVM classifier design," *Machine Learning*, vol. 46, pp. 351–360, 2002.

[14] N. Takahashi and T. Nishi, "Rigorous proof of termination of SMO algorithm for support vector machines," *IEEE Trans. Neural Networks*, vol. 16, no. 3, pp. 774–776, May 2005.

[15] P.-H. Chen, R.-E. Fan, and C.-J. Lin, "A study on SMO-type decomposition methods for support vector machines," *IEEE Trans. Neural Networks*, vol. 17, no. 4, pp. 893–908, July 2006.

[16] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," 2001. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[17] S.-P. Liao, H.-T. Lin, and C.-J. Lin, "A note on the decomposition methods for support vector regression," *Neural Computation*, vol. 14, pp. 1267–1281, 2002.

[18] A. J. Smola and B. Schölkopf, “A tutorial on support vector regression,” Royal Holloway College, London, UK, NeuroCOLT Technical Report NC-TR-98-030, 1998.
[19] S. K. Shevade, S. S. Keerthi, C. Bhattacharyya, and K. R. K. Murthy, “Improvements to the SMO algorithm for SVM regression,” *IEEE Trans. Neural Networks*, vol. 11, no. 5, pp. 1188–1193, Sept. 2000.
[20] G. W. Flake and S. Lawrence, “Efficient SVM regression training with SMO,” *Machine Learning*, vol. 46, pp. 271–290, 2002.
[21] J. Guo, N. Takahashi, and T. Nishi, “A novel sequential minimal optimization algorithm for support vector regression,” in *Neural Information Processing (Proceedings of the 13th International Conference on Neural Information Processing)*, ser. Lecture Notes in Computer Science, vol. 4232, Oct. 2006, pp. 827–836.
[22] R.-E. Fan, P.-H. Chen, and C.-J. Lin, “Working set selection using second order information for training support vector machines,” *Journal of Machine Learning Research*, vol. 6, pp. 1889–1918, 2005.
[23] D. Hush, P. Kelly, C. Scovel, and I. Steinwart, “QP algorithms with guaranteed accuracy and run time for support vector machines,” *Journal of Machine Learning Research*, vol. 7, pp. 733–769, 2006.
[24] T. Glasmachers and C. Igel, “Maximum-gain working set selection for SVMs,” *Journal of Machine Learning Research*, vol. 7, pp. 1437–1466, 2006.
[25] A. Asuncion and D. Newman, “UCI machine learning repository,” 2007. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html
[26] “StatLib – datasets archive.” [Online]. Available: http://lib.stat.cmu.edu/datasets/