2009 International Symposium on Nonlinear Theory and its Applications
NOLTA'09, Sapporo, Japan, October 18-21, 2009

A Two-Phase Decomposition Algorithm for Solving Convex Quadratic Programming Problems

Yuta Kobayashi† and Norikazu Takahashi‡

†Graduate School of Information Science and Electrical Engineering, Kyushu University
‡Faculty of Information Science and Electrical Engineering, Kyushu University
744 Motooka, Nishi-ku, Fukuoka 819-0395 Japan
Email: {kobayasi@kairo.csce, norikazu@inf}.kyushu-u.ac.jp

Abstract—A two-phase decomposition algorithm for convex quadratic programming problems is proposed. Experimental results show that the proposed algorithm can greatly reduce the computation time of the conventional decomposition algorithm when the number of variables is large and the number of equality constraints is small.

1. Introduction

A quadratic programming (QP) problem is an optimization problem in which the objective function is quadratic and the constraints are linear equalities and inequalities. QP has numerous applications such as portfolio selection, data mining, pattern recognition, structural optimization, VLSI placement and so on. Various approaches have been proposed for solving QP problems such as the active-set, the conjugate gradient and the interior-point methods [1].

Recently, the authors have proposed a decomposition algorithm[1] for solving QP problems and shown experimentally that it is very effective when the number of variables is large and the number of constraints is small [4]. However, experimental results also showed that the first part of the algorithm takes more than half of the total computation time. The first part is just to find a feasible solution. This is done by solving an linear programming (LP) problem, which has the same constraints as the original QP problem, by using an existing LP solver such as the simplex method. The second part, on the other hand, is to optimize the solution by using a decomposition method, that is, the following two operations are executed repeatedly until some termination criterion is satisfied: 1) selecting a small number of variables and 2) solving a QP problem with respect to the selected variables.

In this paper, we propose a two-phase decomposition algorithm. The main idea is to apply the decomposition technique not only to the optimization process but also to the selection of a feasible solution. Experimental results show that the computation time is greatly reduced in almost all cases by using the proposed algorithm.

[1]Decomposition algorithm was first proposed by Osuna *et.al* [2] to solve large scale QP problems arising in support vector machines[3].

2. Quadratic Programming Problem

We consider QP problems of the form:

$$\begin{array}{ll} \text{Minimize} & f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^T\boldsymbol{x} \\ \text{Subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{array} \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is a variable; $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$, $\boldsymbol{c} \in \mathbb{R}^n$, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$ are constants. The inequality $\boldsymbol{x} \geq \boldsymbol{0}$ means that all components of $\boldsymbol{x}$ is nonnegative. Note that (1) includes the LP problem as a special case.

If $f(\boldsymbol{x})$ is convex, or, $\boldsymbol{Q}$ is positive semi-definite, a QP problem is called a convex QP problem. The set of optimal solutions of a convex QP problem is completely characterized by the Karush-Kuhn-Tucker (KKT) conditions [1]. In case of (1), a feasible solution $\boldsymbol{x}$ is an optimal solution if and only if there exist $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_m]^T$ and $\boldsymbol{\mu} = [\mu_1, \mu_2, \ldots, \mu_n]^T$ such that the KKT conditions:

$$\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c} + \boldsymbol{A}^T\boldsymbol{\lambda} - \boldsymbol{\mu} = \boldsymbol{0} \tag{2}$$

$$\boldsymbol{\mu} \geq \boldsymbol{0} \tag{3}$$

$$\mu_i x_i = 0, \quad i = 1, 2, \ldots, n \tag{4}$$

are satisfied. If $x_i > 0$ then μ_i must be zero from (4). In this case, the i-th component of (3) is necessarily satisfied, and the i-th component of (2) is written as $[\boldsymbol{Q}\boldsymbol{x}]_i + c_i + [\boldsymbol{A}^T\boldsymbol{\lambda}]_i = 0$, where $[\boldsymbol{Q}\boldsymbol{x}]_i$ ($[\boldsymbol{A}^T\boldsymbol{\lambda}]_i$, resp.) represents the i-th component of the vector $\boldsymbol{Q}\boldsymbol{x}$ ($\boldsymbol{A}^T\boldsymbol{\lambda}$, resp.). If $x_i = 0$ then (4) apparently holds, and the conditions (2) and (3) for the i-th component are rewritten as $[\boldsymbol{Q}\boldsymbol{x}]_i + c_i + [\boldsymbol{A}^T\boldsymbol{\lambda}]_i \geq 0$. Therefore the KKT conditions (2)–(4) can be rewritten as

$$[\boldsymbol{Q}\boldsymbol{x}]_i + c_i + [\boldsymbol{A}^T\boldsymbol{\lambda}]_i \begin{cases} = 0, & \text{if } x_i > 0 \\ \geq 0, & \text{if } x_i = 0 \end{cases}, \quad i = 1, 2, \ldots, n. \tag{5}$$

3. Previous Work

The authors have recently proposed a decomposition algorithm for solving (1) [4]. In the first part of their algorithm, the LP problem

$$\begin{array}{ll} \text{Minimize} & 0 \\ \text{Subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{array} \tag{6}$$

is solved by using one of existing LP solvers such as the simplex method, in order to find a feasible solution of (1). In the second part, a decomposition method is applied to find an optimal solution of (1), that is, two operations are executed repeatedly until some termination criterion is satisfied: one is to select a set of q variables; the other is to minimize $f(\boldsymbol{x})$ by updating only the selected variables. An optimal solution of (6), which is obtained in the first part, is used in the second part as the initial value of $\boldsymbol{x}$.

Experimental results showed that the decomposition algorithm proposed in [4] is faster than the direct application of QP solvers to (1) when the number of variables is large and the number of equality constraints, is small. However, at the same time, it was shown that the computation time for the first part is longer than the second part.

4. Proposed Algorithm

In the following, we will assume that

$$\boldsymbol{b} \geq \boldsymbol{0}. \tag{7}$$

This assumption does not lose generality because any QP problem of the form (1) can be transformed into another QP problem which is equivalent to the original one and satisfies (7) by multiplying the i-th row of $\boldsymbol{Ax} = \boldsymbol{b}$ by -1 for all i such that $b_i < 0$.

4.1. Introduction of Artificial Variables

In order to reduce the computation time for the first part of the decomposition algorithm proposed in [4], we apply decomposition technique not only to the second part but also to the first part. However, this is not straightforward because (6) does not have a trivial feasible solution. Hence we make use of the idea of two-phase method for LP problems, that is, we introduce artificial variables $u_1, u_2, \ldots, u_m$ and transform (6) into the following form:

$$\begin{array}{ll} \text{Minimize} & \sum_{i=1}^{m} u_i \\ \text{Subject to} & \boldsymbol{Ax} + \boldsymbol{u} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \\ & \boldsymbol{u} \geq \boldsymbol{0} \end{array} \tag{8}$$

where $\boldsymbol{u} = [u_1, u_2, \ldots, u_m]^T \in \mathbb{R}^m$. An important property of this LP problem is that it has a trivial feasible solution $(\boldsymbol{x}, \boldsymbol{u}) = (\boldsymbol{0}, \boldsymbol{b})$. So we can easily apply the decomposition method to (8) with the initial solution $(\boldsymbol{x}, \boldsymbol{u}) = (\boldsymbol{0}, \boldsymbol{b})$. Another important property is that (8) has the optimal solution with $\boldsymbol{u} = \boldsymbol{0}$ if and only if (6) has a feasible solution. Hence, by solving (8), we can determine the feasibility of (6).

Let $\tilde{\boldsymbol{x}} = [\boldsymbol{x}^T, \boldsymbol{u}^T]^T \in \mathbb{R}^{n+m}$, $\tilde{\boldsymbol{A}} = [\boldsymbol{A}, \boldsymbol{I}] \in \mathbb{R}^{m\times(n+m)}$ and $\boldsymbol{d} = [\boldsymbol{0}^T, \boldsymbol{1}^T]^T \in \mathbb{R}^{n+m}$ where $\boldsymbol{I}$ is the m dimensional identity matrix, $\boldsymbol{0}$ is the n dimensional zero vector and $\boldsymbol{1}$ is the m dimensional vector with all components equal to one. Then (8) is rewritten as follows:

$$\begin{array}{ll} \text{Minimize} & \boldsymbol{d}^T \tilde{\boldsymbol{x}} \\ \text{Subject to} & \tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}} = \boldsymbol{b} \\ & \tilde{\boldsymbol{x}} \geq \boldsymbol{0} \end{array} \tag{9}$$

The KKT condition for this LP problem can immediately be derived from (5) as follows:

$$d_i + [\tilde{\boldsymbol{A}}^T \tilde{\boldsymbol{\lambda}}]_i \begin{cases} = 0, & \text{if } \tilde{x}_i > 0 \\ \geq 0, & \text{if } \tilde{x}_i = 0 \end{cases}, \quad i = 1, 2, \ldots, n+m. \tag{10}$$

where $\tilde{\boldsymbol{\lambda}} \in \mathbb{R}^m$.

4.2. Decomposition Algorithm

A two-phase decomposition algorithm we propose in this paper is as follows:

---

1. Set $\tilde{\boldsymbol{x}}^{(0)} := [\boldsymbol{0}^T, \boldsymbol{b}^T]^T$ where $\boldsymbol{0}$ is the n dimensional zero vector.

2. Solve the LP problem:

$$\begin{array}{ll} \text{Minimize} & \tilde{\delta} \\ \text{Subject to} & \left| d_i + [\tilde{\boldsymbol{A}}^T \tilde{\boldsymbol{\lambda}}]_i \right| \leq \tilde{\delta}, \quad \forall i \in \tilde{I}_+^{(k)} \\ & d_i + [\tilde{\boldsymbol{A}}^T \tilde{\boldsymbol{\lambda}}]_i \geq -\tilde{\delta}, \quad \forall i \in \tilde{I}_0^{(k)} \end{array} \tag{11}$$

   where $\tilde{I}_+^{(k)} = \{i \,|\, \tilde{x}_i^{(k)} > 0\}$ and $\tilde{I}_0^{(k)} = \{i \,|\, \tilde{x}_i^{(k)} = 0\}$. If the optimal value of $\tilde{\delta}$ is less than $\epsilon_1 (> 0)$ then go to Step 6. Otherwise go to Step 3.

3. For $i = 1, 2, \ldots, n+m$, set

$$\tilde{v}_i^{(k)} := \begin{cases} \left| d_i + [\tilde{\boldsymbol{A}}^T \tilde{\boldsymbol{\lambda}}^*]_i \right|, & \text{if } i \in \tilde{I}_+^{(k)} \\ -\min(0, d_i + [\tilde{\boldsymbol{A}}^T \tilde{\boldsymbol{\lambda}}^*]_i), & \text{if } i \in \tilde{I}_0^{(k)} \end{cases} \tag{12}$$

   where $(\tilde{\boldsymbol{\lambda}}^*, \tilde{\delta}^*)$ is the optimal solution of (11) obtained in Step 2. Sort these values in decreasing order as

$$\tilde{v}_{i_1}^{(k)} \geq \tilde{v}_{i_2}^{(k)} \geq \cdots \geq \tilde{v}_{i_{n+m}}^{(k)}.$$

   Set $\tilde{I}_B^{(k)} := \{i_s\}_{s=1}^{\tilde{q}}$ and $\tilde{I}_N^{(k)} := \{i_s\}_{s=\tilde{q}+1}^{n+m}$.

4. Solve (9) under the additional constraints

$$\tilde{x}_i = \tilde{x}_i^{(k)}, \quad \forall i \in \tilde{I}_N^{(k)},$$

   and set $\tilde{\boldsymbol{x}}^{(k+1)}$ to the obtained optimal solution.

5. Add 1 to k and go to Step 2.

6. If $\tilde{x}_i^{(k)} \geq \epsilon_2 (> 0)$ holds for some $i \in \{n+1, n+2, \ldots, n+m\}$ then stop. Otherwise go to Step 7.

7. Set $\boldsymbol{x}^{(k)} := [\tilde{x}_1^{(k)}, \tilde{x}_2^{(k)}, \ldots, \tilde{x}_n^{(k)}]^T$.

8. Set $\boldsymbol{g}^{(k)} := \boldsymbol{Qx}^{(k)} + \boldsymbol{c}$ and solve the LP problem:

$$\begin{array}{ll} \text{Minimize} & \delta \\ \text{Subject to} & \left| g_i^{(k)} + [\boldsymbol{A}^T \boldsymbol{\lambda}]_i \right| \leq \delta, \quad \forall i \in I_+^{(k)} \\ & g_i^{(k)} + [\boldsymbol{A}^T \boldsymbol{\lambda}]_i \geq -\delta, \quad \forall i \in I_0^{(k)} \end{array} \tag{13}$$

   where $I_+^{(k)} = \{i \,|\, x_i^{(k)} > 0\}$ and $I_0^{(k)} = \{i \,|\, x_i^{(k)} = 0\}$. If the optimal value of δ is less than $\epsilon_3 (> 0)$, then stop. Otherwise go to Step 10.

9. For $i = 1, 2, \dots, n$, set

$$v_i^{(k)} := \begin{cases} \left|g_i^{(k)} + [\boldsymbol{A}^T\boldsymbol{\lambda}^*]_i\right|, & \text{if } i \in I_+^{(k)} \\ -\min(0, g_i^{(k)} + [\boldsymbol{A}^T\boldsymbol{\lambda}^*]_i), & \text{if } i \in I_0^{(k)} \end{cases} \quad (14)$$

where $(\boldsymbol{\lambda}^*, \delta^*)$ is the optimal solution of (13) obtained in Step 9. Sort these values in decreasing order as

$$v_{i_1}^{(k)} \geq v_{i_2}^{(k)} \geq \cdots \geq v_{i_n}^{(k)} .$$

Set $I_B^{(k)} := \{i_s\}_{s=1}^q$ and $I_N^{(k)} := \{i_s\}_{s=q+1}^n$.

10. Solve (1) under the additional constraints

$$x_i = x_i^{(k)}, \quad \forall i \in I_N^{(k)} ,$$

and set $\boldsymbol{x}^{(k+1)}$ to the obtained optimal solution.

11. Add 1 to k and go to Step 8.

In the first phase (Steps 1 to 5), the LP problem (9) is solved by using a decomposition method. In Step 1, the initial solution $\tilde{\boldsymbol{x}}^{(0)}$ is set to a feasible solution $[\boldsymbol{0}^T, \boldsymbol{b}^T]^T$. In Step 2, the LP problem (11), which has $m + 1$ variables and at most $2(n + m)$ inequality constrains, is solved to check whether the KKT condition (10) is satisfied or not for $\tilde{\boldsymbol{x}}^{(k)}$. In Step 3, $\tilde{q}$ variables are selected for the working set based on the degree of violation of the KKT condition defined by (12). In Step 4, the values of the selected variables are updated by solving the LP subproblem. Let us assume for simplicity that $\tilde{I}_B^{(k)} = \{1, 2, \dots, \tilde{q}\}$. Then the LP subproblem is expressed as

$$\begin{array}{ll} \text{Minimize} & \boldsymbol{d}_B^T \tilde{\boldsymbol{x}}_B \\ \text{Subject to} & \tilde{\boldsymbol{A}}_B \tilde{\boldsymbol{x}}_B = \boldsymbol{b} - \tilde{\boldsymbol{A}}_N \tilde{\boldsymbol{x}}_N^{(k)} \\ & \tilde{\boldsymbol{x}}_B \geq \boldsymbol{0} \end{array} \quad (15)$$

where $\tilde{\boldsymbol{x}}_B$, $\tilde{\boldsymbol{x}}_B^{(k)}$ and $\boldsymbol{d}_B$ are the vectors composed of the first $\tilde{q}$ components of $\tilde{\boldsymbol{x}}$, $\tilde{\boldsymbol{x}}^{(k)}$ and $\boldsymbol{d}$, respectively; $\tilde{\boldsymbol{A}}_B$ and $\tilde{\boldsymbol{A}}_N$ are the first $\tilde{q}$ columns and the last $n + m - \tilde{q}$ columns of $\tilde{\boldsymbol{A}}$, respectively. Since (15) is an LP problem with $\tilde{q}$ variables, it can be solved much faster than the original problem (9).

In the second phase (Steps 7 to 11), the QP problem (1) is solved by using a decomposition method. In Step 7, the initial solution $\boldsymbol{x}^{(0)}$ is set to the feasible solution obtained in the first phase. In Step 8, the LP problem (13), which has $m + 1$ variables and at most $2n$ inequality constrains, is solved to check whether the KKT condition (5) is satisfied or not for $\boldsymbol{x}^{(k)}$. In Step 9, q variables are selected for the working set based on the degree of violation of the KKT condition defined by (14). In Step 10, the values of the selected variables are updated by solving the QP subproblem. Let us assume for simplicity that $I_B^{(k)} = \{1, 2, \dots, q\}$. Then the QP subproblem is expressed as

$$\begin{array}{ll} \text{Minimize} & \frac{1}{2}\boldsymbol{x}_B^T \boldsymbol{Q}_{BB} \boldsymbol{x}_B + (\boldsymbol{Q}_{BN}\boldsymbol{x}_N^{(k)} + \boldsymbol{c}_B)^T x_B \\ \text{Subject to} & \boldsymbol{A}_B \boldsymbol{x}_B = \boldsymbol{b} - \boldsymbol{A}_N \boldsymbol{x}_N^{(k)} \\ & \boldsymbol{x}_B \geq \boldsymbol{0} \end{array} \quad (16)$$

where $\boldsymbol{x}_B$, $\boldsymbol{x}_B^{(k)}$ and $\boldsymbol{c}_B$ are the vectors composed of the first q components of $\boldsymbol{x}$, $\boldsymbol{x}^{(k)}$ and $\boldsymbol{c}$, respectively; $\boldsymbol{Q}_{BB}$ is the $q \times q$ matrix composed of the first q rows and the first q columns of $\boldsymbol{Q}$; $\boldsymbol{Q}_{BN}$ is the $q \times (n - q)$ matrix composed of the first q rows and the last $n - q$ columns of $\boldsymbol{Q}$; $\boldsymbol{A}_B$ and $\boldsymbol{A}_N$ are the first q columns and the last $n - q$ columns of $\boldsymbol{A}$, respectively. Since (16) is a QP problem with q variables, it can be solved much faster than the original problem (1).

5. Experiments

The authors have implemented the two-phase decomposition algorithm in Scilab 4.1.2, which has the functions "linpro" for solving LP problems and "quapro" for QP problems, and compared its computation time with those of the conventional decomposition algorithm [4] and the direct method. By the direct method we mean that (1) is solved simply by the function "quapro".

QP problems were randomly generated as follows. Components of the matrix $\boldsymbol{A}$ and the vectors $\boldsymbol{c}$ and $\boldsymbol{b}$ were set to random numbers between -1 and 1. The matrix $\boldsymbol{Q}$ was generated by $\boldsymbol{Q} = \boldsymbol{P}^T\boldsymbol{P}$ so that $\boldsymbol{Q}$ becomes positive semi-definite, where components of the matrix $\boldsymbol{P}$ were set to random numbers between -1 and 1.

Throughout our experiments, ϵ_1, ϵ_2 and ϵ_3 in the proposed algorithm are all set to 10^{-6}. Also, $\tilde{q}$ and q, the sizes of the working set in the first and second phase, respectively, are set to the same value for simplicity. Programs were run on a PC with Intel Core 2 Quad 2.66GHz and 3.25GB RAM.

First, we fixed the number of equality constraints to 10 and measured the computation times of the three methods for $n = 400$, 900 and 1400. Results are shown in Fig.1. When $n = 400$, the two-phase algorithm is a little faster than the conventional algorithm for all q but slower than the direct method except the case where $q = 200$. When $n = 900$, on the other hand, the two-phase algorithm is much faster than the direct method for all q, while the conventional algorithm is still slower than the direct method. When $n = 1400$, the two-phase algorithm is much faster than others and the direct method is the slowest for most values of q.

Next, we fixed the number of equality constraints to 100 and measured the computation times of the three methods for $n = 400$, 900 and 1400. Results are shown in Fig.2. When $n = 400$, the direct method is the fastest and the two-phase algorithm is slower than the conventional algorithm for all q. When $n = 900$ and $n = 1400$, the two-phase algorithm is faster than the conventional algorithm for all q but slower than the direct method for many q.

6. Conclusion

In this paper, we have proposed a two-phase decomposition algorithm for solving general QP problems. Experimental results show that the proposed algorithm is very ef-

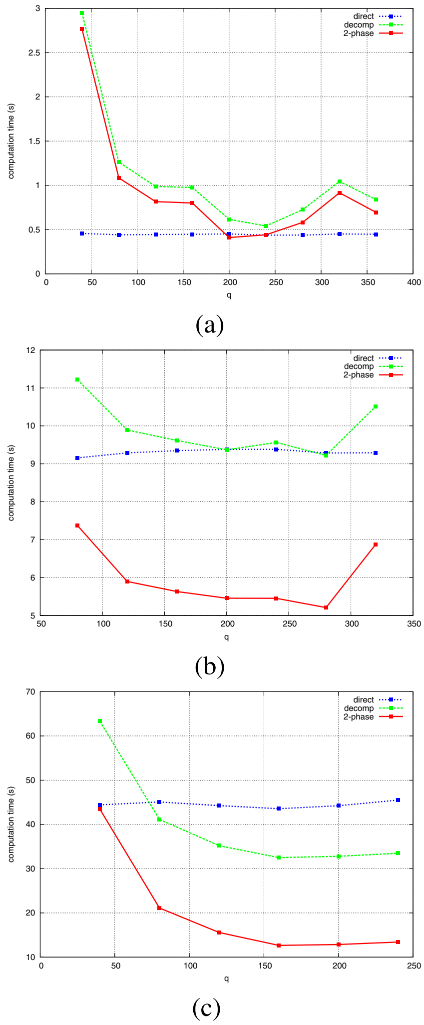

Figure 1: Computation times of the direct method (blue), the conventional decomposition algorithm (green) and the two-phase decomposition algorithm (red) for $m = 10$. (a) $n = 400$. (b) $n = 900$. (c) $n = 1400$.

fective when the number of variables is large and the number of equality constraints is small but not so effective when the number of equality constraints is large. This is because an LP problem with $m + 1$ variables has to be solved every time when the working set is selected. Reducing the computation time for the working set selection is a future problem. Another future problem is to prove theoretically the global convergence of the algorithm.

Acknowledgements

This work was supported in part by Grant-in-Aid for Scientific Research (c) 21560068 from Japan Society for the Promotion of Science.

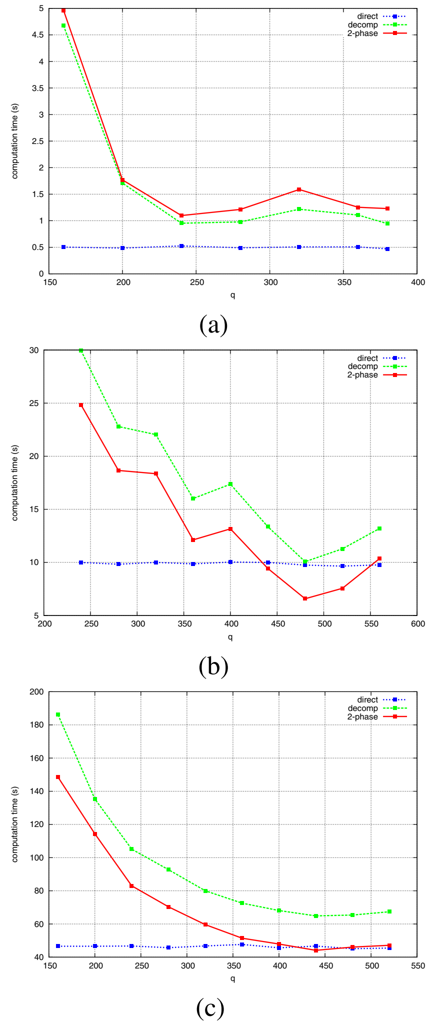

Figure 2: Computation times of the direct method (blue), the conventional decomposition algorithm (green) and the two-phase decomposition algorithm (red) for $m = 100$. (a) $n = 400$. (b) $n = 900$. (c) $n = 1400$.

References

[1] A. Antoniou and W. Lu, *Practical Optimization: Algorithms and Engineering Applications*, Springer, New York, NY, 2007.

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

[3] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol.20, pp.273–297, 1995.

[4] N. Takahashi, Y. Kobayashi and B. Chen, "A new decomposition algorithm for solving convex quadratic programming problems," *Proceedings of 2008 International Symposium on Nonlinear Theory and its Applications*, pp.73–76, 2008.