No. 685

An Algorithm for Strictly Convex Quadratic Programming with Box Constraints

by

Yong-Chang Jiao, Xiaojun Liu and Satoru Fujishige

June 1996

An Algorithm for Strictly Convex Quadratic Programming with Box Constraints

Yong-Chang Jiao*, Xiaojun Liu and Satoru Fujishige

Institute of Policy and Planning Sciences
University of Tsukuba
Tsukuba, Ibaraki 305, Japan

June 1996

Abstract

We propose an efficient algorithm for solving strictly convex quadratic programs with box constraints (i.e., lower and upper bounds on each variable). Our algorithm is based on the active set and the Newton method. We repeatedly compute relevant inverse matrices efficiently and, starting from an initial feasible point, we find an optimal solution of the problem in finitely many steps.

Key words: Quadratic programming, algorithm, active set method, Newton method, box constraints.

1. Introduction

We consider a strictly convex (i.e., positive definite) quadratic programming problem subject to box constraints:

$$\textbf{(QP)} \quad \text{Minimize} \quad f(x) = \frac{1}{2}x^{\top}Ax + b^{\top}x$$

*Visiting Research Fellow supported by Japan Society for the Promotion of Science. E-mail: jiao@kantaro.sk.tsukuba.ac.jp, permanent address: Institute of Antennas and EM Scattering, Xidian University, Xi'an, Shaanxi 710071, P. R. China

$$\text{subject to} \quad c \leq x \leq d. \tag{1.1}$$

where $A = [a_{ij}]$ is an $n \times n$ symmetric positive definite matrix, and b, c and d are n-vectors. Let $g(x)$ be the gradient, $Ax + b$, of $f(x)$ at x. Without loss of generality we assume $c_i < d_i$ for each i with $1 \leq i \leq n$.

Applications of the above-mentioned Problem **(QP)** with box constraints include large linear least squares problem with bounded variables, linear complementarity problems, and dual problems arising in a sequential quadratic programming algorithm. This last application [4] motivated the present work. Other applications can be found in [5].

Yang and Tolle [2] presented conjugate gradient-type algorithms for **(QP)**. Our algriothm is based on the idea of the active set method and we keep the active (and nonactive) set and compute related matrices efficiently.

In Section 2 we give some definitions and notations to be used later and describe the optimality condition for **(QP)**, based on which an algorithm will be constructed. We propose an algorithm for solving **(QP)** in Section 3. The proof of the validity of the proposed algorithm is provided in Section 4. Repeated updating of inverse matrices is required in the algorithm. We give an efficient procedure of updating relevant matrices in Section 5.

2. Definitions and the Optimality Condition

We denote by K the set $\{1, 2, \cdots, n\}$ of the subscript indices of the variable x appearing in (1.1).

Definition: Denote by $x^{(l)}$ the current solution obtained at the lth iteration in the algorithm to be given in Section 3. For the current $x^{(l)}$ with $c \leq x^{(l)} \leq d$ $(l = 0, 1, \cdots)$ we define the *nonactive set*

$$\mathcal{NA}^{(l)} = \{i \,|\, c_i < x_i^{(l)} < d_i, 1 \leq i \leq n\} \tag{2.1}$$

and $n_l = |\mathcal{NA}^{(l)}|$.

For the *nonactive set* $\mathcal{NA}^{(l)}$, also define the $n \times n_l$ matrix

$$P_l = [p_{ij}]_{n \times n_l} \tag{2.2}$$

with the row index set K and the column index set $\mathcal{NA}^{(l)}$ such that $p_{ii} = 1$ if $i \in \mathcal{NA}^{(l)}$ and $p_{ij} = 0$ otherwise. Two related matrices are defined as follows:

$$H_l = P_l^\top A P_l, \quad B_l = H_l^{-1}. \tag{2.3}$$

Note that the row and column index sets of H_l and B_l are both $\mathcal{NA}^{(l)}$.

The Karush-Kuhn-Tucker optimality condition for **(QP)** is given as follows.

Theorem 2.1: *A feasible point x^* is an optimal solution of Problem* **(QP)** *if and only if for $j = 1, 2, \cdots, n$*

$$g_j(x^*) \geq 0, \ \textit{when} \ x_j^* = c_j, \tag{2.4}$$

$$g_j(x^*) \leq 0, \ \textit{when} \ x_j^* = d_j, \tag{2.5}$$

$$g_j(x^*) = 0, \ \textit{when} \ c_j < x_j^* < d_j. \tag{2.6}$$

□

We derive an algorithm for solving **(QP)**, based on Theorem 2.1.

3. An Algorithm

We give an algorithm for solving Problem **(QP)** as follows.

Step 1: Choose an initial feasible point $x^{(0)}$ with $c_i \leq x_i^{(0)} \leq d_i$ $(i \in K)$ such that for some $i_0 \in K$ we have $c_{i_0} < x_{i_0}^{(0)} < d_{i_0}$. Compute $\mathcal{NA}^{(0)}, P_0, H_0, n_0, B_0$, and set $l = 0$.

Step 2: Compute

$$g^{(l)} = Ax^{(l)} + b, \tag{3.1}$$

$$\bar{g}^{(l)} = P_l^\top g^{(l)}, \tag{3.2}$$

$$s^{(l)} = -B_l \bar{g}^{(l)}, \tag{3.3}$$

$$\alpha^{(l)} = \min\{1, \min_{j \in \mathcal{NA}^{(l)},\ s_j^{(l)} < 0} \frac{c_j - x_j^{(l)}}{s_j^{(l)}}, \min_{j \in \mathcal{NA}^{(l)},\ s_j^{(l)} > 0} \frac{d_j - x_j^{(l)}}{s_j^{(l)}}\}, \tag{3.4}$$

$$\tilde{x}^{(l+1)} = x^{(l)} + \alpha^{(l)} P_l s^{(l)}. \tag{3.5}$$

(Here, note that the index set of vector $s^{(l)}$ should be regarded as $\mathcal{NA}^{(l)}$.)
Step 3: If $\alpha^{(l)} < 1$ and $n_l > 1$, choose some $\gamma \in \mathcal{NA}^{(l)}$ such that

$$\alpha^{(l)} = \frac{c_\gamma - x_\gamma^{(l)}}{s_\gamma^{(l)}}, \qquad s_\gamma^{(l)} < 0, \tag{3.6}$$

or

$$\alpha^{(l)} = \frac{d_\gamma - x_\gamma^{(l)}}{s_\gamma^{(l)}}, \qquad s_\gamma^{(l)} > 0, \tag{3.7}$$

set

$$\mathcal{NA}^{(l+1)} = \mathcal{NA}^{(l)} \setminus \{\gamma\}, \quad n_{l+1} = n_l - 1, \tag{3.8}$$

$$x^{(l+1)} = \tilde{x}^{(l+1)}, \tag{3.9}$$

and go to Step 5; otherwise go to Step 4.
Step 4: Compute

$$g^{(l+1)} = A\tilde{x}^{(l+1)} + b, \tag{3.10}$$

$$\Phi_{l1} = \{i \mid \tilde{x}_i^{(l+1)} = c_i, g_i^{(l+1)} < 0\}, \tag{3.11}$$

$$\Phi_{l2} = \{i \mid \tilde{x}_i^{(l+1)} = d_i, g_i^{(l+1)} > 0\}, \tag{3.12}$$

$$\Phi_l = \Phi_{l1} \cup \Phi_{l2}. \tag{3.13}$$

If $\Phi_l = \emptyset$, **STOP** and the current $\tilde{x}^{(l+1)}$ is an optimal solution; otherwise compute for each $i \in \Phi_l$

$$\lambda_i = \frac{g_i^{(l+1)}}{(d_i - c_i)a_{ii}},$$

$$\begin{aligned}\bar{\gamma} = arg\max\{ & \frac{(g_i^{(l+1)})^2}{2a_{ii}} \text{ with } i \in \Phi_{l_1} \text{ and } -\lambda_i < 1; \\ & -\frac{1}{2}(d_i - c_i)^2 a_{ii} - g_i^{(l+1)}(d_i - c_i) \text{ with } i \in \Phi_{l_1} \text{ and } -\lambda_i \geq 1; \\ & \frac{(g_i^{(l+1)})^2}{2a_{ii}} \text{ with } i \in \Phi_{l_2} \text{ and } \lambda_i < 1; \\ & -\frac{1}{2}(d_i - c_i)^2 a_{ii} + g_i^{(l+1)}(d_i - c_i) \text{ with } i \in \Phi_{l_2} \text{ and } \lambda_i \geq 1\}.\end{aligned}$$

There are four cases (a)~(d) with respect to $\bar{\gamma}$ as follows:

(a) If $\bar{\gamma} \in \Phi_{l1}$ and $-\lambda_{\bar{\gamma}} < 1$, set

$$x^{(l+1)} = \begin{cases} \tilde{x}_i^{(l+1)}, & i \neq \bar{\gamma}, \\ c_i - g_i^{(l+1)}/a_{ii}, & i = \bar{\gamma}, \end{cases} \tag{3.14}$$

$$\mathcal{NA}^{(l+1)} = \mathcal{NA}^{(l)} \cup \{\bar{\gamma}\}, \tag{3.15}$$

$$n_{l+1} = n_l + 1, \tag{3.16}$$

and go to Step 6.

(b) If $\bar{\gamma} \in \Phi_{l1}$ and $-\lambda_{\bar{\gamma}} \geq 1$, set

$$\tilde{x}^{(l+1)} = \begin{cases} \tilde{x}_i^{(l+1)}, & i \neq \bar{\gamma}, \\ d_i, & i = \bar{\gamma}, \end{cases} \tag{3.17}$$

and go to the beginning of Step 4.

(c) If $\bar{\gamma} \in \Phi_{l2}$ and $\lambda_{\bar{\gamma}} < 1$, set

$$x^{(l+1)} = \begin{cases} \tilde{x}_i^{(l+1)}, & i \neq \bar{\gamma}, \\ d_i - g_i^{(l+1)}/a_{ii}, & i = \bar{\gamma}, \end{cases} \tag{3.18}$$

$$\mathcal{NA}^{(l+1)} = \mathcal{NA}^{(l)} \cup \{\bar{\gamma}\}, \tag{3.19}$$

$$n_{l+1} = n_l + 1, \tag{3.20}$$

and go to Step 6.

(d) If $\bar{\gamma} \in \Phi_{l2}$, $\lambda_{\bar{\gamma}} \geq 1$, set

$$\tilde{x}^{(l+1)} = \begin{cases} \tilde{x}_i^{(l+1)}, & i \neq \bar{\gamma}, \\ c_i, & i = \bar{\gamma}, \end{cases} \tag{3.21}$$

and go to the beginning of Step 4.

Step 5: Form P_{l+1} by deleting column γ from P_l, $\bar{B}_l$ by deleting row γ and column

γ from B_l and a vector h by deleting component γ from column γ of B_l. Compute

$$\begin{aligned} H_{l+1} &= P_{l+1}^{\top} A P_{l+1}, \\ t &= [B_l]_{\gamma\gamma}, \\ B_{l+1} &= \bar{B}_l - hh^{\top}/t, \end{aligned} \tag{3.22}$$

set $l = l + 1$, and go to Step 2.

Step 6: Compute P_{l+1} and H_{l+1} corresponding to $\mathcal{NA}^{(l+1)}$. Form a vector h by deleting component $\bar{\gamma}$ from column $\bar{\gamma}$ in H_{l+1}, set $t = [H_{l+1}]_{\bar{\gamma}\bar{\gamma}}$, compute

$$\begin{aligned} \alpha &= \frac{1}{t - h^{\top} B_l h}, \\ \hat{h} &= -\alpha B_l h, \\ B_{l+1} &= \begin{bmatrix} B_l + \hat{h}\hat{h}^{\top}/\alpha & \hat{h} \\ \hat{h}^{\top} & \alpha \end{bmatrix}, \end{aligned} \tag{3.23}$$

set $l = l + 1$, and go to Step 2.

4. Validity of the Algorithm

We show the following.

Lemma 4.1: *For all* $l \geq 0$, $0 < \alpha^{(l)} \leq 1$.

(Proof) For all $j \in \mathcal{NA}^{(l)}$ we have $c_j < x_j^{(l)} < d_j$ by definition. Hence,

$$\min_{j \in \mathcal{NA}^{(l)},\ s_j^{(l)} < 0} \frac{c_j - x_j^{(l)}}{s_j^{(l)}} > 0 \tag{4.1}$$

and

$$\min_{j \in \mathcal{NA}^{(l)},\ s_j^{(l)} > 0} \frac{d_j - x_j^{(l)}}{s_j^{(l)}} > 0. \tag{4.2}$$

□

Lemma 4.2: *In* Step 4 *we have*

$$f(x^{(l+1)}) < f(\tilde{x}^{(l+1)}) \quad or \quad f(\tilde{x}_{new}^{(l+1)}) < f(\tilde{x}^{(l+1)}), \tag{4.3}$$

where $\tilde{x}_{new}^{(l+1)}$ *is the new point obtained from* $\tilde{x}^{(l+1)}$ *in Cases* (b) *and* (d) *in* Step 4.

(Proof) We only consider Cases (a) and (b) here; Cases (c) and (d) can be proved in the same way.

In Case (a), since $g_{\bar{\gamma}}^{(l+1)} < 0$, we have

$$0 < -\lambda_{\bar{\gamma}} = \frac{-g_{\bar{\gamma}}^{(l+1)}}{(d_{\bar{\gamma}} - c_{\bar{\gamma}})a_{\bar{\gamma}\bar{\gamma}}} < 1, \tag{4.4}$$

that is,

$$c_{\bar{\gamma}} < c_{\bar{\gamma}} - \frac{-g_{\bar{\gamma}}^{(l+1)}}{a_{\bar{\gamma}\bar{\gamma}}} < d_{\bar{\gamma}}. \tag{4.5}$$

Therefore,

$$c_{\bar{\gamma}} < x_{\bar{\gamma}}^{(l+1)} < d_{\bar{\gamma}}. \tag{4.6}$$

We thus have

$$\begin{aligned}
f(\tilde{x}^{(l+1)}) - f(x^{(l+1)}) &= \frac{1}{2}(\tilde{x}^{(l+1)})^{\top} A\tilde{x}^{(l+1)} + b^{\top}\tilde{x}^{(l+1)} - \frac{1}{2}(x^{(l+1)})^{\top} Ax^{(l+1)} - b^{\top}x^{(l+1)} \\
&= \frac{1}{2}(\tilde{x}^{(l+1)} - x^{(l+1)})^{\top} A(\tilde{x}^{(l+1)} + x^{(l+1)}) + b^{\top}(\tilde{x}^{(l+1)} - x^{(l+1)}) \\
&= -\frac{1}{2}(\tilde{x}^{(l+1)} - x^{(l+1)})^{\top} A(\tilde{x}^{(l+1)} - x^{(l+1)}) \\
&\quad +(A\tilde{x}^{(l+1)} + b)^{\top}(\tilde{x}^{(l+1)} - x^{(l+1)}) \\
&= -\frac{1}{2}\frac{(g_{\bar{\gamma}}^{(l+1)})^2}{a_{\bar{\gamma}\bar{\gamma}}} + \frac{(g_{\bar{\gamma}}^{(l+1)})^2}{a_{\bar{\gamma}\bar{\gamma}}} \\
&= \frac{1}{2}\frac{(g_{\bar{\gamma}}^{(l+1)})^2}{a_{\bar{\gamma}\bar{\gamma}}} > 0.
\end{aligned} \tag{4.7}$$

In Case (b), we also have

$$\begin{aligned}
f(\tilde{x}^{(l+1)}) - f(\tilde{x}_{new}^{(l+1)}) &= -\frac{1}{2}(\tilde{x}^{(l+1)} - \tilde{x}_{new}^{(l+1)})^{\top} A(\tilde{x}^{(l+1)} - \tilde{x}_{new}^{(l+1)}) \\
&\quad +(A\tilde{x}^{(l+1)} + b)^{\top}(\tilde{x}^{(l+1)} - \tilde{x}_{new}^{(l+1)}) \\
&= -\frac{1}{2}(d_{\bar{\gamma}} - c_{\bar{\gamma}})^2 a_{\bar{\gamma}\bar{\gamma}} - g_{\bar{\gamma}}^{(l+1)}(d_{\bar{\gamma}} - c_{\bar{\gamma}}) \\
&= \frac{1}{2}(d_{\bar{\gamma}} - c_{\bar{\gamma}})^2 a_{\bar{\gamma}\bar{\gamma}} - g_{\bar{\gamma}}^{(l+1)}(d_{\bar{\gamma}} - c_{\bar{\gamma}}) - (d_{\bar{\gamma}} - c_{\bar{\gamma}})^2 a_{\bar{\gamma}\bar{\gamma}} \\
&= \frac{1}{2}(d_{\bar{\gamma}} - c_{\bar{\gamma}})^2 a_{\bar{\gamma}\bar{\gamma}} + (d_{\bar{\gamma}} - c_{\bar{\gamma}})^2 a_{\bar{\gamma}\bar{\gamma}}(-\lambda_{\bar{\gamma}} - 1) \\
&\geq \frac{1}{2}(d_{\bar{\gamma}} - c_{\bar{\gamma}})^2 a_{\bar{\gamma}\bar{\gamma}} > 0.
\end{aligned} \tag{4.8}$$

□

Theorem 4.3: *The proposed algorithm strictly decreases the value of the objective function in each iteration.*

(Proof) Since we have in Step 2

$$\begin{aligned}\tilde{x}^{(l+1)} &= x^{(l)} + \alpha^{(l)} P_l s^{(l)} \\ &= x^{(l)} - \alpha^{(l)} P_l B_l \bar{g}^{(l)},\end{aligned} \tag{4.9}$$

we have

$$\begin{aligned}f(x^{(l)}) - f(\tilde{x}^{(l+1)}) &= \frac{1}{2}(x^{(l)})^\top A x^{(l)} + b^\top x^{(l)} - \frac{1}{2}(\tilde{x}^{(l+1)})^\top A \tilde{x}^{(l+1)} - b^\top \tilde{x}^{(l+1)} \\ &= \frac{1}{2}(x^{(l)})^\top A x^{(l)} + b^\top (x^{(l)} - \tilde{x}^{(l+1)}) \\ &\quad -\frac{1}{2}(x^{(l)} - \alpha^{(l)} P_l B_l \bar{g}^{(l)})^\top A (x^{(l)} - \alpha^{(l)} P_l B_l \bar{g}^{(l)}) \\ &= \alpha^{(l)} (x^{(l)})^\top A P_l B_l \bar{g}^{(l)} - \frac{1}{2}(\alpha^{(l)})^2 (P_l B_l \bar{g}^{(l)})^\top A (P_l B_l \bar{g}^{(l)}) \\ &\quad + \alpha^{(l)} b^\top P_l B_l \bar{g}^{(l)} \\ &= \alpha^{(l)} (A x^{(l)} + b)^\top P_l B_l \bar{g}^{(l)} - \frac{1}{2}(\alpha^{(l)})^2 (\bar{g}^{(l)})^\top B_l \bar{g}^{(l)} \\ &= \alpha^{(l)} [P_l^\top (A x^{(l)} + b)]^\top B_l \bar{g}^{(l)} - \frac{1}{2}(\alpha^{(l)})^2 (\bar{g}^{(l)})^\top B_l \bar{g}^{(l)} \\ &= \alpha^{(l)} (\bar{g}^{(l)})^\top B_l \bar{g}^{(l)} - \frac{1}{2}(\alpha^{(l)})^2 (\bar{g}^{(l)})^\top B_l \bar{g}^{(l)} \\ &= \alpha^{(l)} [1 - \frac{1}{2}\alpha^{(l)}] (\bar{g}^{(l)})^\top B_l \bar{g}^{(l)}.\end{aligned}$$

We consider the following two cases :

Case 1: $\bar{g}^{(l)} = 0$. We have $s^{(l)} = 0$ and hence $\tilde{x}^{(l+1)} = x^{(l)}$. Since $\alpha^{(l)} = 1$, we obtain a new $x^{(l+1)}$ in Step 4 where we go when $\Phi_l \neq \emptyset$. From Lemma 4.2, we have

$$f(x^{(l+1)}) < f(\tilde{x}^{(l+1)}) = f(x^{(l)}). \tag{4.10}$$

Case 2: $\bar{g}^{(l)} \neq 0$. We have

$$f(x^{(l)}) - f(\tilde{x}^{(l+1)}) = \alpha^{(l)} [1 - \frac{1}{2}\alpha^{(l)}] (\bar{g}^{(l)})^\top B_l \bar{g}^{(l)}, \tag{4.11}$$

$0 < \alpha^{(l)} \leq 1$ from Lemma 4.1 and B_l is a positive definite matrix. Hence, we have $f(x^{(l)}) - f(\tilde{x}^{(l+1)}) > 0$, that is,

$$f(\tilde{x}^{(l+1)}) < f(x^{(l)}). \tag{4.12}$$

If $\alpha^{(l)} < 1$ and $n_l > 1$, then $x^{(l+1)} = \tilde{x}^{(l+1)}$ and we have

$$f(x^{(l+1)}) = f(\tilde{x}^{(l+1)}) < f(x^{(l)}). \tag{4.13}$$

If $\alpha^{(l)} = 1$ or $n_l = 1$, we have from Lemma 4.2

$$f(x^{(l+1)}) < f(\tilde{x}^{(l+1)}) < f(x^{(l)}). \tag{4.14}$$

□

Lemma 4.4: *For two iteration numbers l_1 and l_2 ($l_1 < l_2$) with $\alpha^{(l_1)} = 1$ or $n_{l_1} = 1$, and $\alpha^{(l_2)} = 1$ or $n_{l_2} = 1$, respectively, if we have*

$$\mathcal{NA}^{(l_1)} = \mathcal{NA}^{(l_2)}, \tag{4.15}$$

then there exists at least one $j_0 \in K \setminus \mathcal{NA}^{(l_1)}$ such that

$$x_{j_0}{}^{(l_1)} \neq x_{j_0}{}^{(l_2)}. \tag{4.16}$$

(Proof) If we have $x_j{}^{(l_1)} = x_j{}^{(l_2)}$ for all $j \in K \setminus \mathcal{NA}^{(l_1)} (= K \setminus \mathcal{NA}^{(l_2)})$, then

$$x^{(l_1)} - P_{l_1} P_{l_1}{}^\top x^{(l_1)} = x^{(l_2)} - P_{l_2} P_{l_2}{}^\top x^{(l_2)}, \tag{4.17}$$

since $\mathcal{NA}^{(l_1)} = \mathcal{NA}^{(l_2)}$ and $n_{l_1} = n_{l_2}$. We consider the following two cases:
Case 1: $\alpha^{(l_1)} = \alpha^{(l_2)} = 1$. Since $\mathcal{NA}^{(l_1)} = \mathcal{NA}^{(l_2)}$, we have

$$P_{l_1} = P_{l_2}, \quad H_{l_1} = H_{l_2}, \quad B_{l_1} = B_{l_2}. \tag{4.18}$$

In Step 2,

$$\begin{aligned}
\tilde{x}^{(l_1+1)} &= x^{(l_1)} + \alpha^{(l_1)} P_{l_1} s^{(l_1)} \\
&= x^{(l_1)} + P_{l_1}[-B_{l_1} \bar{g}^{(l_1)}] \\
&= x^{(l_1)} - P_{l_1} B_{l_1} [H_{l_1} P_{l_1}{}^\top x^{(l_1)} + P_{l_1}{}^\top b + P_{l_1}{}^\top A (I - P_{l_1} P_{l_1}{}^\top) x^{(l_1)}] \\
&= x^{(l_1)} - P_{l_1} P_{l_1}{}^\top x^{(l_1)} - P_{l_1}{}^\top B_{l_1} P_{l_1}{}^\top b - P_{l_1} B_{l_1} P_{l_1}{}^\top A [x^{(l_1)} - P_{l_1} P_{l_1}{}^\top x^{(l_1)}] \\
&= [I - P_{l_1} B_{l_1} P_{l_1}{}^\top A][x^{(l_1)} - P_{l_1} P_{l_1}{}^\top x^{(l_1)}] - P_{l_1} B_{l_1} P_{l_1}{}^\top b.
\end{aligned} \tag{4.19}$$

In the same way, we have

$$\tilde{x}^{(l_2+1)} = [I - P_{l_2} B_{l_2} P_{l_2}{}^\top A][x^{(l_2)} - P_{l_2} P_{l_2}{}^\top x^{(l_2)}] - P_{l_2} B_{l_2} P_{l_2}{}^\top b. \tag{4.20}$$

From (4.17)~(4.20), we have

$$\tilde{x}^{(l_1+1)} = \tilde{x}^{(l_2+1)}. \tag{4.21}$$

Case 2: $n_{l_1} = n_{l_2} = 1$, $\alpha^{(l_1)} < 1$ or $\alpha^{(l_2)} < 1$. In this case, there is only one element in $\mathcal{NA}^{(l_1)}$ and $\mathcal{NA}^{(l_2)}$ respectively. Suppose that $\mathcal{NA}^{(l_1)} = \mathcal{NA}^{(l_2)} = \{k\}$ for some k with $1 \le k \le n$. From (4.17), $x_i^{(l_1)} = x_i^{(l_2)}$ $(i \neq k,\ 1 \le i \le n)$ and from the definition of P_l we have

$$P_{l_1} = P_{l_2} = (0, \cdots, 0, 1, 0, \cdots, 0)^\top, \tag{4.22}$$

where the kth component is one. Then $H_{l_1} = H_{l_2} = [a_{kk}]$, $B_{l_1} = B_{l_2} = [a_{kk}^{-1}]$ and we also have $\tilde{x}_k^{(l_1+1)} = \tilde{x}_k^{(l_2+1)}$. In fact,

$$\begin{aligned}
s^{(l_1)} &= -B_{l_1}\bar{g}^{(l_1)} \\
&= -B_{l_1}[H_{l_1}{P_{l_1}}^\top x^{(l_1)} + {P_{l_1}}^\top b + {P_{l_1}}^\top A(I - P_{l_1}{P_{l_1}}^\top)x^{(l_1)}] \\
&= -x_k^{(l_1)} - a_{kk}^{-1}b_k - a_{kk}^{-1}[a_{k1}x_1^{(l_1)} + \cdots + a_{kk-1}x_{k-1}^{(l_1)} + a_{kk+1}x_{k+1}^{(l_1)} + \cdots + a_{kn}x_n^{(l_1)}] \\
&= -x_k^{(l_1)} - a_{kk}^{-1}b_k - a_{kk}^{-1}\sum_{i=1, i\neq k}^{n} a_{ki}x_i^{(l_1)}.
\end{aligned} \tag{4.23}$$

Since $\alpha^{(l_1)} < 1$, we have

$$s^{(l_1)} < 0 \quad \text{and} \quad \frac{c_k - x_k^{(l_1)}}{s^{(l_1)}} < 1 \tag{4.24}$$

or

$$s^{(l_1)} > 0 \quad \text{and} \quad \frac{d_k - x_k^{(l_1)}}{s^{(l_1)}} < 1. \tag{4.25}$$

When $s^{(l_1)} < 0$ and $(c_k - x_k^{(l_1)})/s^{(l_1)} < 1$, i.e.,

$$x^{(l_1)} - c_k < -s^{(l_1)}, \tag{4.26}$$

we have from (4.23)

$$\begin{aligned}
x^{(l_1)} - c_k &< x_k^{(l_1)} + a_{kk}^{-1}b_k + a_{kk}^{-1}\sum_{i=1, i\neq k}^{n} a_{ki}x_i^{(l_1)}, \\
-c_k &< a_{kk}^{-1}b_k + a_{kk}^{-1}\sum_{i=1, i\neq k}^{n} a_{ki}x_i^{(l_1)}.
\end{aligned} \tag{4.27}$$

In the same way, we have

$$s^{(l_2)} = -x_k^{(l_2)} - a_{kk}^{-1}b_k - a_{kk}^{-1}\sum_{i=1, i\neq k}^{n} a_{ki}x_i^{(l_2)}. \tag{4.28}$$

Since $x_i^{(l_1)} = x_i^{(l_2)}$ $(i \neq k,\ 1 \leq i \leq n)$, we have from (4.27)

$$s^{(l_2)} = -x_k^{(l_2)} - a_{kk}^{-1} b_k - a_{kk}^{-1} \sum_{i=1, i\neq k}^{n} a_{ki} x_i^{(l_1)} < c_k - x_k^{(l_2)} < 0, \tag{4.29}$$

$$\frac{c_k - x_k^{(l_2)}}{s^{(l_2)}} < 1. \tag{4.30}$$

Hence, $\tilde{x}_k^{(l_2+1)} = \tilde{x}_k^{(l_1+1)} = c_k$.

When $s^{(l_1)} > 0$ and $(d_k - x_k^{(l_1)})/s^{(l_1)} < 1$, we can also prove that

$$\tilde{x}_k^{(l_2+1)} = \tilde{x}_k^{(l_1+1)} = d_k. \tag{4.31}$$

Therefore, for Case 2 we also have

$$\tilde{x}^{(l_1+1)} = \tilde{x}^{(l_2+1)}. \tag{4.32}$$

Since $l_1 < l_2$, this is a contradiction to Theorem 4.3. □

Theorem 4.5: *The proposed algorithm solves Problem* (QP) *in finitely many steps.*
(Proof) Since the number of possible nonactive sets (and active sets) is finite, the finiteness of the proposed algorithm follows from Lemma 4.4. □

5. Computing Relevant Matrices

In the algorithm proposed in Section 3 the nonactive set $\mathcal{NA}^{(l)}$ is repeatedly changed by adding or removing a nonactive or active constraint and the relevant matrices are updated accordingly. In this section we describe an efficient way of updating relevant matrices. Before we get into the detail, we first describe how to initiate the data for the algorithm. We choose an initial point given as follows: Choose an index $i_0 \in K$ and t with $c_{i_0} < t < d_{i_0}$ and put

$$x^{(0)} = \begin{cases} t, & i = i_0, \\ c_i \text{ or } d_i, & i \neq i_0. \end{cases}$$

Then we can start the algorithm with very simple parameters:

$$\mathcal{NA}^{(0)} = \{i_0\}, \tag{5.1}$$

$$P_0 = (0, \cdots, 0, 1, 0, \cdots, 0)^{\top}, \tag{5.2}$$

$$H_0 = P_0^{\top} A P_0 = [a_{i_0 i_0}], \quad B_0 = H_0^{-1} = [1/a_{i_0 i_0}]. \tag{5.3}$$

Notice that 1 appearing in P_0 is the i_0th component.

5.1. Removing a nonactive constraint

In this case, $|\mathcal{NA}^{(l)}| = |\mathcal{NA}^{(l-1)}| - 1$, then the number of columns in P_l is one less than that in P_{l-1}. Without loss of generality we assume that $P_{l-1} = [P_l, p_1]$. Then H_{l-1} and H_l have the following relationship:

$$H_{l-1} = \begin{bmatrix} H_l & P_l^\top A p_1 \\ p_1^\top A P_l & p_1^\top A p_1 \end{bmatrix}.$$

Also, we have

$$B_{l-1} = \begin{bmatrix} \bar{B}_{l-1} & h_1 \\ h_1^\top & t_1 \end{bmatrix},$$

and

$$B_l = H_l^{-1} = \bar{B}_{l-1} - h_1 h_1^\top / t_1.$$

5.2. Adding a nonactive constraint

In this case, $|\mathcal{NA}^{(l)}| = |\mathcal{NA}^{(l-1)}| + 1$, then the number of columns in P_l is one more than that in P_{l-1}. We also assume that $P_l = [P_{l-1}, p_2]$. Then,

$$H_l = \begin{bmatrix} H_{l-1} & P_{l-1}^\top A p_2 \\ p_2^\top A P_{l-1} & p_2^\top A p_2 \end{bmatrix}. \tag{5.4}$$

Letting

$$h_2 = P_{l-1}^\top A p_2, \quad t_2 = p_2^\top A p_2, \quad \alpha = 1/(t_2 - h_2^\top B_{l-1} h_2), \quad \hat{h} = -\alpha B_{l-1} h_2, \tag{5.5}$$

we have

$$B_l = H_l^{-1} = \begin{bmatrix} B_{l-1} + \hat{h}\hat{h}^\top/\alpha & \hat{h} \\ \hat{h}^\top & \alpha \end{bmatrix}. \tag{5.6}$$

In this way we can efficiently compute H_l^{-1}.

Acknowledgments

The first author would like to express his sincere appreciation to Professor Yoshitsugu Yamamoto for providing him the opportunity to work at the Institute of Policy and Planning Sciences, University of Tsukuba for four months. Special thanks go to Japan Society for the Promotion of Science (JSPS) for the finacial support to the first author. The third anthor's work is supported by a grant-in-aid of the Ministry of Educatoin, Science and Culture of Japan.

References

[1] D. Goldfarb and A. Idnani: A numerically stable dual metmod for solving strictly convex quadratic programs. *Mathematical Programming* **27** (1983), 1-33.

[2] E. Yang and J. W. Tolle: A class of methods for solving large, convex quadratic programs subject to box constraints. *Mathematical Programming* **51** (1991), 229-245.

[3] Y. C. Jiao: Study on constrained optimization and its application to optimal design of antennas. Ph. D. Dissertation, Xidian University, Xi'an (May 1990).

[4] R. H. Nickel and J. W. Tolle: A sequential quadratic programming algorithm for solving large, sparse nonlinear programs. *Journal of Optimization Theory and Its Application* **60** (1989), 453-473.

[5] R. S. Dembo and U. Tulowitzski: On the minimization of a quadratic function subject to box constraints. Working paper No. 71, Series B, School of Organization and Management, Yale University (New Haven, CT) (1983).