Journal of the Operations Research
Society of Japan
Vol. 36, No. 2, June 1993

A SUCCESSIVE OVER-RELAXATION METHOD FOR QUADRATIC PROGRAMMING PROBLEMS WITH INTERVAL CONSTRAINTS

Yoshikazu Shimazu, *Showa Denko K.K.*
Masao Fukushima, *Advanced Institute of Science and Technology, Nara*
Toshihide Ibaraki, *Kyoto University*

(Received April 6, 1992; Revised November 16, 1992)

Abstract Hildreth's algorithm is a classical iterative method for solving strictly convex quadratic programming problems, which uses the rows of constraint matrix just one at a time. This algorithm is particularly suited to large and sparse problems, because it acts upon the given problem data directly and the coefficient matrix is never modified in the course of the iterations. The original Hildreth's algorithm is mathematically equivalent to Gauss-Seidel method applied to the dual of the given quadratic programming problem. In this paper, we propose an SOR modification of Hildreth's algorithm for solving interval constrained quadratic programming problems. We prove global convergence of the algorithm and show that the rate of convergence is linear. Computational results are also presented to demonstrate the effectiveness of the algorithm.

1 Introduction

Consider the strictly convex quadratic programming problem

$$\begin{array}{lll} \mathrm{P:} & \min & \frac{1}{2} \| x - x^0 \|_Q^2 \\ & \text{s.t.} & \gamma \le Ax \le \delta, \\ & & Bx = d, \end{array}$$

where A and B are given matrices with dimension $m \times n$ and $l \times n$, respectively, $x^0 \in R^n$, $\gamma, \delta \in R^m$, $d \in R^l$ are given vectors, and $\| x \|_Q$ denotes the Q-norm of vector $x \in R^n$ defined by $\| x \|_Q^2 = \langle Qx, x \rangle$, where Q is a given $n \times n$ symmetric positive-definite matrix and $\langle \cdot, \cdot \rangle$ stands for the inner product in R^n. (The ordinary Euclidean norm will be denoted by $\| \cdot \|$.) The pairs of inequality constraints in problem P are referred to as *interval constraints*. Interval constraints often appear in optimization problems that arise in various fields such as network flows [10, 11, 12] and computer tomography [4]. Throughout this paper, we assume that problem P has a solution, which is necessarily unique by the positive-definiteness of matrix Q. We also assume that the constraint matrices do not contain any row of which elements are all zero.

Hildreth's algorithm [5] is a classical iterative method for solving the quadratic programming problem

$$\begin{array}{lll} \mathrm{P}': & \min & \frac{1}{2} \| x - x^0 \|_Q^2 \\ & \text{s.t.} & Ax \le b, \end{array}$$

where A, b, x^0 and Q are as above. This method is particularly suitable for large and sparse problems, because it uses the rows of constraint matrix A just one at a time and acts upon the given problem data directly without changing the original matrix in the course of the iterations [2, 9, 11, 12]. Moreover if the objective function takes a separable form, i.e., the matrix Q is diagonal, then each step of the algorithm can be executed in an extremely simple manner. It is known that the original Hildreth's algorithm is mathematically equivalent to

Gauss-Seidel method applied to the dual of the given quadratic programming problem [1, Section 3.4]. From this viewpoint, Lent and Censor [8] have proposed a successive overrelaxation (SOR) modification of Hildreth's algorithm in order to accelerate its convergence.

Hildreth's algorithm for P′ may naturally be applied to problem P by considering a pair of inequalities to be two separate inequalities. But this approach is by no means the best. By taking into account the special structure of the problem, Herman and Lent [4] extended Hildreth's algorithm to deal with interval constraints directly (see also [3]). The advantage of the latter method over the primitive application of Hildreth's algorithm to problem P may be stated as follows: First, this method needs only one dual variable, rather than two dual variables, associated with each pair of constraints. Second, the number of iterations to solve the problem is reduced by half.

Unlike the ordinary inequality constrained problem P′, however, few attempts have been made to obtain an SOR version of the algorithm for the interval constrained problem P. In this paper, we present an SOR modification of the extended Hildreth's algorithm for solving problem P, which can deal with the interval constraints in a direct manner. We show that the proposed algorithm converges to the solution of the problem and the rate of convergence is linear. We also demonstrate its practical effectiveness by numerical experiments and, in particular, exhibit some results of parallel computation for nonlinear transportation problems.

This paper consists of six sections. As preliminaries, the original Hildreth's algorithm together with its SOR modification for problem P′ and its extension to the interval constrained problem P is described in Section 2. An SOR modification of the extended Hildreth's algorithm for problem P is then presented in Section 3. In Section 4, the proposed algorithm are shown to converge to the solution at a linear rate. In Section 5, computational results with the proposed algorithm are reported. Finally Section 6 concludes the paper.

2 Preliminaries

In this section, we review two variants of Hildreth's algorithm as well as its original form for solving problem P′. One of the variants is the SOR modification of the algorithm for problem P′ presented by Lent and Censor [8], the other is the extension to the interval constrained problem P proposed by Censor and Lent [3]. These algorithms form a basis of the algorithms to be proposed in the next section.

The abovementioned variants of Hildreth's algorithm use the following almost cyclic rule to specify the row of the matrix to be chosen during a given iterative step: Let $I = \{1, 2, \ldots, m\}$. A sequence $\{i_k\}_{k=0}^{\infty}$ is said to be *almost cyclic on I* if $i_k \in I$ for all $k \geq 0$, and there exists an integer $C > 0$ such that for $I \subseteq \{i_{k+1}, \ldots, i_{k+C}\}$ for all $k \geq 0$.

For problem P′, the original Hildreth's algorithm may be stated as follows:

Algorithm 2.1 (Original Hildreth's algorithm).
Initialization: Let $(x^{(0)}, u^{(0)}) := (x^0, 0)$.
Iteration k: Choose an index $i_k \in I$ satisfying the almost cyclic rule and let

$$\begin{aligned}
\alpha_{i_k} &:= \langle a_{i_k}, Q^{-1} a_{i_k} \rangle, \\
\Theta^{(k)} &:= \frac{b_{i_k} - \langle a_{i_k}, x^{(k)} \rangle}{\alpha_{i_k}}, \\
c^{(k)} &:= \min\left(u_{i_k}^{(k)}, \Theta^{(k)}\right) \qquad (1) \\
x^{(k+1)} &:= x^{(k)} + c^{(k)} Q^{-1} a_{i_k},
\end{aligned}$$

$$u^{(k+1)} := u^{(k)} - c^{(k)} e_{i_k},$$

where a_{i_k} denotes the i_kth row of matrix A and e_{i_k} is the i_kth unit vector in R^m. (In the literature, a_{i_k} is often treated as a column vector interchangeably without notice. Strictly speaking, some of the above equations do not make sense if we interpret a_{i_k} as a row vector. Nevertheless we shall admit this misuse in order to avoid confusion with the existing work.)

This algorithm generates two sequences $\{x^{(k)}\}$ and $\{u^{(k)}\}$ in the spaces R^n and R^m, respectively. It is not difficult to observe that $x^{(k)}$ and $u^{(k)}$ always satisfy

$$x^{(k)} = x^0 - Q^{-1} A^T u^{(k)}, \tag{2}$$

where T denotes transposition of a matrix. We may regard $\{x^{(k)}\}$ and $\{u^{(k)}\}$ as sequences of *primal* variables and *dual* variables, respectively.

2.1 SOR modification

Relaxed version [8] of Hildreth's algorithm for problem P′ determines the *step size* $c^{(k)}$ by the following formula instead of (1).

$$c^{(k)} := \min\left(u_{i_k}^{(k)}, \omega\Theta^{(k)}\right) \tag{3}$$

where ω is a relaxation parameter. It is known [6, 8] that this algorithm converges to the solution of problem P′, provided that ω is chosen to satisfy $0 < \omega < 2$. In particular, we may call the algorithm with (3) a successive overrelaxation (SOR) method if $1 < \omega < 2$, because the original Hildreth's algorithm (Algorithm 2.1), which corresponds to the case $\omega = 1$, is essentially solving the dual of problem P′ by Gauss-Seidel method [1, Subsection 3.4.1].

2.2 Extension to interval constrained problems

Herman and Lent [4] extended Algorithm 2.1 to the following quadratic programming problem with interval constraints.

$$\hat{\mathrm{P}}: \begin{array}{ll} \min & \frac{1}{2} \| x - x^0 \|_Q^2 \\ \text{s.t.} & \gamma \le Ax \le \delta. \end{array}$$

Since problem $\hat{\mathrm{P}}$ can be transformed into the form of problem P′ in a trivial manner, it seems quite natural to apply Hildreth's algorithm directly to the transformed problem. However, such an approach would double the number of dual variables compared with the following algorithm [4], which deals only with m dual variables $u \in R^m$.

Algorithm 2.2.

Initialization: Let $(x^{(0)}, u^{(0)}) := (x^0, 0)$.

Iteration k: Choose an index $i_k \in I$ satisfying the almost cyclic rule and let

$$\alpha_{i_k} := \langle a_{i_k}, Q^{-1} a_{i_k} \rangle,$$

$$\Delta^{(k)} := \frac{\delta_{i_k} - \langle a_{i_k}, x^{(k)} \rangle}{\alpha_{i_k}},$$

$$\Gamma^{(k)} := \frac{\gamma_{i_k} - \langle a_{i_k}, x^{(k)} \rangle}{\alpha_{i_k}},$$

$$c^{(k)} := \mathrm{mid}\left(u^{(k)}_{i_k}, \Delta^{(k)}, \Gamma^{(k)}\right),$$
$$x^{(k+1)} := x^{(k)} + c^{(k)}Q^{-1}a_{i_k},$$
$$u^{(k+1)} := u^{(k)} - c^{(k)}e_{i_k},$$

where mid denote the median of three real numbers.

It has been proved that Algorithm 2.2 converges to the solution of problem $\hat{\mathrm{P}}$ [3, 8] and its convergence rate is linear [6].

3 SOR methods for interval constraints

In this section, we present an SOR modification of the extended Hildreth's algorithm (Algorithm 2.2) for solving interval constrained problems. As in Subsection 2.2, we restrict our attention to the problem with pure interval constraints

$$\begin{array}{lll} \hat{\mathrm{P}}: & \min & \frac{1}{2} \| x - x^0 \|^2_Q \\ & \text{s.t.} & \gamma \le Ax \le \delta, \end{array}$$

where γ and δ are assumed to satisfy $\gamma < \delta$. The reason why we consider the purely interval case is just for simplicity of presentation. In fact, presence of equality constraints will not cause any trouble and they are even more tractable than inequality or interval constraints.

The algorithm is stated as follows:

Algorithm 3.1.

Initialization: Let $(x^{(0)}, u^{(0)}) := (x^0, 0)$ and choose a relaxation parameter $\omega \in (0, 2)$.

Iteration k: Choose an index $i_k \in I$ satisfying the almost cyclic rule and let

$$\alpha_{i_k} := \langle a_{i_k}, Q^{-1}a_{i_k}\rangle,$$
$$\Delta^{(k)} := \frac{\delta_{i_k} - \langle a_{i_k}, x^{(k)}\rangle}{\alpha_{i_k}},$$
$$\Gamma^{(k)} := \frac{\gamma_{i_k} - \langle a_{i_k}, x^{(k)}\rangle}{\alpha_{i_k}},$$
$$c^{(k)} := \mathrm{mid}\left(u^{(k)}_{i_k}, \omega\Delta^{(k)}, \omega\Gamma^{(k)}\right),$$
$$x^{(k+1)} := x^{(k)} + c^{(k)}Q^{-1}a_{i_k},$$
$$u^{(k+1)} := u^{(k)} - c^{(k)}e_{i_k}.$$

It should be noted that, since it is assumed that $\gamma < \delta$, the following inequality is always satisfied:

$$\Gamma^{(k)} < \Delta^{(k)}. \tag{4}$$

Moreover, it follows from (4) that the step size $c^{(k)}$ satisfies the inequalities

$$\omega\Gamma^{(k)} \le c^{(k)} \le \omega\Delta^{(k)} \tag{5}$$

Also note that, if $\omega = 1$, then Algorithm 3.1 is reduced to Algorithm 2.2.

4 Convergence Theorems

In this section, we prove that Algorithm 3.1 described in the previous section converges to the solution of problem $\hat{\mathrm{P}}$ at a linear rate. We first prove global convergence of the algorithm and then establish its linear rate of convergence.

Throughout this section, it will be helpful to consider the dual of problem $\hat{\mathrm{P}}$

$$\begin{array}{lll} \hat{\mathrm{D}}: & \max & \Phi(z^+, z^-) \\ & \text{s.t.} & z^+ \geq 0, \\ & & z^- \geq 0, \end{array}$$

where $z^+ \in R^m$ and $z^- \in R^m$ are dual variables and $\Phi : R^{2m} \to R$ is a concave quadratic function defined by

$$\Phi(z^+, z^-) = -\frac{1}{2}\langle z^+ - z^-, AQ^{-1}A^T(z^+ - z^-)\rangle + \langle z^+, Ax^0 - \delta\rangle + \langle z^-, \gamma - Ax^0\rangle. \tag{6}$$

4.1 Global convergence

The proof of global convergence consists of the following steps. First, we present an algorithm which generates sequences $\{z^{+(k)}\}$ and $\{z^{-(k)}\}$ feasible to the dual problem $\hat{\mathrm{D}}$, and show that this algorithm is equivalent to Algorithm 3.1. Next, by proving convergence of the former algorithm, we establish a convergence theorem for Algorithm 3.1.

Now let us consider the following algorithm.

Algorithm 4.1.

Initialization: Let $(x^{(0)}, z^{+(0)}, z^{-(0)}) := (x^0, 0, 0)$ and choose a relaxation parameter $\omega \in (0, 2)$.

Iteration k: Choose an index $i_k \in I$ satisfying the almost cyclic rule and let

$$\alpha_{i_k} := \langle a_{i_k}, Q^{-1}a_{i_k}\rangle,$$

$$\Delta^{(k)} := \frac{\delta_{i_k} - \langle a_{i_k}, x^{(k)}\rangle}{\alpha_{i_k}},$$

$$\Gamma^{(k)} := \frac{\gamma_{i_k} - \langle a_{i_k}, x^{(k)}\rangle}{\alpha_{i_k}},$$

if $z_{i_k}^{+(k)} \geq z_{i_k}^{-(k)}$ then

$$c^{+(k)} := \min\left(z_{i_k}^{+(k)}, \omega\Delta^{(k)}\right),$$

$$c^{-(k)} := \min\left(z_{i_k}^{-(k)}, -\omega\Gamma^{(k)} + c^{+(k)}\right)$$

else

$$c^{-(k)} := \min\left(z_{i_k}^{-(k)}, -\omega\Gamma^{(k)}\right),$$

$$c^{+(k)} := \min\left(z_{i_k}^{+(k)}, \omega\Delta^{(k)} + c^{-(k)}\right)$$

endif

$$x^{(k+1)} := x^{(k)} + (c^{+(k)} - c^{-(k)})Q^{-1}a_{i_k}, \tag{7}$$

$$z^{+(k+1)} := z^{+(k)} - c^{+(k)}e_{i_k}, \tag{8}$$

$$z^{-(k+1)} := z^{-(k)} - c^{-(k)}e_{i_k}. \tag{9}$$

Lemma 4.1 *Let $\{x^{(k)}\}$, $\{z^{+(k)}\}$ and $\{z^{-(k)}\}$ be generated by Algorithm 4.1. Then for all k, we have*

$$x^{(k)} = x^0 - Q^{-1}A^T\left(z^{+(k)} - z^{-(k)}\right), \tag{10}$$

$$z^{+(k)} \geq 0, \tag{11}$$

$$z^{-(k)} \geq 0, \tag{12}$$

$$\langle z^{+(k)}, z^{-(k)}\rangle = 0. \tag{13}$$

Proof. (10)-(12) directly follow from the manner in which $\{x^{(k)}\}$, $\{z^{+(k)}\}$ and $\{z^{-(k)}\}$ are updated in the algorithm. We prove (13) by induction. For $k = 0$, (13) trivially holds. We then assume $\langle z^{+(k)}, z^{-(k)}\rangle = 0$ and show that it is also true for $k+1$. We shall only consider the case where $z_{i_k}^{+(k)} \geq z_{i_k}^{-(k)}$, because a parallel argument is valid for the opposite case. First note that, when $z_{i_k}^{+(k)} \geq z_{i_k}^{-(k)}$, (11) and (12) imply $z_{i_k}^{+(k)} \geq 0$ and $z_{i_k}^{-(k)} = 0$. Moreover, if $z_{i_k}^{+(k)} \geq \omega\Delta^{(k)}$ holds, then we have $c^{+(k)} = \omega\Delta^{(k)}$ and $c^{-(k)} = \min\left(0, \omega(\Delta^{(k)} - \Gamma^{(k)})\right) = 0$, where the last equality follows from (4). Therefore we must have $z_{i_k}^{-(k+1)} = 0$. On the other hand, if $z_{i_k}^{+(k)} < \omega\Delta^{(k)}$ holds, then we have $c^{+(k)} = z_{i_k}^{+(k)}$, which in turn implies $z_{i_k}^{+(k+1)} = 0$. Thus (13) is satisfied for $k+1$. □

For each i, either $z_i^{+(k)} = 0$ or $z_i^{-(k)} = 0$ must always hold by (11), (12) and (13). Moreover, we can deduce the following relations:
If $z_{i_k}^{+(k)} \geq z_{i_k}^{-(k)}$, i.e., $z_{i_k}^{+(k)} \geq 0, z_{i_k}^{-(k)} = 0$, then

$$(c^{+(k)}, c^{-(k)}) = \begin{cases} (\omega\Delta^{(k)}, 0) & \text{if } z_{i_k}^{+(k)} \geq \omega\Delta^{(k)}, \\ (z_{i_k}^{+(k)}, 0) & \text{if } \omega\Delta^{(k)} \geq z_{i_k}^{+(k)} \geq \omega\Gamma^{(k)}, \\ (z_{i_k}^{+(k)}, -\omega\Gamma^{(k)} + z_{i_k}^{+(k)}) & \text{if } \omega\Gamma^{(k)} \geq z_{i_k}^{+(k)}. \end{cases} \tag{14}$$

If $z_{i_k}^{+(k)} \leq z_{i_k}^{-(k)}$, i.e., $z_{i_k}^{+(k)} = 0, z_{i_k}^{-(k)} \geq 0$, then

$$(c^{+(k)}, c^{-(k)}) = \begin{cases} (0, -\omega\Gamma^{(k)}) & \text{if } z_{i_k}^{-(k)} \geq -\omega\Gamma^{(k)}, \\ (0, z_{i_k}^{-(k)}) & \text{if } -\omega\Gamma^{(k)} \geq z_{i_k}^{-(k)} \geq -\omega\Delta^{(k)}, \\ (\omega\Delta^{(k)} + z_{i_k}^{-(k)}, z_{i_k}^{-(k)}) & \text{if } -\omega\Delta^{(k)} \geq z_{i_k}^{-(k)}. \end{cases} \tag{15}$$

The next theorem demonstrates the equivalence between Algorithm 3.1 and Algorithm 4.1.

Theorem 4.1 *Algorithm 3.1 and Algorithm 4.1 are equivalent in the sense that*

$$u^{(k)} = z^{+(k)} - z^{-(k)} \tag{16}$$

$$c^{(k)} = c^{+(k)} - c^{-(k)}, \tag{17}$$

hold for all k and the two algorithms generate an identical sequence $\{x^{(k)}\}$, provided that the same index i_k is chosen at each iteration k.

Proof. The proof is by induction. Since $u^{(0)} = 0$ and $z^{+(0)} = z^{-(0)} = 0$, (16) is obvious for $k = 0$. Now assuming that (16) is true for k, we shall show that (17) and (16) hold for k and $k+1$, respectively.

Since either $z_i^{+(k)} = 0$ or $z_i^{-(k)} = 0$ holds, (14) and (15) imply

$$c^{+(k)} - c^{-(k)} = \mathrm{mid}\left(z_{i_k}^{+(k)} - z_{i_k}^{-(k)}, \omega\Delta^{(k)}, \omega\Gamma^{(k)}\right). \tag{18}$$

Moreover from the manner in which $z^{+(k)}$ and $z^{-(k)}$ are updated, it follows that

$$z^{+(k+1)} - z^{-(k+1)} = z^{+(k)} - z^{-(k)} - \left(c^{+(k)} - c^{-(k)}\right) e_{i_k}. \tag{19}$$

On the other hand, we have

$$c^{(k)} = \mathrm{mid}\left(u_{i_k}^{(k)}, \omega\Delta^{(k)}, \omega\Gamma^{(k)}\right) \tag{20}$$

and

$$u^{(k+1)} = u^{(k)} - c^{(k)} e_{i_k}. \tag{21}$$

Thus, since (16) is true for k, (18) and (20) imply that (17) holds for k. Moreover, (19) and (21) together with (17) and (16) imply that (16) is true for $k+1$. The last part of the theorem is a consequence of (17). □

In the following, we shall prove convergence of Algorithm 4.1. The method of proof is an extension of the one given by Lent and Censor [8] for a relaxed version of Hildreth's algorithm for solving one-sided inequality constrained problems.

Lemma 4.2 *For each k, we have*

$$c^{+(k)}\Delta^{(k)} - c^{-(k)}\Gamma^{(k)} - \frac{1}{2}\left(c^{+(k)} - c^{-(k)}\right)^2 \geq \frac{1}{2}\left(-1 + \frac{2}{\omega}\right)\left\{\left(c^{+(k)}\right)^2 + \left(c^{-(k)}\right)^2\right\}.$$

Proof. Define

$$d^{(k)} = c^{+(k)}\Delta^{(k)} - c^{-(k)}\Gamma^{(k)} - \frac{1}{2}\left(c^{+(k)} - c^{-(k)}\right)^2.$$

Suppose that $z_{i_k}^{+(k)} \geq z_{i_k}^{-(k)}$. Since the value of $\left(c^{+(k)}, c^{-(k)}\right)$ is determined by (14), we may consider the following three cases.

(i) $z_{i_k}^{+(k)} \geq \omega\Delta^{(k)}$: Since $(c^{+(k)}, c^{-(k)}) = (\omega\Delta^{(k)}, 0)$ by (14), direct calculation yields

$$d^{(k)} = \frac{1}{2}\left(-1 + \frac{2}{\omega}\right)\left\{\left(c^{+(k)}\right)^2 + \left(c^{-(k)}\right)^2\right\}. \tag{22}$$

(ii) $\omega\Delta^{(k)} \geq z_{i_k}^{+(k)} \geq \omega\Gamma^{(k)}$: Since $(c^{+(k)}, c^{-(k)}) = (z_{i_k}^{+(k)}, 0)$ by (14), we have

$$\begin{aligned} d^{(k)} &= \frac{1}{2}\left(-1 + \frac{2\Delta^{(k)}}{c^{+(k)}}\right)\left\{\left(c^{+(k)}\right)^2 + \left(c^{-(k)}\right)^2\right\} \\ &\geq \frac{1}{2}\left(-1 + \frac{2}{\omega}\right)\left\{\left(c^{+(k)}\right)^2 + \left(c^{-(k)}\right)^2\right\}, \end{aligned} \tag{23}$$

where the last inequality follows from $c^{+(k)} = z_{i_k}^{+(k)} \leq \omega\Delta^{(k)}$.

(iii) $\omega\Gamma^{(k)} \geq z_{i_k}^{+(k)}$: Since $(c^{+(k)}, c^{-(k)}) = (z_{i_k}^{+(k)}, -\omega\Gamma^{(k)} + z_{i_k}^{+(k)})$ by (14), direct calculation yields

$$d^{(k)} = \frac{1}{2}\left(c^{+(k)}\right)^2 \frac{2\Delta^{(k)} - \omega\Gamma^{(k)}}{z_{i_k}^{+(k)}} + \frac{1}{2}\left(c^{-(k)}\right)^2 \frac{(-2+\omega)\Gamma^{(k)}}{c^{-(k)}}.$$

But since $\omega\Gamma^{(k)} \geq z_{i_k}^{+(k)} \geq 0$ and $\Delta^{(k)} > \Gamma^{(k)}$, we have

$$\frac{2\Delta^{(k)} - \omega\Gamma^{(k)}}{z_{i_k}^{+(k)}} \geq -1 + \frac{2}{\omega}.$$

Also since $\omega\Gamma^{(k)} \geq z_{i_k}^{+(k)} \geq 0$ implies $0 \geq c^{-(k)} = -\omega\Gamma^{(k)} + z_{i_k}^{+(k)} \geq -\omega\Gamma^{(k)}$, we have

$$\frac{(-2+\omega)\Gamma^{(k)}}{c^{-(k)}} \geq -1 + \frac{2}{\omega}.$$

Thus we get

$$d^{(k)} \geq \frac{1}{2}\left(-1 + \frac{2}{\omega}\right)\left\{\left(c^{+(k)}\right)^2 + \left(c^{-(k)}\right)^2\right\}. \tag{24}$$

Consequently (22) – (24) imply that the lemma is true when $z_{i_k}^{+(k)} \geq z_{i_k}^{-(k)}$. Proof for the case $z_{i_k}^{+(k)} \leq z_{i_k}^{-(k)}$ goes through using a similar argument. □

Lemma 4.3 *Let Φ be the objective function of the dual problem $\hat{D}$. Then the following relations hold for Algorithm 4.1:*

(a) $\Phi(z^{+(k+1)}, z^{-(k+1)}) \geq \Phi(z^{+(k)}, z^{-(k)})$ for all k;
(b) $\left(\Phi(z^{+(k+1)}, z^{-(k+1)}) - \Phi(z^{+(k)}, z^{-(k)})\right) \to 0 \quad (k \to \infty)$;
(c) $c^{+(k)} \to 0,\ c^{-(k)} \to 0 \quad (k \to \infty)$;
(d) $\left(x^{(k+1)} - x^{(k)}\right) \to 0 \quad (k \to \infty)$;
(e) $\left(z^{+(k+1)} - z^{+(k)}\right) \to 0,\ \left(z^{-(k+1)} - z^{-(k)}\right) \to 0 \quad (k \to \infty)$.

Proof. Let us denote

$$\Delta\Phi^{(k)} = \Phi(z^{+(k+1)}, z^{-(k+1)}) - \Phi(z^{+(k)}, z^{-(k)}),$$

which, by the definition (6) of Φ, can be written

$$\begin{aligned}\Delta\Phi^{(k)} &= -\frac{1}{2}\langle z^{+(k+1)} - z^{-(k+1)}, AQ^{-1}A^T(z^{+(k+1)} - z^{-(k+1)})\rangle \\ &\quad +\frac{1}{2}\langle z^{+(k)} - z^{-(k)}, AQ^{-1}A^T(z^{+(k)} - z^{-(k)})\rangle \\ &\quad +\langle z^{+(k+1)} - z^{+(k)}, Ax^0 - \delta\rangle + \langle z^{-(k+1)} - z^{-(k)}, \gamma - Ax^0\rangle. \end{aligned} \tag{25}$$

Then by (19), we have

$$\begin{aligned} &-\frac{1}{2}\langle z^{+(k+1)} - z^{-(k+1)}, AQ^{-1}A^T(z^{+(k+1)} - z^{-(k+1)})\rangle \\ &\quad +\frac{1}{2}\langle z^{+(k)} - z^{-(k)}, AQ^{-1}A^T(z^{+(k)} - z^{-(k)})\rangle \\ &= \left(c^{+(k)} - c^{-(k)}\right)\langle e_{i_k}, AQ^{-1}A^T(z^{+(k)} - z^{-(k)})\rangle \\ &\quad -\frac{1}{2}\left(c^{+(k)} - c^{-(k)}\right)^2\langle e_{i_k}, AQ^{-1}A^T e_{i_k}\rangle \\ &= \left(c^{+(k)} - c^{-(k)}\right)\langle a_{i_k}, x^0 - x^{(k)}\rangle - \frac{1}{2}\left(c^{+(k)} - c^{-(k)}\right)^2\alpha_{i_k}, \end{aligned} \tag{26}$$

where the last equality follows from (10) and the definition of α_{i_k}. On the other hand, by (8) and the definition of $\Delta^{(k)}$, we have

$$\begin{aligned}\langle z^{+(k+1)} - z^{+(k)}, Ax^0 - \delta\rangle &= c^{+(k)}\langle e_{i_k}, \delta - Ax^0\rangle \\ &= c^{+(k)}\left(\delta_{i_k} - \langle a_{i_k}, x^{(k)}\rangle\right) - c^{+(k)}\langle a_{i_k}, x^0 - x^{(k)}\rangle \\ &= \alpha_{i_k} c^{+(k)}\Delta^{(k)} - c^{+(k)}\langle a_{i_k}, x^0 - x^{(k)}\rangle. \end{aligned} \tag{27}$$

Similarly we have

$$\langle z^{-(k+1)} - z^{-(k)}, \gamma - Ax^0\rangle = -\alpha_{i_k} c^{-(k)}\Gamma^{(k)} + c^{-(k)}\langle a_{i_k}, x^0 - x^{(k)}\rangle. \tag{28}$$

Thus, it follows from (25) - (28) that

$$\Delta\Phi^{(k)} = \alpha_{i_k}\left(c^{+(k)}\Delta^{(k)} - c^{-(k)}\Gamma^{(k)} - \frac{1}{2}\left(c^{+(k)} - c^{-(k)}\right)^2\right). \tag{29}$$

Now let ϵ be a positive number defined by $\epsilon = \min\{\langle a_i, Q^{-1}a_i\rangle \mid i = 1, ..., m\}$. Then Lemma 4.2 and (29) imply that for all k

$$\Delta\Phi^{(k)} \geq \frac{1}{2}\epsilon\left(-1 + \frac{2}{\omega}\right)\left\{\left(c^{+(k)}\right)^2 + \left(c^{-(k)}\right)^2\right\}, \tag{30}$$

which implies (a). Next note that by the assumption that problem $\hat{\mathrm{P}}$ has a solution, its dual $\hat{\mathrm{D}}$ has an optimal solution. Since $(z^{+(k)}, z^{-(k)})$ are feasible to $\hat{\mathrm{D}}$, $\Phi(z^{+(k)}, z^{-(k)})$ is bounded above by the optimal value of $\hat{\mathrm{D}}$. Therefore (a) implies (b), which together with (30) in turn implies (c). Finally (d) and (e) immediately follow from (c). □

To proceed further, we introduce a sequence of perturbed constraint sets $S^{(k)}$, which converges to the constraint set of problem $\hat{\mathrm{P}}$, say S. Specifically, let $\{q^{+(k)}\}$ and $\{q^{-(k)}\}$ be sequences of vectors defined inductively as follows:
For $k = 0$, let $q^{+(0)} := 0$, $q^{-(0)} := 0$. For $k \geq 1$, if $z_{i_k}^{+(k)} \geq z_{i_k}^{-(k)}$ then

$$q^{+(k+1)} := q^{+(k)} - \left(q_{i_k}^{+(k)} + \frac{\alpha_{i_k}}{\omega}\left\{c^{+(k)} - \omega\Delta^{(k)}\right\}\right)e_{i_k}, \tag{31}$$

$$q^{-(k+1)} := q^{-(k)} - \left(q_{i_k}^{-(k)} + \frac{\alpha_{i_k}}{\omega}\left\{\left(c^{+(k)} - c^{-(k)}\right) - \omega\Gamma^{(k)}\right\}\right)e_{i_k} \tag{32}$$

else

$$q^{+(k+1)} := q^{+(k)} - \left(q_{i_k}^{+(k)} + \frac{\alpha_{i_k}}{\omega}\left\{\left(c^{+(k)} - c^{-(k)}\right) - \omega\Delta^{(k)}\right\}\right)e_{i_k}, \tag{33}$$

$$q^{-(k+1)} := q^{-(k)} - \left(q_{i_k}^{-(k)} + \frac{\alpha_{i_k}}{\omega}\left\{-c^{-(k)} - \omega\Gamma^{(k)}\right\}\right)e_{i_k}, \tag{34}$$

where i_k is the index chosen at iteration k.

Using these vectors, we define the vectors $\delta^{(k)}, \gamma^{(k)}$ by

$$\delta^{(k)} = q^{+(k)} + Ax^{(k)}, \tag{35}$$

$$\gamma^{(k)} = q^{-(k)} + Ax^{(k)}, \tag{36}$$

respectively, and construct a sequence of perturbed constraint sets $S^{(k)}$ by

$$S^{(k)} = \left\{x \in R^n \mid \gamma^{(k)} \leq Ax \leq \delta^{(k)}\right\}. \tag{37}$$

Lemma 4.4 *The following relations hold:*

(a) $q^{+(k)} \geq 0, q^{-(k)} \leq 0$ for all k;
(b) $\delta^{(k)} \to \delta, \gamma^{(k)} \to \gamma \ (k \to \infty)$;
(c) $\langle q^{+(k)}, z^{+(k)} \rangle = 0, \langle q^{-(k)}, z^{-(k)} \rangle = 0$ for all k;
(d) $\| x^{(k)} - x^0 \|_Q \leq \| x - x^0 \|_Q, \forall x \in S^{(k)}$, for all k.

Proof. By (14), (15), (31) – (34), we have (a). We omit the proof of (b), because it can be proved using (c) and (d) of Lemma 4.3 in a way similar to Lemma 4.4 of [8]. (c) can be proved by induction using (14) and (15). For instance, if $z_{i_k}^{+(k)} \geq z_{i_k}^{-(k)}$, then by (31)

$$q_{i_k}^{+(k+1)} = \frac{\alpha_{i_k}}{\omega} \left\{ \omega \Delta^{(k)} - c^{+(k)} \right\}$$

and by the updating rule for $z^{+(k)}$ in the algorithm

$$z_{i_k}^{+(k+1)} = z_{i_k}^{+(k)} - c^{+(k)}.$$

But by (14)

$$c^{+(k)} = \min \left(z_{i_k}^{+(k)}, \omega \Delta^{(k)} \right).$$

Therefore either $q_{i_k}^{+(k+1)}$ or $z_{i_k}^{+(k+1)}$ must vanish. The opposite case can be treated in a similar manner using (15). Finally, by (c) in Lemma 4.4, (d) can be proved in a way similar to Theorem 5.1 of [8]. □

Lemma 4.5 *Let x^* denote the solution of problem $\hat{P}$ and $\{x^{(k)}\}$ be a sequence generated by Algorithm 4.1. Let $\hat{x}^{(k)}$ and $\tilde{x}^{(k)}$ be the closest point in S to $x^{(k)}$ and the closest point in $S^{(k)}$ to x^*, respectively, with respect to the norm $\| \cdot \|_Q$. Then we have*

(a) $\| x^{(k)} - x^0 \|_Q \to \| x^* - x^0 \|_Q \ (k \to \infty)$;
(b) $\| \hat{x}^{(k)} - x^{(k)} \|_Q \to 0 \ (k \to \infty)$;
(c) $\hat{x}^{(k)} \to x^* \ (k \to \infty)$.

Proof. First note that

$$\| x^* - x^0 \|_Q \leq \| \hat{x}^{(k)} - x^0 \|_Q \leq \| \hat{x}^{(k)} - x^{(k)} \|_Q + \| x^{(k)} - x^0 \|_Q . \tag{38}$$

By Lipschitz continuity of the perturbed polyhedral convex sets [8, Theorem 4.6], there exists a constant α such that

$$\| \hat{x}^{(k)} - x^{(k)} \|_Q \leq \alpha \left(\| (\delta^{(k)} - \delta)^+ \| + \| (\gamma^{(k)} - \gamma)^+ \| \right), \tag{39}$$

where for any vector d, d^+ denotes the vector with components $d_i^+ = \max(0, d_i)$, and $\delta^{(k)}$ and $\gamma^{(k)}$ are vectors defined by (35) and (36), respectively. Then by Lemma 4.4 and (38), we have

$$\| x^* - x^0 \|_Q \leq \liminf_{k \to \infty} \| x^{(k)} - x^0 \|_Q . \tag{40}$$

On the other hand, it follows from Lemma 4.4 (d) that

$$\| x^{(k)} - x^0 \|_Q \leq \| \tilde{x}^{(k)} - x^0 \|_Q \leq \| \tilde{x}^{(k)} - x^* \|_Q + \| x^* - x^0 \|_Q . \tag{41}$$

Again by using Lipschitz continuity of the perturbed polyhedral convex sets, we have

$$\| \tilde{x}^{(k)} - x^* \|_Q \leq \alpha \left(\| (\delta^{(k)} - \delta)^+ \| + \| (\gamma^{(k)} - \gamma)^+ \| \right). \tag{42}$$

Thus by Lemma 4.4 and (41), we have

$$\limsup_{k\to\infty} \| x^{(k)} - x^0 \|_Q \leq \| x^* - x^0 \|_Q . \tag{43}$$

From (40) and (43), we obtain (a). Then (b) follows from (39) and Lemma 4.4 (b). Finally, to prove (c), observe that (a), (b) and (38) imply

$$\| \hat{x}^{(k)} - x^0 \|_Q \to \| x^* - x^0 \|_Q \quad (k \to \infty). \tag{44}$$

Since $\hat{x}^{(k)}$ are contained in S and since x^* is the unique solution of $\hat{\mathrm{P}}$, (44) ensures (c). □

Theorem 4.2 *Let x^* be the solution of problem $\hat{P}$ and $\{x^{(k)}\}$ be a sequence generated by Algorithm 4.1. Then*

$$x^{(k)} \to x^* \quad (k \to \infty).$$

Proof. Since

$$\| x^* - x^{(k)} \|_Q \leq \| x^* - \hat{x}^{(k)} \|_Q + \| x^{(k)} - \hat{x}^{(k)} \|_Q,$$

the theorem follows from (b) and (c) in Lemma 4.5. □

Combining Theorems 4.1 and 4.2, we obtain the following convergence theorem.

Theorem 4.3 *Algorithm 3.1 generates a sequence $\{x^{(k)}\}$ that converges to the solution x^* of problem $\hat{P}$.*

4.2 Rate of convergence

Now we turn our attention to the rate of convergence of Algorithm 4.1. Our analysis is a generalization of Iusem and De Pierro [6]. It is noted that Iusem and De Pierro [6] consider the case where the matrix Q is the identity matrix. Nevertheless we shall often make use of the results of [6], because they mostly remain valid for an arbitrary positive definite matrix Q by appropriate modifications.

First we make some definitions. Let

$$U = \left\{ (z^+, z^-) \,\middle|\, z^+ \geq 0, z^- \geq 0, x^* = x^0 - Q^{-1}A^T(z^+ - z^-) \right\}.$$

Let

$$I^+ = \{ i \mid \langle a_i, x^* \rangle = \delta_i \},\ I^- = \{ i \mid \langle a_i, x^* \rangle = \gamma_i \}$$

and

$$I^{+(k)} = \{ i \mid z_i^{+(k+C)} \neq 0 \},\ I^{-(k)} = \{ i \mid z_i^{-(k+C)} \neq 0 \},\ I^{(k)} = I^{+(k)} \cup I^{-(k)},$$

where C is an almost cyclicality constant (see Section 2). From Lemma 4.1, it follows that

$$I^{+(k)} \cap I^{-(k)} = \emptyset. \tag{45}$$

For a subset J of $I = \{1, \ldots, m\}$, let A_J denote the submatrix of A with rows $a_j, j \in J$, and for any vector $d \in R^m$, let d_J denote the subvector of d with components $d_j, j \in J$. Define

$$E^{+(k)} = \{ x \mid A_{I^{+(k)}} x = \delta_{I^{+(k)}} \},\ E^{-(k)} = \{ x \mid A_{I^{-(k)}} x = \gamma_{I^{-(k)}} \},\ E^{(k)} = E^{+(k)} \cap E^{-(k)}$$

and let

$$E = \{ x \in R^n \mid A_{I^+} x = \delta_{I^+},\ A_{I^-} x = \gamma_{I^-} \}.$$

Lemma 4.6 *For all k sufficiently large, we have*

$$\text{(a) } z_i^{+(k)} = 0, \forall i \notin I^+ \text{ and } z_i^{-(k)} = 0, \forall i \notin I^-;$$

(b) *there exists* $(z^+, z^-) \in U$ *such that* $z_i^+ = 0, \forall i \notin I^{+(k)}$ *and* $z_i^- = 0, \forall i \notin I^{-(k)}$;

$$\text{(c) } \| x^{(k+1)} - x^* \|_Q^2 \leq \| x^{(k)} - x^* \|_Q^2 - \left(\frac{2}{\omega} - 1\right) \| x^{(k+1)} - x^{(k)} \|_Q^2 .$$

Proof. Parts (a) and (b) can be proved in a similar manner to [6], and hence omitted here. We shall prove (c). Since $x^{(k+1)} - x^{(k)} = (c^{+(k)} - c^{-(k)})Q^{-1}a_{i_k}$ by (7), and $c^{(k)} = c^{+(k)} - c^{-(k)}$ by (17), we may write

$$\| x^{(k+1)} - x^* \|_Q^2 = \| x^{(k)} - x^* \|_Q^2 + \left(c^{(k)}\right)^2 \alpha_{i_k} + 2c^{(k)} \langle a_{i_k}, x^{(k)} - x^* \rangle,$$

where $\alpha_{i_k} = \langle a_{i_k}, Q^{-1} a_{i_k} \rangle$. Therefore (c) is equivalent to

$$c^{(k)} \left\{ c^{(k)} + \omega \frac{\langle a_{i_k}, x^{(k)} - x^* \rangle}{\alpha_{i_k}} \right\} \leq 0. \tag{46}$$

The following three cases are possible:
(i) $\gamma_{i_k} < \langle a_{i_k}, x^* \rangle < \delta_{i_k}$: Suppose that k is sufficiently large. Then by (a) we have $z_{i_k}^{+(k)} = 0$ and $z_{i_k}^{-(k)} = 0$. Moreover, since $x^{(k)} \to x^*$, $\Delta^{(k)} > 0$ and $\Gamma^{(k)} < 0$. (Recall that we have assumed $\gamma < \delta$.) Consequently (14) and (15) imply $c^{+(k)} = c^{-(k)} = 0$ and hence $c^{(k)} = 0$. Thus (46) holds.
(ii) $\langle a_{i_k}, x^* \rangle = \gamma_{i_k}$: In this case, (c) is equivalent to

$$c^{(k)} \left(c^{(k)} - \omega \Gamma^{(k)} \right) \leq 0 \tag{47}$$

by the definition of $\Gamma^{(k)}$. When k is sufficiently large, we have $z_{i_k}^{+(k)} = 0$, $z_{i_k}^{-(k)} \geq 0$, $\Delta^{(k)} > 0$, so that (14) and (15) imply $c^{(k)} \neq \omega \Delta^{(k)}$. Thus we have either $c^{(k)} = z_{i_k}^{+(k)} - z_{i_k}^{-(k)}$ or $c^{(k)} = \omega \Gamma^{(k)}$. In the former case, since $c^{(k)} = -z_{i_k}^{-(k)} \leq 0$ and since $c^{(k)} \geq \omega \Gamma^{(k)}$ by (5), we obtain (47). In the latter case, (47) is obvious.
(iii) $\delta_{i_k} = \langle a_{i_k}, x^* \rangle$: Proof is similar to the case (ii). □

Note that Lemma 4.6 (c) implies

$$\| x^{(k+C)} - x^* \|_Q^2 \leq \| x^{(k)} - x^* \|_Q^2 - \left(\frac{2}{\omega} - 1\right) \sum_{l=k}^{k+C-1} \| x^{(l+1)} - x^{(l)} \|_Q^2 . \tag{48}$$

Lemma 4.7 *For all k large enough, we have*

$$\| x^{(k+C)} - x^* \|_Q = \min_{x \in E^{(k)}} \| x - x^{(k+C)} \|_Q \tag{49}$$

and

$$x^{(k+C)} \notin E^{(k)}. \tag{50}$$

Proof. To prove (49), it is sufficient to show that x^* together with some vectors $z_{I^{+(k)}}^+, z_{I^{-(k)}}^-$ satisfies the following system of equations:

$$A_{I^{+(k)}} x = \delta_{I^{+(k)}}, \tag{51}$$

$$A_{I^{-(k)}}x = \gamma_{I^{-(k)}}, \tag{52}$$

$$x = x^{(k+C)} - Q^{-1}A^T_{I^{+(k)}}z^+_{I^{+(k)}} + Q^{-1}A^T_{I^{-(k)}}z^-_{I^{-(k)}}. \tag{53}$$

By Lemma 4.6 (a), we see that $I^{(k)} \subset I$ for sufficiently large k, namely, x^* satisfies (51) and (52). Moreover, by Lemma 4.6 (b) and the fact that x^* and $x^{(k+C)}$ satisfy (10), there exists $(\bar{z}^+, \bar{z}^-) \in U$ such that

$$\begin{aligned} x^* &= x^0 - Q^{-1}A^T(\bar{z}^+ - \bar{z}^-) \\ &= x^{(k+C)} - Q^{-1}A^T\left\{(\bar{z}^+ - z^{+(k+C)}) - (\bar{z}^- - z^{-(k+C)})\right\} \end{aligned}$$

and $\bar{z}^+_i = 0, \forall i \notin I^{+(k)}$, and $\bar{z}^-_i = 0, \forall i \notin I^{-(k)}$. But since $z_i^{+(k+C)} = 0, \forall i \notin I^{+(k)}$, and $z_i^{-(k+C)} = 0, \forall i \notin I^{-(k)}$, we have

$$x^* = x^{(k+C)} - Q^{-1}A^T_{I^{+(k)}}\left(\bar{z}^+_{I^{+(k)}} - z^{+(k+C)}_{I^{+(k)}}\right) + Q^{-1}A^T_{I^{-(k)}}\left(\bar{z}^-_{I^{-(k)}} - z^{-(k+C)}_{I^{-(k)}}\right).$$

Therefore, by putting

$$z^+_{I^{+(k)}} = \bar{z}^+_{I^{+(k)}} - z^{+(k+C)}_{I^{+(k)}} \text{ and } z^-_{I^{-(k)}} = \bar{z}^-_{I^{-(k)}} - z^{-(k+C)}_{I^{-(k)}},$$

we get (53). Then we have (50) by similar reasoning to Corollary 2 in [6]. □

The following theorem establishes that the rate of convergence of Algorithm 3.1 is linear.

Theorem 4.4 *Let $\{x^{(k)}\}$ be a sequence generated by Algorithm 4.1, or equivalently by Algorithm 3.1. Then there exists a constant $\rho \in (0,1)$ such that for all k large enough*

$$\| x^{(k+C)} - x^* \|_Q \leq \rho \| x^{(k)} - x^* \|_Q,$$

where C is an almost cyclicality constant associated with the algorithm.

Proof. The main idea of the proof is similar to that of Theorem 1 in [6]. Since complete proof of the theorem is somewhat lengthy, we shall give an outline of the proof with emphasis on the differences between the the arguments of [6] and ours.

For an arbitrary k, consider the system of equations

$$\langle a_i, x\rangle = \delta_i, \ \ i \in I^{+(k)},$$

$$\langle a_i, x\rangle = \gamma_i, \ \ i \in I^{-(k)}$$

and let $j \in I^{+(k)} \cup I^{-(k)}$ be the index that corresponds to the equation such that the hyperplane in R^n defined by that equation lies farthest away from point $x^{(k+C)}$ among all those hyperplanes. Note that by (50), $x^{(k+C)}$ is not contained in the hyperplane j. Then using a similar argument to that in [6, p. 43], we see that there exists a positive constant μ independent of k such that, if $j \in I^{+(k)}$, then

$$\| x^{(k+C)} - x^* \|_Q \leq \frac{1}{\mu} \cdot \frac{|\delta_j - \langle a_j, x^{(k+C)}\rangle|}{\| Q^{-1}a_j \|_Q}, \tag{54}$$

while if $j \in I^{-(k)}$, then

$$\| x^{(k+C)} - x^* \| \leq \frac{1}{\mu} \cdot \frac{|\gamma_j - \langle a_j, x^{(k+C)}\rangle|}{\| Q^{-1}a_j \|_Q}. \tag{55}$$

Now consider the case where $j \in I^{+(k)}$. Since Lemma 4.6 (a) implies $I^{+(k)} \subseteq I^{+}$ for all k large enough, it follows from Lemma 4.6 (b) that $z_j^{-(r)} = 0$. Moreover we have $z_j^{+(r)} \neq c^{+(r)}$, where $r = \max\{l < k+C \mid i_l = j\}$, because if $z_j^{+(r)} = c^{+(r)}$ then (8) would imply $z_j^{+(r+1)} = 0$, which together with the definition of r implies $0 = z_j^{+(r+1)} = z_j^{+(r+2)} = \ldots = z_j^{+(k+C)}$, contradicting the assumption $j \in I^{+(k)}$. Therefore, by (14), we obtain

$$(c^{+(r)}, c^{-(r)}) = (\omega\Delta^{(r)}, 0),$$

i.e.,

$$c^{(r)} = \omega\Delta^{(r)}. \tag{56}$$

In the case where $j \in I^{-(k)}$, we may use a similar argument to obtain

$$c^{(r)} = \omega\Gamma^{(r)}. \tag{57}$$

The relations (54) - (57) are key to proving the theorem and the remaining part of the proof can be done in a way similar to that of Theorem 1 in [6]. □

5 Numerical results

In this section, we present numerical results for the proposed algorithm. The purposes of the numerical experiments are as follows: First we verify that the proposed algorithm is linearly convergent for randomly generated test problems of the form $\hat{\mathrm{P}}$. We also examine how the choice of relaxation parameter ω affects the performance of the algorithm. Finally we demonstrate that the algorithm is well suited to parallel computation for some problems with special structure. The first two issues are considered in Subsection 5.1, while the last is the topic of Subsection 5.2. In the experiments reported below, we adopted the cyclic rule

$$i_k := (k \bmod m) + 1, \quad k = 0, 1, 2, \ldots$$

to choose index i_k at each iteration.

5.1 Behavior of the algorithm

In order to verify the results stated in Theorems 4.3 and 4.4, we have experimented on problems of the form $\hat{\mathrm{P}}$, in which Q is the identity matrix and A is a dense matrix with random elements. Specifically, the elements of A and δ were chosen from the intervals $[-10, 10]$ and $[1, 10]$, respectively, and γ were set equal to $-\delta$. All elements of the constant vector x^0 in the objective function were set to be 10.

Closeness of an iterate $x^{(k)}$ to the solution is measured in terms of the residual of the constraint inequalities of $\hat{\mathrm{P}}$, which is defined by

$$r^{(k)} = \max{}_i \left\{ (\langle a_i, x^{(k)} \rangle - \delta_i)_+, (\gamma_i - \langle a_i, x^{(k)} \rangle)_+ \right\},$$

where $(\cdot)_+ = \max\{\cdot, 0\}$. Because of the primal-dual nature of the algorithm, $x^{(k)}$ is optimal whenever it is feasible to $\hat{\mathrm{P}}$, in which case $r^{(k)}$ vanishes.

Figure 1 illustrates how $\{r^{(k)}\}$ decreases as the iteration proceeds for a test problem with $(n, m) = (75, 50)$. From the figure, we may observe that the algorithm is linearly convergent for any value of relaxation parameter ω.

In order to see how the choice of parameter ω affects the performance of the algorithm, we then solved several test problems of various sizes. The results are summarized in Table

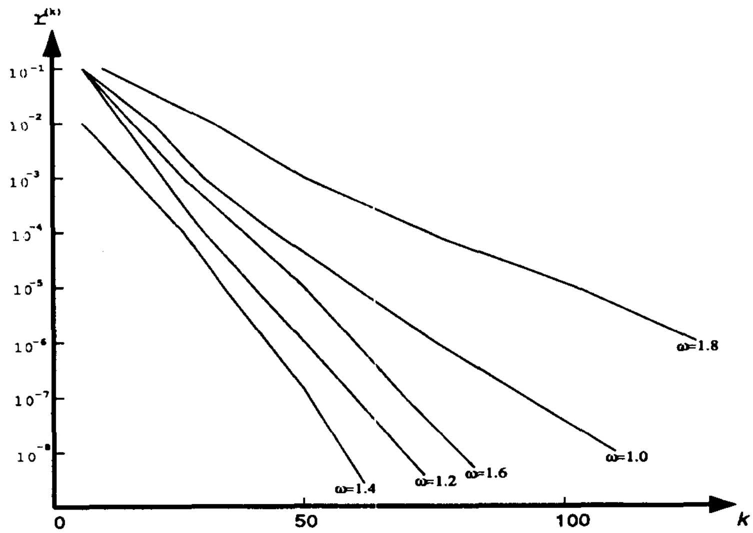

Fig. 1. Behavior of the proposed algorithm

1, in which the numbers indicate how many iterations were required to attain the termination criterion

$$r^{(k)} \leq 10^{-4}.$$

From Table 1, we see that the algorithm converges faster when ω is chosen around $1.2 \sim 1.4$. It is noted that, when $\omega = 1.0$, the algorithm is nothing but the extended Hildreth's algorithm for solving interval constrained problems proposed by Lent and Censor [8].

5.2 Parallel computation for transportation problems

Hildreth type algorithms are most effectively applied to problems with sparse constraint matrix and the network flow problem is a typical example of such problems. Moreover, transportation problems are well suited to parallel implementation of Hildreth type algorithms, because of the bipartite network structure [12].

In the numerical experiments, we have applied Algorithm 3.1 to the following transportation problem with quadratic cost function:

$$\begin{aligned} \min \ & \sum_{i=1}^{M}\sum_{j=1}^{N}\left(\frac{1}{2}w_{ij}x_{ij}^2 + c_{ij}x_{ij}\right) \\ \text{s.t.} \ & \sum_{j=1}^{N} x_{ij} = s_i, \ \ i = 1, \dots, M, \\ & \sum_{i=1}^{M} x_{ij} = d_j, \ \ j = 1, \dots, N, \\ & 0 \leq x_{ij} \leq u_{ij}, \ \ i = 1, \dots, M, \ j = 1, \dots, N. \end{aligned}$$

Note that this class of problems has also been considered by Zenios and Censor [12], who implemented a Hildreth type algorithm on a massively parallel Connection Machine CM-2. In our experiments, we have implemented Algorithm 3.1 on a parallel computer called

Table 1: Computational results for problem $\hat{P}$.

Problem	size		# iterations[1]				
	(n, m)	$\omega =$	1.0	1.2	1.4	1.6	1.8
1	(75, 50)		46	30	28	38	72
2	(100, 50)		27	18	16	25	51
3	(200, 50)		10	9	13	21	44
4	(150, 100)		80	53	30	38	76
5	(200, 100)		27	17	18	27	59
6	(300, 100)		15	10	13	22	46
7	(250, 150)		47	34	28	38	75
8	(300, 150)		25	17	20	28	56
9	(450, 150)		16	11	14	24	48

[1] Termination criterion: $r^{(k)} \leq 10^{-4}$

Table 2: Computational results for transportation problems.

(M, N)	# iterations[1]	CPU sec serial[2]	CPU sec parallel[3]
(100, 100)	37	12.75	0.133
(200, 200)	33	47.14	0.201
(300, 300)	35	114.48	0.698
(400, 400)	34	208.38	0.854
(500, 500)	34	328.44	1.054
(600, 600)	35	497.91	1.894
(700, 700)	34	653.88	2.219

[1] Termination criterion: $\max\left\{\max_i\left\{\left|\sum_j x_{ij}^{(k)} - s_i\right|\right\}, \max_j\left\{\left|\sum_i x_{ij}^{(k)} - d_j\right|\right\}\right\} \leq 10^{-4}$

[2] Serial computation implemented on SUN SPARC1 IPX

[3] Parallel computation implemented on ADENA

ADENA [7] at the Department of Applied Mathematics and Physics, Kyoto University. ADENA was originally designed to solve PDE's and consists of 16×16 PE array, i.e., 256 PEs. Its performance is estimated as 2.56 GFLOPS (peak) or 1.0 GFLOPS (effective). We also implemented the algorithm (serially) on a SUN SPARK1 IPX. By some preliminary experiments, we have found that SUN SPARK IPX performs floating point arithmetic operations about 2 times faster than a single processor element of ADENA.

The results of both implementations are summarized in Table 2, which shows the numbers of iterations and CPU times (sec.) for problems of various sizes with termination criterion

$$\max\left\{\max_i\left\{\left|\sum_j x_{ij}^{(k)} - s_i\right|\right\}, \max_j\left\{\left|\sum_i x_{ij}^{(k)} - d_j\right|\right\}\right\} \leq 10^{-4}.$$

The results strongly suggest that the proposed algorithm is amenable to parallel computation for this type of problems.

6 Conclusion

In this paper, we have proposed an SOR modification of Hildreth's algorithm for solving interval constrained quadratic programming problems. The proposed algorithm is shown to converge at a linear rate. Our limited computational experience demonstrates the effectiveness of the proposed algorithm.

Acknowledgement We are indebted to the referees and Mr. Eiki Yamakawa of Advanced Telecommunications Research Institute International for careful reading of the manuscript. In particular, we are grateful to one of the referees who kindly pointed out that the rate of convergence results of Section 4.3, which were only proved for the special case $Q = I$ in an earlier version of the paper, could be extended to the general case as shown in the finial version.

References

[1] Bertsekas, D.P. and Tsitsiklis, J.N.: *Parallel and Distributed Computation: Numerical Methods.* Prentice Hall, New Jersey, 1989.
[2] Censor, Y.: Row-Action Methods for Huge and Sparse Systems and Their Applications. *SIAM Review*, Vol. 23 (1981), 444-466.
[3] Censor, Y. and Lent, A.: An Iterative Row-Action Method for Interval Convex Programming. *Journal of Optimization Theory and Applications*, Vol. 34 (1981), 321-353.
[4] Herman, G.T. and Lent, A.: A Family of Iterative Quadratic Optimization Algorithms for Pairs of Inequalities, with Application in Diagnostic Radiology. *Mathematical Programming Study*, Vol. 9 (1979), 15-29.
[5] Hildreth, C.: A Quadratic Programming Procedure. *Naval Research Logistics Quarterly*, Vol. 4 (1957), 79-85.
[6] Iusem, A.N. and De Pierro, A.R.: On the Convergence Properties of Hildreth's Quadratic Programming Algorithm. *Mathematical Programming*, Vol. 47 (1990), 37-51.
[7] Kadota, H., Kaneko, K., Tanikawa, Y. and Nogi, T.: VLSI Parallel Computer with Data Transfer Network: ADENA. *Proceedings of the 1989 International Conference on Parallel Processing*, The Pennsylvania State University Press, Vol. 1 (1989), 319-322.
[8] Lent, A. and Censor, Y.: Extensions of Hildreth's Row-Action Method for Quadratic Programming," *SIAM Journal on Control and Optimization*, Vol. 18 (1980), 444-454.
[9] Lin, Y.Y. and Pang, J.S.: Iterative Methods for Large Convex Quadratic Programs: A Survey. *SIAM Journal on Control and Optimization*, Vol. 25 (1987), 383-411.
[10] Nielsen, S.S. and Zenios, S.A.: Massively Parallel Algorithms for Singly Constrained Nonlinear Programs. *ORSA Journal on Computing* Vol. 4 (1992), 166-181.
[11] Zenios, S.A. and Lasken, R.A.: Nonlinear Network Optimization on a Massively Parallel Connection Machine. *Annals of Operations Research*, Vol. 14 (1988), 147-165.
[12] Zenios, S.A. and Censor, Y.: Massively Parallel Row-Action Algorithms for Some Nonlinear Transportation Problems. *SIAM Journal on Optimization*, Vol. 1 (1991), 373-400.

Masao FUKUSHIMA:
Graduate School of Information Science,
Advanced Institute of Science and Technology, Nara,
Takayama, Ikoma, Nara 630-01, Japan