MATHEMATICS OF OPERATIONS RESEARCH
Vol. 19, No. 3, August 1994
Printed in U.S.A.

PARALLEL CONSTRAINT DISTRIBUTION IN CONVEX QUADRATIC PROGRAMMING

MICHAEL C. FERRIS

We consider convex quadratic programs with large numbers of constraints. We distribute these constraints among several parallel processors and modify the objective function for each of these subproblems with Lagrange multiplier information from the other processors. New Lagrange multiplier information is aggregated in a master processor and the whole process is repeated. Linear convergence is established for strongly convex quadratic programs by formulating the algorithm in an appropriate dual space. The algorithm corresponds to a step of an iterative matrix splitting algorithm for a symmetric linear complementarity problem followed by a projection onto a subspace.

1. Introduction. We are concerned with the problem

$$\begin{aligned}&\text{minimize} \quad c^T x + \tfrac{1}{2} x^T Q x\\ &\text{subject to} \quad A_1 x \leqslant a_1, \ldots, A_p x \leqslant a_p,\end{aligned}$$

where $c \in \mathbb{R}^n$, $Q \in \mathbb{R}^{n \times n}$, $A_i \in \mathbb{R}^{m_i \times n}$, $a_i \in \mathbb{R}^{m_i}$ and Q is symmetric and positive definite. Our principal aim is to distribute the p constraint blocks among p parallel processors together with an appropriately modified objective function. We then solve each of these p subproblems independently, aggregate Lagrange multiplier information from the processors and repeat. The method we describe here is closely related to the one given in Ferris and Mangasarian (1991). References to other constraint distribution algorithms can be found in that paper. The key to our approach lies in the precise form of the modified objective function to be optimized by each processor. The modified objectives are made up of the original objective function plus augmented Lagrangian terms involving the constraints handled by the other processors.

In this paper, we show that under the assumption of a strongly convex quadratic objective and linear independence of each of the distributed constraint blocks, the parallel constraint distribution (PCD) algorithm converges linearly from any starting point for a feasible problem. The key to the convergence proof is to show that in the dual space, an iteration of the proposed parallel constraint distribution algorithm is equivalent to a step of an iterative matrix splitting method for a symmetric linear complementarity problem followed by a subspace projection. Another point of view for splitting constraints is presented by Bertsekas and Tseng (1991). Their work appeared after the original version of this paper.

A word about our notation now. For a vector x in the n-dimensional real space $\mathbb{R}^n$, x_+ will denote the vector in $\mathbb{R}^n$ with components $(x_+)_i := \max\{x_i, 0\}$, $i = 1, \ldots, n$. The standard inner product of $\mathbb{R}^n$ will be denoted either by $\langle x, y \rangle$ or $x^T y$. The Euclidean or 2-norm $(x^T x)^{1/2}$ will be denoted by $\| \cdot \|$. For an $m \times n$ real matrix A,

Received May 6, 1991; revised December 21, 1992.
AMS 1991 subject classification. Primary: 90C20.
OR/MS Index 1978 subject classification. Primary: 662 Programming/Quadratic.
Key words. Parallel optimization, augmented Lagrangians, quadratic programs, matrix splitting, linear convergence.

0364-765X/94/1903/0645/$01.25

signified by $A \in \mathbb{R}^{m \times n}$, A^T will denote the transpose. The identity matrix of any order will be given by I. The nonnegative orthant in $\mathbb{R}^n$ will be denoted by $\mathbb{R}^n_+$. The notation $\pi_C(x)$ will be used to define the projection of the point x onto the closed convex set C.

2. Parallel constraint distribution for quadratic programs. For simplicity we consider a quadratic program with three blocks of inequality constraints. Routine extension to p blocks can be achieved by appropriate extension and permutation of subscripts. Equality constraints can also be incorporated in a straightforward manner. Consider then the problem

$$\text{(1)} \qquad \begin{array}{ll} \text{minimize} & c^T x + \tfrac{1}{2} x^T Q x \\ \text{subject to} & A_l x \leqslant a_l, \qquad l = 1,2,3, \end{array}$$

where $c \in \mathbb{R}^n$, $Q \in \mathbb{R}^{n \times n}$, $A_l \in \mathbb{R}^{m_l \times n}$, $a_l \in \mathbb{R}^{m_l}$ and Q is symmetric and positive definite. We now describe the algorithm in detail. At iteration i we distribute the constraints of (1) among three parallel processors ($l = 1,2,3$) as follows:

$$\text{(2)} \qquad \begin{array}{ll} \underset{x_l}{\text{minimize}} & c^T x_l + \dfrac{1}{2} x_l^T Q x_l + \dfrac{1}{2\gamma}\left[\displaystyle\sum_{\substack{k=1 \\ k \neq l}}^{3} \left\| \left(\gamma(A_k x_l - a_k) + t_{kl}^i \right)_+ \right\|^2 \right] \\ \text{subject to} & A_l x_l \leqslant a_l, \end{array}$$

where γ is a positive number and t_{kl}^i are estimates of the Lagrange multipliers from the previous iteration. We note that the subproblems (2) of the algorithm split the constraints of the original quadratic program (1) between them in the form of split explicit constraints as well as augmented Lagrangian terms involving the other constraints. The principal objective that has been achieved is that the *explicit* constraints of each of the subproblems are a subset of the constraints of the original problem. Of course the above approach is applicable for general (not just quadratic) problems.

Although the subproblems (2) of the PCD method have a more complicated objective function than the original problem (1) they have considerably fewer constraints. In the computational work of §3, the subproblems were solved using MINOS with standard default settings without any difficulty. In fact, after the first parallel iteration, MINOS only carried out a few pivots to reach optimality in the newly generated subproblems. Although the objective function (for inequality constraints) is not twice differentiable, the BFGS approximation for the Hessian was very effective for the problems we tested. In all likelihood, this is due to the fact that the gradient is always Lipschitzian. The principal computational advantages of the PCD approach are twofold. Firstly, the manner of splitting the constraints in the algorithm is completely arbitrary. Thus, any suitable structure in the problem can easily be exploited. For example, in multi-commodity network flow problems, the linking constraints could be treated separately, as well as each of the commodities. This would result in a large number of problems with network constraints and a single problem treating the linking constraints. Certainly, the network constraints could be exploited by a state-of-the-art nonlinear network code (for example, Dembo 1986) and the coupling constraints are very simple and thus would also be easy to exploit. Secondly, for extremely large problems, large numbers of processors are envisaged for solving the subproblems in parallel. For example, using 100 processors for problems with one hundred thousand constraints would reduce the number of constraints treated by each processor to 1000. The number of constraints in the subproblem is vastly reduced by this technique, so these very large scale problems become solvable since the amount of memory required to handle the factorizations of the constraint matrices becomes manageable on the separate processors. Furthermore, the computations in §3 suggest that the number of parallel iterations required for the solution of the subproblems is largely independent of the number of processors.

The difference between this algorithm and standard augmented Lagrangian methods (see Bertsekas 1982, Rockafellar 1974, 1976) is that the multiplier update is carried out explicitly rather than with the traditional gradient updating scheme. Each subproblem is then solved and a point $(\bar{x}_l^{i+1}, \bar{s}_l^{i+1}) \in \mathbb{R}^{n+m_l}$, $l = 1,2,3$, which satisfies the Karush-Kuhn-Tucker conditions (Mangasarian 1969) for subproblem (2) is obtained. We define

$$\bar{t}_{jl}^{i+1} = \left(\gamma\left(A_j\bar{x}_l^{i+1} - a_j\right) + t_{jl}^i\right)_+ \qquad j \neq l, \tag{3}$$

and let

$$s_l^{i+1} = \frac{1}{3}\left[\bar{s}_l^{i+1} + \sum_{\substack{k=1\\k\neq l}}^{3} \bar{t}_{kl}^{i+1}\right] \tag{4}$$

and

$$t_{jl}^{i+1} = s_j^{i+1}, \; j \neq l. \tag{5}$$

This completes one iteration of the PCD algorithm. Clearly, steps (2), (3) and (5) should be executed in parallel, while step (4) should be executed on a master processor. For completeness, we give the algorithm below:

PCD *Algorithm*.
Initialization: Start with any s_l^0, $l = 1,2,3$.
Parallel iteration: In parallel ($l = 1,2,3$) perform the following steps.
Having s_l^i compute:
(1) t_{jl}^i using (5);
(2) $(\bar{x}_l^{i+1}, \bar{s}_l^{i+1})$ using (2);
(3) $\bar{t}_{jl}^{i+1}$ using (3).
Synchronization: Evaluate s_l^{i+1}, $l = 1,2,3$ using (4). If converged, then stop, else return to parallel iteration.

In the sequel we will show that s_l^i, $l = 1,2,3$ converge linearly to a set of optimal multipliers for the dual of (1). Furthermore, we show that the solutions of the subproblems $\bar{x}_l^i$ each converge to the optimal solution of (1). In order to facilitate this we will rewrite the iterates of the algorithm exclusively in the dual space. We shall use the following easily established result frequently throughout the paper, so we state it as a lemma.

LEMMA 2.1. *Let b, $d \in \mathbb{R}^n$. Then*

$$b = d_+ \quad \Leftrightarrow \quad b - d \geqslant 0, \quad b^T(b-d) = 0, \quad b \geqslant 0.$$

It follows from Lemma 2.1 that $(\bar{x}_l^{i+1}, \bar{s}_l^{i+1})$ satisfy the following Karush-Kuhn-Tucker conditions:

$$(6)\qquad \begin{aligned} &c + Q\bar{x}_l^{i+1} + \sum_{\substack{k=1\\k\neq l}}^{3} A_k^T\big(\gamma\big(A_k\bar{x}_l^{i+1} - a_k\big) + t_{kl}^i\big)_+ + A_l^T\bar{s}_l^{i+1} = 0,\\ &\bar{s}_l^{i+1} = \big(\gamma\big(A_l\bar{x}_l^{i+1} - a_l\big) + \bar{s}_l^{i+1}\big)_+, \qquad l = 1,2,3, \end{aligned}$$

or equivalently:

$$(7)\qquad \begin{aligned} &\bar{x}_l^{i+1} = -Q^{-1}\left(\sum_{\substack{k=1\\k\neq l}}^{3} A_k^T\bar{t}_{kl}^{i+1} + A_l^T\bar{s}_l^{i+1} + c\right),\\ &\bar{s}_l^{i+1} = \big(\gamma\big(A_l\bar{x}_l^{i+1} - a_l\big) + \bar{s}_l^{i+1}\big)_+,\\ &\bar{t}_{jl}^{i+1} = \big(\gamma\big(A_j\bar{x}_l^{i+1} - a_j\big) + t_{jl}^i\big)_+, \qquad l = 1,2,3,\ j = 1,2,3,\ j \neq l. \end{aligned}$$

We use the first equation of (7) to eliminate $\bar{x}_l^{i+1}$ from the second and third equations of (7). This leads to the following system:

$$(8)\qquad \begin{aligned} &\bar{s}_l^{i+1} = \left(\bar{s}_l^{i+1} - \gamma\left[A_lQ^{-1}A_l^T\bar{s}_l^{i+1} + A_lQ^{-1}\sum_{\substack{k=1\\k\neq l}}^{3} A_k^T\bar{t}_{kl}^{i+1} + A_lQ^{-1}c + a_l\right]\right)_+,\\ &\bar{t}_{jl}^{i+1} = \left(-\gamma\left[A_jQ^{-1}A_l^T\bar{s}_l^{i+1} + A_jQ^{-1}\sum_{\substack{k=1\\k\neq l}}^{3} A_k^T\bar{t}_{kl}^{i+1} + A_jQ^{-1}c + a_j\right] + t_{jl}^i\right)_+, \end{aligned}$$

for $l = 1,2,3$ and $j = 1,2,3$ with $j \neq l$. Let us define new variables in blocks as follows

$$(9)\qquad \begin{aligned} u_1^i &= \big(s_1^i, t_{21}^i, t_{31}^i\big),\\ u_2^i &= \big(t_{12}^i, s_2^i, t_{32}^i\big),\\ u_3^i &= \big(t_{13}^i, t_{23}^i, s_3^i\big) \end{aligned}$$

(with a similar notation for the barred variables). We can then rewrite (8) as follows:

$$\bar{u}_l^{i+1} = \big(\bar{u}_l^{i+1} - \big[(H + J_l)\bar{u}_l^{i+1} - J_lu_l^i + h\big]\big)_+$$

for $l = 1,2,3$. Here

$$(10)\quad H = \gamma\begin{bmatrix} A_1Q^{-1}A_1^T & A_1Q^{-1}A_2^T & A_1Q^{-1}A_3^T\\ A_2Q^{-1}A_1^T & A_2Q^{-1}A_2^T & A_2Q^{-1}A_3^T\\ A_3Q^{-1}A_1^T & A_3Q^{-1}A_2^T & A_3Q^{-1}A_3^T \end{bmatrix}, \qquad h = \gamma\begin{bmatrix} A_1Q^{-1}c + a_1\\ A_2Q^{-1}c + a_2\\ A_3Q^{-1}c + a_3 \end{bmatrix},$$

and J_l is defined by

$$J_1 := \begin{bmatrix} 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}, \qquad J_2 := \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I \end{bmatrix}, \qquad J_3 := \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

where the blocks are partitioned as in the definition of H. If we let

$$(11) \qquad \begin{aligned} \bar{z}^{i+1} &= \left(\bar{u}_1^{i+1}, \bar{u}_2^{i+1}, \bar{u}_3^{i+1}\right), \\ z^i &= \left(u_1^i, u_2^i, u_3^i\right), \end{aligned}$$

and invoke Lemma 2.1, the following symmetric linear complementarity problem in the variable $\bar{z}^{i+1}$ ensues:

$$(12) \qquad B\bar{z}^{i+1} + Cz^i + q \geqslant 0, \qquad \left\langle \bar{z}^{i+1}, B\bar{z}^{i+1} + Cz^i + q \right\rangle = 0, \qquad \bar{z}^{i+1} \geqslant 0.$$

Here

$$(13) \qquad B = \begin{bmatrix} H + J_1 & 0 & 0 \\ 0 & H + J_2 & 0 \\ 0 & 0 & H + J_3 \end{bmatrix},$$

$$(14) \qquad C = \begin{bmatrix} -J_1 & 0 & 0 \\ 0 & -J_2 & 0 \\ 0 & 0 & -J_3 \end{bmatrix},$$

and q is given by

$$(15) \qquad q = \begin{bmatrix} h \\ h \\ h \end{bmatrix}.$$

Consider the linear complementarity problem LCP(M, q),

$$(16) \qquad Mz + q \geqslant 0, \qquad \langle z, Mz + q \rangle = 0, \qquad z \geqslant 0,$$

where M is defined by

$$(17) \qquad M := \begin{bmatrix} H & 0 & 0 \\ 0 & H & 0 \\ 0 & 0 & H \end{bmatrix}.$$

and q by (15). Note that the matrices B and C constitute a splitting of M and hence (12) corresponds to a matrix-splitting algorithm for (16). We observe that due to the structure of B and C, (12) can be performed on three independent processors (which corresponds to the three subproblems in (2)). However, there is no coupling between the subproblems and this will produce poor convergence. We therefore modify the matrix splitting algorithm in order to produce a simple coupling between processors.

If our original quadratic program (1) is feasible, then it is solvable. Hence its Wolfe dual is solvable and this dual is equivalent to the LCP(H, h), given by

$$Hu + h \geqslant 0, \qquad \langle u, Hu + h \rangle = 0, \qquad u \geqslant 0.$$

In fact LCP(M, q) constitutes a replication of the Wolfe dual LCP(H, h) three times. Thus the solvability of LCP(H, h) is equivalent to the solvability of LCP(M, q). It is also clear that the solution set of LCP(M, q) is given by

$$Z^* = U^* \times U^* \times U^*$$

where U^* is the solution set of LCP(H, h). We now construct an algorithm which has two steps. The first step is an iteration of the matrix splitting algorithm

$$\bar{z}^{i+1} = \left(\bar{z}^{i+1} - \left(B\bar{z}^{i+1} + Cz^i + q\right)\right)_+ \tag{18}$$

as described in (12), where we require the splitting (B, C) to be regular, that is

$$M = B + C, \qquad B - C \text{ is positive definite.} \tag{19}$$

The second step is used to force the elements of z^i to converge to a point in Z^* where $u_1 = u_2 = u_3$ and thus introduce a very simple coupling between the subproblems. Hence we define a subspace L by

$$L := \left\{ z = (u_1, u_2, u_3) \middle| u_1 = u_2 = u_3 \right\}$$

and generate z^{i+1} by projecting the iterate given by the matrix splitting step onto L, that is

$$z^{i+1} = \pi_L(\bar{z}^{i+1}). \tag{20}$$

This exactly corresponds to the master processor step described in (4). We note that $M(L) \subseteq L$ and the following important facts concerning projections onto L.

LEMMA 2.2. *Suppose L is a subspace and $a = \pi_L(b)$. Then, for all $c \in L$,*

$$\langle b - a, c \rangle = 0. \tag{21}$$

If M is symmetric and $M(L) \subseteq L$ then

$$\|a - c\|_M^2 = \|b - c\|_M^2 - \|a - b\|_M^2 \tag{22}$$

where $\|x\|_M = (\langle x, Mx \rangle)^{1/2}$.

PROOF. The first statement of the lemma follows directly from the definitions of projections and subspaces. To prove (22), we note from (21) that

$$\begin{aligned} \|a - c\|_M^2 &= \|b - c\|_M^2 + 2\langle a - b, M(b - c)\rangle + \|a - b\|_M^2 \\ &= \|b - c\|_M^2 + 2\langle a - b, M(b - a)\rangle + \|a - b\|_M^2 \\ &= \|b - c\|_M^2 - \|a - b\|_M^2. \quad \square \end{aligned}$$

We now proceed to analyze the algorithm. We will invoke the following merit function to prove linear convergence in the dual space

$$f(z) := \tfrac{1}{2}z^T Mz + q^T z. \tag{23}$$

Our main theorem requires the following result due to Luo and Tseng (1992, Theorem 2.1) which we state here for completeness.

PROPOSITION 2.3. *There exist scalars* $\epsilon > 0$ *and* $\tau > 0$ *such that*

$$\operatorname{dist}(z|Z^*) \leqslant \tau \| z - (z - Mz - q)_+ \|$$

for all $z \geqslant 0$ *with* $\|z - (z - Mz - q)_+\| \leqslant \epsilon$.

We now give our main theorem. The proof of this theorem is modeled after the proof of Theorem 3.1 given in Luo and Tseng (1992).

THEOREM 2.4. *Suppose that* M *is symmetric and positive semidefinite and that* f, *given by* (23), *is bounded from below on* $\mathbb{R}^n_+$. *Let* $\{z^i\}$ *be the iterates generated by the matrix splitting algorithm* (18), (19), (20). *Then* $\{z^i\}$ *converges at least linearly to an element of* $Z^* \cap L$.

PROOF. We first show that

$$f(z^{i+1}) - f(z^i) \leqslant -\nu/2 \| \bar{z}^{i+1} - z^i \|^2, \quad \forall i, \tag{24}$$

where ν is the smallest eigenvalue of $B - C$. To see this, fix any i. The definition of f leads to

$$f(z^{i+1}) - f(z^i) = \langle Mz^i + q, z^{i+1} - z^i \rangle + \tfrac{1}{2}\langle z^{i+1} - z^i, M(z^{i+1} - z^i)\rangle,$$

but $z^{i+1} = \pi_L(\bar{z}^{i+1})$, $Mz^i + q \in L$, so (21) gives $\langle z^{i+1} - \bar{z}^{i+1}, Mz^i + q\rangle = 0$, hence

$$\begin{aligned} f(z^{i+1}) - f(z^i) &= \langle Mz^i + q, \bar{z}^{i+1} - z^i \rangle + \tfrac{1}{2}\langle z^{i+1} - z^i, M(z^{i+1} - z^i)\rangle \\ &= \langle B\bar{z}^{i+1} + Cz^i + q, \bar{z}^{i+1} - z^i\rangle + \tfrac{1}{2}\|z^{i+1} - z^i\|_M^2 - \|\bar{z}^{i+1} - z^i\|_B^2 \end{aligned}$$

the last equality following from $M = B + C$. However, the fact that $z^{i+1} = \pi_L(\bar{z}^{i+1})$ and $z^i \in L$ can be used in (22) to derive

$$\|z^{i+1} - z^i\|_M^2 = \|\bar{z}^{i+1} - z^i\|_M^2 - \|z^{i+1} - \bar{z}^{i+1}\|_M^2. \tag{25}$$

Also, by definition of $\bar{z}^{i+1}$, we have

$$\langle B\bar{z}^{i+1} + Cz^i + q, \bar{z}^{i+1} - z\rangle \leqslant 0, \quad \forall z \geqslant 0. \tag{26}$$

Now, from the above,

$$\begin{aligned} f(z^{i+1}) - f(z^i) &= \langle B\bar{z}^{i+1} + Cz^i + q, \bar{z}^{i+1} - z^i\rangle \\ &\quad + \tfrac{1}{2}\|z^{i+1} - z^i\|_M^2 - \|\bar{z}^{i+1} - z^i\|_B^2 \end{aligned}$$

which using (25) gives

$$f(z^{i+1}) - f(z^i)$$
$$= \langle B\bar{z}^{i+1} + z^i + q, \bar{z}^{i+1} - z^i \rangle - \tfrac{1}{2}\|z^{i+1} - \bar{z}^{i+1}\|_M^2 - \|\bar{z}^{i+1} - z^i\|_{B-M/2}^2$$

and applying (26) with $z = z^i$ and the fact that M is positive semidefinite,

$$f(z^{i+1}) - f(z^i) \leqslant -\|\bar{z}^{i+1} - z^i\|_{B-M/2}^2 \leqslant -\tfrac{1}{2}\|\bar{z}^{i+1} - z^i\|_{B-C}^2,$$

the final inequality following from $M = B + C$. Thus (24) holds.

Now we claim that

$$(27) \qquad \left\| z^i - (z^i - Mz^i - q)_+ \right\| \leqslant \kappa_1 \|\bar{z}^{i+1} - z^i\|, \quad \forall i,$$

for some constant $\kappa_1 > 0$. First, fix an iteration i. From the definition of $\bar{z}^{i+1}$ we have

$$\left\| z^i - (z^i - Mz^i - q)_+ \right\|$$
$$= \left\| z^i - (z^i - Mz^i - q)_+ - \bar{z}^{i+1} + (\bar{z}^{i+1} - B\bar{z}^{i+1} - Cz^i - q)_+ \right\|$$

but projections are nonexpansive, so

$$\leqslant \|z^i - \bar{z}^{i+1}\| + \|z^i - Mz^i - q - \bar{z}^{i+1} + B\bar{z}^{i+1} + Cz^i + q\|$$

and $M = B + C$ implies

$$\leqslant 2\|z^i - \bar{z}^{i+1}\| + \left\| B(\bar{z}^{i+1} - z^i) \right\|$$
$$\leqslant (2 + \|B\|)\|z^i - \bar{z}^{i+1}\|.$$

Thus (27) holds with $\kappa_1 = 2 + \|B\|$.

Since, by assumption, f is bounded below on $\mathbb{R}_+^n$, (24) implies that $\|\bar{z}^{i+1} - z^i\| \to 0$. It then follows from (27) that $\|z^i - (z^i - Mz^i - q)_+\| \to 0$ and so by Proposition 2.3 there exists a scalar constant $\kappa_2 > 0$ and an index $\hat{i}$ such that

$$\operatorname{dist}(z^i | Z^*) \leqslant \kappa_2 \|\bar{z}^{i+1} - z^i\|, \quad \forall i \geqslant \hat{i}.$$

Thus $\operatorname{dist}(z^i|Z^*) \to 0$. It is also well known that f is constant on Z^* so we shall denote this constant value by f^∞.

We now show that $f(z^i) \to f^\infty$ and estimate the speed of convergence. Fix any $i \geqslant \hat{i}$ and suppose (since $z^i \in L$) that $z^i = (u^i, u^i, u^i)$. Let y^i by defined by $y^i := (\hat{u}^i, \hat{u}^i, \hat{u}^i)$ where $\hat{u}^i = \pi_{U^*}(u^i)$. Then $y^i \in Z^*$ and

$$\|y^i - z^i\|^2 = 3\|\hat{u}^i - u^i\|^2 \leqslant 3\operatorname{dist}(z^i|Z^*)^2.$$

Thus

$$(28) \qquad \|y^i - z^i\| \leqslant \sqrt{3}\operatorname{dist}(z^i|Z^*) \leqslant \sqrt{3}\kappa_2\|\bar{z}^{i+1} - z^i\| =: \kappa_3\|\bar{z}^{i+1} - z^i\|.$$

Now

$$f(z^{i+1}) - f^\infty = f(z^{i+1}) - f(y^i)$$
$$= \langle My^i + q, z^{i+1} - y^i \rangle + \tfrac{1}{2}\|z^{i+1} - y^i\|_M^2$$

but $My^i + q \in L$, so (21) implies that $\langle z^{i+1} - \bar{z}^{i+1}, My^i + q \rangle = 0$, giving

$$f(z^{i+1}) - f^\infty = \langle My^i + q, \bar{z}^{i+1} - y^i \rangle + \tfrac{1}{2}\|z^{i+1} - y^i\|_M^2.$$

Invoking (26) with $z = y^i$,

$$f(z^{i+1}) - f^\infty \leqslant \langle My^i + q, \bar{z}^{i+1} - y^i \rangle - \langle B\bar{z}^{i+1} + Cz^i + q, \bar{z}^{i+1} - y^i \rangle$$
$$+ \tfrac{1}{2}\|z^{i+1} - y^i\|_M^2$$

but $M = B + C$, so

$$f(z^{i+1}) - f^\infty = \left\langle C(y^i - z^i), \bar{z}^{i+1} - y^i \right\rangle - \|\bar{z}^{i+1} - y^i\|_B^2 + \tfrac{1}{2}\|z^{i+1} - y^i\|_M^2.$$

Now use $y^i \in L$ in (22) to get

$$\|z^{i+1} - y^i\|_M^2 = \|\bar{z}^{i+1} - y^i\|_M^2 - \|z^{i+1} - \bar{z}^{i+1}\|_M^2$$

so that

$$f(z^{i+1}) - f^\infty \leqslant \left\langle C(y^i - z^i), \bar{z}^{i+1} - y^i \right\rangle - \|\bar{z}^{i+1} - y^i\|_{B-M/2}^2 - \tfrac{1}{2}\|z^{i+1} - \bar{z}^{i+1}\|_M^2.$$

But both M and $B - M/2 = (B - C)/2$ are positive semidefinite, hence

$$f(z^{i+1}) - f^\infty \leqslant \|C\|\,\|y^i - z^i\|\,\|\bar{z}^{i+1} - y^i\|. \tag{29}$$

However, by (28),

$$\|\bar{z}^{i+1} - y^i\| \leqslant \|\bar{z}^{i+1} - z^i\| + \|z^i - y^i\|$$
$$\leqslant \|(\kappa_3 + 1)\|\bar{z}^{i+1} - z^i\|$$

which when substituted in the above inequality and again using (28) gives

$$f(z^{i+1}) - f^\infty \leqslant \|C\|\kappa_3(\kappa_3 + 1)\|\bar{z}^{i+1} - z^i\|^2.$$

Let us define $\kappa_4 := \|C\|\kappa_3(\kappa_3 + 1)$. It now follows that

$$f(z^{i+1}) - f^\infty \leqslant \kappa_4\|\bar{z}^{i+1} - z^i\|^2$$
$$\leqslant (2\kappa_4/\nu)\left(f(z^i) - f(z^{i+1})\right), \quad \forall i \geqslant \hat{i},$$

the last inequality following from (24). If we rearrange terms we find

$$\left(1 + \frac{2\kappa_4}{\nu}\right)\left(f(z^{i+1}) - f^\infty\right) \leqslant \frac{2\kappa_4}{\nu}\left(f(z^i) - f^\infty\right), \quad \forall i \geqslant \hat{i}.$$

Thus $f(z^i)$ converges at least linearly to f^∞. Rearranging (24), we see

$$\|\bar{z}^{i+1} - z^i\|^2 \leqslant \tfrac{2}{\nu}\left[f(z^i) - f(z^{i+1})\right]$$

and since $f(z^{i+1}) \geqslant f^\infty$ for all i,

$$\begin{aligned}\|\bar{z}^{i+1} - z^i\|^2 &\leqslant \tfrac{2}{\nu}\left[f(z^i) - f^\infty\right]\\ &\leqslant \kappa_5 \alpha^i,\end{aligned}$$

for some constant $\kappa_5 > 0$, with the constant $\alpha < 1$ given by the linear convergence of $f(z^i)$. Since the projection operator is nonexpansive, it follows that

$$\|z^{i+1} - z^i\| \leqslant \|\bar{z}^{i+1} - z^i\| \leqslant \sqrt{\kappa_5}\,(\alpha)^{i/2}$$

and hence that $\{z^i\}$ also converges at least linearly (in the root sense) to some z^*. Since $\operatorname{dist}(z^i|Z^*) \to 0$, z^* is an element of Z^*. Furthermore, since L is closed if follows that $z^* \in L$. □

We note that Theorem 2.4 can be extended to cover the case of inexact solution of (18), namely

$$\bar{z}^{i+1} = \left(\bar{z}^{i+1} - \left(B\bar{z}^{i+1} + Cz^i + q + h^i\right)\right)_+$$

provided that the error satisfies

$$\|h^i\| \leqslant \left(\tfrac{\nu}{2} - \epsilon\right)\|\bar{z}^{i+1} - z^i\|$$

for some $\epsilon > 0$. The modification of the proof only requires changes to (24) and (29).

We now relate Theorem 2.4 to our PCD algorithm. The linear independence assumption in the following corollary is somewhat strong—however, it does give an indication of how to distribute the constraints in practice. The computational experiments ignore this assumption and seem entirely satisfactory.

COROLLARY 2.5 (PCD CONVERGENCE FOR QUADRATIC PROGRAMS). *Assume that A_l has linearly independent rows and the feasible region of* (1) *is nonempty. The PCD algorithm defined by* (2), (3), (4) *and* (5) *converges linearly, that is*

$$x_l^i \to x^*, \qquad l = 1, 2, 3$$

and

$$t_{jl}^i \to s_j^*, \qquad j, l = 1, 2, 3,\ j \neq l$$

with x^* a solution of (1).

PROOF. As was shown above, the algorithm defined by (2), (3), (4) and (5) is precisely of the form given in (18) and (20) with B defined in (13) and C defined by (14). Hence, in the dual space, the algorithm corresponds to an iterative matrix splitting algorithm followed by a subspace projection. We now show that the conditions required for convergence of this method as given in Theorem 2.4 are satisfied.

Note that $M = B + C$ and therefore to show that M is positive semidefinite it is sufficient to show (by (17)) that H as defined in (10) is positive semidefinite. This is clear since

$$H = \gamma \begin{bmatrix} A_1 \\ A_2 \\ A_3 \end{bmatrix} Q^{-1} \begin{bmatrix} A_1^T & A_2^T & A_3^T \end{bmatrix}$$

and Q is positive definite (hence also Q^{-1}).

We now show $B - C$ is positive definite. Note that

$$B - C = \begin{bmatrix} H + 2J_1 & 0 & 0 \\ 0 & H + 2J_2 & 0 \\ 0 & 0 & H + 2J_3 \end{bmatrix}$$

so we only show that $H + 2J_1$ is positive definite. Suppose that

$$(x_1, x_2, x_3)^T (H + 2J_1)(x_1, x_2, x_3) = 0.$$

Then

$$\begin{bmatrix} A_1^T & A_2^T & A_3^T \end{bmatrix} (x_1, x_2, x_3) = 0$$

and $x_2 = x_3 = 0$. Since A_1 has linearly independent rows it now follows that $x_1 = 0$, and so $H + 2J_1$ is positive definite as required. Thus $B - C$ is positive definite and so (19) holds.

It remains to show that f is bounded below on the positive orthant. The fact that f is bounded below is equivalent to a solution existing by Frank and Wolfe (1956). Since M is symmetric and positive semidefinite any solution of $\text{minimize}_{z \geq 0} f(z)$ solves LCP(M, q) and conversely. As shown above, any solution of LCP(M, q) leads to a solution of LCP(H, h) which is the dual of (1). The fact that this problem has a solution is equivalent to (1) having a solution which by strong convexity is equivalent to (1) being feasible, as assumed above.

Thus by invoking Theorem 2.4 we see that the dual iterates z^i converge linearly to z^*, a solution of LCP(M, q). Furthermore, $z^* \in L$, so that $z^* = (u^*, u^*, u^*)$. Thus from (9), $t_{jl}^* = s_j^*$, $\forall l \neq j$, and so it follows from (7) that

$$\bar{x}_l^* = -Q^{-1}\left(c + \sum_{k=1}^{3} A_k^T \bar{s}_k^*\right),$$

$$\bar{s}_l^* = \left(\bar{s}_l^* + \gamma(A_l \bar{x}_l^* - a_l)\right)_+, \qquad l = 1, 2, 3.$$

Hence $x_l^* = x^*$ for $l = 1, 2, 3$ and we have

$$\bar{x}^* = -Q^{-1}\left(c + \sum_{k=1}^{3} A_k^T \bar{s}_k^*\right)$$

$$\bar{s}_l^* = \left(\bar{s}_l^* + \gamma(A_l \bar{x}_l^* - a_l)\right)_+, \qquad l = 1, 2, 3.$$

But this implies that $\bar{x}^*$ solves (1). □

3. Computational results. We have tested out the algorithm described above on some linear programming problems. The standard form linear program

$$\begin{aligned} &\text{minimize} && c^T x \\ &\text{subject to} && Ax = b \\ & && x \geqslant 0 \end{aligned}$$

has the dual problem

$$\begin{aligned} &\text{maximize} && b^T y \\ &\text{subject to} && A^T y \leqslant c \end{aligned} \tag{30}$$

and these problems are in precisely the form of our preceding discussion except that the objective is not strongly convex. In order to strongly convexify the objective we have used the least two-norm formulation (Mangasarian and Meyer 1979, Mangasarian 1984), where for $\epsilon \in (0, \bar{\epsilon}]$ for some $\bar{\epsilon} > 0$, the solution of

$$\begin{aligned} &\text{minimize} && -b^T y + \tfrac{\epsilon}{2} y^T y \\ &\text{subject to} && A^T y \leqslant c \end{aligned} \tag{31}$$

is the least two-norm solution of (30). For the purpose of our computation, a value of $\epsilon = 10^{-6}$ was used.

We have split up the problems as follows: firstly the user has specified the number of processors available and the problem has been split randomly into that many blocks of equal size. If the number of constraints in each block is not the same we have added combinations of constraints from other blocks to make the number of constraints in each block equal with the aim of balancing the load in each processor. Specifically, the final block of constraints is augmented with constraints which are the sum of the correspondingly indexed constraints from the other blocks. That is, the kth added constraint in the final block is

$$\left(\sum_{j=1}^{p-1} (A_j)_{k\cdot} \right) x \leqslant \left(\sum_{j=1}^{p-1} (a_j)_k \right)$$

where p is the number of processors.

The PCD algorithm was implemented on the Sequent Symmetry S-81 shared memory multiprocessor. The subproblems were solved on each processor using MINOS 5.3, a more recent version of (Murtagh and Saunders (1983). The explicit constraints in each subproblem remained fixed throughout the computation but the blocks were not chosen to satisfy the linear independence assumption.

The algorithm was terminated whenever the difference in the primal objective value of (30) and its dual objective value (normalized by the sum of their absolute values) differed by less than 10^{-5}. The constraint violation was also required to be less than this tolerance.

We have used the following scheme to update the augmented Lagrangian parameter, γ. Initially it is set at 10 and it is increased by a factor of 4 only when the norm of the violation of the constraints increases. This choice was determined after some experimentation. Other choices are of course possible. In particular, the number of parallel iterations of the method can be decreased by choosing larger values for γ.

TABLE 1

Old PCD *Algorithm with variable* λ

Problem	Variables	Constraints	Blocks 3	6	9	18
Ex6	3	5	2	2		
Ex9	5	11	4	5		
Ex10	6	14	4	4	4	
AFIRO	27	51	13	15	15	14
ADLittle	56	138	12	14	14	15

TABLE 2

New PCD *Algorithm with projection step*

Problem	Variables	Constraints	Blocks 3	6	9	18
Ex6	3	5	2	2		
Ex9	5	11	3	3		
Ex10	6	14	3	2	3	
AFIRO	27	51	8	8	8	8
ADLittle	56	138	9	9	9	9

However, this makes the subproblems more difficult to solve due to conditioning problems and leads to worse computational timings overall. In fact, for some large initial choices of γ, MINOS was observed to fail. If γ is not increased as quickly as the given heuristic, then (in general) the number of parallel iterations increases as the feasibility requirement is harder to satisfy.

We give two tables above for comparison. Table 1 gives the best results that were obtained using the algorithm described in Ferris and Mangasarian (1991) on the Sequent Symmetry S-81 for 5 small linear programs reformulated as in (31). The first three are homemade test problems, while the last two, AFIRO and ADLittle, are from the NETLIB collection (Gay 1985). In the tables, an empty column entry signifies that we did not perform the computation. Note also that these results include a heuristic for calculating a step length.

Table 2 gives the results for the algorithm outlined in this paper. We remark that this algorithm performs uniformly better than the one described in (Ferris and Mangasarian (1991). Furthermore, its implementation is somewhat simpler. Note the strong indication that these results give to the fact that the number of iterations is independent of the number of processors used.

4. Conclusions. We have presented a method for solving strongly convex quadratic programs with large numbers of linear inequality constraints and have shown the method to be linearly convergent. The method is easy to implement and preliminary computational results are encouraging.

Further extensions of this work are possible when the subspace L is modified appropriately. These extensions will be addressed in a future paper.

Acknowledgements. This material is based on research supported by the Air Force Office of Scientific Research Grant AFOSR-89-0410 and the National Science Foundation Grant CCR-9157632.

References

Bertsekas, D. P. (1982). *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, NY.
______ and Tseng, P. (1991). Partial Proximal Minimization for Convex Programming. Manuscript.
Dembo, R. S. (1986). The Performance of NLPNET, a Large-Scale Nonlinear Network Optimizer. *Math. Programming Stud.* **26** 245–248.
Ferris, M. C. and Mangasarian, O. L. (1991). Parallel Constraint Distribution. *SIAM J. Optim.* **1** 487–500.
Frank, M. and Wolfe, P. (1956). An algorithm for quadratic programming. *Naval Research Logistics Quarterly* **3** 95–110.
Gay, D. M. (1985). Electronic Mail Distribution of Linear Programming Test Problems. *COAL Newsletter* **13** 10–12.
Luo, Z.-Q. and Tseng, P. (1992). Error Bound and Convergence Analysis of Matrix Splitting Algorithms for the Affine Variational Inequality Problem. *SIAM J. Optim.* **2** 43–54.
Mangasarian, O. L. (1969). *Nonlinear Programming*. McGraw-Hill, New York.
______ (1984). Normal Solutions of Linear Programs. *Math. Programming Stud.* **22** 206–216.
______ and Meyer, R. R. (1979). Nonlinear Perturbation of Linear Programs. *SIAM J. Control Optim.* **17** 745–752.
Murtagh, B. A. and Saunders, M. A. (1983). MINOS 5.0 User's Guide. Technical Report SOL 83.20, Stanford University.
Rockafellar, R. T. (1974). Augmented Lagrange Multiplier Functions and Duality in Nonconvex Programming. *SIAM J. Control* **12** 268–285.
______ (1976). Augmented Lagrangians and Applications of the Proximal Point Algorithm in Convex Programming. *Math. Oper. Res.* **1** 97–116.

M. C. Ferris: Computer Sciences Department, University of Wisconsin, 1210 West Dayton Street, Madison, Wisconsin 53706; e-mail: ferris@cs.wisc.edu