Appl Math Optim 25:247–262 (1992)

Applied Mathematics and Optimization

A Projection and Contraction Method for a Class of Linear Complementarity Problems and Its Application in Convex Quadratic Programming

Bingsheng He*

Institut für Angewandte Mathematik und Statistik, Universität Würzburg, Am Hubland, 8700 Würzburg, Federal Republic of Germany

Communicated by J. Stoer

Abstract. In this paper we propose a new iterative method for solving a class of linear complementarity problems:

$$u \geq 0, \qquad Mu + q \geq 0, \qquad u^T(Mu + q) = 0,$$

where M is a given $l \times l$ positive semidefinite matrix (not necessarily symmetric) and q is a given l-vector. The method makes two matrix-vector multiplications and a trivial projection onto the nonnegative orthant at each iteration, and the Euclidean distance of the iterates to the solution set monotonously converges to zero. The main advantages of the method presented are its simplicity, robustness, and ability to handle large problems with any start point. It is pointed out that the method may be used to solve general convex quadratic programming problems. Preliminary numerical experiments indicate that this method may be very efficient for large sparse problems.

Key Words. Projection, Féjer-contraction, Linear complementarity problem, Linear programming, Convex quadratic programming.

1. Introduction

Let M be an $l \times l$ matrix and $q \in R^l$, where R^l denotes the l-dimensional Euclidean space. The problem of finding a $u \in R^l$ satisfying

$$\text{(LCP)} \quad u \geq 0, \qquad Mu + q \geq 0, \qquad u^T(Mu + q) = 0, \tag{1}$$

* On leave from the Department of Mathematics, University of Nanjing, Nanjing, People's Republic of China.

is known as a linear complementarity problem (abbreviated to LCP). Let $L = \{1, 2, \ldots, l\}$ and $I \subset L$. In this paper we consider the following generalized linear complementarity problem, which is an LCP with equality restrictions and unrestricted variables: Find a vector $u \in R^l$ such that

$$\text{(GLCP)} \quad \begin{cases} u_i \geq 0, \quad (Mu + q)_i \geq 0, \quad u_i(Mu + q)_i = 0, & \text{for} \quad i \in I, \\ (Mu + q)_i = 0, & \text{otherwise.} \end{cases} \tag{2}$$

Here u_i denotes the ith component of u. Let

$$\Omega = \{u \in R^l \,|\, u_i \geq 0 \text{ for } i \in I\},$$

$$\Omega^* = \{u \in R^l \,|\, u \text{ solves (GLCP)}\}.$$

Throughout this paper we assume that M is positive semidefinite (but not necessarily symmetric) and $\Omega^* \neq \emptyset$.

Some computational methods have been developed for solving LCPs. Most of them are direct and use a sequence of pivoting operations to the system of linear equations $w = Mu + q$ [7], [8], [27]. Recently, Kojima *et al.* [25] proposed a polynomial-time method for LCP. The method is an extension of the new polynomial-time methods for linear programming, which originated with Karmarkar [23] and have been developed by many researchers. Besides the abovementioned finite methods, several iterative methods for solving LCPs are known [1], [29], [31]. These iterative methods can be viewed as an extension similar to the SOR method for solving $Mu + q = 0$. Even though the iterative methods are not finite, they are considerably simpler and well suited for large sparse problems.

Our objective in this paper is to develop a new iterative method for solving problem (2). The method makes a trivial projection onto Ω at each iteration and generates a sequence $\{u^{(k)}\}$ satisfying

$$\operatorname{dist}(u^{(k+1)}, \Omega^*) < \operatorname{dist}(u^{(k)}, \Omega^*), \tag{3}$$

where

$$\operatorname{dist}(u, \Omega^*) = \inf\{\|u - u^*\|_2 \,|\, u^* \in \Omega^*\}. \tag{4}$$

Using the terminology of [5], we call this method a projection and contraction method.

The outline of this paper is as follows. In Section 2 we illustrate some equivalent expressions of (GLCP). Section 3 describes the details of the projection and contraction method, and in Section 4 we show the global convergence and give an error analysis for it. In Section 5 we illustrate how our iterative method is applied to general convex quadratic programming problems. The relationship of this method to other projection methods is described in Section 6. In Section 7 we give a variant of the method—the scaled projection and contraction method. Finally, we give some numerical results for solving the Dirichlet problem with obstacles. In what follows, $P_\Omega(\cdot)$ denotes the (trivial) projection onto the set Ω and $\|\cdot\|_2$, $\|\cdot\|_\infty$ denote the Euclidean and the max-norm, respectively.

2. The Projection Equation and Measure Functions

It is easy to see that $u \in \Omega^*$ if and only if u satisfies the following projection equation:

$$P_\Omega[u - (Mu + q)] = u. \tag{5}$$

See [29] for a proof.

Lemma 1. *For all* $u \in \Omega$,

$$\{u - P_\Omega[u - (Mu + q)]\}^T(Mu + q) \geq \|u - P_\Omega[u - (Mu + q)]\|^2. \tag{6}$$

Proof. Since $\Omega \subset R^l$ is a closed convex set, we know by the properties of a projection on a closed convex set [5] that, for any $v \in R^l$ and $u \in \Omega$,

$$[v - P_\Omega(v)]^T[u - P_\Omega(v)] \leq 0. \tag{7}$$

For $v := u - (Mu + q)$ we obtain the assertion of the lemma. □

Let

$$\varphi(u) := \{u - P_\Omega[u - (Mu + q)]\}^T(Mu + q), \tag{8}$$

$$\psi(u) := \|u - P_\Omega[u - (Mu + q)]\|^2. \tag{9}$$

Actually, by (5) and (6), we have proved the following.

Theorem 1. *Let* $\varphi(u)$ *and* $\psi(u)$ *be defined as in* (8) *and* (9), *respectively. Then*

(i) $\varphi(u) \geq \psi(u) \geq 0$ *for all* $u \in \Omega$.
(ii) $u \in \Omega$ *and* $\varphi(u) = 0 \Leftrightarrow \psi(u) = 0 \Leftrightarrow u \in \Omega^*$.

For $u \in \Omega$, the functions $\varphi(u)$ and $\psi(u)$ can be viewed as measures for the distance of u from the solution set Ω^*.

3. A Search Direction and the Algorithm

For any $u^* \in \Omega^*$, the search direction of the contraction method should be a descent direction of $F(u) = \frac{1}{2}\|u - u^*\|^2$. From (6) and (8), if we put

$$g(u) = M^T\{u - P_\Omega[u - (Mu + q)]\} + (Mu + q), \tag{10}$$

then we have

Theorem 2. *Let* $u^* \in \Omega^*$ *and* $u \in \Omega$, *then*

$$(u - u^*)^T g(u) \geq \varphi(u), \tag{11}$$

i.e., for $u \in \Omega \backslash \Omega^*$, $-g(u)$ *is a descent direction of* $F(u)$ *at* u.

Proof. Since $u^* \in \Omega^*$, we have

$$(Mu^* + q)^T u^* = 0 \tag{12}$$

and, for any $v \in R^l$,

$$(Mu^* + q)^T P_\Omega(v) \geq 0. \tag{13}$$

As M is positive semidefinite, we get

$$\begin{aligned}(u - u^*)^T g(u) &= \{u - P_\Omega[u - (Mu + q)]\}^T[(Mu + q) - (Mu^* + q)] \\ &\quad + (u - u^*)^T(Mu + q) \\ &= \varphi(u) + \{P_\Omega[u - (Mu + q)]\}^T(Mu^* + q) \\ &\quad - u^T(Mu^* + q) + (u - u^*)^T(Mu + q) \\ &\geq \varphi(u) + (u - u^*)^T M(u - u^*) \\ &\geq \varphi(u).\end{aligned}$$

□

For $u \in \Omega$, let

$$N(u) = \{i \in I \,|\, u_i = 0 \text{ and } g_i \geq 0\}, \tag{14}$$

$$B(u) = L \backslash N(u). \tag{15}$$

Correspondingly, denote

$$u = \begin{pmatrix} u_B \\ u_N \end{pmatrix}, \qquad g(u) = \begin{pmatrix} g_B \\ g_N \end{pmatrix}, \qquad g_B(u) = \begin{pmatrix} g_B \\ 0 \end{pmatrix}, \qquad g_N(u) = \begin{pmatrix} 0 \\ g_N \end{pmatrix}. \tag{16}$$

Then

$$(u - u^*)^T g_N(u) \leq 0. \tag{17}$$

Therefore

$$(u - u^*)^T g_B(u) \geq (u - u^*)^T g(u) \geq \varphi(u). \tag{18}$$

Thus, we have the same algorithm as in [21].

Algorithm PC (Projection and Contraction Method)
Given $u^{(0)} \in \Omega$.
For $k = 0, 1, \ldots,$ *if* $u^{(k)} \notin \Omega^*$, then

1. Calculate $\varphi(u^{(k)})$ and $g(u^{(k)})$ by (8) and (10), respectively.
2. Determine $g_B(u^{(k)})$ by (14)–(16) and calculate the step size

$$\rho(u^{(k)}) = \frac{\varphi(u^{(k)})}{\|g_B(u^{(k)})\|^2}. \tag{19}$$

3. Update

$$\bar{u}^{(k)} = u^{(k)} - \rho(u^{(k)}) g_B(u^{(k)}), \tag{20}$$

$$u^{(k+1)} = P_\Omega[\bar{u}^{(k)}]. \tag{21}$$

In the practical computation, we use

$$\varphi(u^{(k)}) \le \varepsilon^2$$

or

$$\frac{\|u - P_\Omega[u - (Mu + q)]\|_\infty}{\|q\|_\infty} \le \varepsilon$$

as the termination criterion.

4. Convergence

Theorem 3. *Let $u^* \in \Omega^*$ and $\{u^{(k)}\}$ be the sequence generated by Algorithm PC. Then*

$$\|u^{(k+1)} - u^*\|^2 \le \|u^{(k)} - u^*\|^2 - \frac{\varphi^2(u^{(k)})}{\|g_B(u^{(k)})\|^2}. \tag{22}$$

Proof. Since $u^* \in \Omega$, we have, by the well-known projection property, for any $v \in R^l$,

$$\|P_\Omega(v) - u^*\| \le \|v - u^*\|, \tag{23}$$

so that

$$\|u^{(k+1)} - u^*\| \le \|\bar{u}^{(k)} - u^*\|. \tag{24}$$

By means of (18)–(20), we get

$$\|\bar{u}^{(k)} - u^*\|^2 \le \|u^{(k)} - u^*\|^2 - \frac{\varphi^2(u^{(k)})}{\|g_B(u^{(k)})\|^2} \tag{25}$$

and thus the theorem is proved. □

Because (22) is true for any $u^* \in \Omega^*$, by Theorem 3 we have actually shown

$$[\mathrm{dist}(u^{(k+1)}, \Omega^*)]^2 \le [\mathrm{dist}(u^{(k)}, \Omega^*)]^2 - \frac{\varphi^2(u^{(k)})}{\|g_B(u^{(k)})\|^2}, \tag{26}$$

i.e., the sequence $\{u^{(k)}\}$ is Féjer-monotone with respect to the solution set Ω^*.

Theorem 4. *If Ω^* is nonempty, then the algorithm is globally convergent, i.e.,*

$$\lim_{k\to\infty} \mathrm{dist}(u^{(k)}, \Omega^*) = 0. \tag{27}$$

Proof. Let u^* be a solution of problem (2). It is easy to check that every Féjer-monotone sequence is bounded. Suppose

$$\lim_{k\to\infty} \mathrm{dist}(u^{(k)}, \Omega^*) = \delta_0 > 0, \tag{28}$$

then

$$\{u^{(k)}\} \subset S := \{u \in \Omega \mid \delta_0 \leq \text{dist}(u, \Omega^*), \|u - u^*\| \leq \|u^{(0)} - u^*\|\} \tag{29}$$

and S is a closed bounded set. Moreover, from the assumption, $S \cap \Omega^* = \emptyset$, then on S

$$T(u) := \frac{\varphi^2(u)}{\|g(u)\|^2} > 0.$$

Since $T(u)$ is continuous on S, we have

$$\min\{T(u) \mid u \in S\} := \varepsilon_0 > 0. \tag{30}$$

From (28), there is a $k_0 > 0$, such that, for all $k > k_0$,

$$[\text{dist}(u^{(k)}, \Omega^*)]^2 < \delta_0^2 + \frac{\varepsilon_0}{2}. \tag{31}$$

On the other hand, from (26), (30), and $\|g_B(u)\| \leq \|g(u)\|$,

$$[\text{dist}(u^{(k+1)}, \Omega^*)]^2 \leq [\text{dist}(u^{(k)}, \Omega^*)]^2 < \delta_0^2 - \frac{\varepsilon_0}{2}. \tag{32}$$

This contradicts (28). □

The reason that we use $\varphi(u^{(k)}) \leq \varepsilon^2$ as our termination criterion in practice is the following.

Theorem 5. *Let $\hat{u} \in \Omega$ and $\varphi(\hat{u}) \leq \varepsilon^2$, then we can find $\tilde{u}$ such that $\|\tilde{u} - \hat{u}\| \leq \varepsilon$ and $\tilde{u}$ is a solution of the perturbed problem*

$$P_\Omega[u - (Mu + \tilde{q})] = u, \tag{33}$$

where $\tilde{q}$ satisfies

$$\|\tilde{q} - q\| \leq (\|M\| + 1)\varepsilon. \tag{34}$$

Proof. Put $\hat{v} = M\hat{u} + q$. By the assumption and Theorem 1

$$\|\hat{u} - P_\Omega(\hat{u} - \hat{v})\| \leq \varepsilon. \tag{35}$$

Let

$$\tilde{u}_i = \begin{cases} 0 & \text{if} \quad i \in I \quad \text{and} \quad \hat{u}_i < \hat{v}_i, \\ \hat{u}_i & \text{otherwise.} \end{cases} \tag{36}$$

By the construction of $\tilde{u}$, if $\tilde{u}_i \neq \hat{u}_i$, then

$$\tilde{u}_i - [P_\Omega(\tilde{u} - \hat{v})]_i = 0,$$

hence

$$(\hat{u} - \tilde{u}) \perp (\tilde{u} - P_\Omega(\tilde{u} - \hat{v})). \tag{37}$$

Similarly, if $\tilde{u}_i \neq \hat{u}_i$, then

$$[P_\Omega(\tilde{u} - \hat{v})]_i = [P_\Omega(\hat{u} - \hat{v})]_i.$$

Therefore

$$(\hat{u} - \tilde{u}) + [\tilde{u} - P_\Omega(\tilde{u} - \hat{v})] = \hat{u} - P_\Omega(\hat{u} - \hat{v}). \tag{38}$$

From (37) and (38) we get

$$\|\hat{u} - \tilde{u}\|^2 + \|\tilde{u} - P_\Omega(\tilde{u} - \hat{v})\|^2 = \|\hat{u} - P_\Omega(\hat{u} - \hat{v})\|^2. \tag{39}$$

It follows that $\|\tilde{u} - \hat{u}\| \leq \varepsilon$ and

$$\|\tilde{u} - P_\Omega(\tilde{u} - \hat{v})\| \leq \varepsilon. \tag{40}$$

From (40) we can easily find a $\tilde{v}$ satisfying $\|\tilde{v} - \hat{v}\| \leq \varepsilon$ and $\|\tilde{u} - P_\Omega(\tilde{u} - \tilde{v})\| = 0$. Let $\tilde{q} = \tilde{v} - M\tilde{u}$, then

$$\|\tilde{q} - q\| = \|\tilde{v} - M\tilde{u} - q\| = \|\tilde{v} - \hat{v} - M(\tilde{u} - \hat{u})\| \leq (\|M\| + 1)\varepsilon. \qquad \square$$

5. Application to Convex Quadratic Programming

We have constructed a projection and contraction algorithm PC for problem (2). The principal formulas in this method are

$$\varphi(u) = \{u - P_\Omega[u - (Mu + q)]\}^T(Mu + q)$$

and

$$g(u) = M^T\{u - P_\Omega[u - (Mu + q)]\} + (Mu + q).$$

If we let $v = u - P_\Omega[u - (Mu + q)]$, then the main work in each iteration is the computation of Mu and $M^T v$. The algorithm can start at any $u^{(0)} \in \Omega$, and (26) shows that the method is robust and stable.

Our primary interest, however, is the efficient solution of convex quadratic problems:

$$\begin{array}{lll} \text{(CQP)} & \text{Minimize} & \tfrac{1}{2}x^T H x + c^T x \\ & \text{subject to} & A_{11}x_I + A_{12}x_{II} \geq b_I, \\ & & A_{21}x_I + A_{22}x_{II} = b_{II}, \\ & & x_I \geq 0, \end{array} \tag{41}$$

where

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \in R^{m \times n}, \qquad b = \begin{pmatrix} b_I \\ b_{II} \end{pmatrix} \in R^m, \qquad c = \begin{pmatrix} c_I \\ c_{II} \end{pmatrix} \in R^n,$$

and

$$H = \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix} \in R^{n \times n}$$

is a symmetric positive semidefinite matrix. Problems of this type arise very often in applications of numerical analysis, in optimal control and operations research, as well as subproblems in general nonlinear optimization problems via successive quadratic programming methods [20], [33], [35]. Many existing algorithms [2], [7], [8], [13], [15], [16], [17], [27], [36] use active set strategies and are direct extensions of the simplex method for linear programming. Under certain conditions most of these algorithms generate the same sequence of points, as shown by Pang [30] and Best [4]. We simultaneously consider the dual problem of (CQP):

$$\begin{array}{lll}\text{(DQP)} & \text{Maximize} & -\tfrac{1}{2}x^T Hx + b^T y \\ & \text{subject to} & H_{11}x_I + H_{12}x_{II} - A_{11}^T y_I - A_{21}^T y_{II} + c_I \geq 0, \\ & & H_{21}x_I + H_{22}x_{II} - A_{12}^T y_I - A_{22}^T y_{II} + c_{II} = 0, \\ & & y_I \geq 0. \end{array} \tag{42}$$

Let

$$\Omega_1 = \{x \in R^n | x_I \geq 0\}, \qquad \Omega_2 = \{y \in R^m | y_I \geq 0\}, \qquad \Omega = \Omega_1 \times \Omega_2.$$

By the duality of convex programming [8], if x^* is a solution of (CQP), then there exists a y^* such that $u^* = (x^*, y^*)$ is a solution of (DQP). Let

$$\Omega^* = \{u^* = (x^*, y^*) | x^* \text{ solves (CQP)}, (x^*, y^*) \text{ solves (DQP)}\}.$$

Then a necessary and sufficient condition for $u = (x, y) \in \Omega^*$ is the following:

$$\begin{cases} A_{11}x_I + A_{12}x_{II} - b_I \geq 0, \\ A_{21}x_I + A_{22}x_{II} - b_{II} = 0, \\ x_I \geq 0 \\ \text{(primal feasibility);} \\ H_{11}x_I + H_{12}x_{II} - A_{11}^T y_I - A_{21}^T y_{II} + c_I \geq 0, \\ H_{21}x_I + H_{22}x_{II} - A_{12}^T y_I - A_{22}^T y_{II} + c_{II} = 0, \\ y_I \geq 0 \\ \text{(dual feasibility);} \\ x_I^T[H_{11}x_I + H_{12}x_{II} - A_{11}^T y_I - A_{21}^T y_{II} + c_I] = 0, \\ y_I^T[A_{11}x_I + A_{12}x_{II} - b_I] = 0 \\ \text{(complementarity).} \end{cases}$$

Let

$$M = \begin{pmatrix} H & -A^T \\ A & 0 \end{pmatrix}, \qquad q = \begin{pmatrix} c \\ -b \end{pmatrix}, \tag{43}$$

then it is easy to see that the necessary and sufficient condition for $u \in \Omega^*$ can be written as

$$P_\Omega[u - (Mu + q)] = u.$$

Note that in this case M, although nonsymmetric, is also positive semidefinite. It is a generalized linear complementarity problem of type (2) and may be solved by Algorithm PC. If $H = 0$, then (41) reduces to a linear program. In particular, for a linear program in standard form, the corresponding formulas become

$$\varphi(u) = \|Ax - b\|^2 + [(x + (A^T y - c))_+ - x]^T (A^T y - c) \tag{44}$$

and

$$g(u) = \begin{pmatrix} g_x \\ g_y \end{pmatrix} = \begin{pmatrix} A^T(Ax - b) - (A^Ty - c) \\ A[(x + (A^Ty - c))_+ - x] + (Ax - b) \end{pmatrix}, \tag{45}$$

which have already been introduced in [21]. Because $u^* = (x^*, y^*)$ is the saddle point of the Lagrangean, sometimes we call the method a saddle-point algorithm [21], [22].

6. Relationship to Other Projection Methods

Let

$$f(x) = \tfrac{1}{2}x^THx + c^Tx, \tag{46}$$

$$C = \{x \mid A_{11}x_I + A_{12}x_{II} \geq b_I,\ A_{21}x_I + A_{22}x_{II} = b_{II},\ x_I \geq 0\}. \tag{47}$$

Other known projection methods use the iteration

$$x^{(k+1)} = P_C[x^{(k)} - \alpha_k g(x^{(k)})]. \tag{48}$$

An example is the Goldstein–Levitin–Polyak gradient projection method [3], [6], [10], [11], [18], [28] and the related projected Newton method [12]. In the original algorithms of Goldstein [18] and Levitin and Polyak [28] the step size α_k was chosen to be constant for all k. An alternative method for selecting the step size α_k was proposed by Bertsekas [3]. His method, using a modified Armijo line search, was a significant contribution to making the gradient projection method useful in practice. As C is a general polytope, however, in order to determine $x^{(k+1)}$ we have to solve a minimum norm problem (possibly more than one),

$$\min\{\|x - [x^{(k)} - \alpha g(x^{(k)})]\| \mid x \in C\}, \tag{49}$$

at each iteration, which is almost as expensive as solving (CQP) itself. Therefore, the overall efficiency of the gradient projection method is seriously affected by the complexity of problem (49).

Uzawa's method [33] is a gradient projection method using the iterative scheme (48) for solving (CQP) without line search. Suppose that x^* is a solution of (CQP) and $g(x)$ is the gradient of $f(x)$. Since $f(x)$ is convex, then, for any $x \in C$,

$$(x - x^*)^Tg(x) \geq f(x) - f(x^*) \geq 0. \tag{50}$$

His ideal step size α_k in (48) is defined by

$$\alpha_k = \frac{f(x^{(k)}) - f(x^*)}{\|g(x^{(k)})\|^2} \tag{51}$$

and the generated sequence $\{x^{(k)}\}$ satisfies

$$\|x^{(k+1)} - x^*\|^2 \leq \|x^{(k)} - x^*\|^2 - \alpha_k[f(x^{(k)}) - f(x^*)]. \tag{52}$$

This method is a beautiful iterative scheme for (CQP), but in practice there are some drawbacks, especially

(a) because $f(x^*)$ is usually unknown and then α_k in (51) is not computable, and
(b) we have to solve a minimum norm problem (49) at each iteration.

The difficulties of these projection methods for solving (CQP) are overcome by Algorithm PC. In the PC method we first transform (CQP) to a (GLCP), define a measure function $\varphi(u)$ (note that $\varphi(u)$ is not differentiable) and a search direction $g(u)$, which is not a gradient of $\varphi(u)$. Nevertheless, similarly as in Uzawa's method (see (50)), we have

$$(u - u^*)^T g_B(u) \geq \varphi(u) > 0$$

and can take

$$\rho(u) = \frac{\varphi(u)}{\|g_B(u)\|^2}$$

as the step size. Here $\varphi(u)$ and $\rho(u)$ are computable and the projection onto Ω is trivial. However, the algorithm usually generates an infinite sequence $\{u^{(k)}\} = \{(x^{(k)}, y^{(k)})\}$, and $\{x^{(k)}\}$ is not necessarily contained in the feasible set C, but, as is shown in Section 4, the sequence $\{x^{(k)}\}$ will be asymptotically feasible as $\text{dist}(u^{(k)}, \Omega^*) \to 0$, and, in fact, converges to a solution of (CQP).

The extra gradient method, which was proposed by Korpelevich [26], is another projection and contraction method for finding saddle points of problem (CQP). His iterative scheme is the following:

$$\hat{u}^{(k)} = P_\Omega[u^{(k)} - \alpha W(u^{(k)})], \tag{53}$$

$$u^{(k+1)} = P_\Omega[u^{(k)} - \alpha W(\hat{u}^{(k)})], \tag{54}$$

where $W(u) = Mu + q$ and α is a constant. For all $0 < \alpha < 1/\|M\|$, the iterates satisfy

$$\|u^{(k+1)} - u^*\|^2 \leq \|u^{(k)} - u^*\|^2 - (1 - \alpha^2\|M\|^2)\|u^{(k)} - \hat{u}^{(k)}\|^2. \tag{55}$$

If $\|M\| < 1$ and we take $\alpha = 1$, then the formula may be written as

$$u^{(k+1)} = P_\Omega[u^{(k)} - \alpha g_K(u^{(k)})], \tag{56}$$

where

$$g_K(u) = -M\{u - P_\Omega[u - (Mu + q)]\} + (Mu + q), \tag{57}$$

but our search direction for (CQP) is

$$g(u) = M^T\{u - P_\Omega[u - (Mu + q)]\} + (Mu + q).$$

For linear programming only, because in this case $M^T = -M$, the directions are equal (but not the step lengths).

7. An Extension—The Scaled PC Method

Let G be an $l \times l$ diagonal positive definite matrix.[1] It is easy to see that u solves

$$P_\Omega[u - (Mu + q)] = u$$

if and only if it is a solution of the following projection equation:

$$P_\Omega[u - G^{-1}(Mu + q)] = u. \tag{58}$$

Based on (58), similarly as (8) and (10) we let

$$\tilde{\varphi}(u, G) = \{u - P_\Omega[u - G^{-1}(Mu + q)]\}^T(Mu + q) \tag{59}$$

and

$$\tilde{g}(u, G) = M^T\{u - P_\Omega[u - G^{-1}(Mu + q)]\} + (Mu + q). \tag{60}$$

Note that, for any $v \in R^l$,

$$P_\Omega(G^{-1/2}v) = G^{-1/2}P_\Omega(v),$$

and if $u \in \Omega$, so is $G^{1/2}u$. Then using Lemma 1 we have, for any $u \in \Omega$,

$$\begin{aligned}\tilde{\varphi}(u, G) &= \{G^{1/2}u - P_\Omega[G^{1/2}u - G^{-1/2}(Mu + q)]\}^T[G^{-1/2}(Mu + q)]\\ &\geq \|G^{1/2}u - P_\Omega[G^{1/2}u - G^{-1/2}(Mu + q)]\|^2\\ &= \|u - P_\Omega[u - G^{-1}(Mu + q)]\|_G^2,\end{aligned}$$

where $\|\cdot\|_G$ denotes a norm in R^l induced by G. Similarly as in Theorem 2, we have

$$\begin{aligned}(u - u^*)^T\tilde{g}(u, G) &= \{u - P_\Omega[u - G^{-1}(Mu + q)]\}^T[(Mu + q) - (Mu^* + q)]\\ &\quad + (u - u^*)^T(Mu + q)\\ &\geq \tilde{\varphi}(u) - u^T(Mu^* + q) + (u - u^*)^T(Mu + q)\\ &= \tilde{\varphi}(u) + (u - u^*)^T M(u - u^*)\\ &\geq \tilde{\varphi}(u, G).\end{aligned}$$

Instead of taking $\varphi(u)$ and $g(u)$ as in Algorithm PC, we now use $\tilde{\varphi}(u, G)$ and $\tilde{g}(u, G)$ and refer to the related method as a scaled projection and contraction method (the SPC algorithm). Then, as indicated in Theorem 3, the sequence $\{u^{(k)}\}$ satisfies

$$\|u^{(k+1)} - u^*\|^2 \leq \|u^{(k)} - u^*\|^2 - \frac{\tilde{\varphi}^2(u^{(k)}, G)}{\|\tilde{g}_B(u^{(k)}, G)\|^2}.$$

Moreover, the scaling matrix G may vary with the iteration index, i.e., $G = G_k$. With minor modifications, the results of this paper are true for variable G_k as well as for fixed G. The best choice of the matrices G_k is the topic of further research.

---

[1] In fact, the results of this section are true for more general matrices G of the form

$$G = \begin{pmatrix} G_1 & \\ & G_2 \end{pmatrix},$$

where $G_1, G_2 > 0$, G_1 is diagonal, and G_1 acts on the indices $i \in I$ and G_2 on $L \backslash I$.

Another modification of the PC method is the following: For any $l \times l$ diagonal positive definite matrix Q, let

$$\bar{g}(u, G, Q) = Q^{-1}\tilde{g}(u, G),$$

$$\bar{\rho}(u) = \frac{\tilde{\varphi}(u, G)}{\|\bar{g}_B(u, G, Q)\|_Q^2}.$$

$$\tilde{u} = P_\Omega[u - \bar{\rho}(u)\bar{g}_B(u, G, Q)].$$

Since

$$(u - u^*)^T Q\bar{g}_B(u, G, Q) = (u - u^*)^T\tilde{g}_B(u, G) \geq \tilde{\varphi}(u, G),$$

then

$$\begin{aligned}\|\tilde{u} - u^*\|_Q^2 &\leq \|u - u^* - \bar{\rho}(u)\bar{g}_B(u, G, Q)\|_Q^2 \\ &= \|Q^{1/2}(u - u^*) - \bar{\rho}(u)Q^{1/2}\bar{g}_B(u, G, Q)\|^2 \\ &= \|u - u^*\|_Q^2 - 2\bar{\rho}(u)(u - u^*)^T\tilde{g}_B(u, G) + \bar{\rho}^2(u)\|\bar{g}_B(u, G, Q)\|_Q^2 \\ &\leq \|u - u^*\|_Q^2 - \frac{\tilde{\varphi}^2(u, G)}{\|\bar{g}_B(u, G, Q)\|_Q^2}.\end{aligned}$$

We call the above method a scaled metric projection and contraction method. Again, the best choice of Q is the topic of further research.

8. Numerical Results

In this section we present some numerical experiments with the PC algorithms (with $G = Q = I$). We consider the Dirichlet Problem with obstacles [24], which can be set in a complementary form:

Given continuous functions f, l, and h on $D = [0, 1] \times [0, 1]$ and w on ∂D, find u: $[0, 1] \times [0, 1] \to R$ such that

$$\begin{aligned}&\max\{u(p) - h(p), \min\{u(p) - l(p), -\Delta u(p) - f(p)\} \mid p \in (0, 1) \times (0, 1)\} = 0, \\ &u(p) = w(p), \qquad p \in \partial D.\end{aligned} \tag{61}$$

To solve (61) by means of a difference method, we put

$$x_i = i\tau, \qquad y_j = j\tau, \qquad i, j = 0, 1, \ldots, N + 1,$$

$$\tau = \frac{1}{N + 1}, \qquad N \geq 1 \quad \text{an integer}$$

$$u_{ij} = u(x_i, y_j),$$

which leads to a linear complementarity problem of the type (5)

$$u = P_\Omega[u - (Mu + q)], \tag{62}$$

where $\Omega = \{u \mid l_{ij} \leq u_{ij} \leq h_{ij},\ i, j = 1, \ldots, N\}$ is defined by box constraints and M

is a block $N^2 \times N^2$ matrix of the form

$$M = \begin{pmatrix} B & -I & 0 & \cdots & \cdots & \cdots & 0 \\ -I & B & -I & \ddots & & & \vdots \\ 0 & -I & B & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & & \ddots & \ddots & \ddots & -I \\ 0 & \cdots & \cdots & \cdots & 0 & -I & B \end{pmatrix}, \tag{63}$$

and B is an $N \times N$ matrix of the following form:

$$B = \begin{pmatrix} 4 & -1 & 0 & \cdots & \cdots & \cdots & 0 \\ -1 & 4 & -1 & \ddots & & & \vdots \\ 0 & -1 & 4 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & & \ddots & \ddots & \ddots & -1 \\ 0 & \cdots & \cdots & \cdots & 0 & -1 & 4 \end{pmatrix}. \tag{64}$$

It is well known that M is positive definite. To investigate the convergence behaviour of Algorithm PC, we form random test problems (62) with matrix (63) and (64) as follows: First, we set $l_{ij} = 0$, and choose h_{ij} randomly in (10, 20), and t_{ij} randomly in (0, 1). Then let

$$u^*_{ij} = \begin{cases} 0 & \text{if } \; t_{ij} \leq 0.25, \\ h_{ij} \times (2t_{ij} - 0.5) & \text{if } \; 0.25 < t_{ij} < 0.75, \\ h_{ij} & \text{otherwise,} \end{cases}$$

$$v_{ij} = \begin{cases} \text{randomly in } (0, 10) & \text{if } \; t_{ij} \leq 0.25, \\ 0 & \text{if } \; 0.25 < t_{ij} < 0.75, \\ \text{randomly in } (-10, 0) & \text{otherwise,} \end{cases}$$

and

$$q := v - Mu^*.$$

In this way we get a random test problem with a given solution u^*.

The code was written in FORTRAN and run on a VAX-8810 computer of the Computing Center of the University of Würzburg. We stop the iteration as soon as

$$\frac{\|u - P_\Omega[u - (Mu + q)]\|_\infty}{\|q\|_\infty} \leq \varepsilon$$

for some $\varepsilon > 0$. The numerical results are given in Tables 1 and 2. Here, $n = N^2$ is the number of variables. The CPU time and the number $\|u - u^*\|_\infty$ refers only to the runs with the highest accuracy $\varepsilon = 10^{-7}$. We used the following starting points: $u^{(0)}_{ij} = 0$ and $u^{(0)}_{ij} = h_{ij}/2$ for the problems in Table 1 and Table 2, respectively.

Table 1

N	n	Number of iteration for $\varepsilon =$ E − 3	E − 5	E − 7	CPU (seconds)	$\|u - u^*\|_\infty$
10	100	40	85	130	0.25	0.48E − 5
20	400	60	85	125	0.98	0.48E − 5
30	900	60	80	120	2.13	0.38E − 5
40	1600	45	85	135	4.41	0.67E − 5
50	2500	55	90	165	9.16	0.79E − 5
60	3600	50	90	135	10.49	0.67E − 5
70	4900	60	95	160	17.64	0.91E − 5
80	6400	55	95	155	22.88	0.83E − 5

Table 2

N	n	Number of iteration for $\varepsilon =$ E − 3	E − 5	E − 7	CPU (seconds)	$\|u - u^*\|_\infty$
10	100	40	75	115	0.23	0.48E − 5
20	400	45	75	115	0.93	0.48E − 5
30	900	50	70	110	2.06	0.67E − 5
40	1600	45	95	150	4.67	0.67E − 5
50	2500	55	95	175	9.19	0.72E − 5
60	3600	40	70	120	9.46	0.74E − 5
70	4900	55	95	165	17.07	0.62E − 5
80	6400	50	90	145	21.10	0.11E − 4

9. Conclusions

This paper decribes a new projection and contraction method for solving a class of generalized linear complementarity problems, which may be used to solve general convex quadratic programs. The related fundamental theory can be found in [8], [29], [19], and [34]. The main advantages of the new method are its simplicity and robustness. The numerical experiments in Section 8 show that the algorithm is efficient in some important practical applications. For an iterative method, however, we cannot expect it to be faster than other direct methods when solving small problems. We believe that a scaled projection and contraction method may also be a suitable method for general nonlinear programs and nonlinear complementarity problems: if these problems are treated by the SQP method, and, e.g., by the inexact Newton method [32], respectively, then the resulting quadratic programs, respectively, the linear complementarity problems, could be solved by Algorithm PC iteratively.

References

1. Ahn, B. H. (1981), Solution of nonsymmetric linear complementarity problem by iterative methods, Journal of Optimization Theory and Applications, 33:175–185.
2. Beale, E. M. L. (1959), On quadratic programming, Naval Research Logistics Quarterly, 6:227–243.
3. Bertsekas, D. P. (1976), On the Goldstein–Levitin–Polyak gradient projection method, IEEE Transactions on Automatic Control, 21:174–184.
4. Best, M. J. (1984), Equivalence of some quadratic programming algorithm, Mathematical Programming, 30:71–87.
5. Blum, E., Oettli, W. (1975), Mathematische Optimierung. Econometrics and Operations Research, Vol. XX. Springer-Verlag, Berlin.
6. Calamai, P. H., Moré, J. J. (1987), Projected gradient method for linearly constrained problems, Mathematical Programming, 39:93–116.
7. Cottle, R. W., Dantzig, G. B. (1968), Complementarity pivot theory of mathematical programming, Linear Algebra and Its Applications, 1:103–125.
8. Dantzig, G. B. (1963), Linear Programming and Extensions. Princeton University Press, Princeton, NJ.
9. Demyanov, V. F., Pevnyi, A. B. (1972), Numerical methods for finding saddle points, USSR Computational Mathematics and Mathematical Physics, 12:11–52.
10. Dunn, J. C. (1980), Global and asymptotic convergence rate estimates for a class of projected gradient processes, SIAM Journal Control and Optimization, 19:368–400.
11. Dunn, J. C. (1987), On the convergence of projected gradient processes to singular critical points, Journal of Optimization Theory and Applications, 55:203–216.
12 Dunn, J. C. (1988), A projected Newton method for minimization problems with nonlinear inequality constraints, Numerische Mathematik, 53:377–409.
13. Fletcher, R. (1971), A general quadratic programming algorithm, Journal of the Institute of Mathematics and Applications, 7:76–91.
14. Fletcher, R. (1981), Practical Methods of Optimization, Vol. 2. Wiley, New York.
15. Gill, P. E., Murray, W. (1978), Numerically stable methods for quadratic programming, Mathematical Programming, 14:343–379.
16. Goldfarb, D. (1972), Extensions of Newton's method and simplex methods for solving quadratic programs. In: Numerical Methods for Nonlinear Optimization (F. A. Lootsma, ed.). Academic Press, London.
17. Goldfarb, D., Idnani, A. (1983), A numerically stable dual method for solving strictly convex quadratic programs, Mathematical Programming, 27:1–33.
18. Goldstein, A. A. (1964), Convex programming in Hilbert space, Bulletin of the American Mathematical Society, 70:709–710.
19. Gubin, L. G., Polyak, B. T., Raik, E. V. (1967), The method of projections for finding the common point of convex sets, USSR Computational Mathematics and Mathematical Physics, 7:1–24.
20. Han, S. P. (1976), Superlinear convergent variable metric algorithms for general nonlinear programming, Mathematical Programming, 11:263–282.
21. He, B. (1989), A saddle point algorithm for linear programming, Nanjing Daxue Xuebao, Shuxue Banniankan, 6:41–48.
22. He, B. (1990), A globally and linearly convergent saddle point algorithm for linear programming, Numerical Mathematics, A Journal of Chinese Universities (to appear).
23. Karmarkar, N. (1984), A new polynomial-time algorithm for linear programming, Combinatorica, 4:373–395.
24. Kinderlehrer, D., Stampacchia, G. (1980), An Introduction to Variational Inequalities and Their Applications. Academic Press, New York.
25. Kojima, M., Mizuno, S., Yoshise, A. (1989), A polynomial-time algorithm for a class of linear complementarity problem, Mathematical Programming, 44:1–26.
26. Korpelevich, G. M. (1976), Ekstragradientnyi method dlia otyskaniia sedlovykh tochek i drugikh zadach, Ekonomika i matematicheski metody, 12:747–756.

27. ·Lemke, C. E. (1965), Bimatrix equilibrium points and mathematical programming, Management Science, 11:681–689.
28. Levitin, E. S., Polyak, B. T. (1966), Constrained minimization problems, USSR Computational Mathematics and Mathematical Physics, 6:1–50.
29. Mangasarian, O. L. (1979), Solution of symmetric linear complementarity problems by iterative methods, Journal of Optimization Theory and Applications, 22:465–485.
30. Pang, J. S. (1981), An equivalence between two algorithms for quadratic programming, Mathematical Programming, 20:152–165.
31. Pang, J. S. (1984), Necessary and sufficient conditions for the convergence of iterative methods for the linear complementarity problem, Journal of Optimization Theory and Applications, 42:1–17.
32. Pang, J. S. (1986), Inexact Newton methods for the nonlinear complementarity problem, Mathematical Programming, 36:54–71.
33. Powell, M. J. D. (1978), A fast algorithm for nonlinearly constrained optimization calculation. In: Numerical Analysis, Dundee, 1977. Lecture Notes in Mathematics, Vol. 630, pp. 144–157. Springer-Verlag, Berlin.
34. Uzawa, H. (1958), Iterative methods for concave programming. In: Studies in Linear and Nonlinear Programming (K. J. Arrow, L. Hurwicz, and H. Uzawa, eds.). Stanford University Press, Palo Alto, CA.
35. Wilson, R. B. (1963), A simplicial algorithm for concave programming. Ph.D. Dissertation, Graduate School of Business Administration, Harvard University, Boston, MA.
36. Wolfe, P. (1959), The simplex method for quadratic programming, Econometrica, 27:382–398.

Accepted 26 February 1991