Volume 29, Number 2, 2004, pp. 119-133

COUPLING THE BANACH CONTRACTION MAPPING PRINCIPLE AND THE PROXIMAL POINT ALGORITHM FOR SOLVING MONOTONE VARIATIONAL INEQUALITES

PHAM NGOC ANH AND LE DUNG MUU

Abstract. In our recent papers [1, 2] we have shown how to find a regularization parameter such that the unique solution of a strongly monotone variational inequality can be approximated by the Banach contraction mapping principle. In this paper we combine this result with the proximal point algorithm to obtain a new projection-type algorithm for solving (not necessarily strongly) monotone variational inequalities. The proposed algorithm does not require knowing any Lipschitz constant of the cost operator. The main subproblem in the proposed algorithm is of computing the projection of a point onto a closed convex set. Application of the proposed algorithm to an equilibrium problem is discussed. Computational results are reported.

1. Introduction

Let H be a real Hilbert space with inner product $\langle .,. \rangle$ and the corresponding norm $||.||$. Let $C \subseteq H$ be a nonempty closed convex set and $F : C \to H$ be a monotone operator. We consider the following variational inequality:

$$\text{(VI)} \qquad \begin{cases} \text{find } \; x^* \in C & \text{such that} \\ \langle F(x^*), x - x^* \rangle \geq 0 & \text{for every } \; x \in C. \end{cases}$$

The mapping F is called the cost operator for (VI). It is well known that if F is the gradient mapping of a convex function f, then x^* is a solution of (VI) if and only if it is an optimal solution of the convex minimization problem

$$\min\{f(x) | x \in C\}.$$

Various iterative methods have been proposed for solving variational inequalities (see e.g. [5, 8, 13, 14, 16, 17, 19, 26] and the references therein). A general scheme for solving variational inequalities is the auxiliary problem-principle which contains the projection method as a special case. In order to guarantee the convergence, this general scheme needs some additional assumptions such as strong monotonicity or cocoercivity (see e.g. [1, 4, 10, 22, 18, 25, 24]). Cohen [3] gave an

Received August 11, 2003; in revised form May 18, 2004.
1991 *Mathematics Subject Classification.* 90 C29.
Key words and phrases. Monotone variational inequality, proximal point algorithm, Banach contraction mapping principle, equilibrium problem, computational results.
This work was support in part by the National Research Program, Grant C12.

example involving a monotone mapping F where the auxiliary problem-principle algorithm does not converge.

The proximal point algorithm [20] is a fundamental iterative procedure for solving the inclusion $0 \in T(x)$ with T being a maximal monotone operator. This inclusion problem contains monotone variational inequality as a special case. In the case of variational inequality (VI), the proximal point algorithm consists of constructing iteratively a sequence $\{x^k\}$ by setting $x^{k+1} := P_k(x^k)$ where P_k is called proximal or resolvent operator defined as $P_k := (I + c_k T)^{-1}$, where $c_k > 0$, I is the identity operator and

$$T(x) = F(x) + N_C(x).$$

Unlike many other iterative methods, for the convergence, besides continuity and monotonicity, the proximal point algorithm does not require any additional assumption on F. However, from a view point of implementation, computing iteration points, in general, is difficult, since it requires evaluating the proximal operator $(I + c_k T)^{-1}$ at each iteration point x^k. It is well known (see e.g. [8]) that computing each iteration point x^k amounts to solving a certain strongly monotone variational inequality.

In our recent papers [1, 2] we have shown how to find a regularization parameter such that the unique solution of a strongly monotone variational inequality with single valued cost operator can be computed by finding the fixed point of a certain contractive mapping. The result in [1] is extended in [2] to the mixed variational inequalities involving cocoercive cost operators and convex functions.

In this paper we continue our work in [1, 2] by coupling the Banach contraction mapping principle with the proximal point algorithm to obtain a projection-type algorithm. It turns out that the proposed algorithm belongs to the class of the modified projection methods. By using the regularization technique, our algorithm, unlike other modified projection algorithms, does not require knowing any Lipschitz constant of the cost operator. It also gives a new analysis to the modified projection method. Moreover, by using the Banach contraction mapping fixed point principle, the convergence theorem is easy to obtain. As in the projection method, the main subproblem in the proposed algorithm is of finding the projection of a point on the feasible set C. In some special cases for instance C is a box, a ball or a simplex, this projection can be given explicitely.

The paper will be organized as follows. The next section contains some preliminaries on the proximal point algorithm applied to monotone variational inequalities. In the third section we describe in detail two algorithms. In the fourth section we illustrate the coupling algorithm by an equilibrium problem. We close the paper with some preliminary computational results on this equilibrium model.

2. Preliminaries

Let $\emptyset \neq A \subseteq H$ and $u \in H$. As usual, the distance from u to A is defined as $d_A(u) = \inf_{x \in A} ||x - u||$. It is well known (see e.g. [7]) that if A is closed and convex, then there exists the unique $v \in A$ such that $d_A(u) = ||v - u||$. We say that v is the projection of u on A.

Let $T : H \to 2^H$ be a multivalued mapping. As usual, we denote the effective domain and the graph of T by dom$T$ and grap$T$, respectively. That is

$$\text{dom}T := \{x | T(x) \neq \emptyset\},$$
$$\text{grap}T := \{(x, y) | y \in T(x)\}.$$

We recall (see e.g. [12, 19]) that the mapping T is said to be *monotone* on C if

$$\langle z - z', x - x' \rangle \geq 0 \quad \forall x, x' \in C, z \in T(x), z' \in T(x').$$

T is said to be *maximal monotone* if it is monotone on H and its graph is not contained properly in the graph of any other monotone mapping.

T is called *strongly monotone* on $C \subseteq H$ with modulus $\beta > 0$ (briefly β-strongly monotone) if

$$\langle z - z', x - x' \rangle \geq \beta ||x - x'||^2 \quad \forall x, x' \in C, z \in T(x), z' \in T(x').$$

A mapping (operator) $M : C \to H$ is said to be *Lipschitz continuous* on C with Lipschitz constant $L \geq 0$ (shortly L-Lipschitz) if

$$||M(x) - M(x')|| \leq L||x - x'|| \quad \forall x, x' \in C. \tag{2.1}$$

If (2.1) is satisfied with $L < 1$, then M is said to be *contractive* on C; it is said to be *nonexpansive* on C if $L = 1$.

The proximal point algorithm [20] can be used to solve an inclusion $0 \in T(u)$, where T may be any maximal multivalued monotone mapping. This algorithm is based on the fact that if T is maximal monotone, then, for any $c > 0$, the proximal operator $P_c := (I + cT)^{-1}$ is defined everywhere, single valued and nonexpansive on the whole space [12]. It is easy to see that $0 \in T(x)$ if and only if $P_c(x) = x$. So the underlying inclusion is reduced to the problem of finding a fixed point of the nonexpansive mapping P_c. To do this, starting from an arbitrary point x^0, the algorithm constructs iteratively a sequence $\{x^k\}$ by setting

$$x^{k+1} = P_k(x^k) \quad k = 0, 1...,$$

where $P_k = (I + c_k T)^{-1}$ and $\{c_k\}$ is a sequence of positive numbers being chosen in advance.

The key question in this algorithm is of evaluating x^{k+1}. In general, it is a difficult task, since it requires computing the inverse mapping $(I + c_k T)^{-1}$. In practice, in stead of computing exact $P_k(x^k)$ one can compute its approximation. In [20] (see also [13]) it has been shown, among others, that if $||x^{k+1} - P_k(x^k)|| \leq \varepsilon_k$ for all k with $\sum_{k=1}^{\infty} \varepsilon_k < +\infty$, then the sequence $\{x^k\}$ weakly converges to a solution of the inclusion $0 \in T(x)$. Moreover the sequence $\{x^k\}$ is asymptotically regular, that is $||x^{k+1} - x^k|| \to 0$ as $k \to +\infty$.

In order to apply the proximal point algorithm to the variational inequality (VI) we take

$$T(x) := F(x) + N_C(x),$$

where $N_C(x)$ denotes the outward normal cone of C at x, i.e.,

$$N_C(x) = \{w | \langle w, y - x\rangle \le 0 \ \ \forall y \in C\}.$$

It has been shown in [19, 20] that if F is continuous and monotone on C, then T defined above is maximal monotone. In this case, starting from a point $x^0 \in C$ the proximal point algorithm constructs a sequence $\{x^k\}$ by setting $x^{k+1} = (I + c_kT)^{-1}(x^k)$. Hence $x^k \in (I + c_kT)(x^{k+1})$. Replacing $T(x^{k+1})$ by $F(x^{k+1}) + N_C(x^{k+1})$ we obtain the inclusion

$$x^k - x^{k+1} - c_kF(x^{k+1}) \in N_C(x^{k+1})$$

which means that

$$x^{k+1} \in C, \ \langle x^{k+1} + c_kF(x^{k+1}) - x^k, x - x^{k+1}\rangle \ge 0 \ \ \forall x \in C.$$

Setting $F_k(x) := x + c_kF(x) - x^k$ we see that x^{k+1} is the solution of the variational inequality

$$\langle F_k(x^{k+1}), x - x^{k+1}\rangle \ge 0 \ \ \forall x \in C.$$

Clearly, if F is monotone, then F_k is strongly monotone with the modulus $\beta = 1$, and if F is L-Lipschitz, then F_k is L_k-Lipschitz with $L_k = 1 + c_kL$.

The proximal point algorithm for (VI) then can be described as follows.

Take $x^0 \in C$ and fix a sequence of positive number $\{c_k\}$ such that $c_k > c > 0$ for every k.

For each $k = 0, 1, ...$, let x^{k+1} be the unique solution of the following strongly monotone variational inequality

$$(VI_k) \qquad \begin{cases} \text{find } \ x^{k+1} \in C \ \text{ such that} \\ \langle c_kF(x^{k+1}) + (x^{k+1} - x^k), x - x^{k+1}\rangle \ge 0 \ \ \forall x \in C. \end{cases}$$

Clearly, if $x^{k+1} = x^k$, then x^k solves (VI). Otherwise, if the algorithm does not terminate, then it has been proved (see e.g. [20]) that $||x^{k+1} - x^k|| \to 0$ as $k \to +\infty$. Moreover the sequence $\{x^k\}$ generated by this algorithm is bounded and weakly convergent to a solution of Problem (VI) whenever it admits a solution.

3. Description of the algorithms

Following Fukushima in [4], for each $x \in C$, we denote by $h(x)$ the unique solution of the strongly convex quadratic problem

$$(P(x)) \qquad \min\Big\{\frac{1}{2}\alpha||y - x||^2 + \langle F(x), y - x\rangle \ |y \in C\Big\},$$

where $\alpha > 0$. As usual, we shall refer to α as a regularization parameter. Since, for each $x \in C$, Problem $(P(x))$ uniquely solvable, h is a single valued mapping on C. It has been shown (see e.g. [4, 22]) that $x^* \in C$ is a solution of (VI) if and only if $h(x^*) = x^*$.

The next proposition says that if F is strongly monotone and Lipschitz on C, then one can choose regularization parameters such that h is contractive on C.

Proposition 3.1. *Suppose that the cost operator F is strongly monotone on C with the modulus β and Lipschitz on C with the constant L. Then h is contractive on C with the modulus*

$$\delta = \sqrt{1 - \frac{2\beta}{\alpha} + \frac{L^2}{\alpha^2}} \tag{3.1}$$

whenever $\alpha > \dfrac{L^2}{2\beta}$.

The proof of this proposition can be found in [1]. Since the reference [1] is unpublished, we give here a direct proof which is simpler than the original one in [1].

Proof. From the definition of h, it follows that

$$h(u) = Pr_C(u - \frac{1}{\alpha}F(u))$$

where $Pr_C(v)$ denotes the projection of v onto C.

Since the projection is nonexpansive, we have

$$\begin{aligned}||h(u) - h(u')||^2 &\le ||u - \frac{1}{\alpha}F(x) - (u' - \frac{1}{\alpha}F(u'))||^2\\ &= ||u - u'||^2 - \frac{2}{\alpha}\langle u - u', F(u) - F(u')\rangle + \frac{1}{\alpha^2}||F(u) - F(u')||^2.\end{aligned}$$

Since F is β-strongly monotone and L-Lipschitz continuous on C, we have

$$\langle u - u', F(u) - F(u')\rangle \ge \beta||u - u'||^2$$

and

$$||F(u) - F(u')||^2 \le L||u - u'||^2.$$

Then it follows that

$$\begin{aligned}||h(u) - h(u')||^2 &\le ||u - u'||^2 + \frac{L^2}{\alpha^2}||u - u'||^2 - \frac{2\beta}{\alpha}||u - u'||^2\\ &= \Big(1 + \frac{L^2}{\alpha^2} - \frac{2\beta}{\alpha}\Big)||u - u'||^2.\end{aligned}$$

Hence h is contractive on C whenever $\alpha > \dfrac{L^2}{2\beta}$. □

Using Proposition 3.1 we can describe an algorithm for solving strongly monotone variational inequalities based upon the Banach contraction mapping principle.

ALGORITHM 1

Initial step. Choose $\alpha > \dfrac{L^2}{2\beta}$ and a tolerance $\varepsilon \ge 0$.

Step 0. Take $u^0 \in C$. Set $k = 0$.

Step 1. Solve the strongly convex quadratic program

$$(P(u^k)) \qquad \min\left\{\frac{1}{2}\alpha||z-u^k||^2 + \langle F(u^k), z-u^k\rangle | z \in C\right\}$$

to obtain its unique solution u^{k+1}.

If $\frac{\delta^{k+1}}{1-\delta}||u^1-u^0|| \leq \varepsilon$, terminate the algorithm: u^{k+1} is an ε-solution to (VI). Otherwise, let $k \leftarrow k+1$ and return to Step 1.

Let u^* denote the unique fixed point of h. Since $u^{k+1} = h(u^k)$, we have

$$||u^{k+1}-u^*|| \leq \frac{\delta^{k+1}}{1-\delta}||u^0-u^1||,$$

where δ is the contraction coefficient of h.

Thus, if the algorithm terminates at some iteration k, then $||u^{k+1}-u^*|| \leq \varepsilon$. Hence u^{k+1} is an ε-solution to (VI). In the case $\varepsilon = 0$, the algorithm may never terminate. However the sequence $\{u^k\}$ generated by the algorithm strongly converges to the unique fixed point u^* of h, and, by the Banach contraction mapping principle, we have the following estimation

$$||u^{k+1}-u^*|| \leq \frac{\delta^{k+1}}{1-\delta}||u^0-u^1||.$$

Remark 3.1. From (3.1) we see that the contraction coefficient δ is a function of the regularization parameter α. An elementary computation from (3.1) shows that δ takes its minimum when $\alpha = \frac{L^2}{\beta}$. Therefore for the convergence, in Algorithm 1 the best way is to choose $\alpha = \frac{L^2}{\beta}$.

Remark 3.2. In the case when the modulus β and the Lipschitz constant L of F are not known in advance, we can justify the regularization parameters α as follows.

At the start, we run the algorithm with $\alpha = \frac{L_0^2}{\beta_0}$, where $L_0 > \beta_0 > 0$, to obtain $u^1 = h(u^0)$ (L_0, β_0 can be considered as approximate values of L and β respectively).

If $||u^1-u^0|| \leq \varepsilon$, then the algorithm has been terminated yielding an ε- fixed point of h. Otherwise, if $||u^1-u^0|| > \varepsilon > 0$, since $||u^{j+1}-u^j|| \leq \delta^j||u^1-u^0||$ for all j, it follows that after j-iteration with

$$j \geq \frac{\log(\frac{\varepsilon}{||u^1-u^0||})}{\log\delta},$$

we must have $||u^{j+1} - u^j|| \leq \varepsilon$. Then we can terminate the algorithm. Otherwise, we increase α by a positive number, for example by one or two, and repeat the procedure. By this way we can avoid computing L and β. However this step may make the algorithm slow. As it will be seen in the next section, by coupling Algorithm 1 with the proximal point algorithm one can avoid completely knowing any Lipschitz constant of the cost operator.

Now we turn to the case where F in (VI) may be any continuous and monotone (not necessarily strongly) on C.

For each variational inequality (VI$_k$) we consider the following strongly convex quadratic program:

$$\min\Big\{\langle F_k(u), y - u\rangle + \frac{\alpha_k}{2}||y - u||^2 \ |y \in C\Big\}, \tag{3.2}$$

where $\alpha_k > 0$. Since C is closed convex and the objective function is strongly convex, this problem uniquely solvable for any u in the domain of F. Let $h_k(u)$ denote the unique solution of Problem (3.2). Then h_k is a mapping from dom$F$ to C.

Note that (see [4]) u^k is the solution to the variational inequality (VI$_k$) if and only if $u^k \in C$ and $h_k(u^k) = u^k$.

The unique solution of (VI$_k$) can be computed by finding the unique fixed point of the contractive mapping h_k by using Proposition 3.1. More precisely we have the following result.

Proposition 3.2. *If F is monotone and L-Lipschitz continuous on C, then h_k is contractive on C with the coefficient* $\delta_k := \sqrt{1 - \dfrac{2}{\alpha_k} + \dfrac{L_k^2}{{\alpha_k}^2}}$ *whenever* $\alpha_k > \dfrac{L_k^2}{2}$, *where $L_k = 1 + c_k L$ is the Lipschitz constant of F_k.*

Proof. Apply Proposition 3.1 to the mapping F_k noting that F_k is strongly monotone on C with the modulus $\beta = 1$ and Lipschitz continuous on C with the Lipschitz constant $L_k = 1 + c_k L$. □

In virtue of Proposition 3.2, the Banach contraction mapping principle can be applied to Subproblem (VI$_k$) by using Algorithm 1. Namely, we construct the sequence $\{u^{k,j}\}$ by setting

$$u^{k,j+1} := h_k(u^{k,j}) \ (j = 0, 1, ...)$$

where $u^{k,0} \in C$ has been chosen in advance.

Note that, at each iteration j, evaluating $h_k(u^{k,j})$ leads to solving the following strongly convex quadratic programming problem

$$\min\Big\{\frac{1}{2}\alpha_k||u - u^{k,j}||^2 + \langle F_k(u^{k,j}), u - u^{k,j}\rangle \ |u \in C\Big\}.$$

Since u^k is the fixed point of the contractive mapping h_k, by the Banach contraction mapping fixed point principle, we have

$$||u^{k,j+1} - u^k|| \leq \frac{\delta_k^{j+1}}{1-\delta_k}||u^{k,0} - u^{k,1}||, \tag{3.3}$$

where $0 < \delta_k < 1$ is the contraction coefficient of h_k. According to Proposition 3.2 one has

$$\delta_k := \sqrt{1 - \frac{2}{\alpha_k} + \frac{L_k^2}{{\alpha_k}^2}}$$

where $L_k = 1 + c_k L$ with L being the Lipschitz constant of F. Moreover, if the original variational inequality problem (VI) admits a solution, then by the proximal point algorithm (see e.g. [20]) the sequence $\{u^k\}$ weakly converges to a solution of (VI) whenever the sequence $\{c_k\}$ is bounded away from zero which means that $c_k > c > 0$ for all k, where c may be any positive number.

In practice we solve Subproblem (VI$_k$) approximately only. Namely, we first choose a decreasing sequence $\{\varepsilon_k\}$ of positive numbers such that $\sum\limits_{k=0}^{\infty} \varepsilon_k < +\infty$. Then instead of computing the exact solution u^k of Subproblem (VI$_k$), we compute an approximate solution $u^{k,j+1}$ such that $||u^{k,j+1} - u^k|| \leq \varepsilon_k$.

Remark 3.3. By Proposition 3.2 the regularization parameter must satisfy $\alpha_k > \frac{(1+c_kL)^2}{2}$. To guarantee convergence for the proximal point algorithm the sequence $\{c_k\}$ must be bounded away from zero which means that $c_k > c > 0$ for every k. Since c can be any positive number, we can choose $c_k > 0$ small enough such that $c_k L < 1$. Then it follows from $\alpha_k = \frac{(1+c_kL)^2}{2}$ that one can choose $\alpha_k = 1$ for every k. In this case, since $\delta_k = \sqrt{1 - \frac{2}{\alpha_k} + \frac{L_k^2}{{\alpha_k}^2}}$, we have

$$\delta_k = \sqrt{1 - 2 + L_k^2} = \sqrt{1 - 2 + (1 + c_kL)^2} = \sqrt{c_k(2L + c_kL^2)}.$$

Thus, if we choose $\alpha_k = 1$ for all k, we can make δ_k small by taking c_k small, but it much be bounded away from zero. A single calculation shows that the same situation occurs when $\alpha_k = L_k^2 = (1 + c_kL)^2$.

Motivated by the fact that $x \in C$ is a solution of (VI) if and only if it is a fixed point of P_k, we agree to say that $x \in C$ is an ε-solution to (VI) if $||x - P_k(x)|| \leq \varepsilon$. The algorithm for solving (VI) with F monotone and Lipschitz continuous on C then can be described as follows.

ALGORITHM 2 (BFP Algorithm)

Choose a tolerance $\varepsilon \geq 0$ and a decreasing sequence $\{\varepsilon_k\}$ of positive numbers such that $\sum_{k=1}^{\infty} \varepsilon_k < +\infty$. Pick $x^0 \in C$ as the starting point.

Iteration k (outer iteration $k = 0, 1, ...$).

Step 0. Take $\alpha_k \geq 1$.

Pick $u^{k,0} := x^k$. Let $j := 0$.

Step 1 (inner iteration). Solve the strongly convex quadratic program

$$\min \Big\{ \frac{1}{2}\alpha_k ||u - u^{k,j}||^2 + \langle F_k(u^{k,j}), u - u^{k,j} \rangle \ |u \in C \Big\} \tag{3.4}$$

to obtain $u^{k,j+1}$.

1a) if

$$u^{k,1} = u^{k,0} \text{ or } j \geq \frac{\log \dfrac{\varepsilon_k(1-\delta_k)}{||u^{k,0} - u^{k,1}||}}{\log \delta_k} - 1,$$

then set $x^{k+1} := u^{k,j+1}$.

If $||x^{k+1} - x^k|| + \varepsilon_k \leq \varepsilon$, then terminate the algorithm: x^k is an ε-solution to (VI).

If $||x^{k+1} - x^k|| + \varepsilon_k > \varepsilon$, then increase k by 1 and go to iteration k.

1b) If

$$u^{k,1} \neq u^{k,0} \text{ and } \ j < \frac{\log \dfrac{\varepsilon_k(1-\delta_k)}{||u^{k,0} - u^{k,1}||}}{\log \delta_k} - 1,$$

then let $j := j + 1$ and go to Step 1.

Remark 3.4. By the proximal point algorithm, $||x^{k+1} - x^k|| \to 0$ as $k \to +\infty$ whenever (VI) admits a solution. Hence, if (VI) is solvable, then the algorithm terminates after a finite iteration whenever $\varepsilon > 0$, because $\varepsilon_k \to 0$ as $k \to +\infty$.

Remark 3.5. The main subproblem in the above algorithm is Problem (3.4). This is a strongly convex quadratic program that can be solved efficiently by the existing codes. Note that the problem (3.4) can be rewritten equivalently as

$$\min \Big\{ ||u - (u^{k,j} - \frac{1}{\alpha_k} F_k(u^{k,j}))||^2 \ | \ u \in C \Big\},$$

which in turn is the problem of finding the projection of $u^{k,j} - \dfrac{1}{\alpha_k} F_k(u^{k,j})$ on C. In some special cases of C such as when C being a box, a simplex or a ball (often occur in applications), the projection of a point on C can be given explicitly.

Convergence. Suppose that (VI) is solvable. First we show that if the algorithm terminates at iteration k, then x^k is in fact an ε-solution to (VI). Indeed, by construction of x^{k+1}, we have

$$||x^{k+1} - P_k(x^k)|| \leq \varepsilon_k \quad \forall k \geq 1.$$

Since

$$||P_k(x^k) - x^k|| \leq ||x^{k+1} - x^k|| + ||P_k(x^k) - x^{k+1}|| \leq \varepsilon - \varepsilon_k + \varepsilon_k = \varepsilon,$$

x^k is an ε-solution of (VI). When $\varepsilon = 0$ the algorithm may never terminate. However, the convergence of the algorithm is guaranteed by the convergence of the proximal point algorithm and the Banach contraction mapping principle. Indeed, suppose that u^k is the exact solution of Subproblem (VI_k), by the proximal point algorithm the sequence $\{u^k\}$ converges weakly to a solution u^* of variational inequality (VI). For any $w \in H$ we have

$$\begin{aligned}\langle w, x^k\rangle - \langle w, x^*\rangle &= \langle w, x^k - u^k\rangle + \langle w, u^k - x^*\rangle \\ &\leq ||w||\ ||u^k - x^k|| + \langle w, u^k - x^*\rangle \\ &\leq \varepsilon_k ||w|| + \langle w, u^k - x^*\rangle.\end{aligned}$$

Since $\varepsilon_k \to 0$ and u^k converges weakly to u^*, it follows that the sequence $\{x^k\}$ weakly converges to x^*.

Remark 3.6. Of course instead of Algorithm 1, we can use Algorithm 2 for solving variational inequalities with strongly monotone cost operators. It is helpful when a Lipschitz constant of the cost operator is difficult to estimate.

Comparison with the Modified Projection Method

As we have mentioned in the introduction part, in order to guarantee the convergence of the projection method, one needs strict monotonicity or cocoercivity of the cost operator F. To avoid this additional condition, the projection method needs some modifications. A well known modified projection method, called extragradient method (see e.g. [9, 15]) for solving variational inequalities involving monotone and L-Lipschitz cost operators can be briefly described as follows.

Start with $x^0 \in C$ and select $0 < \rho < \dfrac{1}{L}$. At each iteration $k = 1, 2...$ find $\overline{x}^{k-1}$ and x^k such that

$$\overline{x}^{k-1} = P_C(x^{k-1} - \rho F(x^{k-1})),$$

and

$$x^k = P_C(x^{k-1} - \rho F(\overline{x}^{k-1})).$$

It has been proven [15] that if the cost operator F is monotone and L-Lipschitz on C, then the sequence $\{x^k\}$ generated by this algorithm weakly converges to a solution of Problem (VI).

Note that the step size ρ in this algorithm plays as the reciprocal of the regularization parameter in Algorithm 1. The difference is that ρ is independent of k while parameter α_k may vary at each iteration k.

The main advantage of Algorithm 2, as comparing to the above described modified projection algorithm, is that Algorithm 2 does not require knowing any Lipschitz constant of the cost operator.

4. An illustrative example and computational results

We illustrate Algorithm BFP by the oligopolistic market equilibrium model considered in [8] (see also [15]). Assume that there are n firms supplying a homogeneous product and that the price p depends on its quantity σ, i.e. $p = p(\sigma)$. Let $h_i(x_i)$ denote the total cost of the firm i of supplying x_i units of the product. Then the profit of firm i is $x_i p(\sigma) - h_i(x_i)$. Naturally, each firm seeks to maximize its own profit by choosing the corresponding production level. Suppose that the strategy set C is a box in R^n given by

$$C := \{x = (x_1, ..., x_n)^T | 0 \le L_i \le x_i \le U_i \ (i = 1, ..., n)\}. \tag{4.1}$$

Thus, the oligopolistic market equilibrium problem can be formulated as a Nash equilibrium noncooperative game, where the ith player has the strategy set C and the utility function

$$f_i(x_1, ..., x_n) = x_i p\Big(\sum_{j=1}^{n} x_j\Big) - h_i(x_i) \quad (i = 1, ..., n). \tag{4.2}$$

As usual, a point $x^* = (x_1^*, ..., x_n^*) \in C$ is said to be equilibrium for this problem if

$$f_i(x_1^*, ..., x_{i-1}^*, y_i, x_{i+1}^*, ..., x_n^*) \le f_i(x_1^*, ..., x_n^*) \quad \forall y_i \in [L_i, U_i] \quad \forall i = 1, ..., n. \tag{4.3}$$

Proposition 4.1. (see e.g. [8]) *A point x^* is equilibrium for the oligopolistic market problem if and only if it is a solution to* (VI), *where C is the box given by* (4.1) *and*

$$F(x) = H(x) - p(\sigma_x)e - p'(\sigma_x)x,$$

with $H(x) = (h_1'(x_1), ..., h_n'(x_n))^T$, $e = (1, ..., 1)^T \in R^n$, $\sigma_x = \langle x, e\rangle$.

Proposition 4.2. (see e.g. [8]) *Let $p : C \to R_+$ be convex, twice continously differentiable and nonincreasing and let the function $\mu_\tau : R_+ \to R_+$, defined by $\mu_\tau(\sigma) = \sigma p(\sigma + \tau)$, be concave for every $\tau \ge 0$. Also, let the functions $h_i : R_+ \to R$ $(i = 1, ..., n)$ be convex and twice continuously differentiable. Then the cost mapping*

$$F(x) = H(x) - p(\sigma_x)e - p'(\sigma_x)x,$$

is monotone on C.

Note that, since in this problem the feasible domain C is a box, the solution of Subproblem (VI$_k$) in each iteration k of Algorithm 2 is given explicitly as follows.

Suppose that the box C is given by (4.1). Let $x \in R^n$ and $y = P_C(x)$. Then it is easy to see that i-th component of y is

$$y_i = \begin{cases} L_i & \text{if } x_i \le L_i, \\ U_i & \text{if } x_i \ge U_i, \\ x_i & \text{if } L_i \le x_i \le U_i. \end{cases}$$

We have tested Algorithm 2 with the following example on a personal computer Intel 845 w, Celeron 1.7 GHZ, Ram 256 Mb.

Example 4.1. In this example

$$C := \Big\{(x_1, ..., x_n)^T | 2 - \frac{1}{i} \le x_i \le 15 + \frac{i}{3i-2}, \quad \forall i = 1, ..., n\Big\}$$

$$H(x) := (\alpha_1 x_1 + \beta_1, ..., \alpha_n x_n + \beta_n)^T,$$

$$p(t) := \frac{\xi}{t}, \quad t \in (0, +\infty),$$

where $\xi > 0$ is fixed. The numbers α_i, β_i $(i = 1, ..., n)$ and ξ are randomly generated in the interval $(0, 20)$. The tolerance $\varepsilon = 10^{-5}$. We have computed this model with several numbers of data and we have the following experiments on the algorithm.

• The choice of the tolerance sequence $\{\varepsilon_k\}$ effects very much on the efficiency of the algorithm, especially on the time for the first outer iteration.

•. To enhance the convergence, we choose the sequence $\{\varepsilon_k\}$ such that $\varepsilon_k \le \lambda\varepsilon$ for all k except several first k, where $\lambda \in (0, 1)$ (of course ε_k much be satisfied the condition in the convergence theorem). In Table 1 below we have chosen $\lambda = 0.5$ for all problems, and $\lambda = 0.99$ for those in Table 2,

Problem	k	$\overline{j_1}$	CPU-times/sec
1	9	1.4444	0.6
2	2	1.5	0.2
3	5	1.2	0.4
4	3	1.3333	0.3
5	4	1.25	0.4
6	5	1.4	0.3
7	4	1.5	0.4
8	5	1.4	0.2
9	8	1.75	0.4
10	4	1.5	0.6
11	3	1433	0.45
12	9	1.6667	0.6
13	81	1.1728	0.51
14	9	1.5556	0.4
15	2	1.5	0.2
16	3	1.6667	0.2
17	8	1.625	0.6
18	82	1.1585	0.21
19	7	1.254	0.25
20	6	1.211	0.2
	$\overline{k} = 12.95$	$\overline{j} = 1.4211$	$\overline{t} = 0.3685$

Table 1 (with $n = 100$)

Problem	k	$\overline{j_1}$	CPU-times/sec
1	196	1.7959	2.31
2	8	1.75	0.1
3	106	1.4717	1.9
4	121	1.570	1.27
5	96	1.063	0.99
6	8	2	0.11
7	151	1.7020	1.69
8	141	1.6667	1.53
9	103	1.4175	0.99
10	132	1.6364	1.43
11	91	1.3407	0.88
12	40	1.075	0.32
13	176	1.7386	2.3
14	101	1.4555	0.98
15	85	1.2824	0.81
16	123	1.5935	1.31
17	163	1.7239	1.81
18	129	1.6202	1.43
19	150	1.68	1.69
20	127	1.6142	1.42
	$\overline{k} = 112.35$	$\overline{j} = 1.5599$	$\overline{t} = 1.2635$

Table 2 (with $n = 800$)

• The coefficient $\delta_k = \sqrt{1 - \frac{2}{\alpha_k} + \frac{L_k^2}{\alpha_k^2}}$ also effects very much the number of inner iterations. Since $\alpha_k \geq \frac{(1 + c_k L)^2}{2}$, we have $c_k \leq \frac{\sqrt{2\alpha_k} - 1}{L}$. Hence if α_k is constant, then the boundedness away from zero of the sequence $\{c_k\}$ can be ensured, by taking, for example, $c_k = \frac{\sqrt{2\alpha_k} - 1}{L}$ for every k. For all tested problems we have chosen $\alpha_k = 1.1$ for every k. Note that, for all computations we do not need to chose c_k.

Some preliminary computational results are reported in the Tables 1 and 2 below. The following headings are used in the tables:

• $\overline{j}$: the average number of inner iterations.
• k: the number of outer iterations for each problem.
• $\overline{k}$: the average number of outer iterations.
• $\overline{j_1}$: the average number of the inner iterations in each outer iterations k.
• $\overline{t}$: the average of CPU times (in second) for each problem.

The results in Table 1 and Table 2 have been computed with the tolerance $\varepsilon_k = \frac{1}{k^2}$. For all computed examples, every component of the starting point is the midpoint of the corresponding edge of the rectangle C, i.e., the ith component is $\frac{L_i + U_i}{2}$ ·

From the computational experience and results we can conclude that Algorithm BFP is efficient for this equilibrium model with problems of several hundreds variables. For problems of thousand variables, the algorithm can be also used efficiently with an appropriate tolerance sequence $\{\varepsilon_k\}$.

Acknowledgment

The authors would like to thank the Associate Editor for his useful comments and remarks that helped them very much in revising the paper.

References

[1] P. N. Anh and L. D. Muu, *The Banach iterative procedure for solving monotone variational inequality*, Hanoi Institute of Mathematics, Preprint 05 (2002).
[2] P. N. Anh, L. D. Muu, V. H. Nguyen and J. J. Strodiot, *On the contraction and nonexpensiveness properties of the marginal mapping in generalized variational inequalities involving cocoercive operators*, to appear in the Proceedings on Generalized Convexity and Monotonicity Conference 7, Lecture Notes in Econ. and Math. Systems, Springer -Verlag, Berlin.
[3] G. Cohen, *Auxiliary problem principle extended to variational inequalities*, J. Optim. Theory Appl. **59** (1988), 325-333.
[4] M. Fukushima, *Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems*, Math. Program. **53** (1992), 99-110.
[5] P. T. Harker and J. S. Pang, *Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms, and applications*, Math. Program. **48** (1990), 161-220.
[6] T. T. Hue, J. J. Strodiot and V. H. Nguyen, *Convergence of the approximate auxiliary problem method of solving generalized variational inequalities*, Preprint.
[7] D. Kinderlehrer and G. Stampacchia, *An Introduction to Variational Inequalities and Their Applications*, Academic Press, 1980.
[8] I. Konnov, *Combined Relaxation Methods for Variational Inequalities*, Springer, 2001.
[9] M. Korpelevich, *The extragradient method for finding saddle points and other problems*, Ekonomicka i Matematcheskie Metody **12** (1976), 747-756.
[10] P. Marcotte, *A new algorithm for solving variational inequalities*, Math. Program. **33** (1995), 339-351.
[11] P. Marcotte and J. H. Wu, *On the convergence of projection methods: application to the decomposition of affine variational inequalities*, J. Optim. Theory Appl. **85** (1995), 347-362.
[12] G. J. Minty, *Monotone operator in Hilbert space*, Duke Math J. **29** (1962), 341-346.
[13] L. D. Muu and D. B. Khang, *Asymptotic regularity and the strong convergence of the proximal point algorithm*, Acta. Math. Vietnam. **8** (1983), 3-11.
[14] L. D. Muu, *An augmented penalty function method for solving a class of variational inequalities*, Soviet. Comput. Math. Physics **12** (1986), 1788-1796.
[15] A. Nagurney, *Network Economics: a Variational Inequality Approach*, Kluwer Academic Publishers, 1993.
[16] M. A. Noor, *Iterative schemes for quasimonotone mixed variational inequalities*, Optimization **50** (2001), 29-44.
[17] M. Patriksson, *Nonlinear Programming and Variational Inequality Problems: A Unified Approach*, Kluwer, Dordrecht, 1999.

[18] M. Patriksson, *Merit function and descent algorithms for a class of variational inequality problems*, Optimization **41** (1997), 37-55.
[19] R. T. Rockafellar, *On the maximality of sums of nonlinear monotone operators*, Trans. Math. Society **149** (1970) 75-87.
[20] R. T. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM J. Control **14** (1976), 877-899.
[21] G. Salmon, V. H. Nguyen and J. J. Strodiot, *Coupling the auxiliary problem principle and epiconvergence theory to solve general variational inequalities*, J. Optim. Theory Appl. **104** (2000), 629-657.
[22] K. Taji and M. Fukushima, *A new merit function and a successive quadratic programming algorithm for variational inequality problem*, SIAM J. Optim. **6** (1996), 704-713.
[23] K. Taji, M. Fukushima and Ibaraki, *A global convergent Newton method for solving monotone variational inequality problem*, Math. Program. **58** (1993), 369-383.
[24] J. H. Wu, M. Florian and P. Marcotte, *A general descent framework for the monotone variational inequality problem*, Math. Program. **53** (1992), 99-110.
[25] D. Zhu and P. Marcotte, *An extended descent framework for variational inequalities*, J. Optim. Theory Appl. **80** (1994), 349-366.
[26] D. Zhu and P. Marcotte, *Cocoercivity and its role in the convergence of iterative schemes for solving variational inequalities*, SIAM J. Optim. **6** (1996), 714-726.

Department of Mathematics
Posts and Telecommunications
Institute of Technology
Hanoi, Vietnam

E-mail address: ngocanhncs@yahoo.com

Optimization and Control Department
Institute of Mathematics
18 Hoang Quoc Viet, 10307 Hanoi, Vietnam

E-mail address: ldmuu@math.ac.vn