Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 6, No. 2, pp. 375–390, May 2010

RESTRICTED-STEP JOSEPHY-NEWTON METHOD FOR GENERAL VARIATIONAL INEQUALITIES WITH POLYHEDRAL CONSTRAINTS*

Mend-Amar Majig and Masao Fukushima

This paper is dedicated to the memory of Professor Alex Rubinov.

Abstract: In this paper, we consider the problem of finding a solution of the general (i.e., not necessarily monotone) variational inequality problem (VIP). We propose a new practical version of globally convergent Josephy-Newton based method for VIP. By means of some additional bound constraint, the method ensures existence of solutions to subproblems, which is an essential property not enjoyed by the classical Josephy-Newton method. Under appropriate conditions, global and locally superlinear convergence properties of the proposed method are established. Furthermore, we develop an evolutionary algorithm for solving general VIPs with bounded polyhedral constraints, which can be used to solve the modified Josephy-Newton subproblems when the constraint set is polyhedral. Numerical results for some test problems are reported to show the practical effectiveness of the proposed methods.

Key words: *variational inequality, merit function, Josephy-Newton method, evolutionary algorithm*

Mathematics Subject Classification: *65K05, 68T20*

1 Introduction

Consider the variational inequality problem (VIP): Find $x^* \in S$ such that

$$\langle F(x^*), x - x^* \rangle \geq 0, \ \forall x \in S, \tag{1.1}$$

where $F : \Re^n \to \Re^n$ is a continuously differentiable mapping, $S \subseteq \Re^n$ is a closed convex set and $\langle \cdot, \cdot \rangle$ denotes the inner product in $\Re^n$. This problem has been studied extensively and many important applications may be found in [6]. Although there are plenty of methods proposed for solving (1.1), most of them rely on the basic assumption that the mapping F is monotone [6]. Some methods [10, 11, 15] may be applied to the general VIP by reformulating the problem as a mixed complementarity problem through its KKT system. But those approaches increase the dimension of the problem by introducing Lagrangian multipliers.

This paper presents two methods for general VIPs, both of which keep the dimension of the problem intact. These methods are designed with quite different ideas, but they are linked in that the second method can be used as a subprocedure within the first method.

*This research was supported in part by a Grant-in-Aid for Scientific Research from Japan Society for the Promotion of Science.

The first method is a modification of the classical Josephy-Newton method for VIPs. The Josephy-Newton method is an iterative method that successively solves sub-linearized variational inequalities and local superlinear convergence of the method has been established [9]. Globalization of the Josephy-Newton method by combining it with some line search methods for equivalent optimization problem of VIP has also been designed [6, 16]. A difficulty of using those approaches in practice is that solvability of the subproblems is not guaranteed, i.e., in some case the subproblems have no solution. Furthermore, even when the subproblems have solutions, it is not easy to find them because the subproblems themselves are general variational inequalities with the same constraint set S.

Our method adopts the idea that an extra bound constraint (such as a box or a ball) is added to the original constraint set in the subproblems so that the Josephy-Newton subproblems have always solutions. In order to get a global convergence property, we combine the Josephy-Newton method with the gradient projection method for the equivalent optimization problem and we employ the regularized gap function for (1.1) as a merit function. Although the solvability of the subproblems is ensured by means of the above mentioned modification, we still need some effective method for solving those subproblems.

The second method we develop is to meet this requirement. It is an evolutionary algorithm designed to solve the VIP with bounded polyhedral constraints and also relies on the equivalent global optimization problem of the VIP. The global minimum value of the equivalent optimization problem is known to be zero and the algorithm uses this fact as a stopping condition. Moreover, crossover and mutation procedures amenable to the polyhedral constraint set are devised as well as the initial population generation procedure. In addition, we use the fitness function modification procedure to increase the validity of the algorithm. In general, an evolutionary algorithm is expensive computationally. However, our target here is to solve VIPs with bounded polyhedral sets, of which size is presumably small. For such problems, we may expect that the algorithm can find a solution with a reasonable amount of computations.

The organization of the paper is as follows. In Section 2, we give a brief review of the equivalent global optimization reformulation of the VIP. In Section 3, we propose the restricted-step Josephy-Newton method and discuss its convergence properties. Section 4 presents the evolutionary algorithm for solving VIPs with bounded polyhedral constraints. Appropriate crossover and mutation procedures as well as the fitness function modification procedure will also be discussed. In Section 5, we report numerical results of the methods for some test problems. Section 6 concludes the paper.

$\boxed{2}$ Equivalent Reformulation of VIP

Many methods for solving the VIP (1.1) adopt the idea of reformulating the problem as a system of nonlinear equations or an optimization problem with zero global minimum value [7, 10, 11, 12, 13, 14, 15, 16]. To reformulate the VIP as an optimization problem, we usually use so-called merit functions. Any global minimum of a merit function, say θ, with zero function value is a solution of VIP (1.1) and vice versa. So the VIP (1.1) is equivalent to the following global optimization problem with zero minimum value:

$$\min \theta(x) \text{ s.t } x \in S. \tag{2.1}$$

Three well known merit functions for the VIP are the following:

- Gap function

$$\theta_{gap}(x) = \sup_{y \in S} \langle F(x), x - y \rangle. \tag{2.2}$$

- Regularized gap function

$$\begin{aligned} \theta_{reg}(x) &= \max_{y \in S} \left(\langle F(x), x - y \rangle - \frac{1}{2}\|x - y\|^2 \right) \\ &= \langle F(x), x - \mathcal{P}_S(x - F(x)) \rangle - \frac{1}{2}\|x - \mathcal{P}_S(x - F(x))\|^2. \end{aligned} \tag{2.3}$$

- Natural residual function

$$\theta_{nat}(x) := \|x - \mathcal{P}_S(x - F(x))\|^2.$$

Here $\mathcal{P}_S$ denotes the projection mapping onto the set S. The gap function and the natural residual function are non-differentiable while the regularized gap function is continuously differentiable whenever so is F, and its gradient can be obtained by the following formula [7]:

$$\nabla \theta_{reg}(x) = F(x) + (F'(x) - I)^T (x - y(x)),$$

where $y(x) = \mathcal{P}_S(x - F(x))$. To compute the value of the regularized gap function or the natural residual function, it is necessary to solve the following convex optimization problem:

$$\min \|y - (x - F(x))\|^2 \quad \text{s.t. } y \in S. \tag{2.4}$$

When S is a polyhedral set, problem (2.4) becomes a convex quadratic programming problem. In the remainder of this paper, we will restrict ourselves to the regularized gap function and simply denote θ_{reg} as θ.

$\boxed{3}$ Restricted-step Josephy-Newton Method

In this section we present the restricted-step Josephy-Newton method for solving the general VIP. First we will briefly review the classical Josephy-Newton method [9]. Let us consider the VIP (1.1). At the k-th iteration, the Josephy-Newton method solves the linearized sub-VIP

$$\langle F^k(x), x' - x \rangle \geq 0, \quad \forall x' \in S \tag{3.1}$$

to get the next iteration point x^{k+1}. Here F^k is the linear approximation of the mapping F at x^k, i.e.,

$$F^k(x) := F(x^k) + F'(x^k)(x - x^k).$$

It has been shown [9] that this algorithm is locally superlinearly convergent and, by combining it with a descent method for a merit function, there have been developed some methods which are globally convergent and locally superlinearly convergent [6, 16]. A drawback of these methods is that the solvability of (3.1) is not guaranteed and even if it has a solution, without some kind of monotonicity assumption on (3.1) we may fail to solve it. We propose here a modification of the Josephy-Newton method which is implementable in practice.

We consider the Josephy-Newton subproblem modified as

$$\langle F^k(x), x' - x \rangle \geq 0, \quad \forall x' \in S \cap B(x^k, \delta), \tag{3.2}$$

where $\delta > 0$ is a fixed scalar and $B(x^k, \delta) := \{x \in \Re^n | \ \|x - x^k\| \leq \delta\}$. We combine this method with the gradient projection method for the equivalent optimization problem to VIP (1.1)

$$\min \theta(x) \text{ s.t } x \in S. \tag{3.3}$$

Lemma 3.1. *A point x^k is a solution of VIP (1.1) if and only if it is a solution of the modified Josephy-Newton subproblem (3.2).*

Proof. Let x^k be a solution to VIP (1.1). Then $\langle F(x^k), x' - x^k\rangle \geq 0$, $\forall x' \in S$, and since $x^k \in S \cap B(x^k, \delta)$ and $F^k(x^k) = F(x^k)$, x^k is solution to (3.2).

Next, let x^k be a solution of the modified Josephy-Newton subproblem (3.2) and show it is a solution of VIP (1.1). Assume to the contrary that there exists a point $\bar{x} \in S$ such that $\langle F(x^k), \bar{x} - x^k\rangle < 0$. Then the point $\tilde{x} := x^k + \min\{\delta, 1\}\dfrac{\bar{x} - x^k}{\|\bar{x} - x^k\|}$ is in $S \cap B(x^k, \delta)$ and

$$\langle F^k(x^k), \tilde{x} - x^k\rangle = \frac{\min\{\delta, 1\}}{\|\bar{x} - x^k\|}\langle F(x^k), \bar{x} - x^k\rangle < 0.$$

This contradicts the fact that x^k is a solution of (3.2). □

The use of the subproblem (3.2) has some advantages. First of all, this problem always has a solution because the constraint set is bounded and closed convex. Secondly, we may develop some effective method for solving it when the parameter δ is of appropriate size. In general, the choice of the parameter δ is important for the efficiency of the algorithm. If δ is inappropriately big, then the procedure of solving the sub-VIP (3.2) may not be very effective. On the other hand, if we choose δ too small, then the improvement of the solution might also be very small, resulting in slow convergence. One possible way to cope with this situation is to adjust the parameter δ appropriately during the iterations in a way similar to the trust region method in nonlinear optimization. We use some upper limit δ_{max} and lower limit δ_{min} for the parameter δ to prevent it from becoming too large or too small. In the next section, we will present an algorithm for solving sub-VIP (3.2).

The restricted-step Josephy-Newton algorithm is formally stated as follows.

Algorithm rJN

1. Initialization. Choose $x^0 \in S$ and set parameters $\varepsilon_1 > 0$, $\varepsilon_2 > 0$, $p > 1$, $\gamma \in (0,1)$, $\sigma \in (0,1)$, $\rho > 0$, $\beta \in (0,1)$, $0 < \delta_{min} < \delta_{max}$ and a negative integer i_{min}. Set $\delta^0 := \delta_{max}$ and the iteration counter $k := 0$.
2. Stopping condition. If $\theta(x^k) \leq \varepsilon_1$ or $\|x^k - \mathcal{P}_S(x^k - \nabla\theta(x^k))\| \leq \varepsilon_2$, then STOP.
3. Sub-VIP. Find a solution $\bar{x}^k$ of the sub-VIP (3.2) with δ^k, and let $d^k := \bar{x}^k - x^k$.
(a) If the condition

$$\theta(x^k + d^k) \leq \sigma \cdot \theta(x^k) \tag{3.4}$$

is satisfied, then set $x^{k+1} := \bar{x}^k$ and $\delta^{k+1} := \begin{cases} \min\{2\delta^k, \delta_{max}\}, & \text{if } \|x^{k+1} - x^k\| = \delta^k; \\ \max\{\delta_{min}, \|x^{k+1} - x^k\|\}, & \text{otherwise.} \end{cases}$
Set $k := k + 1$ and go to Step 2.
(b) Otherwise, check the condition

$$\nabla\theta(x^k)^T d^k \leq -\rho \cdot \|d^k\|^p, \tag{3.5}$$

and if it holds, then find the smallest nonnegative integer i_k such that

$$\theta(x^k + \beta^{i_k} d^k) \leq \theta(x^k) + \gamma \beta^{i_k} \nabla\theta(x^k)^T d^k. \tag{3.6}$$

If $i_k = 0$, then redefine i_k by finding the largest nonpositive integer in $[i_{min}, 0]$ satisfying (3.6) and

$$\theta(x^k + \beta^{i_k - 1} d^k) > \theta(x^k + \beta^{i_k} d^k), \; x^k + \beta^{i_k} d^k \in S. \tag{3.7}$$

Set $x^{k+1} := x^k + \beta^{i_k} d^k$, $\delta^{k+1} := \mathrm{mid}\{\delta_{min}, \|x^{k+1} - x^k\|, \delta_{max}\}$. Set $k := k+1$ and go to Step 2.
(c) If either of conditions (3.4) and (3.5) is not satisfied, then go to Step 4.
4. Gradient projection step. Find x^{k+1} by the following gradient projection procedure: Define $x^k(\alpha) := \mathcal{P}_S(x^k - \frac{\alpha\delta^k}{\|\nabla\theta(x^k)\|}\nabla\theta(x^k))$ and find the smallest nonnegative integer i_k such that

$$\theta(x^k(\beta^{i_k})) \leq \theta(x^k) + \gamma \nabla\theta(x^k)^T (x^k(\beta^{i_k}) - x^k). \tag{3.8}$$

If $i_k = 0$, then redefine i_k by finding the largest nonpositive integer in $[i_{min}, 0]$ satisfying (3.8) and

$$\theta(x^k(\beta^{i_k - 1})) > \theta(x^k(\beta^{i_k})). \tag{3.9}$$

Set $x^{k+1} := x^k(\beta^{i_k})$, $\delta^{k+1} := \mathrm{mid}\{\delta_{min}, \|x^{k+1} - x^k\|, \delta_{max}\}$, $k := k+1$ and go to Step 2.

Remark 3.2. In Step 3(b) and Step 4, when $i_k = 0$ is accepted by (3.6) or (3.8), we try to find a stepsize larger than unity in the hope of achieving a further reduction in the function value. Note that this modification does not affect the theoretical convergence properties of the algorithm.

In the algorithm, we use a gradient projection step when the Josephy-Newton subproblem fails to give a useful direction. The idea of combining Newton-type methods with gradient related methods is an important technique to make Newton-type methods globally convergent and can be found in many literatures [6, 11, 15, 16]. When the iterative point approaches a solution satisfying a certain regularity condition, Newton type methods will take effect and make the convergence faster.

Convergence of the algorithm

Theorem 3.3. *Let F be a continuously differentiable mapping and S be a closed convex set. Let $\{x^k\}$ be an infinite sequence generated by Algorithm rJN. Then*

(a) *every accumulation point of $\{x^k\}$ is a stationary point of the optimization problem (3.3).*

(b) *if x^* is an accumulation point of $\{x^k\}$ such that $F'(x^*)$ is positive definite, then x^* is a solution of the VIP (1.1) and the whole sequence $\{x^k\}$ converges to this point. Furthermore, if $p > 2$ and $\gamma < \frac{1}{2}$ in Algorithm rJN, then the following statements hold:*

   (i) *eventually the unit step size for the direction d^k is accepted so that $x^{k+1} = \bar{x}^k$.*

   (ii) *the convergence rate is Q-superlinear; furthermore, if the Jacobian $F'(x)$ is Lipschitz continuous in a neighborhood of x^*, the convergence rate is Q-quadratic.*

Proof. Part (a) can be proved by using standard arguments on descent methods. For part (b), if we show that eventually the subproblems (3.1) and (3.2) have identical solution sets, then everything will essentially follow from Theorem 10.4.23 of [6].

Since F is continuously differentiable and $F'(x^*)$ is positive definite, there exists a scalar $\delta_1 > 0$ such that $F'(x^k)$ is positive definite for any $x^k \in B(x^*, \delta_1)$. This implies that the mapping $F^k(x) = F(x^k) + F'(x^k)(x - x^k)$ is strongly monotone for all k large enough, and hence the subproblems (3.1) and (3.2) both have unique solutions.

Theorem 7.3.3 of [6] states that there exists a positive scalar $\bar{\varepsilon}$ such that for every $\varepsilon \in (0, \bar{\varepsilon}]$ a $\delta_2 > 0$ exists such that for every $x^k \in S \cap B(x^*, \delta_2)$, the problem (3.1) has a unique solution in $B(x^k, \varepsilon)$. If we choose $\varepsilon = \min\{\bar{\varepsilon}, \frac{\delta_{min}}{2}\}$, then we can ensure that the solution of (3.1) is in $B(x^k, \delta_{min})$. Therefore, this solution is also a solution of (3.2), and since they both have unique solutions, their solution sets are identical. □

$\boxed{4}$ Evolutionary Algorithm for VIP with a Bounded Constraint Set

In this section, we present a method for solving the following VIP: Find $x \in D$ such that

$$\langle \bar{F}(x), x' - x \rangle \geq 0, \ \forall x' \in D, \tag{4.1}$$

where $\bar{F} : \Re^n \to \Re^n$ and $D \subseteq \Re^n$. If we let $\bar{F}(x) := F^k(x)$ and $D := S \cap B(x^k, \delta)$, then this VIP reduces to the Josephy-Newton subproblems (3.1) in the previous section. We will confine ourselves to the case where S is a polyhedral set and the norm used to define $B(x^k, \delta)$ is the l_∞ norm. Then the set D is a bounded polyhedral set represented by a system of linear inequalities, that is,

$$D := \{x \in \Re^n | Ax \leq b\},$$

where $A \in \Re^{m \times n}$ and $b \in \Re^m$.

Then the VIP (4.1) can be reformulated as the following global optimization problem:

$$\min \bar{\theta}(x) \text{ s.t } x \in D, \tag{4.2}$$

where $\bar{\theta}$ is the gap function (2.2) associated with (3.2). The method presented here follows the main framework of the evolutionary algorithm [4, 8] and uses some additional procedures to exploit the special features of the problem such as the known minimum value and the polyhedrality of the constraint set.

Firstly, the global minimum value of the problem is known to be zero if the VIP (4.1) has a solution. We use this fact not only as a stopping condition but also to improve the validity of the evolutionary algorithm. The evolutionary algorithm is a population-based algorithm. It makes use of a fitness function to evaluate the goodness of a solution, thereby keeping the population set consisting of promising solutions. If, during the search, the population set gets stuck around a point which is not a global minimum, we need to direct the population set to another possible region which may contain a global solution. But if we use the same fitness function all the time, even with a different initial population set, the algorithm is very likely to converge back to this non-global solution. So we modify the fitness function by increasing the function value around this non-global solution. More specifically, we use the tunneling function technique [2] and consider a new fitness function

$$f_t(x, \bar{x}) := f_c(x) \cdot \exp\Big(\frac{1}{\|x - \bar{x}\|^2}\Big), \tag{4.3}$$

where $f_c(x)$ is the current fitness function and $\bar{x}$ is the above-mentioned non-global solution. The modified fitness function $f_t(x, \bar{x})$ will have the same exact global minimum points as the original function.

Next we use crossover and mutation procedures that are amenable to the polyhedral constraint set $D = \{x \in \Re^n | Ax \le b\}$. Let $a_i^T,\ i = 1, \ldots, m$ be the rows of matrix A, and $b_i,\ i = 1, \ldots, m$ be the components of vector b. Suppose we have two points $x \notin D$ and $y \in \text{int } D$, where int D denotes the interior of D. First we determine the point where the line segment $[x, y]$ intersects the boundary of D. Let us define the index sets $I(x) := \{i \mid a_i^T x - b_i > 0\}$. Then the point of intersection is determined as

$$\bar{x} := y + \frac{-a_{i(x,y)}^T y + b_{i(x,y)}}{a_{i(x,y)}^T (x - y)} (x - y),$$

where $i(x, y) := \operatorname*{argmin}_{i \in I(x)} \dfrac{-a_i^T y + b_i}{a_i^T x - b_i}$. During the evolutionary search, points generated by the crossover and mutation procedures may go outside the constraint set D, and we need to bring them back into the set D. Because the projection operator may project many different points onto the same point on the boundary, it may be helpful to use points of intersection computed by the above mentioned procedure instead of projection.

To describe our procedures, it will be convenient to define the mapping $P : \Re^n \times \text{int } D \to D$ by

$$P(x, y) := \begin{cases} x, & \text{for } x \in D; \\ y + \dfrac{-a_{i(x,y)}^T y + b_{i(x,y)}}{a_{i(x,y)}^T (x - y)} (x - y), & \text{for } x \notin D. \end{cases} \tag{4.4}$$

Remark 4.1. The coefficient $\dfrac{-a_{i(x,y)}^T y + b_{i(x,y)}}{a_{i(x,y)}^T (x - y)}$ in (4.4) is well defined and actually bounded. In fact, we have

$$\frac{-a_{i(x,y)}^T y + b_{i(x,y)}}{a_{i(x,y)}^T (x - y)} = \frac{-a_{i(x,y)}^T y + b_{i(x,y)}}{-a_{i(x,y)}^T y + b_{i(x,y)} + a_{i(x,y)}^T x - b_{i(x,y)}} = \Big(1 + \frac{a_{i(x,y)}^T x - b_{i(x,y)}}{-a_{i(x,y)}^T y + b_{i(x,y)}}\Big)^{-1},$$

which is bounded by the definition of $i(x, y)$.

Initial population generation. First we determine the analytical center x^0 of the set D and a box D_1 containing D inside. Finding such a box can be done easily by solving the $2n$ linear programming problems

$$l_i := \min_{x \in D} e_i^T x, \quad u_i := \max_{x \in D} e_i^T x, \quad i = 1, 2, \ldots, n,$$

where e_i is the i-th unit vector, and then letting $D_1 := \{x \in \Re^n | \ l_i \le x_i \le u_i,\ i = 1, \ldots, n\}$. Moreover the analytical center x^0 of D will be determined by

$$x^0 := \operatorname*{argmin}_{x \in D} \sum_{i=1}^{m} -\log(b_i - a_i^T x).$$

Then we will generate points randomly in the box D_1 and bring them inside the set D. Specifically, let $x \in D_1$ be a generated point. Then we compute a new point x_{new} as follows and add it to the population set:

$$x_{new} := x^0 + \alpha \cdot (P(x, x^0) - x^0),$$

where $P(\cdot,\cdot)$ is defined by (4.4) and $\alpha \in (0,1)$ is a random scalar. Clearly all points in the population set lie in the interior of D.

Crossover. Let x^1 and x^2 be parents to mate and $x_c \in \Re^n$ be a point created by the intermediate recombination of x^1 and x^2, whose components are corresponding components of either x^1 or x^2. Then we produce two children as follows:

$$c^1 := x^1 + \alpha_1 \cdot (P(x^1, x_c) - x^1), \quad c^2 := x^2 + \alpha_2 \cdot (P(x^2, x_c) - x^2),$$

where $\alpha_1, \alpha_2 \in (0,1)$ are random scalars.
Mutation. We choose some component of a newly produced child and change it randomly in a certain range.

Now we state our evolutionary algorithm for problem (4.2), thereby solving the general VIP (4.1) with a bounded polyhedral constraint.

Algorithm EA

1. Initialization. Choose a population size M and fix parameters $l_N, l_s, \bar{N}, \eta, NF_{max}, \varepsilon > 0$. Construct an initial population set $\mathcal{P}$ and let the initial fitness function be the original objective function of (4.2),

$$f_c(x) := \bar{\theta}(x).$$

Evaluate the trial points in $\mathcal{P}$ and order them according to their fitness function values so that x^1 is the best solution and x^M is the worst, i.e.,

$$f_c(x^1) \le f_c(x^2) \le \cdots \le f_c(x^M).$$

Set the generation counters $t := 1$ and $s := 0$.
2. Parents Pool Generation. Generate a parents pool $\mathcal{P}'$, which consists of all different pairs from the population set $\mathcal{P}$.
3. Crossover and Mutation. Select a pair (p^1, p^2) from the parents pool $\mathcal{P}'$. Apply the Crossover and Mutation Procedure to the pair (p^1, p^2) to obtain two new trial points c^1, c^2.
4. Survival Selection. If the newly produced point c^1 or c^2 is better than any point $\bar{p}$ in the population, then replace $\bar{p}$ by that new point. Delete the pair (p^1, p^2) from the parents pool $\mathcal{P}'$. If $\mathcal{P}' = \emptyset$, then reorder the population set according to their fitness function value and let

$$N := \min\{s, \bar{N}\}, \ B := \{b^1, b^2, \ldots, b^N\} \leftarrow \{x^1, b^1, \ldots, b^{(N-1)}\}, \ s := s + 1$$

and go to Step 5; otherwise go to Step 3.
5. Intensification. If, during the last $\bar{N}$ generations of evolution, the fitness function has not been modified and the best point in the population set has not been improved enough, i.e.,

$$s \ge \bar{N} \text{ and } f_c(b^{\bar{N}}) - f_c(b^1) \le \eta \cdot (1 + f_c(b^1)),$$

then choose $x^1, x^2, ..., x^{l_N} \in \mathcal{P}$ and for each x^i, $i = 1, 2, ..., l_N$, perform a local search on the original objective function $\bar{\theta}(x)$:

$$\bar{x}^i \longleftarrow \textit{Local Search } (\bar{\theta}(x), x^i, l_s),$$

where l_s is the maximum number of steps in the local search. If $f_c(x^i) < f_c(\bar{x}^i)$, i.e, after the local search the current fitness function value increases, then construct a new fitness function by

$$f_c(x) := f_c(x) \cdot \exp\Big(\frac{1}{\|x - x^i\|^2}\Big).$$

Otherwise, let $\mathcal{P} := \mathcal{P} \cup \{\bar{x}^i\} \backslash \{x^i\}$. If the fitness function is modified at least once during the above procedure, then set $s := 1$. Reorder the points in the population set according to their fitness function values and let $t := t + 1$. Go to Step 6.

6. Stopping Condition. If $f_c(x^1) < \varepsilon$ or the number of function evaluations exceeds the pre-specified limit NF_{max}, then terminate the algorithm and refine the global solution x^1 by some local search method.

If the fitness function of the problem has not been modified and $x^1 \in \mathcal{P}$ has not been improved enough during the last $\bar{N}$ generations of evolution and a local search, i.e.,

$$s \geq \bar{N} \text{ and } f_c(b^{\bar{N}}) - f_c(x^1) \leq \eta(1 + f_c(x^1)),$$

then construct a new fitness function by

$$f_c(x) := f_c(x) \cdot \exp\Big(\frac{1}{\|x - x^1\|^2}\Big)$$

and set $s := 1$. Otherwise, let $B := \{b^1, b^2, \ldots, b^{\bar{N}}\} \leftarrow \{x^1, b^1, \ldots, b^{(\bar{N}-1)}\}$. Proceed to Step 2 with $(f_c(x), \mathcal{P})$.

$\boxed{5}$ Numerical Results

The performance of the restricted-step Josephy-Newton method was tested on a number of test problems. The lack of test problems for general variational inequality problems enforces us to generate test problems by ourselves. We did this task by modifying, i.e., adding some extra constraints to, well known mixed complementarity test problems in the MCPLIB library [5]. The test problems used in our numerical experiments are included in the Appendix. The programming code for the algorithm was written in MATLAB and run on a computer Pentium 4 Microprocessor.

The algorithm is terminated if one of the following two conditions is satisfied:

$$\theta(x^*) \leq \varepsilon_1, \quad \|x^* - \mathcal{P}_S(x^* - \nabla\theta(x^*))\| \leq \varepsilon_2.$$

In the implementation, we have used the following parameter settings:

$$\beta = 0.5,\ \sigma = 0.5,\ \gamma = 0.49,\ p = 2.1,\ \rho = 0.5,\ \delta_{min} = 0.2n,\ \delta_{max} = 10\delta_{min},\ i_{min} = -10$$

and

$$\varepsilon_1 = 10^{-12}, \quad \varepsilon_2 = 10^{-6}.$$

The parameter settings of the evolutionary algorithm EA used to solve sub-VIPs are displayed in Table 1. The last two parameters in Table 1 are used in the stopping conditions of EA, and they both depend on the current objective value in the main iteration, where n is the dimension of the problem and x^k is the current point of the main iteration. In the early stage of the main iterations, i.e, when $\theta(x^k)$ is relatively large, solving subproblems

Table 1: Parameter settings.

Parameters	Definition	Value
M	number of elements in population	10
l_N	number of best points for which local search is used	1
l_s	maximum number of steps in local search	20
N_k, η	parameters controlling local search in EA	3, 0.995
ε_t, ρ_t	tunneling parameters	0.1, 2
ε_{EA}	tolerance parameters for the objective function in EA	$\min\{10^{-6}, 10^{-2} \cdot \theta(x^k)\}$
NF_{max}	maximum number of function evaluations in EA	$\mathrm{mid}\{100n, 400n, \dfrac{400n}{\theta(x^k)^{0.25}}\}$

with high precision is not so effective from the viewpoint of computational expense. As the iteration process converges to a solution, the tolerance ε_{EA} in EA will decrease, while the maximum number of function evaluations allowed will increase.

The results for the test problems are shown in Table 2. The columns in this table have the following meanings:

Problem:	name of the test problem,
n:	dimension of the test problem,
t:	number of main iterations,
N_θ:	number of rJN directions accepted by the condition (3.4),
N_d:	number of rJN directions accepted by the condition (3.6),
N_{mon}:	number of monotone sub-VIPs solved,
N_{inact}:	number of sub-VIPs where the constraint $\|x - x^k\| \leq \delta$ was inactive at the solution,
N_{unit}:	number of times rJN unit stepsize was taken,
$\theta(x^*)$:	regularized gap function value at the obtained solution,
NF:	number of function evaluations,
NF_{EA}:	number of function evaluations required to solve the sub-VIPs by EA,
CPU(sec):	total CPU time to solve the problems.

As shown in Table 2, the main termination criterion

$$\theta(x^*) \leq \varepsilon_1$$

was satisfied for all test problems. Moreover, the number of iterations was quite small overall and the solutions of sub-VIPs were used very often. All the problems except *choi** made use of the directions determined from the solution of sub-VIPs. As for the problem *choi**, the method solved sub-VIPs 8 times, and 7 of them gave successful directions.

The column 6 (N_{mon}) shows the number of monotone sub-VIPs we encountered. Most of the test problems have non-monotone sub-VIPs, while every sub-VIP of the problem *nash** was monotone. This suggests that the problem *nash** itself is monotone. The column 7 (N_{inact}) shows the number of sub-VIPs that had solutions interior to additional box constraints. The column 8 (N_{unit}) gives the number of times the unit step size or a longer step size was accepted in the direction determined by the solutions of sub-VIPs. For all problems except *mathisum** and *choi**, such a step size was accepted all the times.

The numbers of function evaluations required to solve all the sub-VIPs are shown in column 10 (*NF*) and column 11 (NF_{EA}) of the table. The column *NF* shows the number of actual function evaluations of F, while NF_{EA} shows the number of merit function evaluations for sub-VIPs (3.2). Since the evaluation of those merit functions requires to solve a linear programming problem, these numbers equal the number of linear programming problems solved. The last column of the table shows the CPU time in seconds used to solve the problems.

Table 2: Numerical results for the restricted-step Josephy-Newton method.

Problem	n	t	N_θ	N_d	N_{mon}	N_{inact}	N_{unit}	$\theta(x^*)$	NF	NF_{EA}	CPU(sec)
badfree*	5	3	3	0	0	2	3	4.5391e-18	4	6556	4.7656
choi*	13	8	7	0	5	4	7	7.4270e-13	19	29095	80.0469
explcp*	16	12	11	1	0	0	12	8.8818e-16	13	4527	23.8906
josephy*	4	4	4	0	0	3	4	2.8813e-13	5	3968	5.7081
kojshin*	4	4	4	0	0	3	4	1.4279e-16	5	2979	7.0156
mathinum*	3	8	6	2	0	5	8	2.9116e-16	9	7944	9.8906
mathisum*	4	10	8	2	0	3	9	6.0942e-16	12	11321	22.1406
nash*	10	8	8	0	8	6	8	1.0027e-13	9	18871	45.1101

6 Conclusions

In this paper we have proposed a practically implementable, globally convergent and locally superlinearly convergent method for solving the general variational inequality problem with polyhedral constraints. Our restricted-step Josephy-Newton method is a modification of the classical Josephy-Newton method and it solves the linearized sub-VIPs with additional box constraints. Moreover, an evolutionary algorithm for solving those sub-VIPs is presented that can effectively deal with bounded polyhedral constraints. Numerical results show that the proposed approach is able to solve various test problems of non-monotone VIPs successfully.

References

[1] F.A. Al-Khayyal, An implicit enumeration procedure for the general linear complementarity problem, *Mathematical Programming Study* 31 (1987) 1–20.

[2] C. Barron and S. Gomez, The exponential tunneling method, *Reporte de Investigacio'n IIMAS* 1 (1991) 1–23.

[3] S.Ch. Choi, W.S. Desarbo and P.T. Harker, Product positioning under price competition, *Management Science* 36 (1990) 175–199.

[4] K.A. De Jong, *Evolutionary Computation*, The MIT Press, Cambridge, MA, 2005.

[5] S.P. Dirkse and M.C. Ferris, Mcplib: a collection of nonlinear mixed complementarity problems, *Optimization Methods and Software* 5 (1995) 319–345. http://www.gams.com/mpec/mcplib.htm.

[6] F. Facchinei and J.S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*, Volumes I and II, Springer-Verlag, New York, 2003.

[7] M. Fukushima, Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems, *Mathematical Programming* 53 (1992) 99–110.

[8] D.E. Goldberg, *Genetic Algorithm in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, 1989.

[9] N.H. Josephy, Newton's method for generalized equations, *Techincal Summary Report 1965*, Mathematics Research Center, University of Wisconsin, 1979.

[10] C. Kanzow, Global optimization techniques for mixed complementarity problems, *Journal of Global Optimization* 16 (2000) 1–21.

[11] C. Kanzow and H. Pieper, Jacobian smoothing methods for nonlinear complementarity problems, *SIAM Journal on Optimization* 9 (1999) 342–373.

[12] N.M. Kostreva and Q. Zheng, Integral global optimization method for solution of nonlinear complementarity problems, *Journal of Global Optimization* 5 (1994) 181–193.

[13] M. Majig, A.R. Hedar and M. Fukushima, Hybrid evolutionary algorithm for solving general variational inequalities, *Journal of Global Optimization* 38 (2007) 637–651.

[14] P.M. Pardalos and J.B. Rosen, Global optimization approach to the linear complementarity problem, *SIAM Journal on Scientific and Statistical Computing* 6 (1988) 341–353.

[15] H.D. Qi and L.Z. Liao, A smoothing Newton method for general nonlinear complementarity problems, *Computational Optimization and Applications* 17 (2000) 231–253.

[16] K. Taji, M. Fukushima and T. Ibaraki, A globally convergent Newton method for solving strongly monotone variational inequalities, *Mathematical Programming* 58 (1993) 369–383.

$\boxed{7}$ Appendix

The test problems used in the numerical experiments are displayed here. We have used some standard test problems from the MCPLIB [5] to construct our test problems by modifying the constraint sets. In particular, modifications have been made in such a way that the solutions of the original test problems become infeasible to the new test problems.

- **badfree***
  $n = 5$ and $F(x) = Mx + q$, where

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}, \quad q = \begin{pmatrix} -1 \\ -1 \\ -0.5 \\ -0.5 \\ -1 \end{pmatrix}.$$

  In this example, the original constraint set X was

$$X := \{x \in \Re^n |\ x_i \geq 0,\ i = 1, \ldots, n-1\}.$$

We have modified this by adding two linear constraints:

$$S = \{x \in X |\ \sum_{i=1}^{n} x_i \leq n, \quad \sum_{i=1}^{n} i x_i \geq n+1\}.$$

We use the feasible starting point $x^0 = \Big(\frac{n-1}{n}, \frac{n-1}{n}, \ldots, \frac{n-1}{n}\Big)^T$.

- **choi***
$n = 13$ and $F(x) = (\nabla_{x_j} \mathcal{P}_j(x))_{j=1}^{n}$, where

$$\mathcal{P}_j(x) = (x_j - C_j)\frac{Q}{I}\sum_{i=1}^{I} Pr_{ij},$$

with

$$Pr_{ij} = \frac{\exp\{-\chi DU_{ij}\}}{\sum\limits_{m=1}^{J} \exp\{-\chi DU_{im}\} + K}$$

and

$$DU_{ij} = \sum_{n=1}^{N} v_{in}(a_{jn} - b_{in})^2 + w_i x_j + b_i.$$

For the details of the coefficients given above, see [3]. In this example, the original constraint set X was

$$X := \{x \in \Re^n |\ l_i \leq x_i \leq u_i\},$$

where

$$l = (0.40.13280.40.12750.09750.11720.15410.40.3010.40.40.260.2383),$$

$$u = (1.20.39841.20.38250.29250.35160.46231.20.9031.21.20.780.7149).$$

We have modified this by adding two linear constraints:

$$S = \{x \in X |\ \sum_{i=1}^{n} x_i \geq 5, \quad \sum_{i=1}^{n} x_i \leq 10\}.$$

We use the feasible starting point $x^0 = \frac{(l+u)}{2}$.

- **explcp***
$n = 16$ and $F(x) = Mx + q$, where

$$M = \begin{pmatrix} 1 & 2 & \ldots & 2 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 2 \\ 0 & \ldots & 0 & 1 \end{pmatrix}, \quad q = \begin{pmatrix} -1 \\ \vdots \\ -1 \end{pmatrix}.$$

In this example, the original constraint set X was

$$X := \{x \in \Re^n |\ x_i \geq 0,\ i = 1, \ldots, n\}.$$

We have modified this by adding two linear constraints:

$$S = \{x \in X| \sum_{i=1}^{n} x_i \geq 2, \quad \sum_{i=1}^{n} x_i \leq n\}.$$

We use the feasible starting point $x^0 = \left(\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2}\right)^T$.

- **josephy***
  $n = 4$ and

$$F(x) = \begin{pmatrix} 3x_1^2 + 2x_2^2 + 2x_1x_2 + x_3 + 3x_4 - 6 \\ 2x_1^2 + x_2^2 + x_1 + 3x_3 + 2x_4 - 2 \\ 3x_1^2 + 2x_2^2 + x_1x_2 + 2x_3 + 3x_4 - 1 \\ x_1^2 + 3x_2^2 + 2x_3 + 3x_4 - 3 \end{pmatrix}.$$

In this example, the original constraint set X was

$$X := \{x \in \Re^n| \ x_i \geq 0, \ i = 1, \ldots, n\}.$$

We have modified this by adding two linear constraints:

$$S = \{x \in X| \sum_{i=1}^{n} ix_i \geq n, \quad \sum_{i=1}^{n} x_i \leq n-1\}.$$

We use the feasible starting point $x^0 = \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)^T$.

- **kojshin***
  $n = 4$ and

$$F(x) = \begin{pmatrix} 3x_1^2 + 2x_2^2 + 2x_1x_2 + x_3 + 3x_4 - 6 \\ 2x_1^2 + x_2^2 + x_1 + 10x_3 + 2x_4 - 2 \\ 3x_1^2 + 2x_2^2 + x_1x_2 + 2x_3 + 9x_4 - 9 \\ x_1^2 + 3x_2^2 + 2x_3 + 3x_4 - 3 \end{pmatrix}.$$

In this example, the original constraint set X was

$$X := \{x \in \Re^n| \ x_i \geq 0, \ i = 1, \ldots, n\}.$$

We have modified this by adding two linear constraints:

$$S = \{x \in X| \sum_{i=1}^{n} ix_i \geq n, \quad \sum_{i=1}^{n} x_i \leq n-1\}.$$

We use the feasible starting point $x^0 = \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)^T$.

- **mathinum*** and **mathisum***
  Both *mathinum* and *mathisum* test problems originate from the Mathiesen's Walrasian equilibrium problem with

$$F(x) = \begin{pmatrix} -x_1 + x_2 + x_3 \\ x_4 - \dfrac{0.9(5x_2 + 3x_3)}{x_1} \\ 5 - x_4 - \dfrac{0.1(5x_2 + 3x_3)}{x_2} \\ 3 - x_4 \end{pmatrix}.$$

The dimension of the problem is $n = 4$ and the original constraint set X is given by

$$X := \{x \in \Re^n |\ x_i \geq 0,\ i = 1, \dots, n\}.$$

(a). **mathinum***
Setting $x_1 = 4.2$, we have *mathinum* test problem with dimension $n = 3$. We have modified the problem by adding two linear constraints:

$$S = \{x \in X|\ \sum_{i=1}^{n} x_i \geq 10, \quad \sum_{i=1}^{n} x_i \leq 10n\}.$$

We use the feasible starting point $x^0 = (4, 4, 4)^T$.
(b). **mathisum***
Setting $\sum_{i=1}^{n-1} x_i = 1$, we have *mathisum* test problem with dimension $n = 4$. We have modified the problem by adding a linear constraint:

$$S = \{x \in X|\ \sum_{i=1}^{n} x_i \leq n\}.$$

We use the feasible starting point $x^0 = \Big(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\Big)^T$.

- **nash***
$n = 10,\ \gamma = 1.2,\ c = (5,\ 3,\ 8,\ 5,\ 1,\ 3,\ 7,\ 4,\ 6,\ 3),\ L = (10,\ 10, \dots,\ 10),$
$\beta = (1.2,\ 1,\ 0.9,\ 0.6,\ 1.5,\ 1,\ 0.7,\ 1.1,\ 0.95,\ 0.75)$ and

$$F(x) = \Big(c_i + (L_i x_i)^{\frac{1}{\beta_i}} - P(x) + x_i \frac{P(x)}{\gamma Q(x)}\Big)_{i=1}^{n},$$

where

$$Q(x) = \sum_{j=1}^{n} x_j, \quad P(x) = 5000^{\frac{1}{\gamma}} Q(x)^{-\frac{1}{\gamma}}.$$

In this example, the original constraint set X was

$$X := \{x \in \Re^n |\ x_i \geq 0,\ i = 1, \dots, n\}.$$

We have modified this by adding two linear constraints:

$$S = \{x \in X|\ \sum_{i=1}^{n} x_i \geq 1, \quad \sum_{i=1}^{n} x_i \leq 4n\}.$$

We use the feasible starting point $x^0 = (1, 1, \dots, 1)^T$.

Manuscript received 9 March 2008
revised 13 August 2008
accepted for publication 18 August 2008

Mend-Amar Majig
Department of Applied Mathematics
School of Mathematics and Computer Science
National University of Mongolia
Ulaanbaatar, Mongolia
E-mail address: `mendamarm@num.edu.mn`

Masao Fukushima
Department of Applied Mathematics and Physics
Graduate School of Informatics, Kyoto University
Kyoto 606-8501, Japan
E-mail address: `fuku@i.kyoto-u.ac.jp`