A new step size rule in Yan et al.'s self-adaptive projection method

Bin Gao*

April 2, 2015

Abstract

In this paper, we propose a new step size rule to accelerate Yan et al.'s self-adaptive projection method. Under the new step size strategy, the superiority of modified projection method is verified through theory to numerical experiments.

Keywords: Variational inequalities; Projection methods; Extragradient methods.

1 Introduction

This paper concerns the so-called variational inequalities (VI(F,Ω)): Find $u^* \in \Omega$ such that

$$F(u^*)^T(u-u^*) \geq 0, \quad \forall u \in \Omega, \tag{1}$$

where Ω is a nonempty closed convex subset of R^n, and F is a mapping from R^n into itself.

A benchmark solver for (1) is the following projection-type method, proposed by Goldstein,Levitin and Polyak [1, 2], and further studied extensively in [4, 5]:

$$u^{k+1} = P_\Omega[u^k - \beta_k F(u^k)], \tag{2}$$

where $P_\Omega(\cdot)$ is the projection from R^n onto Ω. The projection method maintains the global convergence under strong condition that the mapping F is strongly monotone and Lipschitz continuous. In fact, the estimation of the lipschitz constant L and strongly monotone parameter α are difficult and expensive. To overcome the shortcomings, Korpelevich [3] proposed the extragradient method which can be viewed as a prediction-correction method, i.e.,

$$\tilde{u}^k = P_\Omega[u^k - \beta_k F(u^k)] \tag{3}$$
$$u^{k+1} = P_\Omega[u^k - \beta_k F(\tilde{u}^k)] \tag{4}$$

*College of Communications Engineering, PLA University of Science and Technology, Nanjing, 210007, China, **E-mail:** feimaxiao123@gmail.com. The research of this author was partially supported by the National Natural Science Foundation of China grant 70971136.

Under the assumption that F is monotone and Lipschitz continuous, the extragradient method converges globally with appropriate β_k.

Recently, some researchers have investigated numerous modified projection and extragradient-type methods for solving (1). Most of these methods were committed to improve the efficiency by changing the step size parameters under some appropriate strategies and removing the Lipschitz constant with line search technique. Among them, the modified extragradient method proposed by He and Liao [6] adopted the following iterative scheme:

Algorithm 1.1. For given $x^k \in R^n$ and $\beta_k > 0$, the next iterative point is generated by

$$u^{k+1} = P_\Omega[u^k - \gamma\alpha_k g(u^k, \beta_k)] \quad 0 < \gamma < 2, \tag{5}$$

where

$$e(u^k, \beta_k) = u^k - P_\Omega[u^k - \beta_k F(u^k)], \tag{6}$$

$$g(u^k, \beta_k) = e(u^k, \beta_k) - \beta[F(u^k) - F(u^k - e(u^k, \beta_k))], \tag{7}$$

$$\alpha_k = \frac{e(u^k, \beta_k)^T g(u^k, \beta_k)}{\|g(u^k, \beta_k)\|^2}. \tag{8}$$

The convergence of this method can be guaranteed under monotone F and appropriate β_k. However, the direction $g(u^k, \beta_k)$ tends to zero when u^k approaches the solution u^*. In view of this, Yan et al. [14] described a modifed extra-gradient-type method as follows:

Algorithm 1.2. For given $x^k \in R^n$ and $\beta_k > 0$, the new iterative point is obtained through

$$u^{k+1} = P_\Omega[u^k - \gamma\alpha_k d(u^k, \beta_k)], \tag{9}$$

where

$$d(u^k, \beta_k) = e(u^k, \beta_k) + \beta_k F(u^k - e(u^k, \beta_k)), \tag{10}$$

$$\alpha_k = \min\left\{(1 - \frac{\beta_k}{4\tau}, \frac{g(u^k, \beta_k)^T e(u^k, \beta_k)}{\|g(u^k), \beta_k)\|^2 + 2e(u^k, \beta_k)^T g(u^k, \beta_k)}\right\}, \tag{11}$$

the method maintains global convergence under the condition that F is co-coercive.

Inspired by the optimal step size strategy adopted by Li and Yuan [15], a new step size is proposed in this paper, which can be viewed as a combination of Li and Yuan's manipulation idea of step size and descending direction presented by Yan et al. [14].

This paper is organized as follows. In the next section, some preliminaries are given. In section 3, the details of the new step size together with its global convergence are presented. The numerical experiments are carried out in Section 4. Finally, conclusions are summarized in the last section.

2 Preliminaries

In this section, some definitions and results from the literature are presented which are used throughout the paper.

Lemma 2.1 ([11]) *Let $\Omega \subset R^n$ be a convex closed set, then it holds that*

$$(1)\quad (\upsilon - P_\Omega(\upsilon))^T(u - P_\Omega(\upsilon)) \le 0, \quad \forall u \in \Omega, \upsilon \in R^n, \tag{12}$$
$$(2)\quad \|u - P_\Omega(\upsilon)\|^2 \le \|u - \upsilon\|^2 - \|\upsilon - P_\Omega(\upsilon)\|^2, \quad \forall u \in \Omega, \upsilon \in R^n, \tag{13}$$
$$(3)\quad \|P_\Omega(u) - P_\Omega(\upsilon)\|^2 \le (u - \upsilon)^T(P_\Omega(u) - P_\Omega(\upsilon)), \quad \forall u, \upsilon \in R^n. \tag{14}$$

Lemma 2.2 ([17]) *Assuming $\beta > 0$, u^* is a solution of $VI(F, \Omega)$ if and only if*

$$u^* = P_\Omega[u^* - \beta F(u^*)]. \tag{15}$$

From Lemma 2.2, it is evident that solving $VI(F, \Omega)$ is equivalent to finding a zero point of the residual function

$$e(u, \beta) := u - P_\Omega[u - \beta F(u)]. \tag{16}$$

Lemma 2.3 ([12, 13, 14]) *For all $u \in R^n$ and $0 < \beta_1 \le \beta_2$, it follows that*

$$\|e(u, \beta_1)\| \le \|e(u, \beta_2)\|, \tag{17}$$

and

$$\frac{\|e(u, \beta_1)\|}{\beta_1} \ge \frac{\|e(u, \beta_2)\|}{\beta_2}. \tag{18}$$

3 Algorithms and convergence analysis

The proposed algorithm is as follows.
Algorithm 3.1.(A new self-adaptive projection method)
Step 0. Give $\varepsilon > 0$, choose $u^0 \in \Omega$, $\mu \in (0, 1), \beta_0 = \rho = 1$, $\tau, L \in (0, 1)$, and $\gamma \in (0, 2)$. Set $k := 0$.
Step 1. Compute $e(u^k, \beta_k)$ by (6). If $\|e(u^k, \beta_k)\| < \varepsilon$, stop; otherwise go to step 2.
Step 2. Find the smallest nonnegative integer $m_k, \beta_k = \rho\mu^{m_k}$ such that

$$\beta_k\|F(u^k) - F(u^k - e(u^k, \beta_k))\| \le L\|e(u^k, \beta_k)\|. \tag{19}$$

Step 3. Calculate

$$\alpha_k = \frac{g(u^k, \beta_k)^T e(u^k, \beta_k) + (1 - \frac{\beta_k}{4\tau})\|e(u^k, \beta_k)\|^2}{\|g(u^k, \beta_k) + e(u^k, \beta_k)\|^2} \quad \textbf{(Improvement part)}$$

where $g(u^k, \beta_k)$ is defined by (7).
Step 4. Compute

$$u^{k+1} = P_\Omega[u^k - \gamma\alpha_k d(u^k, \beta_k)] \tag{20}$$

where $d(u^k, \beta_k)$ is defined by (10).
Step 5. If $\beta_k\|F(u^k) - F(u^k - e(u^k, \beta_k))\| \le 0.4\|e(u^k, \beta_k)\|$, then $\rho = \beta_k/0.7$; else $\rho = \beta_k$. Set $k := k + 1$; go to step 1.

Remark 3.1 *It is clear that Algorithm 3.1 and Algorithm 1.2 are almost the same and just distinguish different step size in each iteration.*

Lemma 3.2 ([10]) *If $\|e(u,1)\| \neq 0$, then there exist $0 < L < 1$ and $\tilde{\beta} > 0$, such that*

$$\beta\|F(u) - F(u - e(u,\beta))\| \leq L\|e(u,\beta)\| \quad \forall 0 < \beta \leq \tilde{\beta}. \tag{21}$$

For convenience, u^k, α_k, β_k is represented by u, α, β in this section, respectively.

Lemma 3.3 ([9]) *Under the assumption (21), we have*

$$\begin{aligned} e(u,\beta)^T g(u,\beta)) &\geq (1-L)\|e(u,\beta)\|^2, & (22)\\ e(u,\beta)^T g(u,\beta)) &\geq \frac{1}{2}\|g(u,\beta)\|^2. & (23) \end{aligned}$$

Lemma 3.4 *Assume the condition (21) are met, we have*

$$\alpha^{**} \geq \frac{1-L}{5-4L} > 0 \tag{24}$$

where

$$\alpha^{**} = \frac{g(u,\beta)^T e(u,\beta) + (1-\frac{\beta}{4\tau})\|e(u,\beta)\|^2}{\|g(u,\beta)+e(u,\beta)\|^2}. \tag{25}$$

Proof: Utilizing (22) and (23), we get

$$\begin{aligned} \alpha^{**} &= \frac{g(u,\beta)^T e(u,\beta) + (1-\frac{\beta}{4\tau})\|e(u,\beta)\|^2}{\|g(u,\beta)+e(u,\beta)\|^2} \\ &\geq \frac{g(u,\beta)^T e(u,\beta)}{\|g(u,\beta)\|^2 + \|e(u,\beta)\|^2 + 2g(u,\beta)^T e(u,\beta)} \\ &\geq \frac{1-L}{5-4L} > 0. \end{aligned}$$

This completes the proof. □

From Lemma 3.4, it holds that new step size is bounded away from zero.

Lemma 3.5 ([14]) *Denote $u(\alpha) = P_\Omega(u - \alpha d(u,\beta))$, let u^* be a solution of problem (1), under the assumption $F(\cdot)$ is co-coercive with modulus $\mu > 0$, i.e.,*

$$(v-u)^T(F(u) - F(v)) \geq \mu\|u-v\|^2 \quad \forall u, v \in \Omega, \tag{26}$$

then we get

$$\Theta(\alpha) \geq \Upsilon(\alpha) \tag{27}$$

where

$$\begin{aligned} \Theta(\alpha) &:= \|u-u^*\|^2 - \|u(\alpha)-u^*\|^2, & (28)\\ \Upsilon(\alpha) &:= \Psi(\alpha) + \alpha[2(1-\frac{\beta}{4\tau}) - \alpha]\|e(u,\beta)\|^2, & (29)\\ \Psi(\alpha) &:= 2\alpha d(u,\beta)^T e(u,\beta) - \alpha^2[\|d(u,\beta)\|^2 + 2d(u,\beta)^T e(u,\beta)]. & (30) \end{aligned}$$

Remark 3.6 *As can be seen from formulas (11) and (30), Yan et al.'s step size is based on the manipulation of $\Psi(\alpha)$ by reaching its maximum with respect to α. Intuitively, $\Theta(\alpha)$ represents the progress in each iteration. It is more efficient to manipulate $\Upsilon(\alpha)$ rather than maximizing $\Psi(\alpha)$ because $\Upsilon(\alpha)$ is a direct lower bound of $\Theta(\alpha)$.*

The following lemmas indicate that in each iterative step, we may expect Algorithm 3.1 with new step size derived from maximizing $\Upsilon(\alpha)$ to get more progress than Algorithm 1.2.

Lemma 3.7 ([14]) *Let β_u is upper bound of β. Then, there exists a constant $C_0 > 0$, such that*

$$\Upsilon(\alpha^*) \geq C_0 \|e(u,\beta)\|^2, \tag{31}$$

where

$$\alpha^* = \min\Big\{(1 - \frac{\beta}{4\tau}, \frac{g(u,\beta)^T e(u,\beta)}{\|g(u),\beta)\|^2 + 2e(u,\beta)^T g(u,\beta)}\Big\}. \tag{32}$$

Lemma 3.8 *Let ρ_u is upper bound of ρ. Then, there exists a constant $C_0 > 0$, satisfying*

$$\Theta(\alpha^{**}) \geq \Upsilon(\alpha^{**}) \geq \Upsilon(\alpha^*) \geq C_0 \|e(u,\beta\|^2. \tag{33}$$

where α^* and α^{**} is defined in (32) and (25), respectively.

Proof: Since $\Upsilon(\alpha)$ is a quadratic function with respect to α, from (29) and $\frac{\partial \Upsilon(\alpha)}{\partial \alpha} = 0$ we get

$$\alpha = \frac{g(u,\beta)^T e(u,\beta) + (1 - \frac{\beta}{4\tau})\|e(u,\beta)\|^2}{\|g(u,\beta) + e(u,\beta)\|^2}. \tag{34}$$

Therefore, we get $\Upsilon(\alpha^{**}) \geq \Upsilon(\alpha^*)$, and by Lemma 3.5 and Lemma 3.7, we have

$$\Theta(\alpha^{**}) \geq \Upsilon(\alpha^{**}) \geq \Upsilon(\alpha^*) \geq C_0 \|e(u,\beta)\|^2.$$

This completes the proof. □

In the following, we analyze the convergence of our method.

Theorem 3.9 *Suppose $F(\cdot)$ is co-coercive on Ω, and the solution set Ω^* is nonempty, then the sequence $\{u^k\}$ generated by the Algorithm 3.1 converges to a solution point u^*.*

Proof: From (28), we conclude that (33) is equivalent to

$$\|u^{k+1} - u^*\|^2 \leq \|u^k - u^*\|^2 - C_0 \|e(u^k, \beta_k)\|^2. \tag{35}$$

It follows from (35) that

$$C_0 \sum_{k=0}^{\infty} \|e(u^k, \beta_k)\|^2 \leq \|u^0 - u^*\|^2. \tag{36}$$

Thus we have

$$\lim_{k\to\infty} e(u^k, \beta_k) = 0. \tag{37}$$

Since $\beta_k \geq \beta_{min}$, it follows from (17) that

$$\lim_{k\to\infty} e(u^k, \beta_{min}) = 0. \tag{38}$$

Since $\{u^k\}$ is bounded, it has at least one cluster point. Let u^* be a cluster point of u^k and $\{u^{k_j}\}$ is the subsequence converging to u^*. Because $e(u^k, \beta_{min})$ is continuous, then

$$e(u^*, \beta_{min}) = \lim_{j\to\infty} e(u^{k_j}, \beta_{min}) = 0. \tag{39}$$

Then, if follows from Lemma 2.2 that u^* is a solution of $VI(F, \Omega)$. □

4 Numerical experiments

We implement Algorithm 3.1 in Matlab 7.1 and run on a Lenovo ThinkPad T61 PC, and also test Algorithm 1.2 [14] for comparison of their performance.

To generate a set of test instances we proceeded as follows: we set $F(u) = D(u) + Mu + q$, where $D(u)$ and $Mu + q$ are the nonlinear part and the linear part of $F(u)$, respectively. We form the linear part $Mu + q$ similarly as in [9]l . The matrix $M = A^T A + B$, where A is an $n \times n$ matrix whose entries are randomly generated in the interval $(-5, 5)$ and a skew-symmetric matrix B is generated in the same way. The vector q is generated from a uniform distribution in the interval $(-500, 500)$ or $(-500, 0)$. In $D(u)$, the nonlinear part of $F(u)$, the components are $D_j(u) = a_j \times arctan(u_j)$ and a_j is a random variable in $(0, 1)$. In all tests, we choose $\gamma = 1.8, L = 0.8, \mu = 0.7$, and $\tau = 0.9, \varepsilon = 10^{-6}$. We report the numbers of iteration and the CPU-time for various problem sizes in Tables 1 and 2.

It can be seen that the proposed method can outperform yan et al.'s method dramatically. This can be evidenced by the results in Tables 1 and 2. For example, the number of iterations in Algorithm 3.1 is about 49.8% of that in Algorithm 1.2 in average. The CPU time in Algorithm 3.1 is about 50.3% of that in Algorithm 1.2 in average.

5 Concluding remarks

In this paper, a modified extra-gradient method under new step size for solving a class of co-coercive monotone variational inequalities is proposed on the basis of the algorithm in [14]. In theory, it can be proved that new step size make more progress than step size adopted in [14]. On numerical experiments, the algorithm with new step size can outperform the algorithm in [14].

Table 1: The numerical results for $q \in (-500, 0)$.

n	Algorithm 1.2		Algorithm 3.1	
	No. It.	CPU (s)	No. It.	CPU (s)
100	420	0.035	135	0.018
200	680	0.155	194	0.037
300	548	0.237	187	0.191
400	673	0.525	199	0.438
500	741	1.164	214	0.857
600	722	2.202	207	0.974
700	733	3.297	201	1.243
800	688	4.635	204	2.079
1000	791	7.653	258	3.926
2000	963	51.728	299	26.694

Table 2: The numerical results for $q \in (-500, 500)$.

n	Algorithm 1.2		Algorithm 3.1	
	No. It.	CPU (s)	No. It.	CPU (s)
100	195	0.014	98	0.006
200	264	0.055	134	0.021
300	297	0.112	153	0.049
400	298	0.202	152	0.095
500	348	0.445	171	0.215
600	289	0.807	151	0.400
700	337	1.399	174	0.701
800	283	1.700	140	0.917
1000	321	2.675	164	1.430
2000	425	19.710	220	10.052

References

[1] A.A. Goldstein, *Convex programming in Hilbert space*, Bulletin of the American Mathematical Society 70 (1964) 709–710.

[2] E.S. Levitin, B.T. Polyak, *Constrained minimization problems*, USSR Computational Mathematics and Mathematical Physics 6 (1966) 1–50.

[3] G.M. Korpelevich, *The extragradient method for finding saddle points and other problems*, Matecon 12 (1976) 747–756.

[4] B.S. He, H. Yang, Q. Meng, D.R. Han,*Modified Goldstein-Levitin-Polyak projection method for asymmetric strongly monotone variational inequalities*, Journal of Optimization Theory and Applications 112 (2002) 129–143.

[5] D.R. Han, W.Y. Sun, *A new modified Goldstein-Levitin-Polyak projection method for variational inequality problems*, Computers & Mathematics with Applications 47 (2004) 1817–1825.

[6] B.S. He, *A class of projection and contraction methods for monotone variational inequalities*, Applied Mathematics and Optimization 35 (1997) 69–76.

[7] D.F. Sun, *A class of iterative methods for solving nonlinear projection equations*, Journal of Optimization Theory and Applications 91 (1) (1996) 123–140.

[8] M.V. Solodov, P. Tseng, *Modified projection-type methods for monotone variational inequalities*, SIAM Journal of Control and Optimization 34 (1996) 1814–1830.

[9] B.S. He and L.Z. Liao, *Improvement of some projection methods for monotone nonlinear variational inequalities*, Journal of Opimization Theory and Application 112 (2002) 111–128.

[10] D.R. Han, H.K. Lo, *Two new self-adaptive projection methods for variational inequality problems*, Computers & Mathematics with Applications 43 (2002) 1529–1537.

[11] E.H. Zarantonello, *Projections on convex sets in Hilbert space and spectral theory*, in: E.H. Zarantonello (Ed.), Contributions to Nonlinear Functional Analysis, Academic Press, New York, NY, 1971.

[12] E.H. Gafni, D.P. Bertsekas, *Two-metric projection methods for constrained optimization*, SIAM Journal of Control and Optimization 22 (1984) 1814–1830.

[13] H. Yin, *An iterative method for general variational inequalities*, Journal of Industrial and Management Optimization 1 (2005) 201–209.

[14] X.H. Yan, D.R. Han, W.Y. Sun, *A self-adaptive projection method with improved step-size for solving variational inequalities*, Computers & Mathematics with Applications 55 (2008) 819–832.

[15] M. Li, X.M. Yuan, *An APPA-based descent method with optimal step-sizes for monotonevariational inequalities*, European Journal of Operational Research, 186 (2008) 486–495.

[16] T. Zhu, Z.G. Yu, *A simple proof for some important properties of the projection mapping*, Mathematical Inequalities and Applications 7 (2004) 453–456.

[17] D.P. Bertsekas, J.N. Tsitsiklis, *Parallel and Distributed Computation*, Numerical Methods, Prentice-Hall, Englewood Cliffs, NJ, 1989.