

Proximal-like contraction methods for monotone variational inequalities in a unified framework II: general methods and numerical experiments

Bingsheng He · Li-Zhi Liao · Xiang Wang

Received: 27 May 2009 / Published online: 23 November 2010
© Springer Science+Business Media, LLC 2010

Abstract Approximate proximal point algorithms (abbreviated as APPAs) are classical approaches for convex optimization problems and monotone variational inequalities. In Part I of this paper (He et al. in Proximal-like contraction methods for monotone variational inequalities in a unified framework I: effective quadruplet and primary methods, 2010), we proposed a unified framework consisting of an effective quadruplet and a corresponding accepting rule. Under the framework, various existing APPAs can be grouped in the same class of methods (called primary or elementary methods) which adopt one of the geminate directions in the effective quadruplet and take the unit step size. In this paper, we extend the primary methods by using the same effective quadruplet and the accepting rule. The extended (general) contraction methods need only minor extra even negligible costs in each iteration, whereas having better properties than the primary methods in sense of the distance to the solution set. A set of matrix approximation examples as well as six other groups of numerical experiments are constructed to compare the performance between the primary (elementary) and extended (general) methods. As expected, the numerical results show the efficiency of the extended (general) methods are much better than that of the primary (elementary) ones.

This work was supported by the National Natural Science Foundation of China (Grant No. 10971095), the Natural Science Foundation of Jiangsu Province (BK2008255), the Cultivation Fund of the Key Scientific and Technical Innovation Project, Ministry of Education of China (708044), and the Research Grant Council of Hong Kong.

B. He (✉) · X. Wang
Department of Mathematics, Nanjing University, Nanjing 210093, China
e-mail: hebma@nju.edu.cn

B. He
National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China

L.-Z. Liao
Department of Mathematics, Hong Kong Baptist University, Hong Kong, China

Keywords Variational inequality · Monotone · Contraction methods

1 Introduction

Let Ω be a nonempty closed convex subset of R^n and F be a continuous mapping from R^n into itself. Variational inequality problem is to determine a vector $u^* \in \Omega$ such that

$$VI(\Omega, F) \quad (u - u^*)^T F(u^*) \geq 0, \quad \forall u \in \Omega. \tag{1.1}$$

For any $\beta > 0$, it is well known ([1], p. 267) that

$$u^* \text{ is a solution of } VI(\Omega, F) \iff u^* = P_\Omega[u^* - \beta F(u^*)], \tag{1.2}$$

where $P_\Omega(\cdot)$ denotes the projection onto Ω with respect to the Euclidean norm, i.e.,

$$P_\Omega(v) = \operatorname{argmin}\{\|u - v\| \mid u \in \Omega\}.$$

For solving monotone variational inequality, a classical method is the proximal point algorithm (abbreviated as PPA) [13], while extensive developments on approximate proximal point algorithms (abbreviated as APPAs) are followed [3, 4]. In Part I of this paper [12], we proposed a unified framework consisting of an effective quadruplet and an accepting rule for APPAs. For a symmetric positive definite matrix G and a scalar $\kappa > 0$, the effective quadruplet $(d_1(u^k, v^k, \tilde{u}^k), d_2(u^k, v^k, \tilde{u}^k), \varphi(u^k, v^k, \tilde{u}^k), \phi(u^k, v^k, \tilde{u}^k))$ satisfies

$$\tilde{u}^k = P_\Omega\{\tilde{u}^k - [d_2(u^k, v^k, \tilde{u}^k), -Gd_1(u^k, v^k, \tilde{u}^k)]\} \tag{1.3a}$$

$$(\tilde{u}^k - u^*)^T d_2(u^k, v^k, \tilde{u}^k) \geq \varphi(u^k, v^k, \tilde{u}^k) - (u^k - \tilde{u}^k)^T G d_1(u^k, v^k, \tilde{u}^k), \tag{1.3b}$$

$$\varphi(u^k, v^k, \tilde{u}^k) \geq \frac{1}{2}\{\|d_1(u^k, v^k, \tilde{u}^k)\|_G^2 + \phi(u^k, v^k, \tilde{u}^k)\}, \tag{1.3c}$$

and

$$\phi(u^k, v^k, \tilde{u}^k) \geq \kappa \|u^k - \tilde{u}^k\|^2. \tag{1.3d}$$

In [12], many kinds of VIs and APPAs are studied under the unified framework. By constructing the effective quadruplets and/or the accepting rules for those studied APPAs, all of them can be grouped into the primary (elementary) methods, i.e.,

$$u^{k+1} = u^k - d_1(u^k, v^k, \tilde{u}^k), \quad \text{for all positive definite } G \text{ in (1.3),} \tag{1.4a}$$

$$u^{k+1} = P_\Omega[u^k - d_1(u^k, v^k, \tilde{u}^k)], \quad \text{when } G \text{ in (1.3) is the identity matrix,} \tag{1.4b}$$

and

$$u^{k+1} = P_\Omega[u^k - d_2(u^k, v^k, \tilde{u}^k)], \quad \text{when } G \text{ in (1.3) is the identity matrix.} \tag{1.4c}$$

A clear but important view of these three primary methods is that, all of these three recursions adopt the unit step size. For the primary method (1.4a), we can construct

extended methods with minor extra computational loads for a better step size. An application is proposed to a kind of matrix approximation problems to compare the proximal alternating directions method and its extended one. The numerical experiments strongly demonstrate the efficiency of the extended method.

In case of $G = I$, both $d_1(u^k, v^k, \tilde{u}^k)$ and $d_2(u^k, v^k, \tilde{u}^k)$ are the descent directions of the distance function $\|u - u^*\|^2$ at the current point u^k , and both primary methods (1.4b) and (1.4c) can be applied to solve VIs. It is interesting to compare the performance of these two different directions. For this purpose, according to the unified framework, we show lower bounds for the descent progress $\|u^k - u^*\|^2 - \|u^{k+1} - u^*\|^2$ along these two directions. With the analysis on the lower bounds, better directions as well as step sizes can be adopted. A series of preliminary numerical experiments are constructed for comparing the performances with different directions and step sizes. With selected step sizes and the better directions, the efficiencies are improved significantly.

The projection mapping is a tool in the analysis of this paper. We list two used properties of projections in Lemmas 1.1 and 1.2. The proof of Lemma 1.1 can be found in textbooks, e.g., [2]. For a simple proof of Lemma 1.2, the readers can consult [16].

Lemma 1.1 *Let $\Omega \subset R^n$ be a closed convex set, then we have*

$$\|u - P_\Omega(u')\|^2 \leq \|u - u'\|^2 - \|u' - P_\Omega(u')\|^2, \quad \forall u' \in R^n, \forall u \in \Omega. \quad (1.5)$$

Lemma 1.2 *Let $\Omega \subset R^n$ be a closed convex set, u and d be any given vectors in R^n . Then $\|P_\Omega(u - td) - u\|$ is a non-decreasing function of t for $t \geq 0$.*

The rest of the paper is organized as follows. In Sect. 2, we propose a simple extended version of the primary method (1.4a) according to the same effective quadruplet. We then compare the proximal alternating directions method and its extended version through a set of numerical experiments on a certain kind of matrix approximation problems. The numerical results strongly demonstrate the efficiency of the extended method. In Sect. 3, we present the general contraction methods, which generalize the descent directions used in the primary methods (1.4b) and (1.4c). Such generalized descent directions can be any convex combination of the geminate directions. For the efficiency of the general methods, theoretical considerations are then provided, which address the better and more reasonable directions as well as step sizes. In Sect. 4, the theoretical results obtained in Sect. 3 are verified by a series of numerical experiments for the four types of monotone VIs, see Table 2.1 in [12]. Finally, some concluding remarks are drawn in Sect. 5.

2 Simple extended contraction method and its numerical results

For the primary method (1.4a), the new iterate can be computed straightforwardly with the obtained $d_1(u^k, v^k, \tilde{u}^k)$ and the current point u^k . It needs no additional projections, which are involved in the other two primary methods (1.4b) and (1.4c). In

some cases, computing projections would be time consuming, e.g., the projection of the symmetric matrix onto the positive semi-definite cone. In this section, we consider the first kind primary methods (1.4a) and give its simple extension with the application to the proximal alternating directions methods.

2.1 The extended contraction method

Under the unified framework, instead of the primary method (1.4a), we take the new iterate by

$$\text{(Extended contraction method)} \quad u^{k+1} = u^k - \gamma \alpha_k^* d_1(u^k, v^k, \tilde{u}^k), \quad (2.1)$$

where

$$\alpha_k^* = \frac{\varphi(u^k, v^k, \tilde{u}^k)}{\|d_1(u^k, v^k, \tilde{u}^k)\|_G^2} \quad \text{and} \quad \gamma \in [1, 2). \quad (2.2)$$

Involving only a few extra computational loads for the step size, the extended method needs no additional projections also. Note that from (1.3c) and (1.3d) we have

$$\alpha_k^* \geq \frac{1}{2}, \quad (2.3)$$

whenever u^k is not a solution point. With this lower bounded step size, the extended contraction method has the following contraction property.

Theorem 2.1 *Assume that the quadruplet $(d_1(u^k, v^k, \tilde{u}^k), d_2(u^k, v^k, \tilde{u}^k), \varphi(u^k, v^k, \tilde{u}^k), \phi(u^k, v^k, \tilde{u}^k))$ is effective under a certain accepting rule and the sequence $\{u^k\}$ is generated by the extended method (2.1)–(2.2). Then we have*

$$\|u^{k+1} - u^*\|_G^2 \leq \|u^k - u^*\|_G^2 - \frac{\gamma(2 - \gamma)}{2} \varphi(u^k, v^k, \tilde{u}^k), \quad \forall u^* \in \Omega^*. \quad (2.4)$$

Proof Since $u^* \in \Omega$, it follows from (2.1)–(2.2) and Lemma 2.1 in [12] that

$$\begin{aligned} \|u^{k+1} - u^*\|_G^2 &= \|u^k - \gamma \alpha^* d_1(u, v, \tilde{u}) - u^*\|_G^2 \\ &= \|u - u^*\|_G^2 - 2\gamma \alpha^* (u - u^*)^T G d_1(u, v, \tilde{u}) + (\gamma \alpha^*)^2 \|d_1(u, v, \tilde{u})\|_G^2 \\ &\leq \|u - u^*\|_G^2 - 2\gamma \alpha^* \varphi(u, v, \tilde{u}) + (\gamma \alpha^*)^2 \|d_1(u, v, \tilde{u})\|_G^2 \\ &= \|u - u^*\|_G^2 - \gamma(2 - \gamma) \alpha^* \varphi(u, v, \tilde{u}). \end{aligned}$$

The assertion (2.4) follows from the above inequality and $\alpha_k^* > \frac{1}{2}$ immediately. \square

The convergence of the extended method can be proved as the proof of Theorem 2.1 in [12]. We have shown in [12] that, both Solodov-Svaiter’s APPA (see Algorithm 2 in [14]) and the proximal alternating directions method (see [9]) can be

viewed as the primary method (1.4a). In [10], the updating form of Solodov-Svaiter’s APPA is extended by

$$u^{k+1} = u^k - \alpha_k(u^k - \tilde{u}^k).$$

This recursion is just the extended contraction method (2.1)–(2.2) related to the unified framework. Numerical experiments in [10] show that, such extended method is much more efficient than its primary version, i.e., Solodov-Svaiter’s APPA. In this section, the efficiency of the extended method can be reconfirmed by comparison on the performance of the proximal alternating directions method and its extended version.

2.2 Test problems and the equivalent structured variational inequality

As an application of the proximal alternating directions method in [12] (see Sect. 8 therein), we consider the matrix approximation problems arising from finance and statistics. We form the test problems similarly as in [5]. Let H_L, H_U and C be given $n \times n$ symmetric matrices, $H_L \leq H_U$ in element-wise. The problem considered in this subsection is

$$\min \left\{ \frac{1}{2} \|X - C\|_F^2 \mid X \in S_+^n \cap \mathcal{B} \right\}, \tag{2.5}$$

where $\|\cdot\|_F$ is the matrix Fröbenius norm, i.e., $\|C\|_F = (\sum_{i=1}^n \sum_{j=1}^n |C_{ij}|^2)^{1/2}$,

$$S_+^n = \{H \in R^{n \times n} \mid H^T = H, H \geq 0\}$$

is the semi-definite cone and

$$\mathcal{B} = \{H \in R^{n \times n} \mid H_L \leq H \leq H_U\}. \tag{2.6}$$

Note that the matrix Fröbenius norm is induced by the inner product

$$\langle A, B \rangle = \text{Trace}(A^T B).$$

We convert the problem (2.5) to the following equivalent one:

$$\begin{aligned} \min \quad & \frac{1}{2} \|X - C\|_F^2 + \frac{1}{2} \|Y - C\|_F^2 \\ \text{s.t} \quad & X - Y = 0, \\ & X \in S_+^n, \quad Y \in \mathcal{B}. \end{aligned} \tag{2.7}$$

The mathematical form of the equivalent structured variational inequality is to find

$$u = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \in \Omega, \quad \begin{cases} \langle X' - X, (X - C) - Z \rangle \geq 0, \\ \langle Y' - Y, (Y - C) + Z \rangle \geq 0, \\ X - Y = 0, \end{cases} \quad \forall u' = \begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} \in \Omega, \tag{2.8}$$

where

$$\Omega = S_+^n \times \mathcal{B} \times R^{n \times n}. \tag{2.9}$$

2.3 Implementation of the proximal alternating directions method

If we use the proximal alternating directions method in [12] to solve the structured variational inequality (2.8)–(2.9), for given triplet $u^k = (X^k, Y^k, Z^k)$, $\tilde{u}^k = (\tilde{X}^k, \tilde{Y}^k, \tilde{Z}^k)$ is obtained by

$$\begin{aligned} \tilde{X}^k &\in S_+^n, & \langle X' - \tilde{X}^k, (\tilde{X}^k - C) - [Z^k - \beta(\tilde{X}^k - Y^k)] + r(\tilde{X}^k - X^k) \rangle &\geq 0, \\ \forall X' &\in S_+^n, & & \end{aligned} \tag{2.10a}$$

$$\begin{aligned} \tilde{Y}^k &\in \mathcal{B}, & \langle Y' - \tilde{Y}^k, (\tilde{Y}^k - C) + [Z^k - \beta(\tilde{X}^k - \tilde{Y}^k)] + s(\tilde{Y}^k - Y^k) \rangle &\geq 0, \\ \forall Y' &\in \mathcal{B}, & & \end{aligned} \tag{2.10b}$$

and

$$\tilde{Z}^k = Z^k - \beta(\tilde{X}^k - \tilde{Y}^k). \tag{2.10c}$$

Using the principle (1.2) and the special structures of (2.10a) and (2.10b), \tilde{X}^k and \tilde{Y}^k can be directly obtained by

$$\tilde{X}^k = P_{S_+^n} \left\{ \frac{1}{1 + \beta + r} (\beta Y^k + Z^k + C + r X^k) \right\} \tag{2.11}$$

and

$$\tilde{Y}^k = P_{\mathcal{B}} \left\{ \frac{1}{1 + \beta + s} (\beta \tilde{X}^k - Z^k + C + s Y^k) \right\}, \tag{2.12}$$

respectively. For given symmetric matrix $A \in R^{n \times n}$, let the eigenvalue decomposition be

$$A = V \Lambda V^T, \tag{2.13}$$

where $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$. Then

$$P_{S_+^n}(A) = V \tilde{\Lambda} V^T, \tag{2.14}$$

where

$$\tilde{\Lambda} = \text{diag}(\tilde{\lambda}_1, \dots, \tilde{\lambda}_n), \quad \tilde{\lambda}_i = \max\{0, \lambda_i\}.$$

Note that the computational loads of (2.13) and (2.14) are about $9n^3$ and n^3 flops, respectively [6]. The projection $P_{\mathcal{B}}(A)$ is easy to be carried out, namely, in element-wise

$$P_{\mathcal{B}}(A) = \max\{H_L, \min\{A, H_U\}\}.$$

For this matrix optimization problem, using the notations in Sect. 8 of [12], we have

$$u = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad \text{and} \quad G = \begin{pmatrix} rI & & \\ & (\beta + s)I & \\ & & \frac{1}{\beta}I \end{pmatrix}. \tag{2.15}$$

According to the effective quadruplet defined in Theorem 8.1 of [12], for this problem

$$d_1(u, v, \tilde{u}) = u - \tilde{u}, \tag{2.16}$$

and

$$\varphi(u, v, \tilde{u}) = r\|X - \tilde{X}\|_F^2 + (\beta + s)\|Y - \tilde{Y}\|_F^2 + \frac{1}{\beta}\|Z - \tilde{Z}\|_F^2 - \langle Y^k - \tilde{Y}^k, Z^k - \tilde{Z}^k \rangle. \tag{2.17}$$

In line with primary method (1.4a),

$$u^{k+1} = u^k - d_1(u^k, v^k, \tilde{u}^k).$$

Since $d_1(u, v, \tilde{u}) = u - \tilde{u}$, the proximal alternating directions method adopts \tilde{u}^k as the new iterate. And the iteration form can be written as

$$\begin{pmatrix} X^{k+1} \\ Y^{k+1} \\ Z^{k+1} \end{pmatrix} = \begin{pmatrix} X^k \\ Y^k \\ Z^k \end{pmatrix} - \begin{pmatrix} X^k - \tilde{X}^k \\ Y^k - \tilde{Y}^k \\ Z^k - \tilde{Z}^k \end{pmatrix} = \begin{pmatrix} \tilde{X}^k \\ \tilde{Y}^k \\ \tilde{Z}^k \end{pmatrix}.$$

It is the computation of \tilde{X}^k (which is about $10n^3$ flops) that the most time consuming operation is among the computations of the above three new iterate matrices.

2.4 Implementation of the extended contraction method

Let $(\tilde{X}^k, \tilde{Y}^k, \tilde{Z}^k)$ be generated by (2.10) from given (X^k, Y^k, Z^k) . In line with the extended contraction method with updating form (2.1)–(2.2), the new iterate is generated by

$$\begin{pmatrix} X^{k+1} \\ Y^{k+1} \\ Z^{k+1} \end{pmatrix} = \begin{pmatrix} X^k \\ Y^k \\ Z^k \end{pmatrix} - \gamma \alpha_k^* \begin{pmatrix} X^k - \tilde{X}^k \\ Y^k - \tilde{Y}^k \\ Z^k - \tilde{Z}^k \end{pmatrix}. \tag{2.18}$$

According to (2.2) and (2.15)–(2.17), we have

$$\alpha_k^* = \frac{\|u^k - \tilde{u}^k\|_G^2 - \langle Y^k - \tilde{Y}^k, Z^k - \tilde{Z}^k \rangle}{\|u^k - \tilde{u}^k\|_G^2}$$

where

$$\|u^k - \tilde{u}^k\|_G^2 = r\|X^k - \tilde{X}^k\|_F^2 + (\beta + s)\|Y^k - \tilde{Y}^k\|_F^2 + \frac{1}{\beta}\|Z^k - \tilde{Z}^k\|_F^2.$$

Note that the computational load for α_k^* is $4n^2$. In comparison with the cost for getting \tilde{X}^k in (2.11), the extra work in the extended contraction method is negligible.

Remark 2.1 The procedures (2.10) and (2.18) are only provided for the comparison of the primary proximal alternating directions method and its extended version with the effective quadruplet proposed in Sect. 8 in [12]. For solving the similar optimization problem (2.5) (with less than 5% inequality constraints on the off-diagonal elements), the most recent method in [5] reported the better numerical performance.

Table 1 Numerical results for $r = s = 1, \beta = 10$ and $\gamma = 1.5$

$n \times n$ Matrix	Primary Method		Extended Method	
	Updating form (1.4a)		Updating form (2.1)	
n	No. It	CPU Sec.	No. It	CPU Sec.
100	71	1.07	46	0.85
200	67	4.08	44	2.97
500	79	40.44	49	28.28
1000	91	367.46	56	250.58

2.5 Numerical results

The tested problems have the following structures.

- The entries of diagonal elements of C are randomly generated in the interval $(0, 2)$, the entries of off-diagonal elements of C are randomly generated in the interval $(-1, 1)$.
- $(H_U)_{jj} = (H_L)_{jj} = 1$, and $(H_U)_{ij} = -(H_L)_{ij} = 0.1, \forall i \neq j, i, j = 1, 2, \dots, n$.

We take $u^0 = (X^0, Y^0, Z^0) = (I_n, I_n, 0_n)$ as the initial point in the tests. The code was written in MATLAB and run on a Lenovo X200 Computer with 2.53 GHz. The iteration is stopped whenever

$$\frac{\max(\text{abs}(u^k - \tilde{u}^k))}{\max(\text{abs}(u^0 - \tilde{u}^0))} \leq \varepsilon = 10^{-6}.$$

For $r = s = 1, \beta = 10$ and the relaxation factor $\gamma = 1.5$, Table 1 reports the iteration numbers and the CPU times of the two methods. Since the complexity of each iteration is $O(n^3)$ (about $10n^3$), the CPU time is proportional to the product of the iteration number by n^3 . From Table 1 we can find

$$\frac{\text{It. No. of the extended contraction method}}{\text{It. No. of the proximal alternating directions method}} \approx 0.60.$$

The performance improvement by the extended contraction method is significant.

For these tested problems, Table 2 shows the influence of the different proximal coefficients. The iteration numbers of the extended contraction method by using different relaxation factors are reported in Table 3. As the numerical experiences in [8, 11], a good experiential choice of the relaxation factor is $\gamma \in [1.2, 1.8]$.

3 The general contraction methods

In the case $G = I$, the primary methods (1.4b) and (1.4c) take the similar iterations. Besides the primary methods, this section considers the construction of the *general*

¹Since $-S^{-1}d(u, v, \tilde{u}, t)$ is a descent direction of $\|u - u^*\|_S^2$ for any given symmetric positive definite matrix S , the analysis for the general case $u(\alpha, t) = P_{\Omega, S}[u - \alpha S^{-1}d(u, v, \tilde{u}, t)]$ is similar.

Table 2 Iteration numbers for proximal coefficients $r = s = 1$ and different β

$n \times n$ Matrix	Primary Method				Extended Method			
	Updating form (1.4a)				Updating form (2.1), $\gamma = 1.5$			
n	$\beta = 2$	$\beta = 10$	$\beta = 30$	$\beta = 50$	$\beta = 2$	$\beta = 10$	$\beta = 30$	$\beta = 50$
100	285	71	167	269	187	46	108	176
200	371	67	149	240	245	44	96	156
500	443	79	146	235	292	49	94	153
1000	499	91	149	240	329	56	95	156

Table 3 Iteration numbers for different relaxation factor γ with $r = s = 1$ and $\beta = 10$

$n \times n$ Matrix	Extended method with updating form (2.1)						
	$\gamma = 0.8$	$\gamma = 1.0$	$\gamma = 1.2$	$\gamma = 1.5$	$\gamma = 1.8$	$\gamma = 1.9$	$\gamma = 2.0$
100	85	71	57	46	69	145	div
200	82	67	54	44	71	147	div
500	97	79	63	49	69	145	div
1000	111	91	72	56	68	140	div

contraction methods. In the general methods, instead of $d_1(u, v, \tilde{u})$ and/or $d_2(u, v, \tilde{u})$, we use their convex combination

$$d(u, v, \tilde{u}, t) = (1 - t)d_1(u, v, \tilde{u}) + td_2(u, v, \tilde{u}) \quad t \in [0, 1] \tag{3.1}$$

as the search direction. Since $G = I$, it follows from Lemmas 2.1 and 2.2 in [12] that $-d(u, v, \tilde{u}, t)$ is a descent direction of the unknown distance function $\|u - u^*\|^2$ for any $u \in \Omega \setminus \Omega^*$. Let the new iterate be given by

$$u(\alpha, t) = P_\Omega[u - \alpha d(u, v, \tilde{u}, t)]^1. \tag{3.2}$$

We discuss how to select a reasonable step length α and analyze which direction is better. For these purposes, we define

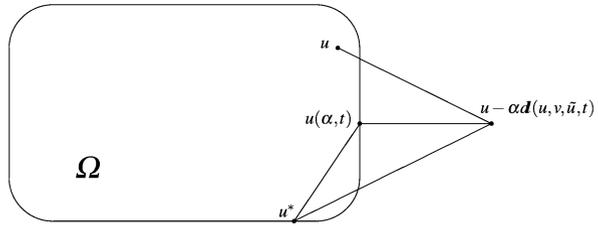
$$\theta(\alpha, t) = \|u - u^*\|^2 - \|u(\alpha, t) - u^*\|^2 \tag{3.3}$$

as the progress function in the k -th iteration. In order to achieve more progress in each iteration, the ideal thought is to maximize $\theta(\alpha, t)$. Unfortunately, because $\theta(\alpha, t)$ involves the unknown vector u^* , we cannot maximize it directly. The following theorem provides a lower bound for $\theta(\alpha, t)$, namely, $\vartheta(\alpha, t)$, which does not include the unknown solution u^* .

Theorem 3.1 For any $u^* \in \Omega^*$, $t \in [0, 1]$ and $\alpha \geq 0$, we have

$$\theta(\alpha, t) \geq \vartheta(\alpha, t), \tag{3.4}$$

Fig. 1 Geometric interpretation of inequality (3.8) with $G = I$



where

$$\vartheta(\alpha, t) = q(\alpha) + \varpi(\alpha, t), \tag{3.5}$$

$$q(\alpha) = 2\alpha\varphi(u, v, \tilde{u}) - \alpha^2\|d_1(u, v, \tilde{u})\|^2 \tag{3.6}$$

and

$$\varpi(\alpha, t) = \|u(\alpha, t) - [u - \alpha d_1(u, v, \tilde{u})]\|^2. \tag{3.7}$$

Proof First, since $u(\alpha, t) = P_\Omega[u - \alpha d(u, v, \tilde{u}, t)]$ and $u^* \in \Omega$, it follows from (1.5) that

$$\|u(\alpha, t) - u^*\|^2 \leq \|u - \alpha d(u, v, \tilde{u}, t) - u^*\|^2 - \|u - \alpha d(u, v, \tilde{u}, t) - u(\alpha, t)\|^2. \tag{3.8}$$

Consequently, using the definition of $\theta(\alpha, t)$, we get

$$\begin{aligned} \theta(\alpha, t) &\geq \|u - u^*\|^2 - \|u - \alpha d(u, v, \tilde{u}, t) - u^*\|^2 + \|u - \alpha d(u, v, \tilde{u}, t) - u(\alpha, t)\|^2 \\ &= \|u - u(\alpha, t)\|^2 + 2\alpha(u - u^*)^T d(u, v, \tilde{u}, t) + 2\alpha(u(\alpha, t) - u)^T d(u, v, \tilde{u}, t) \\ &= 2\alpha(u(\alpha, t) - u^*)^T d(u, v, \tilde{u}, t) + \|u - u(\alpha, t)\|^2. \end{aligned} \tag{3.9}$$

Since $G = I$, it follows from Lemma 2.1 of [12] that

$$(u(\alpha, t) - u^*)^T d_1(u, v, \tilde{u}) \geq \varphi(u, v, \tilde{u}) + (u(\alpha, t) - u)^T d_1(u, v, \tilde{u}).$$

In addition, setting $u' = u(\alpha)$ in (2.8) of [12] and using $G = I$, we have

$$(u(\alpha, t) - u^*)^T d_2(u, v, \tilde{u}) \geq \varphi(u, v, \tilde{u}) + (u(\alpha, t) - u)^T d_1(u, v, \tilde{u}).$$

Since $d(u, v, \tilde{u}, t)$ is a convex combination of $d_1(u, v, \tilde{u})$ and $d_2(u, v, \tilde{u})$, it follows from the above two inequalities that

$$(u(\alpha, t) - u^*)^T d(u, v, \tilde{u}, t) \geq \varphi(u, v, \tilde{u}) + (u(\alpha, t) - u)^T d_1(u, v, \tilde{u}). \tag{3.10}$$

Substituting (3.10) into the right-hand-side of (3.9), we obtain

$$\begin{aligned} \theta(\alpha, t) &\geq 2\alpha\varphi(u, v, \tilde{u}) + 2\alpha(u(\alpha, t) - u)^T d_1(u, v, \tilde{u}) + \|u - u(\alpha, t)\|^2 \\ &= 2\alpha\varphi(u, v, \tilde{u}) - \alpha^2\|d_1(u, v, \tilde{u})\|^2 + \|u - u(\alpha, t) - \alpha d_1(u, v, \tilde{u})\|^2 \\ &= q(\alpha) + \varpi(\alpha, t). \end{aligned} \tag{3.11}$$

The proof is complete. □

In general, $\vartheta(\alpha, t)$ is a tight lower bound of $\theta(\alpha, t)$ (for an example, see [11]). Note that $\vartheta(\alpha, t)$ involves two parts. The first part, $q(\alpha)$, offers us the rule to determine the step length and the second part, $\varpi(\alpha, t)$, tells us how to choose the search directions.

3.1 Selecting reasonable step lengths and the convergence

When $\Omega = R^n$ and $d(u, v, \tilde{u}, 0) = d_1(u, v, \tilde{u})$, we have $\varpi(\alpha, 0) = 0$ (see (3.2)) for any $\alpha \geq 0$. Therefore, in the process of determining the step length α , we ignore $\varpi(\alpha, t)$ and use the function $q(\alpha)$ only. Note that $q(\alpha)$ (independent of t) is a quadratic of α , it reaches its maximum at

$$\alpha^* = \frac{\varphi(u, v, \tilde{u})}{\|d_1(u, v, \tilde{u})\|^2}. \tag{3.12}$$

Clearly, α^* is just the same as in the extended contraction method (2.2). For $u^k \notin \Omega^*$, it follows from condition (1.3c) and (1.3d) that

$$\alpha^* = \frac{\varphi(u, v, \tilde{u})}{\|d_1(u, v, \tilde{u})\|^2} \geq \frac{\varphi(u, v, \tilde{u})}{\|d_1(u, v, \tilde{u})\|^2 + \phi(u^k, v^k, \tilde{u}^k)} \geq \frac{1}{2}.$$

The step size α^* is only dependent on $\varphi(u, v, \tilde{u})$ and $d_1(u, v, \tilde{u})$, regardless of which convex combination factor t is selected. In other words, it is an ‘optimal’ unified step size that we can use along various search directions.

Since some inequalities are used in the proof of Theorem 3.1, the ‘optimal’ step size (3.12) is usually conservative for contraction methods. We can use a relaxation factor $\gamma \in (0, 2)$ and the new iterate is updated by

$$u^{k+1} = P_\Omega[u^k - \gamma\alpha_k^*d(u^k, v^k, \tilde{u}^k, t)]. \tag{3.13}$$

For updating form (3.13), using Theorem 3.1, by a simple manipulation we get

$$\begin{aligned} \|u^{k+1} - u^*\|^2 &\leq \|u^k - u^*\|^2 - q(\gamma\alpha_k^*) \\ &= \|u^k - u^*\|^2 - 2\gamma\alpha_k^*\varphi(u^k, v^k, \tilde{u}^k) + \gamma^2(\alpha_k^*)^2\|d_1(u^k, v^k, \tilde{u}^k)\|^2 \\ &= \|u^k - u^*\|^2 - \gamma(2 - \gamma)\alpha_k^*\varphi(u^k, v^k, \tilde{u}^k). \end{aligned} \tag{3.14}$$

Since $\alpha_k^* > \frac{1}{2}$, from (3.14) we obtain

$$\|u^{k+1} - u^*\|^2 \leq \|u^k - u^*\|^2 - \frac{\gamma(2 - \gamma)}{2}\varphi(u^k, v^k, \tilde{u}^k). \tag{3.15}$$

As a conclusion, we get the following convergence results.

Theorem 3.2 *For given $u^k \in \Omega$ and $\beta_k \geq \beta_L > 0$, assume that the quadruplet $(d_1(u^k, v^k, \tilde{u}^k), d_2(u^k, v^k, \tilde{u}^k), \varphi(u^k, v^k, \tilde{u}^k), \phi(u^k, v^k, \tilde{u}^k))$ is effective under a certain accepting rule and the sequence $\{u^k\}$ is generated by the general methods. In*

addition, if

$$\lim_{k \rightarrow \infty} \{d_2(u^k, v^k, \tilde{u}^k) - \beta_k F(\tilde{u}^k)\} = 0, \tag{3.16}$$

then $\{u^k\}$ converges to some u^∞ which is a solution point of $VI(\Omega, F)$.

Proof Based on the proof of Theorem 2.1 in [12], we need only to verify

$$\lim_{k \rightarrow \infty} d_1(u^k, v^k, \tilde{u}^k) = 0. \tag{3.17}$$

From (3.14), it follows that

$$\lim_{k \rightarrow \infty} \varphi(u^k, v^k, \tilde{u}^k) = 0$$

and (3.17) is satisfied due to the condition (1.3c). The proof is complete. □

According to our numerical experience, it is recommended to set $\gamma \in [1, 2)$. Instead of choosing a fixed $\gamma \in (0, 2)$, we could adopt a dynamical relaxation factor $\gamma_k = 1/\alpha_k^*$ in (3.13),

$$u^{k+1} = P_\Omega[u^k - \mathbf{d}(u^k, v^k, \tilde{u}^k, t)]. \tag{3.18}$$

For this updating form, it follows from (3.14) and (3.12) that

$$\begin{aligned} \|u^{k+1} - u^*\|^2 &\leq \|u^k - u^*\|^2 - \gamma(2 - \gamma)\alpha_k^* \varphi(u^k, v^k, \tilde{u}^k) \\ &= \|u^k - u^*\|^2 - (2 - 1/\alpha_k^*)\varphi(u^k, v^k, \tilde{u}^k) \\ &= \|u^k - u^*\|^2 - (2\varphi(u^k, v^k, \tilde{u}^k) - \|d_1(u, v, \tilde{u})\|^2) \\ &\leq \|u^k - u^*\|^2 - \phi(u^k, v^k, \tilde{u}^k). \end{aligned}$$

This is the same result as the ones in Propositions 2.1 and 2.2 in [12]. Thus (3.18) can be viewed as a primary method.

3.2 Choosing the better directions

This subsection focuses on which direction is better for getting more progress in the k -th iteration. Since the first part of $\vartheta(\alpha, t)$, namely $q(\alpha)$, is independent of t (see (3.6)), we need only to investigate the magnitude $\varpi(\alpha, t)$ for different $t \in [0, 1]$.

Theorem 3.3 *For any $\alpha \geq 0$ and $t \in [0, 1]$, $\varpi(\alpha, t)$ is a nondecreasing function of t . Especially, we have*

$$\varpi(\alpha, t) - \varpi(\alpha, 0) \geq \|u(\alpha, t) - u(\alpha, 0)\|^2, \quad \forall \alpha \geq 0. \tag{3.19}$$

Proof Note that $\mathbf{d}(u, v, \tilde{u}, t)$ in (3.1) can be rewritten as

$$\mathbf{d}(u, v, \tilde{u}, t) = d_1(u, v, \tilde{u}) + t(d_2(u, v, \tilde{u}) - d_1(u, v, \tilde{u})), \tag{3.20}$$

and thus $u(\alpha, t)$ in (3.2) is

$$u(\alpha, t) = P_\Omega\{[u - \alpha d_1(u, v, \tilde{u})] - t\alpha[d_2(u, v, \tilde{u}) - d_1(u, v, \tilde{u})]\}.$$

By using the notation

$$\bar{u}(\alpha) = u - \alpha d_1(u, v, \tilde{u}), \tag{3.21}$$

we have (see (3.2))

$$u(\alpha, t) = P_\Omega\{\bar{u}(\alpha) - t\alpha[d_2(u, v, \tilde{u}) - d_1(u, v, \tilde{u})]\},$$

and (see (3.7))

$$\varpi(\alpha, t) = \|P_\Omega\{\bar{u}(\alpha) - t\alpha[d_2(u, v, \tilde{u}) - d_1(u, v, \tilde{u})]\} - \bar{u}(\alpha)\|^2.$$

It follows from Lemma 1.2 that $\varpi(\alpha, t)$ is a nondecreasing function of t for $t \geq 0$. According to (3.7) and (3.21)

$$\varpi(\alpha, t) - \varpi(\alpha, 0) = \|u(\alpha, t) - \bar{u}(\alpha)\|^2 - \|u(\alpha, 0) - \bar{u}(\alpha)\|^2. \tag{3.22}$$

Note that (see the notation (3.2), (3.20) and (3.21))

$$u(\alpha, 0) = P_\Omega[\bar{u}(\alpha)]. \tag{3.23}$$

Since $u(\alpha, t) \in \Omega$, by using (1.5) we obtain

$$\|u(\alpha, t) - u(\alpha, 0)\|^2 \leq \|\bar{u}(\alpha) - u(\alpha, t)\|^2 - \|\bar{u}(\alpha) - u(\alpha, 0)\|^2. \tag{3.24}$$

The assertion of this theorem follows directly from (3.22) and (3.24). □

Theorem 3.3 indicates that in each iterative step, we may expect the contraction methods using the direction $d_2(u, v, \tilde{u})$ will get more progress than the one using $d_1(u, v, \tilde{u})$. Abide by the recommendation proposed in the previous subsection, the new iterate should be updated by

$$u^{k+1} = P_\Omega[u^k - \gamma\alpha_k^* d_2(u^k, v^k, \tilde{u}^k)], \tag{3.25}$$

where $\gamma \in (0, 2)$ and α_k^* is defined in (3.12). Note that any relaxation factor $\gamma \in (0, 2)$ could be used in (3.25). Nevertheless, the experiments [8, 11] have shown that better numerical results could be obtained by setting $\gamma \in [1.8, 1.95]$.

4 Numerical experiments

Based on the effective quadruplets in Sects. 3–6 of [12], this section compares the primary methods and the related general methods described in Sect. 3. The quadruplets and the accepting rules will be satisfied by choosing a suitable parameter β_k . In

Table 4 The mapping F and the geminate directions d_1 and d_2 in [12]

Type	$F(u)$	$d_1(u, u, \tilde{u})$ and which is given in	$d_2(u, u, \tilde{u})$ and which is given in
Type 1	$Hu + q$	$(u - \tilde{u})$ (3.4) of [12]	$\beta(Hu + q)$ (3.5) of [12]
Type 2	$Mu + q$	$(u - \tilde{u}) + \beta M^T(u - \tilde{u})$ (4.4) of [12]	$\beta(Mu + q) + \beta M^T(u - \tilde{u})$ (4.5) of [12]
Type 3	$\nabla f(u)$	$(u - \tilde{u})$ (5.4) of [12]	$\beta F(u)$ (5.5) of [12]
Type 4	$F(u)$	$(u - \tilde{u}) - \beta(F(u) - F(\tilde{u}))$ (6.2) of [12]	$\beta F(\tilde{u})$ (6.3) of [12]

other words, set $v^k = u^k$ in (1.6) of [12] and the basic equation of APPAs ((1.7) in [12]) is simplified to

$$\tilde{u}^k = P_\Omega[u^k - \beta_k F(u^k)].$$

Thus, the main computational load in each iteration of the methods is the evaluation of the mapping F . In this case, the geminate directions $d_1(u, u, \tilde{u})$ and $d_2(u, u, \tilde{u})$ for different types of VIs in [12] are reduced as in Table 4.

We differentiate the test methods by using the following abbreviations:

- L and NL are abbreviations for Linear VI and Nonlinear VI, respectively.
- The directions $d_1(u, v, \tilde{u})$ and $d_2(u, v, \tilde{u})$ are abbreviated as D1 and D2, respectively;
- P and G denote the primary methods and the general methods, respectively.

Remark 4.1 The VIs of Types 1 and 3 are equivalent to the constrained convex optimization problems and are called as symmetric VIs. For such problems, since

$$\tilde{u}^k = P_\Omega[u^k - \beta_k F(u^k)], \quad G = I$$

and

$$d_1(u^k, v^k, \tilde{u}^k) = u^k - \tilde{u}^k,$$

the three update forms of the primary method are equal (see (1.4)). In other words, $u^{k+1} = \tilde{u}^k$ is the new iterate. Therefore, for such optimization problems, we suggest only to use the primary methods. We use the abbreviations

$$\text{SLD-P} \quad \text{and} \quad \text{SNLD-P}$$

for the primary methods for symmetric linear VIs and symmetric nonlinear VIs, respectively.

Remark 4.2 Only for the asymmetric VIs of Type 2 and 4, we test the primary methods and the general methods. All of the recursions are in the form

$$u^{k+1} = P_\Omega[u^k - \alpha_k d(u^k, v^k, \tilde{u}^k)].$$

In the primary methods, $\alpha_k \equiv 1$, while in the general methods (see (3.12)),

$$\alpha_k = \gamma \alpha_k^*, \quad \text{where } \alpha_k^* = \frac{\varphi(u, v, \tilde{u})}{\|d_1(u, v, \tilde{u})\|^2} \text{ and } \gamma = 1.8.$$

Table 5 The functions $\varphi(u, u, \tilde{u})$ for asymmetric VIs

Type	$F(u)$	$\varphi(u, u, \tilde{u})$ and which is given in
Type 2	$Mu + q$	$\ u - \tilde{u}\ ^2$ (4.8) of [12]
Type 4	$F(u)$	$\ u - \tilde{u}\ ^2 - (u - \tilde{u})^T \beta(F(u) - F(\tilde{u}))$ (6.5) of [12]

Table 6 The marks of different methods for asymmetric VIs

Step size α	VIs of Type 2		VIs of Type 4		VIs of Type 1	VIs of Type 3
	$d_1(u, v, \tilde{u})$	$d_2(u, v, \tilde{u})$	$d_1(u, v, \tilde{u})$	$d_2(u, v, \tilde{u})$		
$\alpha_k \equiv 1$	LD1-P	LD2-P	NLD1-P	NLD2-P	SLD-P	SNLD-P
$\alpha_k = \gamma \alpha_k^*$	LD1-G	LD2-G	NLD1-G	NLD2-G	–	–

Table 7 The test order for VIs of different types

Test order	VIs of Type	$F(u)$	In Sect.
1	Type 4	$F(u)$	Sect. 4.1
2	Type 3	$\nabla f(u)$	Sect. 4.2
3	Type 2	$Mu + q$	Sect. 4.3
4	Type 1	$Hu + q$	Sect. 4.4

The functions $\varphi(u, u, \tilde{u})$ for asymmetric VIs in are restated in Table 5. In conclusion, the abbreviations for different methods are listed in Table 6. Because

$$\{\text{Symmetric Nonlinear VIs}\} \subset \{\text{Nonlinear VIs}\}$$

and

$$\{\text{Symmetric Linear VIs}\} \subset \{\text{Linear VIs}\},$$

the test order for VIs of different types is listed in Table 7.

We use No. it and No. F to denote the numbers of iterations and the evaluations of the mapping F , respectively. The size of the tested problems is from 100 to 1000. All codes are written in Matlab and run on a notebook computer. The iterations begin with $u^0 = 0, \beta = 1$ and stop as soon as

$$\frac{\|u^k - P_\Omega[u^k - F(u^k)]\|_\infty}{\|u^0 - P_\Omega[u^0 - F(u^0)]\|_\infty} \leq 10^{-6}.$$

4.1 The numerical results for nonlinear variational inequalities

Test examples of nonlinear VIs The mapping $F(u)$ in the tested nonlinear VIs is given by

$$F(u) = D(u) + Mu + q, \tag{4.1}$$

where $D(u) : R^n \rightarrow R^n$ is the nonlinear part, M is an $n \times n$ matrix, and $q \in R^n$ is a vector.

– In $D(u)$, the nonlinear part of $F(u)$, the components are

$$D_j(u) = d_j \cdot \arctan(a_j \cdot u_j),$$

where a and d are random vectors² whose elements are in $(0, 1)$.

– The matrix M in the linear part is given by $M = A^T A + B$. A is an $n \times n$ matrix whose entries are randomly generated in the interval $(-5, +5)$, and B is an $n \times n$ skew-symmetric random matrix ($B^T = -B$) whose entries³ are in the interval $(-5, +5)$. Note that $\forall x \in R^n$,

$$x^T Bx = (x^T Bx)^T = x^T B^T x = -x^T Bx,$$

which implies $x^T Bx = 0$. As a consequence, $\forall x \in R^n$, it holds that

$$x^T Mx = x^T A^T A x + x^T Bx = \|Ax\|_2^2 \geq 0,$$

which implies M is positive semi-definite.

It is clear that the mapping composed in this way is monotone. We construct the following 6 sets of test examples by different combinations of Ω and q .

1. In the first set of test examples, $\Omega = R_+^n$ is the non-negative orthant. The elements of vector q is generated from a uniform distribution in the interval $(-1000, 1000)$.
2. The second set of test examples is modified from the first set, the only difference is $\Omega = \{u \in R^n \mid 0 \leq u \leq b\}$. Each elements of b equals u_{\max}^* by a positive factor less than 1, where u_{\max}^* is the maximal element of the solution of the related problem in the first set.
3. The 3-rd set⁴ of test examples is similar to the first set. Instead of $q \in (-1000, 1000)$, the vector q is generated from a uniform distribution in the interval $(-1000, 0)$.
4. The 4-th set of test examples is modified from the third set, the only difference is $\Omega = \{u \in R^n \mid 0 \leq u \leq b\}$. Each elements of b equals u_{\max}^* by a positive factor less than 1, where u_{\max}^* is the maximal element of the solution of the related problem in the 3-rd set.
5. The 5-th set of test examples has a known solution u^* and $\Omega = R_+^n$. Let vector p be generated from a uniform distribution in the interval $(-10, 10)$ and

$$u^* = \max(p, 0). \tag{4.2}$$

²A similar type of (small) problems was tested in [15] where the components of the nonlinear mapping $D(u)$ are $D_j(u) = c \cdot \arctan(u_j)$.

³In the paper by Harker and Pang [7], the matrix $M = A^T A + B + D$, where A and B are the same matrices as what we use here, and D is a diagonal matrix with uniformly distributed random entries $d_{jj} \in (0.0, 0.3)$.

⁴In [7], the similar problems in the first set are called easy problems while the third set problems are called hard problems.

For given $\eta > 0$, by setting

$$w = \eta \cdot \max(-p, 0) \quad \text{and} \quad q = w - (D(u^*) + Mu^*),$$

we have $F(u^*) = D(u^*) + Mu^* + q = w = \eta \cdot \max(-p, 0)$. Since $\eta > 0$ and

$$(u^*)^T F(u^*) = (\max(-p, 0))^T (\max(p, 0)) = 0,$$

we then have constructed a test problem

$$0 \leq u \perp F(u) \geq 0$$

with a known solution u^* described in (4.2).

6. The 6-th set of test examples: The test problem has a known solution u^* and

$$\Omega = \{u \in R^n \mid 0 \leq u_j \leq 10, j = 1, \dots, n\}$$

is a box. Let vector p be generated from a uniform distribution in the interval $(-5, 15)$ and

$$u^* = \max(0, \min(p, 10)). \tag{4.3}$$

For given $\eta_1, \eta_2 > 0$, by setting

$$w = \max(-p, 0) \cdot \eta_1 - \max(p - 10, 0) \cdot \eta_2,$$

we have

$$u^* = P_\Omega[u^* - w].$$

Therefore, in order to form a test problem

$$u = P_\Omega[u - F(u)]$$

with $F(u)$ described in (4.1) and a known solution u^* given in (4.3), we need only to set

$$q = w - (D(u^*) + Mu^*).$$

The tested methods and the numerical results For nonlinear variational inequalities, we test the problems by using the accepting rule (5.10) in [12] which is fulfilled by Procedure 5.1 in [12]. The quadruplet $(d_1(u, v, \tilde{u}), d_2(u, v, \tilde{u}), \varphi(u, v, \tilde{u}), \phi(u, v, \tilde{u}))$ is described in (5.9) in [12] with $v = u$. It is worth observing the effectiveness of different search directions and the different step-size rules. Thus, we compare the convergence behaviors of the following 4 methods:

NLD1-P, NLD2-P, NLD1-G and NLD2-G.

Since both $F(u^k)$ and $F(\tilde{u}^k)$ are involved in those methods recursions, each iteration of the test methods needs at least 2 times of evaluations of the mapping F . The

Table 8 Numerical results for nonlinear VIs of the 1-st set examples

Problem size n	Method NLD1-P		Method NLD2-P		Method NLD1-G		Method NLD2-G		max element of u^*
	No. It	No. F							
100	466	1011	391	815	236	490	200	434	5.1207
200	640	1361	568	1184	316	668	282	611	3.3988
500	697	1471	596	1244	343	728	310	672	1.4260
800	565	1203	483	999	278	589	259	546	0.8282
1000	601	1265	520	1086	297	631	267	580	0.5933

Table 9 Numerical results for nonlinear VIs of the 2-nd set examples

Problem size n	Method NLD1-P		Method NLD2-P		Method NLD1-G		Method NLD2-G		The vector b in $u \in [0, b]$
	No. It	No. F							
100	576	1243	473	986	285	591	243	527	4.0
200	676	1441	594	1237	329	695	296	641	3.0
500	730	1547	627	1302	361	764	325	704	1.0
800	713	1527	624	1296	358	755	350	727	0.6
1000	787	1663	677	1411	384	814	349	757	0.5

Table 10 Numerical results for nonlinear VIs of the 3-rd set examples

Problem size n	Method NLD1-P		Method NLD2-P		Method NLD1-G		Method NLD2-G		max element of u^*
	No. It	No. F							
100	952	2021	884	1841	478	1018	448	969	14.437
200	1189	2246	1105	2296	594	1270	561	1214	9.0339
500	1453	3000	1402	2922	733	1571	711	1538	3.7623
800	1434	2952	1344	2802	730	1560	683	1478	2.5715
1000	1532	3159	1424	2968	772	1652	720	1557	2.4738

test results for the 6 sets of nonlinear variational inequalities are given in Tables 8–13. Because u^* in the 5-th and 6-th sets of test examples is known, the difference $\|u^k - u^*\|$ is reported when the stopping criterium is satisfied.

The numerical results coincide with our theoretical results and analysis.

- In both of the primary methods and general methods, the methods with direction $d_2(u, v, \tilde{u})$ require fewer iterations than the corresponding methods with direction $d_1(u, v, \tilde{u})$. In particular,

$$\frac{\text{Computational load of NLD2-P}}{\text{Computational load of NLD1-P}}, \frac{\text{Computational load of NLD2-G}}{\text{Computational load of NLD1-G}} < 95\%.$$

Table 11 Numerical results for nonlinear VIs of the 4-th set examples

Problem size n	Method NLD1-P		Method NLD2-P		Method NLD1-G		Method NLD2-G		The vector b in $u \in [0, b]$
	No. It	No. F							
100	932	1961	837	1741	453	958	418	904	10.0
200	1346	2767	1246	2593	682	1457	635	1374	6.0
500	1723	3554	1626	3390	875	1873	830	1795	3.0
800	1759	3619	1639	3416	889	1898	830	1795	2.0
1000	1962	4046	1850	3855	994	2127	940	2032	2.0

Table 12 Numerical results for nonlinear VIs of the 5-th set examples

Problem size n	Method NLD1-P		Method NLD2-P		Method NLD1-G		Method NLD2-G		The bounds of $\ u^k - u^*\ _\infty$
	No. It	No. F							
100	762	1550	670	1393	379	804	337	730	2.0e-6
200	964	1972	869	1812	478	1017	442	957	8.0e-7
500	969	1962	868	1809	487	1031	442	956	3.0e-7
800	1030	2106	928	1933	526	1120	470	1017	1.6e-7
1000	1026	2074	926	1928	518	1101	473	1023	1.2e-7

Table 13 Numerical results for nonlinear VIs of the 6-th set examples

Problem size n	Method NLD1-P		Method NLD2-P		Method NLD1-G		Method NLD2-G		The bounds of $\ u^k - u^*\ _\infty$
	No. It	No. F							
100	1221	2507	1141	2374	618	1318	573	1238	2.0e-7
200	1157	2406	1052	2191	571	1216	533	1153	7.0e-8
500	1433	2908	1311	2729	723	1535	665	1437	3.0e-8
800	1312	2683	1212	2524	654	1391	613	1324	1.8e-8
1000	1133	2311	1034	2152	564	1198	521	1126	1.2e-8

- For the methods adopting the same direction, the general methods converge much faster than the primary methods.

$$\frac{\text{Computational load of NLD1-G}}{\text{Computational load of NLD1-P}}, \frac{\text{Computational load of NLD2-G}}{\text{Computational load of NLD2-P}} \in [0.5, 0.55].$$

Therefore, for nonlinear variational inequalities, we suggest to use the method NLD2-G with $\gamma = 1.8$.

4.2 The numerical results for symmetric nonlinear variational inequalities

Test examples of Symmetric Nonlinear VIs Since the nonlinear part of $F(u)$, namely $D(u)$ in (4.1), is symmetric, the test problems of symmetric nonlinear variational inequalities in this subsection are formed by deleting the asymmetric part of the matrix

Table 14 Numerical results for symmetric NL-VIs of the 1-st set examples

Problem size n	Method NLD1-G		Method NLD2-G		Method SNLD-P		max element of u^*
	No. It	No. F	No. It	No. F	No. It	No. F	
100	235	485	205	445	113	146	5.1150
200	292	618	252	546	125	181	3.2853
500	345	732	308	667	142	202	1.4302
800	278	589	526	538	110	158	0.8389
1000	293	623	267	580	122	172	0.6027

M , in (4.1). In details,

$$F(u) = D(u) + Mu + q, \quad M = A^T A.$$

In these test problems, the Jacobian of $F(u)$ is symmetric and $F(u)$ can be viewed as the gradient of a certain convex function. The other data in the test problems are the same as those described in Sect. 4.1.

The tested methods and the numerical results We use the accepting rule (5.8) in [12] which is fulfilled by Procedure 5.1 in [12]. The quadruplet $(d_1(u, v, \tilde{u}), d_2(u, v, \tilde{u}), \varphi(u, v, \tilde{u}), \phi(u, v, \tilde{u}))$ is described in (5.10) in [12] with $v = u$. Because

$$\{\text{Symmetric nonlinear VIs}\} \subset \{\text{Nonlinear VIs}\},$$

and the general contraction methods outperform the primary methods, we test the symmetric nonlinear problems with the method for nonlinear problem

NLD1-G and NLD2-G.

It is worth comparing the effectiveness of the following 3 methods:

NLD1-G, NLD2-G and SNLD-P.

Without the trial computations for finding the suitable parameter β_k , each iteration of SNLD-P needs only one evaluation of the mapping F . The test results for the 6 sets of symmetric nonlinear variational inequalities are given in Tables 14–19. Also, in the 5-th and 6-th sets of test examples, because u^* is known, we also report the difference $\|u^k - u^*\|$ when the stopping criterium is satisfied.

As in Sect. 4.1, the numerical results coincide with our theoretical results and analysis.

- Again, the general method NLD2-G requires fewer iterations than NLD1-G.

$$\frac{\text{Computational load of NLD2-G}}{\text{Computational load of NLD1-G}} < 0.95.$$

Table 15 Numerical results for symmetric NL-VIs of the 2-nd set examples

Problem size n	Method NLD1-G		Method NLD2-G		Method SNLD-P		The vector b in $u \in [0, b]$
	No. It	No. F	No. It	No. F	No. It	No. F	
100	299	630	265	575	156	187	4.0
200	307	649	274	593	140	194	3.0
500	351	713	319	691	181	246	1.0
800	356	751	340	710	167	222	0.6
1000	385	817	349	757	168	238	0.5

Table 16 Numerical results for symmetric NL-VIs of the 3-rd set examples

Problem size n	Method NLD1-G		Method NLD2-G		Method SNLD-P		max element of u^*
	No. It	No. F	No. It	No. F	No. It	No. F	
100	617	1307	565	1222	249	336	17.2269
200	648	1386	609	1318	247	343	8.8365
500	738	1580	692	1497	318	427	3.8270
800	703	1502	657	1422	284	382	2.6666
1000	769	1666	725	1568	275	375	2.5822

Table 17 Numerical results for symmetric NL-VIs of the 4-th set examples

Problem size n	Method NLD1-G		Method NLD2-G		Method SNLD-P		The vector b in $u \in [0, b]$
	No. It	No. F	No. It	No. F	No. It	No. F	
100	514	1079	462	998	210	299	12
200	779	1667	735	1590	294	406	6
500	929	1990	884	1912	362	486	3
800	869	1855	803	1737	296	408	2
1000	956	2046	910	1967	347	474	2

- For symmetric nonlinear VIs, the method SNLD-P converges much faster than the method NLD2-G.

$$\frac{\text{Computational load of SNLD-P}}{\text{Computational load of NLD2-G}} \in [0.25, 0.35],$$

it means that we should use symmetry when the mapping F is the gradient of certain convex function.

4.3 The numerical results for asymmetric linear variational inequalities

Test examples of Linear VIs In the linear variational inequalities (4.1) in [12], the mapping

$$F(u) = Mu + q. \tag{4.4}$$

Table 18 Numerical results for symmetric NL-VIs of the 5-th set examples

Problem size n	Method NLD1-G		Method NLD2-G		Method SNLD-P		The bounds of $\ u^k - u^*\ _\infty$
	No. It	No. F	No. It	No. F	No. It	No. F	
100	419	890	374	810	190	275	1.4e-6
200	507	1078	464	1004	208	295	7.8e-7
500	491	1039	446	965	213	302	2.3e-7
800	524	1116	473	1023	233	330	1.0e-6
1000	522	1109	474	1025	220	311	1.3e-7

Table 19 Numerical results for symmetric NL-VIs of the 6-th set examples

Problem size n	Method NLD1-G		Method NLD2-G		Method SNLD-P		The bounds of $\ u^k - u^*\ _\infty$
	No. It	No. F	No. It	No. F	No. It	No. F	
100	794	1695	723	1562	349	480	1.4e-7
200	619	1318	576	1067	286	394	5.6e-8
500	734	1560	684	1478	361	496	2.7e-8
800	654	1392	611	1320	291	391	1.6e-8
1000	570	1210	528	1141	243	333	1.1e-8

The test problems are formed by deleting the nonlinear part $D(u)$ in (4.1). The other data in the test problems are same as those described in Sect. 4.1.

The tested methods and the numerical results We use the accepting rule (4.17) in [12] which is fulfilled by Procedure 4.1 in [12]. The quadruplet $(d_1(u, v, \tilde{u}), d_2(u, v, \tilde{u}), \varphi(u, v, \tilde{u}), \phi(u, v, \tilde{u}))$ is described in (4.13) in [12] with $v = u$. Because

$$\{\text{Linear VIs}\} \subset \{\text{Nonlinear VIs}\},$$

and the general contraction methods outperform the primary methods, we test the linear problems with nonlinear methods

$$\text{NLD1-G and NLD2-G.}$$

In addition, by using the linearity, we use the methods

$$\text{LD1-G and LD2-G.}$$

It is worth comparing the effectiveness of the methods:

$$\text{NLD1-G, NLD2-G, LD1-G and LD2-G.}$$

Since both $F(u^k)$ and $F(\tilde{u}^k)$ are involved in those methods recursions, each iteration of the test methods needs at least 2 times of evaluations of the mapping F . The test results for the 6 sets of (asymmetric) linear variational inequalities are given in

Table 20 Numerical results for asymmetric LVIs of the 1-st set examples

Problem size n	Method NLD1-G		Method NLD2-G		Method LD1-G		Method LD2-G		max element of u^*
	No. It	No. F	No. It	No. F	No. It	No. F	No. It	No. F	
100	236	490	200	434	348	783	178	363	5.1242
200	315	666	282	611	520	1132	261	529	3.4029
500	342	726	310	672	596	1269	279	563	1.4266
800	278	589	259	546	417	964	228	463	0.8284
1000	296	629	267	580	470	1051	240	487	0.5933

Table 21 Numerical results for asymmetric LVIs of the 2-nd set examples

Problem size n	Method NLD1-G		Method NLD2-G		Method LD1-G		Method LD2-G		The vector b in $u \in [0, b]$
	No. It	No. F	No. It	No. F	No. It	No. F	No. It	No. F	
100	286	593	244	529	429	959	216	439	4.0
200	329	695	297	643	538	1182	273	553	3.0
500	359	760	325	704	618	1323	291	587	1.0
800	358	755	351	729	505	1235	287	580	0.6
1000	384	814	349	757	648	1406	312	631	0.5

Table 22 Numerical results for asymmetric LVIs of the 3-rd set examples

Problem size n	Method NLD1-G		Method NLD2-G		Method LD1-G		Method LD2-G		max element of u^*
	No. It	No. F	No. It	No. F	No. It	No. F	No. It	No. F	
100	479	1020	448	969	822	1792	432	874	14.449
200	595	1272	561	1214	927	2142	521	1047	9.0412
500	740	1586	711	1538	1293	2832	636	1301	3.7641
800	730	1560	684	1480	1256	2737	641	1296	2.5727
1000	771	1650	720	1557	1405	2963	665	1356	2.4747

Tables 20–25. In the 5-th and 6-th sets of test examples, because u^* is known, the difference $\|u^k - u^*\|$ is reported when the stopping criterium is satisfied.

Similarly to the previous two subsections, the numerical results coincide with our theoretical results and analysis.

- Using the general methods for nonlinear VIs in Sect. 5 in [12] to solve the linear VIs, the method with direction $d_2(u, v, \tilde{u})$ requires fewer iterations than the corresponding methods with direction $d_1(u, v, \tilde{u})$,

$$\frac{\text{Computational load of NLD2-G}}{\text{Computational load of NLD1-G}} \approx 0.9.$$

- For the general methods with either direction $d_1(u, v, \tilde{u})$ or direction $d_2(u, v, \tilde{u})$ for linear VIs in Sect. 4 in [12], the method with direction $d_2(u, v, \tilde{u})$ converges

Table 23 Numerical results for asymmetric LVIs of the 4-th set examples

Problem size n	Method NLD1-G		Method NLD2-G		Method LD1-G		Method LD2-G		The vector b in $u \in [0, b]$
	No. It	No. F	No. It	No. F	No. It	No. F	No. It	No. F	
100	453	958	418	904	775	1680	415	840	10.0
200	684	1461	640	1385	1207	2593	589	1183	6.0
500	875	1873	830	1795	1553	3349	768	1555	3.0
800	894	1909	835	1806	1590	3397	781	1576	2.0
1000	995	2129	940	2032	1819	3852	875	1767	2.0

Table 24 Numerical results for asymmetric LVIs of the 5-th set examples

Problem size n	Method NLD1-G		Method NLD2-G		Method LD1-G		Method LD2-G		The bounds of $\ u^k - u^*\ _\infty$
	No. It	No. F	No. It	No. F	No. It	No. F	No. It	No. F	
100	379	804	334	723	544	1268	320	646	2.0e-6
200	478	1017	442	957	796	1754	416	840	8.0e-7
500	487	1031	442	956	710	1667	417	840	3.0e-7
800	525	1118	470	1017	841	1859	442	905	1.6e-7
1000	518	1101	473	1023	880	1904	412	861	1.2e-7

Table 25 Numerical results for asymmetric LVIs of the 6-th set examples

Problem size n	Method NLD1-G		Method NLD2-G		Method LD1-G		Method LD2-G		The bounds of $\ u^k - u^*\ _\infty$
	No. It	No. F	No. It	No. F	No. It	No. F	No. It	No. F	
100	617	1316	573	1238	1062	2301	553	1120	2.0e-7
200	572	1218	533	1153	983	2137	505	1022	7.0e-8
500	724	1537	666	1439	1226	2666	613	1251	3.0e-8
800	657	1398	616	1331	1029	2361	570	1161	1.8e-8
1000	565	1200	521	1126	947	2074	493	997	1.2e-8

much faster than the corresponding method with direction $d_1(u, v, \tilde{u})$,

$$\frac{\text{Computational load of LD2-G}}{\text{Computational load of LD1-G}} \approx 0.5.$$

- For linear VIs, the general method with direction $d_2(u, v, \tilde{u})$ in Sect. 4 in [12] requires fewer iterations than the corresponding method in Sect. 5 in [12],

$$\frac{\text{Computational load of LD2-G}}{\text{Computational load of NLD2-G}} \approx 0.9.$$

The method LD2-G converges faster than all other tested methods, which implies that we should use the linearity when the variational inequality is linear.

Table 26 Numerical results for symmetric LVIs of the 1-st set examples

Problem size n	Method NLD2-G		Method LD2-G		Method SLD-P		max element of u^*
	No. It	No. F	No. It	No. F	No. It	No. F	
100	205	445	185	375	74	103	5.1184
200	256	555	232	471	83	116	3.2891
500	308	667	279	563	97	123	1.4308
800	256	538	226	459	139	160	0.8391
1000	267	580	210	497	134	160	0.6028

4.4 The numerical results for symmetric linear variational inequalities

Test examples of Linear VIs In the symmetric linear variational inequalities (3.1) in [12], the mapping

$$F(u) = Hu + q. \tag{4.5}$$

The test problems are formed by $H = A^T A$. The other data in the test problems are the same as those described in Sect. 4.3.

The tested methods and the numerical results We use the accepting rule (3.12) in [12] which is fulfilled by Procedure 3.1 in [12]. The quadruplet $(d_1(u, v, \tilde{u}), d_2(u, v, \tilde{u}), \varphi(u, v, \tilde{u}), \phi(u, v, \tilde{u}))$ is described in (3.10) in [12] with $v = u$. Because

$$\{\text{Symmetric Linear VIs}\} \subset \{\text{Linear VIs}\} \subset \{\text{Nonlinear VIs}\},$$

and the general contraction methods outperform the primary methods, we test the symmetric linear problems with the methods for nonlinear and asymmetric linear problems

$$\text{NLD2-G and LD2-G.}$$

In addition, by using the symmetry, we use the method

$$\text{SLD-P.}$$

It is worth comparing the effectiveness of the following 3 methods:

$$\text{NLD2-G, LD2-G and SLD-P.}$$

Without the trial computations for finding the suitable parameter β_k , each iteration of SLD-P needs only one evaluation of the mapping F (here is $Hu + q$). The test results for the 6 sets of symmetric linear variational inequalities are given in Tables 26–31. Also, in the 5-th and 6-th sets of test examples, because u^* is known, we also report the difference $\|u^k - u^*\|$ when the stopping criterium is satisfied.

As in the previous three subsections, the numerical results coincide with our theoretical results and analysis.

Table 27 Numerical results for symmetric LVIs of the 2-nd set examples

Problem size n	Method NLD2-G		Method LD2-G		Method SLD-P		The vector b in $u \in [0, b]$
	No. It	No. F	No. It	No. F	No. It	No. F	
100	377	817	345	704	115	157	4.0
200	468	1013	436	890	164	223	3.0
500	602	1303	553	1112	153	204	1.0
800	510	1104	475	957	137	192	0.6
1000	585	1266	547	1103	184	240	0.5

Table 28 Numerical results for symmetric LVIs of the 3-rd set examples

Problem size n	Method NLD2-G		Method LD2-G		Method SLD-P		max element of u^*
	No. It	No. F	No. It	No. F	No. It	No. F	
100	566	1224	531	1068	163	220	17.2432
200	609	1318	577	1159	182	251	8.8438
500	692	1497	638	1298	150	209	3.8289
800	658	1424	613	1244	289	349	2.6678
1000	725	1568	674	1362	161	219	2.5832

Table 29 Numerical results for symmetric LVIs of the 4-th set examples

Problem size n	Method NLD2-G		Method LD2-G		Method SLD-P		The vector b in $u \in [0, b]$
	No. It	No. F	No. It	No. F	No. It	No. F	
100	461	999	426	867	146	198	12
200	839	1814	793	1600	166	225	6
500	830	1795	786	1577	172	236	3
800	859	1858	806	1628	230	292	2
1000	1008	2179	956	1928	267	352	2

- For symmetric linear VIs, the general method LD2-G requires fewer iterations than NLD2-G.

$$\frac{\text{Computational load of LD2-G}}{\text{Computational load of NLD2-G}} < 0.95,$$

this means that we should use the linearity if the variational inequality is linear.

- For symmetric linear VIs, the method SLD-P converges much faster than the method LD2-G.

$$\frac{\text{Computational load of SLD-P}}{\text{Computational load of LD2-G}} \in [0.20 - 0.25],$$

it means that we should use symmetry when the linear VIs are symmetric.

Table 30 Numerical results for symmetric LVIs of the 5-th set examples

Problem size n	Method NLD2-G		Method LD2-G		Method SLD-P		The bounds of $\ u^k - u^*\ _\infty$
	No. It	No. F	No. It	No. F	No. It	No. F	
100	370	802	352	715	151	190	2.0e-6
200	464	1004	438	881	170	219	8.0e-7
500	446	965	417	857	174	230	3.0e-7
800	477	1032	451	909	179	225	1.6e-7
1000	474	1025	430	883	150	197	1.2e-7

Table 31 Numerical results for symmetric LVIs of the 6-th set examples

Problem size n	Method NLD2-G		Method LD2-G		Method SLD-P		The bounds of $\ u^k - u^*\ _\infty$
	No. It	No. F	No. It	No. F	No. It	No. F	
100	725	1566	704	1415	208	274	2.0e-6
200	577	1248	513	1067	189	253	8.0e-7
500	684	1478	634	1287	204	268	3.0e-7
800	612	1322	555	1145	216	270	1.6e-7
1000	528	1141	492	1003	155	210	1.2e-7

5 Concluding remarks

In this paper, we first extend the primary methods proposed in Part I of this paper [12] to more efficient ones (called extended methods) under our framework [12]. The extended methods need only minor extra costs. As an application, we test a matrix approximation problem to compare the efficiency of the proximal alternating directions method and its extended version. From the numerical results, the improvement on the efficiency of the extended method is significant and convincing.

Then in case of $G = I$, following the unified framework, we introduce the extended and general contraction methods. In such methods, we can use the convex combinations of the geminate directions in the quadruplet as the search directions with selected step lengths. Besides the theoretical comparisons on the efficiency of the different directions and step lengths, we present numerous numerical results confirming our theoretical results clearly. From the numerical results, the numbers of the iterations and function evaluations are reduced significantly for the general contraction methods. Our numerical experiments also indicate that, special properties such as symmetry, linearity, etc., should be considered in solving these problems.

References

1. Bertsekas, D.P., Tsitsiklis, J.N.: Parallel and Distributed Computation, Numerical Methods. Prentice-Hall, Englewood Cliffs (1989)
2. Blum, E., Oettli, W.: Mathematische Optimierung. Grundlagen und Verfahren. Ökonometrie und Unternehmensforschung. Springer, Berlin (1975)

3. Eckstein, J.: Nonlinear proximal point algorithms using Bregman functions, with applications to convex programming. *Math. Oper. Res.* **18**, 202–226 (1993)
4. Eckstein, J.: Approximate iterations in Bregman-function-based proximal algorithms. *Math. Program.* **83**, 113–123 (1998)
5. Gao, Y., Sun, D.F.: Calibrating least squares covariance matrix problems with equality and inequality constraints. *SIAM J. Matrix Anal. Appl.* **31**, 1432–1457 (2009)
6. Golub, G.H., van Loan, C.F.: *Matrix Computations*, 3rd edn. Johns Hopkins University Press, Baltimore (1996)
7. Harker, P.T., Pang, J.S.: A Damped-Newton method for the linear complementarity problem. *Lect. Appl. Math.* **26**, 265–284 (1990)
8. He, S.B., Liao, L.-Z.: Improvements of some projection methods for monotone nonlinear variational inequalities. *J. Optim. Theory Appl.* **112**, 111–128 (2002)
9. He, B.S., Liao, L.-Z., Han, D.R., Yang, H.: A new inexact alternating directions method for monotone variational inequalities. *Math. Program.* **92**, 103–118 (2002)
10. He, B.S., Yang, Z.H., Yuan, X.M.: An approximate proximal-extragradient type method for monotone variational inequalities. *J. Math. Anal. Appl.* **300**, 362–374 (2004)
11. He, B.S., Yuan, X.M., Zhang, J.J.Z.: Comparison of two kinds of prediction-correction methods for monotone variational inequalities. *Comput. Optim. Appl.* **27**, 247–267 (2004)
12. He, B.S., Liao, L.-Z., Wang, X.: Proximal-like contraction methods for monotone variational inequalities in a unified framework I: effective quadruplet and primary methods (2010, preprint)
13. Rockafellar, R.T.: Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Math. Oper. Res.* **1**, 97–116 (1976)
14. Solodov, M.V., Svaiter, B.F.: Error bounds for proximal point subproblems and associated inexact proximal point algorithms. *Math. Program.* **88**, 371–389 (2000)
15. Taji, K., Fukushima, M., Ibaraki, I.: A globally convergent Newton method for solving strongly monotone variational inequalities. *Math. Program.* **58**, 369–383 (1993)
16. Zhu, T., Yu, Z.G.: A simple proof for some important properties of the projection mapping. *Math. Inequal. Appl.* **7**, 453–456 (2004)