A globally convergent descent method for nonsmooth variational inequalities*

Barbara Panicucci† Massimo Pappalardo† Mauro Passacantando†

Abstract. We propose a descent method via gap functions for solving nonsmooth variational inequalities with a locally Lipschitz operator. Assuming monotone operator (not necessarily strongly monotone) and bounded domain, we show that the method with an Armijo-type line search is globally convergent. Finally, we report some numerical experiments.

Keywords. Nonsmooth variational inequality, monotone map, gap function, descent method.

AMS Subject classification. 90C30, 90C33, 65K10.

1 Introduction

Let X be a nonempty, closed and convex subset of $\mathbb{R}^n$ and $F : \mathbb{R}^n \to \mathbb{R}^n$ a given map. The variational inequality (VI) problem is to find a point $x^* \in X$ such that

$$\langle F(x^*), x - x^* \rangle \geq 0 \qquad \forall\, x \in X, \tag{1}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product in $\mathbb{R}^n$.

This problem, which includes nonlinear complementarity problems and systems of nonlinear equations, has many important applications in a wide variety of scientific and engineering fields including network economics, transportation science, and game theory.

Various iterative methods have been developed for solving VI problems; see [2] and the references therein for a survey of theory and algorithms. A very useful approach for solving a VI entails reformulating it into an equivalent optimization problem through gap (or merit) functions (see e.g. [3, 7]). Descent methods which utilize those functions have been studied in the case when F is smooth. Recently, two descent methods via gap functions for solving a VI with nonsmooth (locally Lipschitz) and strongly monotone operator have been proposed in [5, 6]. In this paper, we present a modification, in the spirit of [10], of the method described in [5] for solving a nonsmooth monotone (not necessarily strongly monotone) VI with a bounded feasible set. Using nonsmooth analysis we prove global convergence of the proposed method.

Nonsmooth analysis has also been used in [4], where the authors propose a locally superlinearly convergent method for solving nonsmooth VI, which extends the classical Newton method for smooth VI. To obtain the convergence result, the operator F needs to be semismooth and locally strongly monotone at the solution. In this paper, instead, we get the global convergence under the monotonicity assumption only.

*This work has been supported by the National Research Program PRIN/2005017083 "Innovative Problems and Methods in Nonlinear Optimization". This paper has been published in Computational Optimization and Applications, vol. 43 (2009), pp. 197–211.

†Department of Applied Mathematics - University of Pisa, via Buonarroti 1/c, 56127 Pisa - Italy (panicucc@mail.dm.unipi.it), (pappalardo@dma.unipi.it), (passacantando@dma.unipi.it).

For the sake of convenience, we first recall some definitions and notations which will be used in the subsequent sections. Given a symmetric positive definite matrix G, we denote by $\|\cdot\|_G$ the norm in $\mathbb{R}^n$ defined by $\|x\|_G = \sqrt{\langle x, G\,x\rangle}$. In particular, $\|\cdot\|$ denotes the classical Euclidean norm induced by unit matrix I. The projection of a point $x \in \mathbb{R}^n$ onto the closed convex set X with respect to $\|\cdot\|_G$, denoted by $\Pi_{X,G}(x)$, is defined as the unique solution of the problem

$$\min_{y \in X} \ \|y - x\|_G.$$

It is well known that $\Pi_{X,G}(x)$ is characterized by the following condition:

$$\langle x - \Pi_{X,G}(x), G\,[z - \Pi_{X,G}(x)]\rangle \leq 0 \qquad \forall\ z \in X.$$

We recall that x^* is a solution of VI if and only if $x^* = \Pi_{X,G}(x^* - G^{-1}\,F(x^*))$.

For a function $f : \mathbb{R}^n \to \mathbb{R}$ Lipschitz near $x \in \mathbb{R}^n$, the *generalized directional derivative* of f at x in the direction $d \in \mathbb{R}^n$ is defined as [1]:

$$f^\circ(x; d) := \limsup_{\substack{y \to x \\ \lambda \downarrow 0}} \ \frac{f(y + \lambda\,d) - f(y)}{\lambda},$$

and the *generalized gradient* of f at x is

$$\partial f(x) = \{\xi \in \mathbb{R}^n :\ f^\circ(x; d) \geq \langle \xi, d\rangle, \qquad \forall\ d \in \mathbb{R}^n\}.$$

In the rest of the paper we utilize the following assumptions.

(A1) *The set $X \subseteq \mathbb{R}^n$ is nonempty, closed, and convex.*

(A2) *The map $F : Y \to \mathbb{R}^n$ is locally Lipschitz at each point of an open convex set Y such that $X \subset Y$.*

(A3) *The map $F : Y \to \mathbb{R}^n$ is monotone on X, i.e. $\langle F(x) - F(y), x - y\rangle \geq 0 \quad \forall\ x, y \in X$.*

(A4) *The set X is bounded.*

2 Gap functions

Given $\alpha > 0$ and the matrix G, we consider the following gap function introduced in [3]:

$$\varphi_\alpha(x) = \max_{y \in X} \left[\langle F(x), x - y\rangle - \frac{\alpha}{2}\|x - y\|_G^2\right] = \langle F(x), x - y_\alpha(x)\rangle - \frac{\alpha}{2}\|x - y_\alpha(x)\|_G^2, \tag{2}$$

where $y_\alpha(x) = \Pi_{X,G}(x - (\alpha\,G)^{-1}F(x))$ is the unique maximizer.

Thus

- $\varphi_\alpha(x) \geq 0, \quad \forall\ x \in X$;
- x^* solves VI if and only if $x^* \in X$ and $\varphi_\alpha(x^*) = 0$;

hence VI can be reformulated as the following constrained minimization problem:

$$\min_{x \in X} \ \varphi_\alpha(x). \tag{3}$$

Under assumption assumptions $(A1) - (A2)$, the gap function φ_α is locally Lipschitz at any point $x \in X$ and its generalized gradient is given by the formula [9]:

$$\partial\varphi_\alpha(x) = \left\{F(x) - (V^T - \alpha\,G)\,(y_\alpha(x) - x), \qquad V \in \partial F(x)\right\}, \tag{4}$$

where ∂F is the classical Clarke generalized Jacobian of F [1].

In the following we generalize the Proposition 4.1 in [6] regarding strongly monotone maps to the case where F is monotone.

Theorem 2.1. *Let assumptions* $(A1)-(A3)$ *be fulfilled. Then, for each* $x \in X$*, the vector* $y_\alpha(x) - x$ *satisfies the following condition:*

$$\varphi_\alpha^\circ(x; y_\alpha(x) - x) \le -\varphi_\alpha(x) + \frac{\alpha}{2} \|x - y_\alpha(x)\|_G^2 \le 0. \tag{5}$$

Proof. Let $g \in \partial\varphi_\alpha(x)$. From (4) we have:

$$g = F(x) - (V^T - \alpha\, G)\,(y_\alpha(x) - x), \qquad \text{for some } V \in \partial F(x).$$

Since F is monotone and $(A1)-(A2)$ holds, each matrix $V \in \partial F(x)$ is positive semidefinite [4], then:

$$\begin{aligned}
\langle g, y_\alpha(x) - x\rangle &= \langle F(x), y_\alpha(x) - x\rangle + \alpha\, \|x - y_\alpha(x)\|_G^2 - \langle y_\alpha(x) - x, V\,(y_\alpha(x) - x)\rangle \\
&\le \langle F(x), y_\alpha(x) - x\rangle + \alpha\, \|x - y_\alpha(x)\|_G^2 \\
&= -\varphi_\alpha(x) + \frac{\alpha}{2}\, \|x - y_\alpha(x)\|_G^2.
\end{aligned}$$

Moreover, we have [1]:

$$\varphi_\alpha^\circ(x; y_\alpha(x) - x) = \max_{g \in \partial\varphi_\alpha(x)} \langle g, y_\alpha(x) - x\rangle,$$

thus

$$\varphi_\alpha^\circ(x; y_\alpha(x) - x) \le -\varphi_\alpha(x) + \frac{\alpha}{2}\, \|x - y_\alpha(x)\|_G^2\,.$$

It is also known [6] that

$$\varphi_\alpha(x) \ge \frac{\alpha}{2}\, \|x - y_\alpha(x)\|_G^2, \qquad \forall\, x \in X.$$

Hence

$$-\varphi_\alpha(x) + \frac{\alpha}{2}\, \|x - y_\alpha(x)\|_G^2 \le 0.$$

□

This result will be exploited in Section 3 where we modify, in the spirit of [10], the descent method for solving nonsmooth strongly monotone VI proposed in [5], in order to solve the nonsmooth monotone VI.

3 Descent method

The basic idea of the method is to use (5) to obtain a descent direction as follows: if $x \in X$ satisfies the condition

$$-\varphi_\alpha(x) + \frac{\alpha}{2}\, \|x - y_\alpha(x)\|_G^2 < -\eta\, \varphi_\alpha(x), \tag{6}$$

where $\eta \in (0, 1)$, then from (5) and (6) we get

$$\varphi_\alpha^\circ(x; y_\alpha(x) - x) \le -\varphi_\alpha(x) + \frac{\alpha}{2}\, \|x - y_\alpha(x)\|_G^2 < -\eta\, \varphi_\alpha(x).$$

Thus $d = y_\alpha(x) - x$ is a descent direction for φ_α and we can perform a line search with respect to d. Otherwise, if x does not solve VI and does not satisfy (6), we reduce the parameter α. In the following, we describe the descent method with an Armijo-type line search.

Algorithm

0. (Initial step)
   Let G be a symmetric positive definite matrix, $\eta, \gamma \in (0,1)$, and $\beta \in (0,\eta)$.
   Let $\{\alpha_k\}$ be a sequence strictly decreasing to 0.
   Choose any $x^0 \in X$ and set $k = 0$.
1. (Stopping criterion)
   If $\varphi_{\alpha_k}(x^k) = 0$

   then STOP,

   else set $k = k + 1$.
2. (Minimization of φ_{α_k})

   2a. (Initialization)
   Set $i = 0$ and $z^0 = x^{k-1}$.

   2b. **If** $-\varphi_{\alpha_k}(z^i) + \dfrac{\alpha_k}{2}\,\|z^i - y_{\alpha_k}(z^i)\|_G^2 < -\eta\,\varphi_{\alpha_k}(z^i)$

   then (line search)

   compute $y^i = \Pi_{X,G}(z^i - (\alpha_k\, G)^{-1} F(z^i))$
   set $d^i = y^i - z^i$
   compute the smallest nonnegative integer m such that:

$$\varphi_{\alpha_k}(z^i + \gamma^m\, d^i) - \varphi_{\alpha_k}(z^i) \le -\beta\,\gamma^m\,\varphi_{\alpha_k}(z^i)$$

   set $t_i = \gamma^m$,

   else (update of x^k)

   set $x^k = z^i$ and return to step 1.

   2c. (Update of z^i)
   Set $z^{i+1} = z^i + t_i\, d^i$, $i = i + 1$, and return to step 2b.

Theorem 3.1. (Global convergence)
If assumptions $(A1) - (A4)$ are fulfilled, then the algorithm either stops at a solution of VI after a finite number of iterations, or generates a bounded sequence $\{x^k\}$ such that any of its cluster points solves VI, or generates a bounded sequence $\{z^i\}$, for some fixed α_k, such that any of its cluster points solves VI.

Proof. First, we show that the algorithm is well defined, i.e. that the line search procedure is always finite. Assume, by contradiction, that there are $i, k \ge 0$ such that the inequality

$$\varphi_{\alpha_k}(z^i + \gamma^m\, d^i) - \varphi_{\alpha_k}(z^i) > -\,\beta\,\gamma^m\,\varphi_{\alpha_k}(z^i),$$

holds for all $m \in \mathbb{N}$. Then we have:

$$\varphi^\circ_{\alpha_k}(z^i; d^i) \ge \limsup_{m\to+\infty} \frac{\varphi_{\alpha_k}(z^i + \gamma^m\, d^i) - \varphi_{\alpha_k}(z^i)}{\gamma^m} \ge -\,\beta\,\varphi_{\alpha_k}(z^i).$$

Combining (5) and step 2b, we get:

$$\varphi^\circ_{\alpha_k}(z^i; d^i) \le -\varphi_{\alpha_k}(z^i) + \frac{\alpha_k}{2}\,\|d^i\|_G^2 < -\,\eta\,\varphi_{\alpha_k}(z^i),$$

therefore

$$(\eta - \beta)\,\varphi_{\alpha_k}(z^i) < 0,$$

which is impossible because $\eta > \beta$ and $\varphi_{\alpha_k}(z^i) \ge 0$. So the line search procedure is always finite.

There are three possible cases.

Case 1. The algorithm stops at x^k after a finite number of iterations. From the stopping criterion at step 1 it follows that $\varphi_{\alpha_k}(x^k) = 0$, thus x^k solves VI.

Case 2. The algorithm generates an infinite sequence $\{x^k\}$. From step 2b we have that

$$\varphi_{\alpha_k}(x^k) \leq \frac{\alpha_k}{2\,(1-\eta)}\,\|x^k - y_{\alpha_k}(x^k)\|_G^2 \qquad \forall\, k \in \mathbb{N}.$$

Since x^k and $y_{\alpha_k}(x^k)$ belong to X which is bounded, the norm $\|x^k - y_{\alpha_k}(x^k)\|_G^2$ is bounded above. Moreover $\lim\limits_{k\to\infty} \alpha_k = 0$, thus

$$\lim_{k\to\infty} \varphi_{\alpha_k}(x^k) = 0. \tag{7}$$

Since X is bounded, the sequence $\{x^k\}$ has cluster points. Let x^* be any cluster point of $\{x^k\}$. From the definition of φ_{α_k} it follows that

$$\varphi_{\alpha_k}(x^k) \geq \langle F(x^k), x^k - y\rangle - \frac{\alpha_k}{2}\,\|x^k - y\|_G^2 \qquad \forall\, k \in \mathbb{N}, \forall\, y \in X.$$

Passing to the limit and taking a subsequence if necessary, we obtain

$$0 \geq \langle F(x^*), x^* - y\rangle \qquad \forall\, y \in X,$$

on account of the continuity of F, $\lim\limits_{k\to\infty} \alpha_k = 0$, and (7). It follows that x^* solves VI.

Case 3. The algorithm generates an infinite sequence $\{z^i\}$ for a fixed $\alpha_k = \alpha$. Let us consider two possible subcases: either $\limsup\limits_{i\to\infty} t_i > 0$, or $\limsup\limits_{i\to\infty} t_i = 0$.

Subcase 3a. If $\limsup\limits_{i\to\infty} t_i > 0$, then there exists $t^* > 0$ and a subsequence $\{t_{i_s}\}$ such that $t_{i_s} \geq t^* > 0$ for all $s \in \mathbb{N}$. Since the sequence $\{z^i\}$ is infinite, we have:

$$\varphi_\alpha(z^{i_s}) - \varphi_\alpha(z^{i_s+1}) \geq \beta\, t_{i_s}\, \varphi_\alpha(z^{i_s}) \geq \beta\, t^*\, \varphi_\alpha(z^{i_s}) \geq 0. \tag{8}$$

The sequence $\{\varphi_\alpha(z^i)\}$ is monotone decreasing and bounded below, hence

$$\lim_{i\to\infty} (\varphi_\alpha(z^i) - \varphi_\alpha(z^{i+1})) = 0,$$

and in particular

$$\lim_{s\to\infty} (\varphi_\alpha(z^{i_s}) - \varphi_\alpha(z^{i_s+1})) = 0. \tag{9}$$

Using (8) and (9), we obtain $\lim\limits_{s\to\infty} \varphi_\alpha(z^{i_s}) = 0$ and thus $\lim\limits_{i\to\infty} \varphi_\alpha(z^i) = 0$. Let z^* be any cluster point of $\{z^i\}$. From the continuity of φ_α it follows that $\lim\limits_{i\to\infty} \varphi_\alpha(z^i) = \varphi_\alpha(z^*)$, hence $\varphi_\alpha(z^*) = 0$, i.e. z^* is a solution of VI.

Subcase 3b. If $\limsup\limits_{i\to\infty} t_i = 0$, then $\lim\limits_{i\to\infty} t_i = 0$. From the step length rule it follows that for all $i \in \mathbb{N}$,

$$\varphi_\alpha\left(z^i + \frac{t_i}{\gamma}\, d^i\right) - \varphi_\alpha(z^i) > -\,\beta\,\frac{t_i}{\gamma}\,\varphi_\alpha(z^i).$$

By the mean value theorem we have

$$\varphi_\alpha\left(z^i + \frac{t_i}{\gamma}\, d^i\right) - \varphi_\alpha(z^i) = \langle \xi^i, \frac{t_i}{\gamma}\, d^i\rangle,$$

where $\xi^i \in \partial\varphi_\alpha(z^i + \theta_i\, \frac{t_i}{\gamma}\, d^i)$ for some $\theta_i \in (0,1)$. From (4) it follows that

$$\xi^i = F\left(z^i + \theta_i\,\frac{t_i}{\gamma}\, d^i\right) - \left(V_i^T - \alpha\, G\right)\left[y_\alpha\left(z^i + \theta_i\,\frac{t_i}{\gamma}\, d^i\right) - \left(z^i + \theta_i\,\frac{t_i}{\gamma}\, d^i\right)\right],$$

for some $V_i \in \partial F(z^i + \theta_i \frac{t_i}{\gamma} d^i)$. Therefore, for all $i \in \mathbb{N}$, we have:

$$\langle F\left(z^i + \theta_i \frac{t_i}{\gamma} d^i\right) - (V_i^T - \alpha G)\left[y_\alpha\left(z^i + \theta_i \frac{t_i}{\gamma} d^i\right) - \left(z^i + \theta_i \frac{t_i}{\gamma} d^i\right)\right], d^i\rangle > - \beta \varphi_\alpha(z^i).$$

The sequences $\{z^i\}$ and $\{d^i\}$ are bounded thus, since F is continuous and ∂F is upper semicontinuous, the sequence $\{V_i\}$ is bounded. Let z^* be any cluster point of $\{z^i\}$. Since $\lim_{i \to \infty} t_i = 0$, passing to the limit and taking a subsequence if necessary, we get:

$$\langle F(z^*) - (V_*^T - \alpha G)(y_\alpha(z^*) - z^*), d^*\rangle \geq - \beta \varphi_\alpha(z^*), \tag{10}$$

where $d^* = y_\alpha(z^*) - z^*$ and $V_* \in \partial F(z^*)$. Since

$$\varphi^\circ_{\alpha_k}(z^*; d^*) = \max_{g \in \partial \varphi_\alpha(z^*)} \langle g, d^*\rangle,$$

from (4) and (10) it follows that:

$$\varphi^\circ_\alpha(z^*; d^*) \geq \langle F(z^*) - (V_*^T - \alpha G)(y_\alpha(z^*) - z^*), d^*\rangle \geq - \beta \varphi_\alpha(z^*). \tag{11}$$

Moreover, for all $i \in \mathbb{N}$,we have:

$$- \varphi_\alpha(z^i) + \frac{\alpha}{2} \|z^i - y_\alpha(z^i)\|_G^2 < - \eta \varphi_\alpha(z^i),$$

hence passing to the limit and taking a subsequence if necessary, and using Theorem 2.1 we obtain:

$$\varphi^\circ_\alpha(z^*; d^*) \leq -\varphi_\alpha(z^*) + \frac{\alpha}{2} \|d^*\|_G^2 \leq - \eta \varphi_\alpha(z^*). \tag{12}$$

From (11) and (12) we get

$$(\eta - \beta) \varphi_\alpha(z^*) \leq 0.$$

Since $\eta > \beta$ and $\varphi_\alpha(z^*) \geq 0$, it follows that $\varphi_\alpha(z^*) = 0$. i.e. z^* solves VI.

□

Remark 3.1. *In the algorithm we can choose the sequence $\{\alpha_k\}$ adaptively, for example (see also [8]) such as:*

$$\alpha_k = \begin{cases} \alpha_{k-1} & \text{if } \varphi_{\alpha_{k-1}}(x^{k-1}) \leq \nu_{k-1}, \\ \mu \alpha_{k-1} & \text{otherwise}, \end{cases} \tag{13}$$

where $0 < \mu < 1$ and $\{\nu_k\}$ is a sequence decreasing to 0. Indeed, if the algorithm generates an infinite sequence $\{x^k\}$ with $\{\alpha_k\}$ chosen by (13), then either $\lim_{k \to \infty} \alpha_k = 0$, which can be treated as in the Case 2 of Theorem 3.1, or α_k is constant for every k greater than some $\bar{k}$, i.e.

$$\alpha_k = \bar{\alpha} \quad \text{and} \quad \varphi_{\bar{\alpha}}(x^k) \leq \nu_k \qquad \forall\ k > \bar{k},$$

hence $\lim_{k \to \infty} \varphi_{\bar{\alpha}}(x^k) = 0$. Then for each cluster point x^* of $\{x^k\}$ we have $\varphi_{\bar{\alpha}}(x^*) = 0$, that is x^* solves VI.

4 Numerical experiments

In this section we show some preliminary numerical results for the algorithm proposed in Section 3. We implemented the algorithm in MATLAB 7.0.4 and we set the matrix G as the identity matrix. As stopping criterion we used the natural residual: $\|x - \Pi_{X,I}(x - F(x))\| < 10^{-4}$. In the test problems we chose the feasible set $X = [1, 7] \times \cdots \times [1, 7]$ and the map $F : \mathbb{R}^n \to \mathbb{R}^n$ as

$$F(x) = (A + D)x + H(x),$$

where A is a skew-symmetric random matrix, D is a diagonal matrix with nonnegative entries, and H_i is a nonsmooth nondecreasing function of the only variable x_i, for all $i = 1, \ldots, n$. Under these conditions, it is easy to show that F is a nonsmooth monotone (but not strongly monotone) map.

Example 4.1. Consider the VI problem where the components of $F(x)$ are given as follows:

$$
\begin{aligned}
F_1(x) &= -2.3443\,x_2 - 0.2079\,x_3 - 3.4258\,x_4 - 1.4208\,x_5 + \max\{\log(x_1), 1\}\\
F_2(x) &= 2.3443\,x_1 + x_2 + 4.5392\,x_3 - 1.63211\,x_4 + 1.3325\,x_5 + \max\{\log(x_2), 1\}\\
F_3(x) &= 0.2079\,x_1 - 4.5392\,x_2 + x_3 - 1.0441\,x_4 - 4.1165\,x_5 + \max\{\log(x_3), 1\}\\
F_4(x) &= 3.4258\,x_1 + 1.6321\,x_2 + 1.0441\,x_3 + 2.5772\,x_5 + \max\{\log(x_4), 1\}\\
F_5(x) &= 1.4208\,x_1 - 1.3325\,x_2 + 4.1165\,x_3 - 2.5772\,x_4 + x_5 + \max\{\log(x_5), 1\}
\end{aligned}
$$

Preliminary computational results show that setting $\alpha_k = 1/10^k$, $\gamma = 0.2$, $\beta = 0.2$, and $\eta = 0.5$ provides a good parameter choice. We applied the algorithm, with such choice of parameters, to solve this example with several vertices of X as initial points. Numerical results are summarized in Table 1 containing seven columns: starting point, number of outer iterations, number of inner iterations, number of projections, number of evaluations of the operator, natural residual at x, and the found approximate solution x.

Table 1: Numerical results for Example 4.1 with $\alpha_k = 1/10^k$, $\gamma = 0.2$, $\beta = 0.2$, $\eta = 0.5$.

starting point	outer iter.	inner iter.	proj.	eval. of F	natural residual	approximate solution
$(1,1,1,1,1)$	4	8	45	57	3.38E-05	$(7,1,6.389768,1,1)$
$(1,1,1,7,1)$	4	10	50	64	3.38E-05	$(7,1,6.389768,1,1)$
$(1,1,7,1,1)$	4	8	45	57	3.38E-05	$(7,1,6.389768,1,1)$
$(1,1,7,7,1)$	4	8	45	57	3.38E-05	$(7,1,6.389768,1,1)$
$(1,7,1,1,1)$	4	9	47	60	3.38E-05	$(7,1,6.389768,1,1)$
$(1,7,1,7,1)$	4	9	47	60	3.38E-05	$(7,1,6.389768,1,1)$
$(1,7,7,1,1)$	4	8	45	57	3.38E-05	$(7,1,6.389768,1,1)$
$(1,7,7,7,1)$	4	8	45	57	3.38E-05	$(7,1,6.389768,1,1)$
$(7,1,1,1,1)$	4	8	45	57	3.38E-05	$(7,1,6.389768,1,1)$
$(7,1,1,7,1)$	4	9	48	61	3.38E-05	$(7,1,6.389768,1,1)$
$(7,1,7,1,1)$	4	7	43	54	3.38E-05	$(7,1,6.389768,1,1)$
$(7,1,7,7,1)$	4	8	45	57	3.38E-05	$(7,1,6.389768,1,1)$
$(7,7,1,1,1)$	4	8	45	57	3.38E-05	$(7,1,6.389768,1,1)$
$(7,7,1,7,1)$	4	9	47	60	3.38E-05	$(7,1,6.389768,1,1)$
$(7,7,7,1,1)$	4	8	45	57	3.38E-05	$(7,1,6.389768,1,1)$
$(7,7,7,7,1)$	4	8	45	57	3.38E-05	$(7,1,6.389768,1,1)$

Table 1 shows that the algorithm is quite robust with respect to the starting point, as it always converges to the same solution. Besides, the number of outer iterations is always equal to 4 and the number of inner iterations is always between 8 and 10. The number of projections and of evaluations is stable as well, always between 43 and 50, and between 54 and 64, respectively.

Computational tests have been then carried out to show the behavior of the algorithm with different parameter values. First, the behavior with respect to different choices of sequence α_k is shown in Table 2: $\alpha_k = 1/k$, $\alpha_k = 1/k^2$, $\alpha_k = 1/2^k$, and $\alpha_k = 1/10^k$. The other parameters are set as in Table 1: $\gamma = 0.2$, $\beta = 0.2$, $\eta = 0.5$.

Different choices of the sequence α_k are tested over the set of vertices as starting points. In Table 2, for each choice of the sequence α_k, the average and maximum number of outer iterations, inner iterations, projections, and evaluations of F are given. Computational results show that exponential sequences, such as $1/2^k$ or $1/10^k$, seem to provide a good behavior while choosing polynomial sequences seems to lead to a greater number of iterations.

In Table 3 results with different values of parameter γ (from 0.1 to 0.9) are shown. Results are obtained keeping values of other parameters as in Table 1. For each choice of γ, the average and maximum number of outer iterations, inner iterations, projections, and evaluation of F are given.

According to the results the algorithm seems to perform well for $0.2 \leq \gamma \leq 0.5$, while the number of needed iterations, projections, and evaluations increases for $\gamma = 0.1$ and for $\gamma \geq 0.6$.

Finally, keeping values of α_k and γ as in Table 1, we show in Table 4 results for different values of parameters β and η between 0.1 and 0.9, with $\beta < \eta$.

For the considered example the choice of β and η seems to be less important than the choice of γ: in fact the average number of projections is between 45 and 50 for any chosen value, and the number of needed evaluations is between 57 and 63.

For the first considered example the most important role seems to be played by α_k, while β and η seem to be not so important. Although γ does not influence as α_k, its value has a quite significant importance.

Table 2: Behavior of the algorithm with respect to different α_k choice.

	outer iterations		inner iterations		projections		evaluations of F	
α_k	avg.	max	avg.	max	avg.	max	avg.	max
$1/k$	9741	9741	8.3	10	29256.7	29261	39006.0	39012
$1/k^2$	99	99	8.3	10	330.7	335	438.0	444
$1/2^k$	14	14	8.3	10	75.6	80	97.9	104
$1/10^k$	4	4	8.3	10	45.6	50	57.8	64

Table 3: Behavior of the algorithm with respect to different γ choice.

	outer iterations		inner iterations		projections		evaluations of F	
γ	avg.	max	avg.	max	avg.	max	avg.	max
0.1	5	5	25.3	27	135.6	140	165.8	172
0.2	4	4	8.3	10	45.6	50	57.8	64
0.3	4	4	10.3	12	66.6	72	80.8	88
0.4	3	3	9.3	11	58.6	64	70.8	78
0.5	4	4	7.3	9	55.6	61	66.8	74
0.6	5	5	13.3	15	139.6	146	157.9	166
0.7	4	4	7.3	9	91.7	99	102.9	112
0.8	4	4	13.2	14	173.7	181	190.9	199
0.9	5	5	37.3	39	1190.1	1206	1232.3	1250

Table 4: Behavior of the algorithm with respect to different β and η choice.

		outer iterations		inner iterations		projections		evaluations of F	
β	η	avg.	max	avg.	max	avg.	max	avg.	max
0.1	0.2	4	4	8.3	10	45.6	50	57.8	64
0.1	0.3	4	4	8.3	10	45.6	50	57.8	64
0.1	0.4	4	4	8.3	10	45.6	50	57.8	64
0.1	0.5	4	4	8.3	10	45.6	50	57.8	64
0.1	0.6	5	5	8.3	10	48.6	53	61.8	68
0.1	0.7	5	5	8.3	10	48.6	53	61.8	68
0.1	0.8	5	5	8.3	10	48.6	53	61.8	68
0.1	0.9	5	5	8.3	10	48.6	53	61.8	68
0.2	0.3	4	4	8.3	10	45.6	50	57.8	64
0.2	0.4	4	4	8.3	10	45.6	50	57.8	64
0.2	0.5	4	4	8.3	10	45.6	50	57.8	64
0.2	0.6	5	5	8.3	10	48.6	53	61.8	68
0.2	0.7	5	5	8.3	10	48.6	53	61.8	68
0.2	0.8	5	5	8.3	10	48.6	53	61.8	68
0.2	0.9	5	5	8.3	10	48.6	53	61.8	68
0.3	0.4	4	4	8.3	10	45.6	52	57.9	66
0.3	0.5	4	4	8.3	10	45.6	52	57.9	66
0.3	0.6	5	5	8.3	10	48.6	55	61.9	70
0.3	0.7	5	5	8.3	10	48.6	55	61.9	70
0.3	0.8	5	5	8.3	10	48.6	55	61.9	70
0.3	0.9	5	5	8.3	10	48.6	55	61.9	70
0.4	0.5	4	4	8.3	10	45.6	52	57.9	66
0.4	0.6	5	5	8.3	10	48.6	55	61.9	70
0.4	0.7	5	5	8.3	10	48.6	55	61.9	70
0.4	0.8	5	5	8.3	10	48.6	55	61.9	70
0.4	0.9	5	5	8.3	10	48.6	55	61.9	70
0.5	0.6	5	5	8.3	10	48.7	55	61.9	70
0.5	0.7	5	5	8.3	10	48.7	55	61.9	70
0.5	0.8	5	5	8.3	10	48.7	55	61.9	70
0.5	0.9	5	5	8.3	10	48.7	55	61.9	70
0.6	0.7	5	5	8.3	10	48.8	56	62.1	71
0.6	0.8	5	5	8.3	10	48.8	56	62.1	71
0.6	0.9	5	5	8.3	10	48.8	56	62.1	71
0.7	0.8	5	5	8.5	14	49.7	73	63.2	92
0.7	0.9	5	5	8.5	14	49.7	73	63.2	92
0.8	0.9	5	5	8.6	14	50.1	75	63.7	94

Example 4.2. Consider the VI problem where the components of $F(x)$ are given as follows:

$$
\begin{aligned}
F_1(x) = \;& -1.8897\,x_2 - 1.8640\,x_3 + 0.9461\,x_4 + 2.1910\,x_5 \\
& +1.9724\,x_6 - 0.1430\,x_7 - 2.2689\,x_8 + 3.3547\,x_9 - 0.1707\,x_{10} + \max\{e^{x_1-4}, 4\} \\
F_2(x) = \;& 1.8897\,x_1 + x_2 - 0.3930\,x_3 + 0.5227\,x_4 - 0.1551\,x_5 \\
& -2.2249\,x_6 - 0.9974\,x_7 + 1.6434\,x_8 + 0.0714\,x_9 + 0.9947\,x_{10} + \max\{e^{x_2-4}, 4\} \\
F_3(x) = \;& 1.8640\,x_1 + 0.3930\,x_2 + x_3 - 0.6498\,x_4 + 1.8380\,x_5 \\
& -2.7493\,x_6 - 2.5758\,x_7 - 2.3058\,x_8 + 2.9067\,x_9 + 3.3159\,x_{10} + \max\{e^{x_3-4}, 4\} \\
F_4(x) = \;& -0.9461\,x_1 - 0.5227\,x_2 + 0.6498\,x_3 + x_4 + 3.0704\,x_5 \\
& +1.1716\,x_6 - 1.5065\,x_7 + 1.4465\,x_8 + 1.6084\,x_9 + 4.4847\,x_{10} + \max\{e^{x_4-4}, 4\} \\
F_5(x) = \;& -2.1910\,x_1 + 0.1551\,x_2 - 1.8380\,x_3 - 3.0704\,x_4 + x_5 \\
& -1.7578\,x_6 + 0.1742\,x_7 + 1.3372\,x_8 + 1.0249\,x_9 + 2.9095\,x_{10} + \max\{e^{x_5-4}, 4\} \\
F_6(x) = \;& -1.9724\,x_1 + 2.2249\,x_2 + 2.7493\,x_3 - 1.1716\,x_4 + 1.7578\,x_5 \\
& + x_6 + 0.4999\,x_7 - 0.3121\,x_8 + 2.3238\,x_9 + 1.5032\,x_{10} + \max\{e^{x_6-4}, 4\} \\
F_7(x) = \;& 0.1430\,x_1 + 0.9974\,x_2 + 2.5758\,x_3 + 1.5065\,x_4 - 0.1742\,x_5 \\
& -0.4999\,x_6 + x_7 - 0.7091\,x_8 + 0.4407\,x_9 - 0.6773\,x_{10} + \max\{e^{x_7-4}, 4\} \\
F_8(x) = \;& 2.2689\,x_1 - 1.6434\,x_2 + 2.3058\,x_3 - 1.4465\,x_4 - 1.3372\,x_5 \\
& +0.3121\,x_6 + 0.7091\,x_7 + x_8 + 0.5291\,x_9 - 2.1871\,x_{10} + \max\{e^{x_8-4}, 4\} \\
F_9(x) = \;& -3.3547\,x_1 - 0.0714\,x_2 - 2.9067\,x_3 - 1.6084\,x_4 - 1.0249\,x_5 \\
& -2.3238\,x_6 - 0.4407\,x_7 - 0.5291\,x_8 + x_9 - 1.1628\,x_{10} + \max\{e^{x_9-4}, 4\} \\
F_{10}(x) = \;& 0.1707\,x_1 - 0.9947\,x_2 - 3.3159\,x_3 - 4.4847\,x_4 - 2.9095\,x_5 \\
& -1.5032\,x_6 + 0.6773\,x_7 + 2.1871\,x_8 + 1.1628\,x_9 + x_{10} + \max\{e^{x_{10}-4}, 4\}
\end{aligned}
$$

Preliminary computational results show that a good choice of parameters for this example is given by $\alpha_k = 1/2^k$, $\gamma = 0.4$, $\beta = 0.5$, $\eta = 0.6$. Results for such parameters with several vertices of X as initial points are shown in Table 5.

Table 5: Numerical results for Example 4.2 with $\alpha_k = 1/2^k$, $\gamma = 0.4$, $\beta = 0.5$, $\eta = 0.6$.

starting point	outer iter.	inner iter.	proj.	eval. of F	natural residual	approximate solution
$(1,1,1,7,1,1,1,7,1,1)$	17	19	171	207	2.33E-05	$(1,1,1,1,1,1,1,1,6.003977,1)$
$(1,1,1,7,1,1,7,7,7,1)$	6	15	76	97	1.41E-05	$(1,1,1,1,1,1,1,1,6.003981,1)$
$(1,1,1,7,7,1,1,7,1,1)$	15	13	116	144	3.50E-05	$(1,1,1,1,1,1,1,1,6.003984,1)$
$(1,1,1,7,7,1,7,7,1,1)$	11	10	85	106	6.84E-05	$(1,1,1,1,1,1,1,1,6.003971,1)$
$(1,1,7,7,1,1,1,7,1,1)$	11	10	85	106	6.84E-05	$(1,1,1,1,1,1,1,1,6.003971,1)$
$(1,1,7,7,1,1,7,7,1,1)$	11	12	90	113	6.84E-05	$(1,1,1,1,1,1,1,1,6.003971,1)$
$(1,1,7,7,7,1,1,7,1,1)$	14	14	129	157	5.34E-05	$(1,1,1,1,1,1,1,1,6.003973,1)$
$(1,1,7,7,7,1,7,7,1,1)$	15	16	145	176	6.01E-05	$(1,1,1,1,1,1,1,1,6.003972,1)$
$(7,1,1,7,1,1,1,7,1,1)$	15	20	157	192	2.05E-06	$(1,1,1,1,1,1,1,1,6.003979,1)$
$(7,1,1,7,1,1,7,7,1,1)$	15	15	122	152	3.50E-05	$(1,1,1,1,1,1,1,1,6.003984,1)$
$(7,1,1,7,7,1,1,7,1,1)$	16	23	210	249	6.02E-05	$(1,1,1,1,1,1,1,1,6.003972,1)$
$(7,1,1,7,7,1,7,7,1,1)$	15	14	118	147	3.50E-05	$(1,1,1,1,1,1,1,1,6.003984,1)$
$(7,1,7,7,1,1,1,7,1,1)$	11	11	87	109	6.84E-05	$(1,1,1,1,1,1,1,1,6.003971,1)$
$(7,1,7,7,1,1,7,7,1,1)$	15	22	186	223	8.57E-05	$(1,1,1,1,1,1,1,1,6.003969,1)$
$(7,1,7,7,7,1,1,7,1,1)$	11	11	87	109	6.84E-05	$(1,1,1,1,1,1,1,1,6.003971,1)$
$(7,1,7,7,7,1,7,7,1,1)$	11	11	87	109	6.84E-05	$(1,1,1,1,1,1,1,1,6.003971,1)$

Although the number of iterations, projections, and evaluations changes starting from different initial point, the algorithm seems to be robust also for the second example. In fact the number of outer iterations is between 6 and 17 for any starting point, the number of inner iterations is between 10 and 23, the number of projections is between 76 and 210, and the number of evaluations of F is between 97 and 249.

In Tables 6–8 the sensitivity of the algorithm to the parameter values is studied. Table 6 is devoted to the α_k sequence and shows results for $\alpha_k = 1/k$, $\alpha_k = 1/k^2$, $\alpha_k = 1/2^k$, and $\alpha_k = 1/10^k$; Table 7 is devoted to parameter γ which varies between 0.1 and 0.9; finally Table 8 gives results on parameters β and η between 0.1 and 0.9 with $\beta < \eta$.

Table 6: Behavior of the algorithm with respect to different α_k choice.

	outer iterations		inner iterations		projections		evaluations of F	
α_k	avg.	max	avg.	max	avg.	max	avg.	max
$1/k$	5902.0	28188	41.1	361	17931.4	84654	23874.5	112858
$1/k^2$	63.2	242	42.1	361	417.0	2143	522.2	2507
$1/2^k$	13.1	17	13.5	23	113.5	210	140.0	249
$1/10^k$	3.3	5	156.0	391	935.3	2314	1094.5	2706

Table 7: Behavior of the algorithm with respect to different γ choice.

	outer iterations		inner iterations		projections		evaluations of F	
γ	avg.	max	avg.	max	avg.	max	avg.	max
0.1	12.8	15	26.6	151	137.3	561	176.7	716
0.2	12.3	16	31.3	120	175.0	415	218.6	536
0.3	14.5	16	24.6	110	161.1	592	200.1	716
0.4	13.1	17	13.5	23	113.5	210	140.0	249
0.5	9.8	17	45.2	75	318.5	461	373.5	539
0.6	2.0	15	125.4	390	806.9	3425	934.3	3818
0.7	5.7	17	2525.5	12423	52952.3	314862	55483.5	327295
0.8	1.6	5	251.7	1333	4100.5	31704	4353.8	33042
0.9	6.2	17	4012.8	19411	276331.0	1577768	280350.0	1597189

According to the computational results on the sequence α_k, the best behavior is obtained with $\alpha_k = 1/2^k$, while $\alpha_k = 1/10^k$ does not seem to perform well in the second example. As for Example 4.1, $\alpha_k = 1/k$ does not provide good results. The influence of γ is more relevant than for Example 4.1. A good choice is to set $0.1 \leq \gamma \leq 0.4$ while with $\gamma \geq 0.5$ the number of projections and evaluations increases significantly. The impact of β and η seems to be important, as well. Results show that setting $\beta \geq 0.6$ and $\eta \geq 0.7$ may cause a significant increase of the number of projections and evaluations. Although there is not a clear rule, it seems that choosing $\beta \leq 0.5$, $\eta \leq 0.6$, and β and η quite close to each other provides a quite good behavior.

Computational tests over the two considered examples show that the behavior of the algorithm is sensitive to the choice of α_k, γ, β, and η. The algorithm behaves differently for the two examples, however some suggestions for the parameter setting can be derived. The most important choice seems to be the choice of the sequence α_k: according to the results exponential sequence is to be recommended. The importance of γ is relevant too: small values of γ (between 0.2 and 0.4) seem to provide good behavior for both examples. Although β and η seem to be less relevant, however they have an impact especially for Example 4.2. Small and close values of β and η seem to perform well.

Table 8: Behavior of the algorithm with respect to different β and η choice.

		outer iterations		inner iterations		projections		evaluations of F	
β	η	avg.	max	avg.	max	avg.	max	avg.	max
0.1	0.2	14.7	16	20.6	29	162.4	220	197.8	265
0.1	0.3	14.7	16	20.4	29	160.3	220	195.3	265
0.1	0.4	8.5	16	203.2	343	1191.4	1966	1403.1	2312
0.1	0.5	8.5	16	203.2	343	1191.4	1966	1403.1	2312
0.1	0.6	8.5	16	203.2	343	1191.4	1966	1403.1	2312
0.1	0.7	4.8	16	555.3	1029	3941.6	8059	4501.8	9094
0.1	0.8	5.1	16	374.5	453	2434.7	2893	2814.3	3350
0.1	0.9	14.0	16	19.0	28	140.3	209	173.3	252
0.2	0.3	13.9	15	15.8	31	126.9	184	156.6	228
0.2	0.4	7.8	15	198.6	342	1157.9	1964	1364.3	2309
0.2	0.5	7.8	15	198.6	342	1158.0	1964	1364.4	2309
0.2	0.6	7.8	15	208.9	359	1217.2	2051	1433.8	2413
0.2	0.7	7.6	16	217.7	359	1276.0	2051	1501.2	2413
0.2	0.8	8.2	16	276.7	447	1779.5	2859	2064.4	3310
0.2	0.9	13.2	18	16.5	26	125.4	222	155.0	264
0.3	0.4	10.6	16	34.0	340	221.0	1959	265.6	2302
0.3	0.5	10.6	17	34.0	340	221.1	1959	265.8	2302
0.3	0.6	12.3	17	33.1	340	221.6	1959	267.0	2302
0.3	0.7	13.0	17	17.8	170	131.5	839	162.3	1011
0.3	0.8	13.1	17	18.6	170	136.4	839	168.1	1011
0.3	0.9	11.4	18	382.4	1191	3000.2	9329	3393.9	10526
0.4	0.5	10.4	15	23.8	340	160.8	1959	195.0	2302
0.4	0.6	12.3	15	22.8	340	161.1	1959	196.1	2302
0.4	0.7	12.8	16	17.6	173	129.7	853	160.1	1028
0.4	0.8	13.0	17	17.9	173	132.4	853	163.3	1028
0.4	0.9	12.8	18	186.2	3947	1559.9	34806	1758.9	38760
0.5	0.6	13.1	17	13.5	23	113.5	210	140.0	249
0.5	0.7	13.6	17	13.5	23	115.1	213	142.2	253
0.5	0.8	13.4	18	25.0	378	180.5	2179	218.9	2560
0.5	0.9	13.1	18	104.5	1071	779.7	8354	897.3	9431
0.6	0.7	11.1	17	121.3	356	732.9	2056	865.3	2415
0.6	0.8	11.1	18	155.3	1060	997.1	8311	1163.5	9377
0.6	0.9	10.5	18	367.4	3316	2760.4	29133	3138.4	32456
0.7	0.8	11.7	18	131.0	1103	869.4	8657	1012.1	9766
0.7	0.9	12.1	18	277.2	3152	2135.5	27650	2424.7	30809
0.8	0.9	14.5	18	424.3	5509	3953.7	54651	4392.5	60169

5 Conclusions

In this paper a modified descent method, with the use of Armijo-type line search, via gap functions for solving nonsmooth variational inequalities is proposed and its global convergence, with less strict assumptions as usual in literature, is proved. Moreover, preliminary computational tests are reported, which provide useful considerations on the parameter settings. It would be very interesting to investigate the convergence rate of the method and to extend it to generalized variational inequalities (i.e. with multivalued operator). Much more examples could be tested to confirm its robustness.

References

[1] F. H. Clarke, Optimization and nonsmooth analysis, John Wiley and Sons: New York, 1983.

[2] F. Facchinei and J.-S. Pang, Finite-dimensional variational inequalities and complementarity problems, Springer: Berlin, 2003.

[3] M. Fukushima, "Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems", Mathematical Programming, vol. 53, pp. 99–110, 1992.

[4] H. Jiang and L. Qi, "Local uniqueness and convergence of iterative methods for nonsmooth variational inequalities", Journal of Mathematical Analysis and Applications, vol. 196, pp. 314–331, 1995.

[5] I. Konnov, "Descent methods for nonsmooth variational inequalities", Computational Mathematics and Mathematical Physics, vol. 46, pp. 1186–1192, 2006.

[6] I. Konnov, B. Panicucci, M. Passacantando, "A derivative-free descent method for nonsmooth variational inequalities", Technical Report of the Department of Applied Mathematics, University of Pisa, vol. 13, 2006.

[7] J.-M. Peng, "Equivalence of variational inequality problems to unconstrained minimization", Mathematical Programming, vol. 78, pp. 347–355, 1997.

[8] M. V. Solodov and P. Tseng, "Some methods based on the D-gap function for solving monotone variational inequalities", Computational Optimization and Applications, vol. 17, pp. 255–277, 2000.

[9] H. Xu, "Regularized gap function and D-gap function for nonsmooth variational inequalities", in "Optimization and Related topics" edited by A. Rubinov and B. Glover, Kluwer Academic Publishers: Dordrecht, pp. 153–176, 2001.

[10] D. L. Zhu and P. Marcotte, "Modified descent methods for solving the monotone variational inequality problem", Operations Research Letters, vol. 14, pp. 111–120, 1993.