

Particle Swarm Optimization for Absolute Value Equations

Longquan YONG[†]

Department of Mathematics, Shaanxi University of Technology, Hanzhong 723001, China

Abstract

We investigate the NP-hard absolute value equation (AVE) $Au - |u| = b$, where A is an arbitrary square matrix. In this paper, we present a smoothing method for the AVE. First, we replace the absolute value function by a smooth one, called aggregate function. With this smoothing technique, we formulate the non-smooth AVE as a smooth nonlinear equations, furthermore, an unconstrained differentiable optimization problem. Then we adopt Particle Swarm Optimization (PSO) to AVE. The numerical experiments show that the proposed algorithm is effective in dealing with the AVE.

Keywords: Absolute Value Equation; Particle Swarm Optimization; Smoothing Method; Aggregate Function

1. Introduction

We consider the absolute value equation (AVE)

$$Ax - |x| = b \quad (1)$$

where $A \in \mathbb{R}^{n \times n}$, $x, b \in \mathbb{R}^n$, and $|x|$ denotes the vector with absolute values of each component of x . A slightly more general form of the AVE was introduced in [1] and investigated in a more general context in [2].

As was shown in [3,4], the general NP-hard linear complementarity problem (LCP) that subsumes many mathematical programming problems can be formulated as an absolute value equation such as (1). This implies that AVE (1) is NP-hard in its general form. Theoretical analysis focuses on the theorem of alternatives, various equivalent reformulations, and the existence and nonexistence of solutions. Reference [1] provides a theorem of the alternatives for a more general form of AVE, $Ax + B|x| = b$, and enlightens the relation between the AVE and the interval matrix. In [5], the AVE is shown to be equivalent to the bilinear program, the generalized LCP, and the standard LCP if 1 is not an eigenvalue of A . Based on the LCP reformulation, sufficient conditions for the existence and nonexistence of solutions are given.

Oleg proved in [6] that the AVE (1) can be equivalently reformulated as a standard LCP without any assumption on A and B , and discussed unique solvability of AVE (1). In [7], Hu and Huang reformulated a system of absolute value equation as a standard linear complementarity problem without any assumption and give some existence and convexity results for the solution set of the AVE (1).

It is worth mentioning that any LCP can be reduced to the AVE [5-7], which owns a very special and simple structure. Hence how to solve the AVE directly attracts much attention. Based on a new

[†] Corresponding author.

Email addresses: yonglongquan@sohu.com (Longquan YONG).

reformulation of the AVE (1) as the minimization of a parameter-free piecewise linear concave minimization problem on a polyhedral, Mangasarian proposed a finite computational algorithm that is solved by a finite succession of linear programs in [8]. In the recent interesting paper [9] of Mangasarian, a semismooth Newton method is proposed for solving the AVE, which largely shortens the computation time than the SLP method. It shows that the semismooth Newton iterations are well defined and bounded when the singular values of A exceed 1. However, the global linear convergence of the method is only guaranteed under more stringent condition than the singular values of A exceed 1. In [10] Mangasarian formulated the NP-hard n -dimensional knapsack feasibility problem as an equivalent absolute value equation in an n -dimensional noninteger real variable space and proposed a finite succession of linear programs for solving the AVE (1).

In [11], a generalized Newton method, which has global and finite convergence, was proposed for the AVE. The method utilizes both the semismooth and the smoothing Newton steps, in which the semismooth Newton step guarantees the finite convergence and the smoothing Newton step contributes to the global convergence. In [12], a smoothing Newton algorithm to solve the AVE (1.1) was presented. The algorithm was proved to be globally convergent and the convergence rate was quadratic under the condition that the singular values of A exceed 1. This condition was weaker than the one used in [9].

Recently, AVE (1) has been investigated in the literature [13-15].

In this paper, we present a new method for solving absolute value equations. We replace the absolute value function by a smooth one, called aggregate function. With this smoothing technique, we formulate the non-smooth AVE as a smooth nonlinear equations, furthermore, an unconstrained differentiable optimization problem. Then we adopt Particle Swarm Optimization (PSO) to AVE. The numerical experiments show that the proposed algorithm is effective in dealing with the AVE.

In section 2, we give a smoothing function and study its properties that will be used in the next section. Meanwhile, we give some propositions or lemmas for AVE that will be used later. In section 3 we describe and present Particle Swarm Optimization (PSO) to AVE. Effectiveness of the method is demonstrated in section 4 by solving some randomly generated AVE problems with singular values of A exceeding 1. Section 5 concludes the paper.

We now describe our notation. All vectors will be column vectors unless transposed to a row vector. The scalar (inner) product of two vectors x and y in the n -dimensional real space R^n will be denoted by $x^T y$. For $x \in R^n$ the 2-norm will be denoted by $\|x\|$, while $|x|$ will denote the vector with absolute values of each component of x . The notation $A \in R^{m \times n}$ will signify a real $m \times n$ matrix. For such a matrix A^T will denote the transpose of A . We write I for the identity matrix, e for the vector of all ones (I and e are suitable dimension in context). A vector of zeros in a real space of arbitrary dimension will be denoted by 0 . For $x = (x_1, x_2, \dots, x_n)^T \in R^n$, $x_{\min} = \min\{x_1, x_2, \dots, x_n\}$, i.e. the minimal component of x . $X = \text{diag}\{x_i\}$ for the diagonal matrix whose elements are the coordinates x_i of $x \in R^n$.

2. Theoretical Background and a Smoothing Function

Define $H: R^n \rightarrow R^n$ by

$$H(x) := Ax - |x| - b. \quad (2)$$

It is clear that x is a solution of the AVE (1) if and only if $H(x) = 0$. H is a nonsmooth function due to the non-differentiability of the absolute value function. In this section we give a smoothing function

of H and study its properties. We first give some properties of H which will be used in the next section.

The following results by Mangasarian [5] and Jiri Rohn [13] characterize solvability of AVE.

Proposition 2.1 (Mangasarian). (Existence of AVE solution)

(i) If 1 is not an eigenvalue of A and the singular values of A are merely greater or equal to 1, then the AVE (1) is solvable if the set $S \neq \emptyset$, where

$$S = \{x \mid (A+I)x - b \geq 0, (A-I)x - b \geq 0\}.$$

(ii) If $b < 0$ and $\|A\|_\infty < \gamma/2$, where $\gamma = \min_i |b_i|/\max_i |b_i|$, then AVE (1) has exactly 2^n distinct solutions, each of which has no zero components and a different sign pattern.

Proposition 2.2 (Mangasarian). (Unique solvability of AVE).

(i) The AVE (1) is uniquely solvable for any $b \in R^n$ if the singular values of A exceed 1.

(ii) The AVE (1) is uniquely solvable for any $b \in R^n$ if $\|A^{-1}\| < 1$.

Proposition 2.3 (Mangasarian). (Existence of nonnegative solution).

Let $A \geq 0$, $\|A\| < 1$ and $b \leq 0$, then a nonnegative solution to the AVE (1) exists.

Proposition 2.4 (Jiri Rohn). If the interval matrix $[A-I, A+I]$ is regular, then for each right-hand side b the equation $Ax - |x| = b$ has a unique solution.

Lemma 2.1[5] For a matrix $A \in R^{n \times n}$, the following conditions are equivalent.

(i) The singular values of A exceed 1.

(ii) The minimum eigenvalue of $A^T A$ exceeds 1.

(iii) $\|A^{-1}\| < 1$.

Lemma 2.2 [16] Suppose that A is nonsingular and $\|A^{-1}B\| < 1$. Then, $A+B$ is nonsingular.

Proof We first show that $I + A^{-1}B$ is nonsingular. For, if not, then for some non-zero vector $x \in R^n$ we have that $(I + A^{-1}B)x = 0$, which shows $\|x\| \leq \|A^{-1}Bx\| \leq \|A^{-1}B\|\|x\|$, so $1 \leq \|A^{-1}B\|$, too, which gives the contradiction. Since $A(I + A^{-1}B)$ is nonsingular, we have $A+B$ is nonsingular.

Lemma 2.3 Let $D = \text{diag}(d)$ with $d_i \in [-1, 1], i = 1, 2, \dots, n$. Suppose that $\|A^{-1}\| < 1$. Then, $A+D$ is nonsingular.

Proof Since $\|A^{-1}D\| \leq \|A^{-1}\|\|D\| < \|D\| \leq 1$, by Lemma 2.2, we have $A+D$ is nonsingular.

Definition 2.1[17] A function $H_\mu := R^n \rightarrow R^n, \mu > 0$ is called a uniformly smoothing approximation function of a non-smooth function $H := R^n \rightarrow R^n$ if, for any $x \in R^n$, H_μ is continuously differentiable, and there exists a constant κ such that

$$\|H_\mu(x) - H(x)\| \leq \kappa\mu, \quad \forall \mu > 0.$$

Where $\kappa > 0$ is constant that does not depend on x .

Obviously absolute value function $|x|$ is non-differentiable. Let $\varphi(x) := |x|$. Since

$$\varphi(x_i) := |x_i| = \max\{x_i, -x_i\}, i = 1, 2, \dots, n,$$

we can adopt the aggregate function introduced in Ref. [18] to smooth the max function. The smoothing approximation function to the function $\varphi(x_i)$ is derived as

$$\varphi_\mu(x_i) = \mu \ln \left\{ \exp\left(\frac{x_i}{\mu}\right) + \exp\left(-\frac{x_i}{\mu}\right) \right\}, i = 1, 2, \dots, n.$$

According to Theorem 3 of Ref. [18], we have

$$0 \leq \varphi_\mu(x_i) - \varphi(x_i) \leq \ln 2 \cdot \mu, \quad i = 1, 2, \dots, n.$$

Thus $\varphi_\mu(x_i)$ is a uniformly smoothing approximation function of $\varphi(x_i)$.

For any $\mu > 0$, let $\varphi_\mu(x) = (\varphi_\mu(x_1), \varphi_\mu(x_2), \dots, \varphi_\mu(x_n))^T$. Define $H_\mu := \mathbb{R}^n \rightarrow \mathbb{R}^n$ by

$$H_\mu(x) = Ax - \varphi_\mu(x) - b. \quad (3)$$

Clearly, H_μ is a smoothing function of H . Now we give some properties of H_μ , which will be used in the next section.

By simple computation, we have

Lemma 2.4 For any $\mu > 0$, the Jacobian of H_μ at $x \in \mathbb{R}^n$ is

$$H'_\mu(x) = A - \text{diag} \left(\frac{\exp\left(\frac{x_i}{\mu}\right) - \exp\left(-\frac{x_i}{\mu}\right)}{\exp\left(\frac{x_i}{\mu}\right) + \exp\left(-\frac{x_i}{\mu}\right)}, i = 1, 2, \dots, n \right).$$

Now we investigate the nonsingularity of the matrix $H'_\mu(x)$. Note that for any $\mu > 0$,

$$\left| \frac{\exp\left(\frac{x_i}{\mu}\right) - \exp\left(-\frac{x_i}{\mu}\right)}{\exp\left(\frac{x_i}{\mu}\right) + \exp\left(-\frac{x_i}{\mu}\right)} \right| = \left| 1 - 2 \frac{\exp\left(-\frac{x_i}{\mu}\right)}{\exp\left(\frac{x_i}{\mu}\right) + \exp\left(-\frac{x_i}{\mu}\right)} \right| < 1, i = 1, 2, \dots, n.$$

Hence, by Lemma 2.3, we obtain the following result.

Theorem 2.1 Suppose that $\|A^{-1}\| < 1$. Then,

$$H'_\mu(x) = A - \text{diag} \left(\frac{\exp\left(\frac{x_i}{\mu}\right) - \exp\left(-\frac{x_i}{\mu}\right)}{\exp\left(\frac{x_i}{\mu}\right) + \exp\left(-\frac{x_i}{\mu}\right)}, i = 1, 2, \dots, n \right).$$

is nonsingular.

Theorem 2.2 Let $H(x)$ and $H_\mu(x)$ be defined as (2) and (3), respectively. Then, $H_\mu(x)$ is a uniformly smoothing approximation function of $H(x)$.

Proof For any $\mu > 0$,

$$\|H_\mu(x) - H(x)\| = \|\varphi_\mu(x) - \varphi(x)\| = \sqrt{\sum_{i=1}^n |\varphi_\mu(x_i) - \varphi(x_i)|^2} \leq \sqrt{n} \ln 2 \cdot \mu.$$

Thus, $H_\mu(x)$ is a uniformly smoothing approximation function of $H(x)$.

Define $\theta := \mathbb{R}^n \rightarrow \mathbb{R}$ by

$$\theta(x) = \frac{1}{2} \|H(x)\|^2.$$

For any $\mu > 0$, Define $\theta := \mathbb{R}^n \rightarrow \mathbb{R}$ by

$$\theta_\mu(x) = \frac{1}{2} \|H_\mu(x)\|^2.$$

we can get the following theorem.

Theorem 2.3 Suppose that $\|A^{-1}\| < 1$. Then, for any $\mu > 0$ and $x \in \mathbb{R}^n$, $\nabla \theta_\mu(x) = 0$ implies that $\theta_\mu(x) = 0$.

Proof For any $\mu > 0$ and $x \in \mathbb{R}^n$,

$$\nabla \theta_{\mu}(x) = [H'_{\mu}(x)]^T H_{\mu}(x).$$

By Theorem 2.1, $H'_{\mu}(x)$ is nonsingular. Hence, if $\nabla \theta_{\mu}(x) = 0$, then $H_{\mu}(x) = 0$ and $\theta_{\mu}(x) = 0$.

3. PSO Algorithm for AVE

In this section, we give Particle swarm optimization (PSO) method for solving $H(x) = 0$. In past several years, PSO has been successfully applied in many research and application areas. It is demonstrated that PSO gets better results in a faster, cheaper way compared with other methods. Another reason that PSO is attractive is that there are few parameters to adjust. One version, with slight variations, works well in a wide variety of applications. Particle swarm optimization has been used for approaches that can be used across a wide range of applications, as well as for specific applications focused on a specific requirement. Now we state this algorithm as follows.

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling [19]. PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles.

Each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called *gbest*. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighbors of the particle. This location is called *pbest*. When a particle takes all the population as its topological neighbors, the best value is a global best and is called *gbest*.

The particle swarm optimization concept consists of, at each time step, changing the velocity of (accelerating) each particle toward its *gbest* and *pbest* locations (local version of PSO). Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward *gbest* and *pbest* locations.

PSO concept is based on a metaphor of social interaction such as bird flocking and fish schooling. Akin to GA, PSO is also population-based and evolutionary in nature, with one major difference from genetic algorithms that it does not implement filtering; that is, all members in the population survive through the entire search process. PSO simulates a commonly observed social behavior, where members of a group tend to follow the lead of the best of the group. The procedure of PSO is illustrated as follows.

i. **Initialization.** Randomly generate a population of the potential solutions, called "particles," and each particle is assigned a randomized velocity.

ii. **Velocity Update.** The particles then "fly" through search hyperspace while updating their own velocity, which is accomplished by considering its own past flight and those of its companions.

The particle's velocity and position are dynamically updated by the following equations:

$$v_{id}^{NEW} = w_i \cdot v_{id}^{OLD} + C_1 \cdot r_1 \cdot (x_{pbest} - x_{id}^{OLD}) + C_2 \cdot r_2 \cdot (x_{gbest} - x_{id}^{OLD}) \quad (4)$$

$$x_{id}^{NEW} = x_{id}^{OLD} + v_{id}^{NEW} \quad (5)$$

where the acceleration coefficients C_1 and C_2 are two positive constants; w_i is an inertia weight and r_1, r_2 is a uniformly generated random number from the range $[0, 1]$ which is generated every time for each iteration. Eberhart and Shi [20] and Hu and Eberhart [21] suggested using $C_1 = C_2 = 2$ and $w_i = 0.5 + rand/2$. Equation (4) shows that, when calculating the new velocity for a particle, the previous velocity of the particle (v_{id}), their own best location that the particles have discovered previously (x_{id}) and the global best location (x_{gbest}) all contribute some influence on the outcome of velocity update. The global best location (x_{gbest}) is identified, based on its fitness, as the best particle among the population. All particles are then accelerated towards the global best particle as well as in the directions of their own best solutions that have been visited previously. While approaching the current best particle from different directions in the search space, all particles may encounter by chance even better particles en route, and the global best solution will eventually emerge. Equation (5) shows how each particle's position (x_{id}) is updated in the search of solution space.

Following we give PSO steps for solving the AVE.

Algorithm 3.1 PSO method for AVE

Step 1. Given $\mu_0 > 0, k = 0$. Establish the fitness value function

$$\theta_{\mu_k}(x) = \frac{1}{2} \|H_{\mu_k}(x)\|^2$$

by using aggregate function $\varphi_{\mu_k}(x)$.

Step 2. Apply PSO to solve $\min_x \theta_{\mu_k}(x)$. Let $x_k = \arg \min_x \theta_{\mu_k}(x)$.

Step 3. Check whether the stopping rule is satisfied. If satisfied, stop.

Step 4. Let $\mu_{k+1} = \mu_k + (1 - e^{\mu_k}) / e^{\mu_k}$, $k := k + 1$. Return to step 2.

Remark. The origin of formula $\mu_{k+1} = \mu_k + (1 - e^{\mu_k}) / e^{\mu_k}$ is the Newton iteration of equation

$e^\mu - 1 = 0$. So μ_k converges to zero quadratically.

4. Computational Results

First we consider one AVE problem where the data (A, b) are

$$A = \begin{bmatrix} 4 & 1 & 0 & 0 \\ 0 & 5 & 1 & 0 \\ 0 & 0 & 6 & 1 \\ 0 & 0 & 0 & 7 \end{bmatrix}, \quad b = \begin{bmatrix} 4 \\ 5 \\ 6 \\ 6 \end{bmatrix}.$$

Since singular values $\text{svd}(A) = [7.2443, 6.0643, 5.0285, 3.8024]$, the AVE is uniquely solvable by proposition 2.2. Solve by Algorithm 3.1, the unique solution to this AVE problem is $x^* = [1, 1, 1, 1]^T$.

Following we consider some randomly generated AVE problem with singular values of A exceeding 1 where the data (A, b) are generated by the Matlab scripts:

```
n=input('dimension of matrix A=');
rand('state',0);
R=rand(n,n);
b=rand(n,1);
A=R'*R+n*eye(n);
```

and we set the random-number generator to the state of 0 so that the same data can be regenerated. More detail of numerical results are presented in Table 1.

Table 1 Computational Results by Algorithm I

Dimension	Iterations	Elapsed Time (in seconds)
4	2	2.2300
8	2	3.3400
16	3	3.7900
32	2	4.1200
64	3	6.6900
128	3	12.4500
256	3	34.6700
512	5	76.5700
1024	5	157.1200

All the experiments were performed on Windows XP system running on a Hp540 laptop with Intel(R) Core(TM) 2×1.8GHz and 2GB RAM, and the codes were written in Matlab6.5.

In all instances the Algorithm 3.1 performs extremely well, and finally converges to an optimal solution for the AVE. We also note that the Algorithm 3.1 converges in very few iterations.

5. Conclusions and Future Work

We have proposed a new smooth method for solving the NP-hard absolute value equation $Ax - |x| = b$ under the less stringent condition that the singular values of A exceed 1. The effectiveness of the algorithm is demonstrated by its ability to solve some randomly generated problems. Possible future work may consist of investigating other intelligent optimization algorithm and improvement of the proposed algorithm here.

Acknowledgement

The author is very grateful to the referees for their valuable comments and suggestions. This work is supported by Natural Science Foundation of Shaanxi Educational Committee under Grant No.09JK381.

References

- [1] Jiri Rohn. A Theorem of the Alternatives for the Equation $Ax + B|x| = b$ [J]. Linear and Multilinear Algebra. 2004, 52(6):421-42
- [2] Mangasarian O.L. Absolute value programming [J]. Computational Optimization and Applications. 2007,36(1):

43-53

- [3] R. W. Cottle and G. B. Dantzig. Complementary pivot theory of mathematical programming, in Mathematics of Decision Sciences, G. B. Dantzig and A. F. Veinott, Jr. (eds.), American Mathematical Society, Providence, R.I., 1968, pp:115-136
- [4] R.W. Cottle, J.S Pang, and R E Stone. The Linear Complementarity Problems. Academic Press, New York, 1992
- [5] Mangasarian O.L., Meyer, R.R. Absolute value equations[J]. Linear Algebra and its Applications. 2006, 419(5): 359-367
- [6] Oleg Prokopyev. On equivalent reformulations for absolute value equations [J]. Computational Optimization and Applications. 2009, 44(3): 363-372
- [7] Shen-Long Hu, Zheng-Hai Huang. A note on absolute value equations [J]. Optim. Lett. 2006, 4(3):417-424
- [8] Mangasarian O.L. Absolute value equation solution via concave minimization [J]. Optim. Lett. 2007, 1(1): 3-8
- [9] Mangasarian O.L. A generalized newton method for absolute value equations [J]. Optim. Lett. 2009, (3): 101-108
- [10] Mangasarian O.L. Knapsack feasibility as an absolute value equation solvable by successive linear programming [J]. Optim. Lett. 2009, 3(2): 161-170
- [11] C. Zhang, Q. J. Wei. Global and Finite Convergence of a Generalized Newton Method for Absolute Value Equations. Journal of Optimization Theory and Applications. 2009, 143(2):391-403
- [12] Louis Caccetta, Biao Qu, Guanglu Zhou. A globally and quadratically convergent method for absolute value equations [J]. Computational Optimization and Applications. (2010) doi: 10.1007/s10589-009-9242-9
- [13] Jiri Rohn. An Algorithm for Solving the Absolute Value Equation [J]. Electronic Journal of Linear Algebra. 2009 (18): 589-599
- [14] Jiri Rohn. On Unique Solvability of the Absolute Value Equation [J]. Optim. Lett. 2009, 3(4): 603-606
- [15] L Yong. Mixed integer linear programming method for absolute value equations[C]. In Proceedings of the 2009 International Symposium on Information Processing, 2009, pp: 316-318
- [16] Stewart, G.W. Introduction to Matrix Computations. Academic Press, San Diego, 1973
- [17] Qi, L., Sun, D. Smoothing functions and smoothing Newton method for complementarity and variational inequality problems [J]. J. Optim. Theory Appl. 2002, 113(1):121-147
- [18] LI Xing-si. An efficient method for non-differentiable optimization problem[J]. Science in China-Series A. 1994, 37 (4): 371-377
- [19] Kennedy, J. and R. C. Eberhart. Particle Swarm Optimization [C]. In Proceedings of the IEEE International Conference on Neural Networks. 1995, pp:1942-1948.
- [20] Eberhart, R. C. and Y. Shi. Tracking and optimizing dynamic systems with particle swarms [C]. In Proceedings of the Congress on Evolutionary Computation. 2001, pp:94-97
- [21] Hu, X. and R. C. Eberhart. Tracking dynamic systems with PSO: where's the cheese?" [C]. In Proceedings of The Workshop on Particle Swarm Optimization. 2001