Absolute Value Equation Solution via Dual Complementarity

Olvi L. Mangasarian *

Abstract

By utilizing a dual complementarity condition, we propose an iterative method for solving the NP-hard absolute value equation (AVE): $Ax - |x| = b$, where A is an $n \times n$ square matrix. The algorithm makes no assumptions on the AVE other than solvability and consists of solving a succession of linear programs. The algorithm was tested on 500 consecutively generated random solvable instances of the AVE with n =10, 50, 100, 500 and 1,000. The algorithm solved 90.2% of the test problems to an accuracy of 10^{-8} .

Keywords: absolute value equation, complementarity, linear programming

1 Introduction

We consider the absolute value equation (AVE):

$$Ax - |x| = b, \tag{1.1}$$

where $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$ are given, and $|\cdot|$ denotes absolute value. A slightly more general form of the AVE, $Ax + B|x| = b$ was introduced in [14] that was investigated algorithmically in [15] and in a more general context in [8]. The AVE (1.1) was investigated in detail theoretically in [11], and a bilinear program in the *primal* space of the problem was prescribed there for the special case when the singular values of A are not less than one. No computational results were given in either [11] or [8]. In contrast in [7], computational results were given for a linear-programming-based successive linearization algorithm utilizing a concave minimization model and a primal-dual bilinear model in [10]. As was shown in [11], the general NP-hard linear complementarity problem (LCP) [2, 3, 1], which subsumes many mathematical programming problems, can be formulated as an AVE (1.1). This implies that (1.1) is NP-hard in its general form. More recently a generalized Newton method was proposed for solving the AVE (1.1) [9], while a uniqueness result for the AVE (1.1) is presented in [16] and for a more general version of the AVE (1.1) in [17]. Finally, existence and convexity results are given in [4], while an algorithm for computing all solutions of an AVE (1.1) is given in [18], and equivalent formulations of the AVE (1.1) are given in [13].

Our present approach consists of formulating the AVE (1.1) as a linear program with a certain requirement on its dual. This requirement ensures, through the dual complementarity condition, that the linear program generates a solution to the AVE (1.1). This requirement on the dual linear program is achieved by successively modifying the primal linear program based on a prior dual program. In most instances this approach eventually obtains a solution to the AVE (1.1). In Section 2 we outline the theory behind our approach and in Section 3 we state our iterative algorithm that consists of solving a succession of linear programs with modified objective functions. In Section 4 we give computational

*Computer Sciences Department, University of Wisconsin, Madison, WI 53706 and Department of Mathematics, University of California at San Diego, La Jolla, CA 92093.*olvi@cs.wisc.edu*.

results that show the effectiveness of our approach by solving 90.2% of a sequence of 500 randomly generated consecutive AVEs in $\mathbb{R}^{10}$ to $\mathbb{R}^{1,000}$ to an accuracy of 10^{-8} . Section 5 concludes the paper.

We describe our notation now. All vectors will be column vectors unless transposed to a row vector by a prime $'$. For a vector $x \in \mathbb{R}^n$ the notation x_j will signify the j-th component. The scalar (inner) product of two vectors x and y in the n-dimensional real space $\mathbb{R}^n$ will be denoted by $x'y$. For $x \in \mathbb{R}^n$, $\|x\|_\infty$ will denote the ∞-norm: $\max\limits_{i=1,\ldots,n} |x_i|$. The notation $A \in \mathbb{R}^{m\times n}$ will signify a real $m \times n$ matrix. For such a matrix, A' will denote the transpose of A. A vector of ones in a real space of arbitrary dimension will be denoted by e, while I will denote the identity matrix. Thus for $e \in \mathbb{R}^m$ and $y \in \mathbb{R}^m$ the notation $e'y$ will denote the sum of the components of y. A vector of zeros in a real space of arbitrary dimension will be denoted by 0. The abbreviation "s.t." stands for "subject to".

2 AVE Solution via Dual Based Successive Linear Programs

We begin with the linear program:

$$\min_{y,z}\ h'(y+z)\ \text{ s.t. }\ A(y-z)-(y+z)=b,\ y\geq 0,\ z\geq 0, \tag{2.2}$$

and its dual:

$$\max_{u}\ b'u\ \text{ s.t. }\ (A'-I)u\leq h,\ \ -(A'+I)u\leq h, \tag{2.3}$$

where the variable x of AVE (1.1) is represented as $x = y - z$ with $y \geq 0,\ z \geq 0$. Here h is an appropriately chosen vector in $\mathbb{R}^n$ that will play a key role in our linear programming formulation. We note immediately that the following proposition gives sufficient conditions for the choice of h, in terms of an optimal dual variable u, that will ensure that a solution of the linear program (2.2) indeed solves AVE (1.1).

Proposition 2.1. *Let (y, z) be a solution of the primal problem (2.2) and u be a solution of the corresponding dual problem (2.3). Then:*

$$h+u>0 \implies Ax-|x|=b,\ \text{where } x=y-z. \tag{2.4}$$

Proof From the dual complementarity condition we have that:

$$y'(h-(A'-I)u)+z'(h+(A'+I)u)=0. \tag{2.5}$$

Hence, if $h-(A'-I)u+h+(A'+I)u = 2(h+u) > 0$, it follows that either $y_i = 0$, or $z_i = 0$, for $i = 1,\ldots,n$. Hence, setting $x = y - z$ it follows that $y + z = |x|$ and from the constraint $A(y - z) - (y + z) = b$, it follows that $Ax - |x| = b$. □

We note that Proposition 2.1 is based on a fundamental idea first utilized in [6, Lemma 1] for solving a general linear complementarity problem by a single linear program.

Based on Proposition 2.1 we have that if $h + u > 0$ then $x = y - z$ solves AVE (1.1). This leads to the following choice of for h:

$$h=\max\{-u+\epsilon e,\epsilon e\}>0, \tag{2.6}$$

for some small positive ϵ. The inclusion of the last ϵe term in the above expression ensures against the possibility when the term $-u+\epsilon e$ includes some negative components which may lead to a primal linear program (2.2) that is unbounded below. Since condition (2.6) cannot be imposed *a priori* because of the lack of knowledge of what the optimal dual variable u will be, this leads us immediately to the iterative procedure that we are proposing in the next section, where u is a dual optimal u of a previous linear programming iteration.

3 Successive Linear Programming Algorithm for the Absolute Value Equation

We begin by stating our successive linear programming algorithm as follows.

ALGORITHM 3.1. *Choose parameter value ϵ for the constraint of (2.6) (typically $\epsilon = 10^{-6}$), tolerance (typically tol=10^{-8}), and maximum number of iterations itmax (typically itmax= 20).*

(I) Initialize the algorithm by determining an initial optimal primal (y^0, z^0) and its corresponding dual solution u^0 by solving the following linear program:

$$\begin{array}{lrl} \min\limits_{y,z} & e'(y+z) & \\ \text{s.t.} & A(y-z)-(y+z) & = b \\ & (y,z) & \geq \; 0. \end{array} \tag{3.7}$$

Set iteration number $i = 0$.

(II) **Set $x^i = y^i - z^i$. While $\|Ax^i - |x^i| - b\|_\infty > tol$ and $i < itmax$ perform the following two steps.**

(III) Obtain a dual optimal variable u^{i+1} by solving the following primal linear program:

$$\begin{array}{lrl} \min\limits_{y,z} & \max\{-u^i+\epsilon, \epsilon e\}'(y+z) & \\ \text{s.t.} & A(y-z)-(y+z) & = b \\ & (y,z) & \geq \; 0. \end{array} \tag{3.8}$$

(IV) Set $i = i + 1$ and go to Step (II).

We note that since the feasible region of (3.8) has a finite number of vertices, then for an infinite sequence $\{u^i\}$ in the above algorithm, one such vertex of (3.8) must appear an infinite number of times if $itmax = \infty$. Hence the corresponding $u^{\bar{i}}$ to that vertex is repeated an infinite number of times also. Thus $u^{\bar{i}}$ is an accumulation of the sequence $\{u^i\}$. This fact in itself does not necessarily ensure that $u^{\bar{i}}$ leads to a solution of AVE (1.1). However, if at any iteration i in the above algorithm, including $\bar{i}$, $u^{i+1} = u^i$, then $x^{i+1} = y^{i+1} - z^{i+1}$ solves AVE (1.1). This is so because for $h = \max\{-u^i + \epsilon e, \epsilon e\}$, we have that:

$$h + u^{i+1} = \max\{-u^i + \epsilon e, \epsilon e\} + u^i \geq \epsilon e > 0. \tag{3.9}$$

Hence it follows by Proposition 2.1 that $x^{i+1} = y^{i+1} - z^{i+1}$ solves AVE (1.1). We thus have the following proposition.

PROPOSITION 3.2. *If for some i in Algorithm 3.1, $u^{i+1} = u^i$, then $x^{i+1} = y^{i+1} - z^{i+1}$ solves AVE (1.1) and Algorithm 3.1 terminates at this solution.*

This proposition does not always guarantee termination of Algorithm 3.1 at an AVE (1.1) solution. However, computationally this appears to be the case in most instances, as demonstrated in the next section where exact solutions are obtained for over 90% of 500 randomly generated problems to an accuracy of 10^{-8}.

4 Computational Results

We implemented our algorithm by solving 500 solvable random instances of the absolute value equation (1.1) consecutively generated. Elements of the matrix A were random numbers picked from a uniform distribution in the interval $[-5, 5]$. A random solution x with random components from [-.5,.5] was generated and the right hand side b was computed as $b = Ax - |x|$. All computation was performed on 4 Gigabyte machine with a 3159MHz CPU running i386 rhe15 Linux. We utilized the CPLEX linear programming code [5] within MATLAB [12] to solve our linear programs.

Of the 500 test problems, over 90% were solved exactly to an ∞-norm tolerance of 10^{-8}. The maximum number of iterations was set at 20 for $n = 10, 50, 100, 500$ and at 40 for $n = 1,000$. The computational results are summarized in Table 1 and are better in the number of problems solved accurately and in total solution time than those given in [10] for a similar set of problems where a bilinear minimization approach was used but only 88.6% of the problems were solved to a lower accuracy of 10^{-6}. Furthermore, the present approach is much more general than that of [9] which is applicable only to AVEs with singular values of A greater than 1.

It is interesting to note that an average of 90.2% solution rate was achieved among the 500 problems described in Table 1 even though Proposition 3.2 does not guarantee a solution of AVE (1.1) by Algorithm 3.1 unless $u^{i+1} = u^i$. One possible justification for our solution rate is that all our test problems have a solution and it is not a simple matter to generate an AVE (1.1) that has *no* solution, to test our algorithm on, nor can our algorithm determine with any certainty that no solution exists for such a problem.

Problem Size n	Number of AVEs out of 100 Solved with ∞-norm Error $\leq tol = 10^{-8}$	Average # of Iterations per Problem	Time in Seconds for Solving 100 AVEs
10	92	6.29	1.7356
50	93	9.15	3.1598
100	91	9.38	13.560
500	85	8.68	2,128.9
1,000	90	7.91	17,053

Table 1: Computational Results for 500 Randomly Generated Consecutive AVEs

5 Conclusion and Outlook

We have proposed a dual-complementarity-based linear programming formulation for solving the NP-hard absolute value equation. The method consists of solving a succession of linear programs. In 90.2% of 500 consecutive instances of solvable random test problems, the proposed algorithm solved the problem to an accuracy of 10^{-8}. Possible future work may consist of precise sufficient conditions under which the proposed formulation and solution method is guaranteed to terminate in a finite number of steps.

Acknowledgments The research described here, based on Data Mining Institute Report 11-03, September 2011, was supported by the Microsoft Corporation and ExxonMobil.

References

[1] S.-J. Chung. NP-completeness of the linear complementarity problem. *Journal of Optimization Theory and Applications*, 60:393–399, 1989.

[2] R. W. Cottle and G. Dantzig. Complementary pivot theory of mathematical programming. *Linear Algebra and its Applications*, 1:103–125, 1968.

[3] R. W. Cottle, J.-S. Pang, and R. E. Stone. *The Linear Complementarity Problem.* Academic Press, New York, 1992.

[4] Sheng-Long Hu and Zheng-Hai Huang. A note on absolute equations. *Optimization Letters*, 4(3):417–424, 2010.

[5] ILOG, Incline Village, Nevada. *ILOG CPLEX 9.0 User's Manual*, 2003. http://www.ilog.com/products/cplex/.

[6] O. L. Mangasarian. Linear complementarity problems solvable by a single linear program. *Mathematical Programming*, 10:263–270, 1976.

[7] O. L. Mangasarian. Absolute value equation solution via concave minimization. *Optimization Letters*, 1(1):3–8, 2007. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/06-02.pdf.

[8] O. L. Mangasarian. Absolute value programming. *Computational Optimization and Applications*, 36(1):43–53, 2007. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/05-04.ps.

[9] O. L. Mangasarian. A generalized Newton method for absolute value equations. Technical Report 08-01, Data Mining Institute, Computer Sciences Department, University of Wisconsin, May 2008. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/08-01.pdf. Optimization Letters 3(1), January 2009, 101-108. Online version: http://www.springerlink.com/content/c076875254r7tn38/.

[10] O. L. Mangasarian. Primal-dual bilinear programming solution of the absolute value equation. Technical report, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, February 2011. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/11-01.pdf, Optimization Letters, to appear.

[11] O. L. Mangasarian and R. R. Meyer. Absolute value equations. *Linear Algebra and Its Applications*, 419:359–367, 2006. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/05-06.pdf.

[12] MATLAB. *User's Guide.* The MathWorks, Inc., Natick, MA 01760, 1994-2006. http://www.mathworks.com.

[13] O. A. Prokopyev. On equivalent reformulations for absolute value equations. *Computational Optimization and Applications*, 44(3):363–372, 2009.

[14] J. Rohn. A theorem of the alternatives for the equation $Ax + B|x| = b$. *Linear and Multilinear Algebra*, 52(6):421–426, 2004. http://www.cs.cas.cz/ rohn/publist/alternatives.pdf.

[15] J. Rohn. An algorithm for solving the absolute value equation. *Electronic Journal of Linear Algebra*, 18:589–599, 2009.

[16] J. Rohn. On unique solvability of the absolute value equation. *Optimization Letters*, 3:603–606, 2009.

[17] J. Rohn. A residual existence theorem for linear equations. *Optimization Letters*, 4(2):287–292, 2010.

[18] J. Rohn. An algorithm for computing all solutions of an absolute value equation. *Optimization Letters*, 2011. To appear.