Optimization Letters, , 1?? (2006)

Absolute Value Equation Solution via Concave Minimization

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

Received March 2006

Editor: Panos Pardalos

Abstract. The NP-hard absolute value equation (AVE) $Ax - |x| = b$ where $A \in R^{n\times n}$ and $b \in R^n$ is solved by a succession of linear programs. The linear programs arise from a reformulation of the AVE as the minimization of a piecewise-linear concave function on a polyhedral set and solving the latter by successive linearization. A simple MATLAB implementation of the successive linearization algorithm solved 100 consecutively generated 1000-dimensional random instances of the AVE with only five violated equations out of a total of $100,000$ equations.

Keywords: absolute value equation, concave minimization, successive linear programming

1. Introduction

We consider the absolute value equation (AVE):

$$Ax - |x| = b, \tag{1}$$

where $A \in R^{n\times n}$, $b \in R^n$ and $|\cdot|$ denotes absolute value. A slightly more general form of the AVE, $Ax + B|x| = b$ was introduced in [11] and investigated in a more general context in [6]. The AVE (1) was investigated in detail theoretically in [7] and a bilinear program 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 [7] or [6] where a parametric successive linearization algorithm was proposed in the latter. As was shown in [7], the general NP-hard linear complementarity problem (LCP) [2, 3, 1] which subsumes many mathematical programming problems can be formulated as an AVE (1). This implies that (1) is NP-hard in its general form.

The basic contribution of the present work is a finite computational algorithm based on a new reformulation of the AVE (1) as the minimization of a parameter-free piecewise linear concave minimization problem on a polyhedral that is solved by a finite succession of linear programs. We prove that the algorithm terminates at a point satisfying a necessary optimality condition. This turns out to be an effective way for solving the AVE as indicated by computational results on one hundred consecutively generated random 1000-dimensional AVEs, 95 of which were solved exactly to an accuracy of 10^{-6}, while each of the remaining five instances had only one equation out of 1000 violated.

In Section 2 of the paper we formulate the AVE as the minimization of a piecewise-linear concave function on a polyhedral set and establish the existence of a vertex solution. In Section 3 we give a successive linearization algorithm (SLA) for the solution of the AVE (1) and establish its finite termination at a vertex satisfying a necessary optimality condition. Section 4 gives our computational results. Section 5 concludes the paper.

A word about our notation and background material. The scalar product of two vectors x and y in the n-dimensional real space will be denoted by $x'y$ in conformity with MATLAB [8] notation. For $x \in R^n$, the norm $\|x\|$ will denote the 2-norm $(x'x)^{\frac{1}{2}}$, while $|x|$ will denote the vector in R^n of absolute values of components of x and $sign(x)$ will denote a vector with components equal to 1, 0 or -1 depending on whether the corresponding component of x is positive, zero or negative. For an $m \times n$ matrix A, A_i will denote the ith row of A. The identity matrix in a real space of arbitrary dimension will be denoted by I, while a column vector of ones of arbitrary dimension will be denoted by e. For a concave function $f : R^n \to R$ the supergradient $\partial f(x)$ of f at x is a vector in R^n satisfying

$$f(y) - f(x) \leq \partial f(x)(y - x) \tag{2}$$

for any $y \in R^n$. The set $D(f(x))$ of supergradients of f at the point x is nonempty, convex, compact and reduces to the ordinary gradient $\nabla f(x)$, when f is differentiable at x [9, 10].

2. The Absolute Value Equation as a Piecewise-Linear Concave Minimization Problem

We show now the equivalence of the AVE to the following concave minimization problem.

Proposition 1 AVE as Concave Minimization *The AVE (1) is equivalent to the following minimization problem:*

$$\begin{array}{rl} \min\limits_{x \in R^n} & e'(Ax - b) - e'|x| \\ s.t. & -(A + I)x \ \leq\ -b, \\ & (-A + I)x \ \leq\ -b, \end{array} \tag{3}$$

having a zero minimum.

Proof: The linear constraints of the concave minimization problem (3) are equivalent to:

$$-Ax + b \ \leq x \ \leq Ax - b, \tag{4}$$

which in turn are equivalent to:

$$|x| \ \leq Ax - b. \tag{5}$$

Hence (3) is feasible if the AVE (1) is solvable and its concave objective is bounded below by zero. Thus, by [10, Corollary 32.3.3] the minimization problem (3) has a solution. This minimum is zero if and only if the AVE (1) is solvable. ■

We now establish that the minimization problem (3) has a vertex solution when it is solvable.

Proposition 2 *The minimization problem (3) has a vertex solution if its feasible region is nonempty, which is the case if the AVE (1) is solvable.*

Proof: By [10, Corollary 32.3.4] the minimization problem (3) has a vertex solution if its nonempty feasible region has no straight lines going to infinity in both directions. We will show that this is indeed the case for the feasible region of (3) in its equivalent form (4). If such a line exists then for some fixed points x and $y \neq 0$ in R^n and for $\lambda \uparrow \infty$ the constraints (4) must be satisfied as:

$$-A(x \pm \lambda y) + b \leq x \pm \lambda y \leq A(x \pm \lambda y) - b. \tag{6}$$

Dividing by λ and letting $\lambda \uparrow \infty$ we have that:

$$\mp Ay \leq \pm y \leq \pm Ay, \tag{7}$$

which is equivalent to:

$$\begin{aligned} (A+I)y &= 0, \\ (A-I)y &= 0. \end{aligned} \tag{8}$$

Subtracting the last equation from the previous one gives that $y = 0$, which contradicts $y \neq 0$. Hence there are no lines going to $\pm\infty$ in the feasible region of (3) and the minimization problem (3) must have a vertex solution. ■

We describe now our successive linearization algorithm for solving the concave minimization problem (3).

3. The Successive Linearization Algorithm (SLA)

Algorithm 1 SLA Algorithm *Start with a random $x^0 \in R^n$. Having x^i determine x^{i+1} as a vertex solution of the following linear program:*

$$\begin{aligned} \min_{x \in R^n} \ & e'(Ax - b) - sign(x^i)'x \\ s.t. \ -(A+I)x &\leq -b, \\ (-A+I)x &\leq -b. \end{aligned} \tag{9}$$

Stop when $(e'A - sign(x^i)')(x^{i+1} - x^i) = 0$.

Remark 2 *Note that minimizing the objective function of (9) is equivalent to minimizing a supporting plane of the convex hypograph of the concave objective function of (3). This is given by the scalar product of a supergradient of the concave objective* *function of (3) at the point x^i times $(x - x^i)$ added to the function value at x^i as follows:*

$$e'(Ax^i - b) - e'|x^i| + (e'A - sign(x^i)')(x - x^i). \tag{10}$$

The finite termination of Algorithm 1 follows from the following theorem.

THEOREM 3 **SLA Finite Termination Theorem** *Let f denote the concave function of the minimization problem (3) which is bounded below by zero on the feasible region of (3) when the AVE (1) is solvable. The Successive Linearization Algorithm 1 generates a finite sequence of feasible vertices $\{x^1, x^2, \ldots, x^{\bar{i}}\}$ with strictly decreasing objective function values: $f(x^1) > f(x^2) > \ldots > f(x^{\bar{i}})$, such that $x^{\bar{i}}$ satisfies the minimum principle necessary optimality condition:*

$$(e'A - sign(x^{\bar{i}}))(x - x^{\bar{i}}) \geq 0, \quad \forall x \in \{x \mid -Ax + b \leq x \leq Ax - b\}. \tag{11}$$

Proof: Follows from Theorem 3 of [5]. ■

We turn now to our computational testing.

4. Computational Results

We used MATLAB 7.1 to test Algorithm 1 on 100 consecutively generated solvable random problems with fully dense matrices $A \in R^{1000\times1000}$ as follows. We first chose a random A from a uniform distribution on [-10,10], then we chose a random x from a uniform distribution on [-1,1]. Finally we computed $b = Ax - |x|$. We broke the 100 problems into 10 groups each containing 10 problems (instances). Code 1 gives the MATLAB code for generating "imax" problems and solving at most "itmax" (typically 40) iterations of of Algorithm 1 by using the CPLEX 9.0 [4] linear programming solver. The maximum number of iterations used per instance was itmax=40. An absolute value equation component i is considered satisfied provided $|A_ix - |x_i| - b_i| \leq 10^{-6}$. This was used as a alternate stopping criterion when satisfied for $i = 1, \ldots, n$ instead of that of the SLA Algorithm 1 of $|(e'A - sign(x^i)')(x^{i+1} - x^i)| = 0$ in order to ensure a 10^{-6} accuracy in satisfying the AVE (1).

Code 1 SLA: Successive Linearization Algorithm Code

```
%generate imax instances of solvable AVE Ax-|x|=b & solve by SLA:
%min  e'(Ax-b)-e'|x| s.t. -Ax+b =< x =< Ax-b
%input: n,imax,itmax (itmax=maximum no. of iterations typically 40)
%output: total number of violated equations for all instnances (nnztot),
%max no. of violated equations per instance (nnzx), total LPs & time (toc)
%lpxt(c,B,d) is CPLEX call to solve  min c'x s.t. Bx =< d
I=eye(n);e=ones(n,1);
nnztot=0;nnzx=0;k=0;tic; %k is total number of LPs for all imax instances
```

```
for i=1:imax
A=10*(rand(n,n)-rand(n,n));x=rand(n,1)-rand(n,1);b=A*x-abs(x);%generate AVE
  j=1;y=rand(n,1)-rand(n,1);%generate initial point y
  while(j<itmax & norm(A*y-abs(y)-b)>1e-6)%itmax or AVE error stop
  c=[A'*e-sign(y)];B=[-(A+I) ;(-A+I) ];d=[-b;-b];[t,z,u]=lpxt(c,B,d);
  y=z; j=j+1;
  end
err=(A*y-abs(y)-b);nnz1=nnz(find(abs(err)>1e-6));
nnztot=nnztot+nnz1;nnzx=max([nnz1 nnzx]);k=k+j-1;
end
[nnztot nnzx k toc]
```

Computational results are summarized in Table 1 and were obtained using a 3.00GHz Pentium 4 processor running i386_tao10 Linux. We note the following:

(i) Out of 100 instances, 95 instances were solved to an accuracy of 10^{-6}.

(ii) For each of the 5 unsolved instances, only one equation out of a thousand equations was violated per instance.

(iii) The average number of LPs per instance varied from 2.6 to 9.1 with the overall average for the 100 problems being 4.81.

(iv) The overall average time for solving each instance of 1000 fully dense equations was 248 seconds.

(v) The average time to solve each LP was 52 seconds.

5. Conclusion

We have proposed a finite successive linear programming algorithm for solving the the NP-hard absolute value equation $Ax-|x|=b$. The effectiveness of the algorithm is demonstrated by its ability to solve 95 out of 100 consecutively generated random problems each consisting of 1000 equations in 1000 unknowns. One fascinating feature of the AVE is its simplicity despite its NP-hardness. Possible future work may consist of investigating other algorithms and improvement of the proposed algorithm here and those in [6, 7].

Acknowledgments

The research described in this Data Mining Institute Report 06-02, March 2006, was supported by National Science Foundation Grants CCR-0138308 and IIS-0511905. Thanks to Michael Thompson for helpful comments on this paper.

Table 1. **Results from Algorithm 1 on 100 consecutive random AVEs with n=1000, shown in groups of 10. nnztot denotes the total number of violated equations in each group of 10 problems. nnzx denotes the maximum number of violated equations per individual problem. k denotes the total number of iterations (LPs) utilized for each group of 10 problems. toc denotes the total time for solving each group of 10 problems. seonds per LP denotes the time for each LP obtained by dividing the previous column by the column preceding it.**

Instances	nztot	nnzx	k	toc(seonds)	seonds per LP
1-10	1	1	80	4236	53
11-20	0	0	26	1361	52
21-30	1	1	74	3804	51
31-40	0	0	26	1351	52
41-50	0	0	26	1354	52
51-60	0	0	26	1364	52
61-70	1	1	80	4207	53
71-80	2	1	91	4379	48
81-90	0	0	26	1358	52
91-100	0	0	26	1359	52

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. ILOG, Incline Village, Nevada. *ILOG CPLEX 9.0 User's Manual*, 2003. http://www.ilog.com/products/cplex/.
5. O. L. Mangasarian. Solution of general linear complementarity problems via nondifferentiable concave minimization. *Acta Mathematica Vietnamica*, 22(1):199–205, 1997. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/96-10.ps.
6. O. L. Mangasarian. Absolute value programming. Technical Report 05-04, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, September 2005. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/05-04.ps. *Computational Optimization and Applications, to appear.*
7. *O. L. Mangasarian and R. R. Meyer. Absolute value equations. Technical Report 05-06, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, December 2005. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/05-06.ps.*
8. *MATLAB.* User's Guide. *The MathWorks, Inc., Natick, MA 01760, 1994-2001. http://www.mathworks.com.*
9. *B. T. Polyak.* Introduction to Optimization. *Optimization Software, Inc., Publications Division, New York, 1987.*
10. *R. T. Rockafellar.* Convex Analysis. *Princeton University Press, Princeton, New Jersey, 1970.*
11. *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.*