

DIAGONAL SCALING OF ILL-CONDITIONED MATRIXES BY GENETIC ALGORITHM

BEHROUZ FATHI VAJARGAH

and

MOJTABA MORADI

Abstract

The purpose of this article is to use genetic algorithm for finding two invertible diagonal matrices D_1 and D_2 such that the scaled matrix D_1AD_2 approaches to minimum condition number.

Mathematics Subject Classification 2000: 65F35,68W20

Additional Key Words and Phrases: Ill conditioned matrix, Condition number, Diagonal scaling, Genetic algorithm.

1. INTRODUCTION

Let $\|\cdot\|$ be a subordinate matrix norm and let A be an invertible matrix. The number

$$\kappa(A) = \text{cond}(A) = \|A\| \|A^{-1}\| \quad (1)$$

is called the condition number of the matrix A , relative to the given matrix norm.

The number $\kappa(A)$ measures the sensitivity of the solution u of the linear system $Ax = b$ to variations in the data A and b ; a feature which is referred to as the condition of the linear system in question. The preceding, therefore, gives sense to a statement such as 'a linear system is well-conditioned or ill-conditioned', according as the condition number of its matrix is 'small' or 'large'.

THEOREM 1.1. *For every matrix A , we have*

$$(1) \text{ cond}(A) \geq 1$$

$$(2) \text{ cond}(\alpha A) = \text{cond}(A); \text{ for every scalar } \alpha \neq 0 .$$

The property $\text{cond}(\alpha A) = \text{cond}(A)$ shows that it is futile to expect to be able to improve the condition of a linear system by multiplying all the equations by a scalar. On the contrary, it is indeed possible to diminish $\text{cond}_2(A)$ by multiplying every row and every column by a suitable number; this is the task of the equilibration of a matrix which may be stated as follows. Given a matrix A , find two invertible diagonal matrices D_1 and D_2 such that

$$\text{cond}(D_1AD_2) = \inf_{\Delta_1, \Delta_2 \in \Upsilon} \text{cond}(\Delta_1A\Delta_2) \quad (2)$$

(Υ denoting the set of all diagonal matrices) relative to a given matrix norm. With this step carried out, the solution of the system $Ax = b$ is effected by solving the system $(D_1AD_2)y = D_1b$ and then calculating $x = D_2y$.

Scaling or equilibration of data in linear systems of equations is a topic of great importance that has already been the subject of many scientific publications, with many different developments depending on the properties one wants to obtain after scaling. It has given rise to several well known algorithms (see [8], [14], for instance). Scaling consists in pre- and post-multiplying a matrix by two diagonal matrices, D_1 and D_2 , respectively. Classical scalings are the well known row and column scaling. A more general purpose scaling method is the one used in the HSL 2000 routine MC29, which aims to make the nonzeros of the scaled matrix close to one by minimizing the sum of the squares of the logarithms of the moduli of the nonzeros (see [7]). MC29 reduces this sum in a global sense and therefore should be useful on a wide range of sparse matrices. Any combination of these scalings is also a possibility. Scaling can also be combined with permutations (see [9] and the HSL 2000 routine MC64). The matrix is first permuted so that the product of absolute values of entries on the diagonal of the permuted matrix is maximized (other measures such as maximizing the minimum element are also options). Then the matrix is scaled so that the diagonal entries are one and the offdiagonals are less than or equal to one. This then provides a useful tool for a good pivoting strategy for sparse direct solvers, as well as for building good preconditioners for an iterative method. In the 1960s, Bauer and van der Sluis, in particular, showed some optimal properties in terms of conditions numbers for scaled matrices with all rows or all columns of equal norm of 1 ([1], [2], [15]). The problem of optimal scaling of matrices with respect to the condition number κ_2 has been extensively studied, as seen in papers presented by Bauer [1], Braatz and Morari [3], Businger [4], Rump [13], Watson [16], Chiang and Chandler [5].

The purpose of this article is to propose genetic algorithm for finding two invertible diagonal matrices D_1 and D_2 such that the scaled matrix D_1AD_2 , decrease the condition number of matrix A .

2. GENETIC ALGORITHM

Genetic algorithms (GAs) are stochastic global search and optimization methods that mimic the metaphor of natural biological evolution. Genetic algorithms operate on a population of potential solutions applying the principle of survival of the fittest to produce successively better approximations to a solution. At each generation of a GA, a new set of approximations is created by the process of selecting individuals according to their level of fitness in the problem domain and reproducing them using operators borrowed from natural genetics. This process leads to the evolution of populations of individuals that are better suited to their environment than the individuals from which they were created, just as in natural adaptation.

Genetic operators control the evolution of successive generations. The three basic genetic operators are reproduction, crossover and mutation. The probability of a given solution's being chosen for reproduction is proportional to the fitness of that solution. Crossover implies that parts of two randomly chosen chromosomes will be swapped to create a new individual. Mutation involves randomly changing an allele

in a solution to look for new points in the solution space. Although there are more elaborate versions of these operators, the basic principles remain similar for most Genetic algorithms. A genetic algorithm starts by generating a number of possible solutions to a problem, evaluates them and applies the basic genetic operators to that initial population according to the individual fitness of each individual. This process generates a new population with higher average fitness than the previous one, which in turn will be evaluated. The cycle is repeated for the number of generations set by the user, which is dependent on problem complexity.

We use GAs algorithm because:

- An easily understood approach that can be applied to a wide range of problems with little or no modification. Other approaches have required substantial alteration to be successfully used in building applications. For example, dynamic programming was applied to the problem of selecting the number, location, and power of lamps along a hallway to minimize the electrical power needed to produce the required illuminance [10]
- The choice of the location and power of a lamp affected decisions made about previous lamps, the sequential decision-making approach inherent in dynamic programming could not be made. It was necessary to suspend earlier decisions and reconsider them, substantially increasing computation time and data storage.
- Publicly available, easily implemented GA codes. Reduced set-up time makes them attractive relative to other optimization methods that may offer better performance but must be identified, obtained and properly configured.
- Inherent capability to work with complex simulation programs. Simulation does not need to be simplified to accommodate optimization.
- Methods to permit Genetic algorithms to handle constraints that would make some solutions unattractive or entirely infeasible.
- Proven effectiveness in solving complex problems that cannot be readily solved with other optimization methods. The mapping of the objective function for a daylighting design problem showed the existence of local minima that would potentially trap a gradient-based method[6].

3. NUMERICAL RESULTS

We start with a simple example to show that GA can find invertible diagonal matrices D_1 and D_2 such that the scaled matrix D_1AD_2 , has the $cond(D_1AD_2)$ close to 1.

Example 3.1. Let

$$A = \begin{pmatrix} 10^5 & 0 \\ 0 & 10^{-5} \end{pmatrix},$$

then $\kappa_2(A) = 10^{10}$. Applying GA (number of generations=1000 and Population size =200), we obtain

$$D_1 = \begin{pmatrix} -0.0014899606376 & 0 \\ 0 & 261.6031905051186 \end{pmatrix}$$

$$D_2 = \begin{pmatrix} 0.0046900908854 & 0 \\ 0 & -267.1240676478366 \end{pmatrix}$$

where $\kappa_2(D_1AD_2) = 1.000000010883026$.

Example 3.2. From BAUER[1] we know that the Optimal scaling of

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 6 \end{pmatrix}$$

is $\kappa_2(A) = (3 + \sqrt{10})^2 = 37.973665961010283$. This is the minimum possible condition number following scaling. In GA we obtain $\kappa_2(D_1AD_2) = 37.973666669263956$, where

$$D_1 = \begin{pmatrix} 978.1747835410918 & 0 & 0 \\ 0 & 892.8483795597657 & 0 \\ 0 & 0 & 461.1142062448835 \end{pmatrix}$$

and

$$D_2 = \begin{pmatrix} -947.1370305923301 & 0 & 0 \\ 0 & 864.4901198012243 & 0 \\ 0 & 0 & -446.3812683112591 \end{pmatrix}$$

Example 3.3. The matrix

$$A = \begin{pmatrix} 0.0926612 & 17.0784926 & 0.3127063 & 12.7526810 \\ 1.7811361 & 54.0213314 & 1.4953060 & 14.7655003 \\ 0.3460217 & 0.0680433 & 0.2626770 & 0.0227214 \\ 1.3745248 & 45.1500312 & 0.0505958 & 1.4314422 \end{pmatrix}$$

has the condition number equal to 460.2704442412975. Chiang and Chandler [5] reduced the condition number of this matrix to 14.4856257 by an iterative algorithm which is called SCALGM. Our result using GA is $\kappa_2(D_1AD_2) = 14.485476520157350$ where

$$D_1 = \begin{pmatrix} -1022.446636286579 & 0 & 0 & 0 \\ 0 & 495.973718177785 & 0 & 0 \\ 0 & 0 & 2743.200397567089 & 0 \\ 0 & 0 & 0 & 545.478809477854 \end{pmatrix}$$

and

$$D_2 = \begin{pmatrix} 2335.779656989865 & 0 & 0 & 0 \\ 0 & 74.778044753772 & 0 & 0 \\ 0 & 0 & -2923.32215707509 & 0 \\ 0 & 0 & 0 & -209.320088460085 \end{pmatrix}$$

4. CONCLUSION

Genetic algorithms supply a very good opportunity for reducing the condition number of a given ill condition matrix.

DIAGONAL SCALING OF ILL-CONDITIONED MATRIXES BY GENETIC ALGORITHM

REFERENCES

- [1] F. L. BAUER, (1963), Optimally scaled matrices, *Numerische Mathematik*, 5, 7387.
- [2] F. L. BAUER, (1969), Remarks on optimally scaled matrices, *Numerische Mathematik*, 13, 13.
- [3] D. R. Braatz, Manfred Morari, (1994), Minimizing the Euclidean Condition Number. *SIAM Journal on Control and Optimization*, 32(6):17631768.
- [4] P. A. Businger, (1968), Matrices Which Can Be Optimally Scaled. *Numerische Mathematik*, 12:346348.
- [5] Chin-Chieh Chiang, John P. Chandler, An Approximate Equation for the Condition Numbers of Well-scaled Matrices Proceedings of The 2008 IAJC-IJME International Conference.
- [6] A. Chutarat, (2001), Experience of Light: The Use of an Inverse Method and a Genetic Algorithm in Daylighting Design, Ph.D. Thesis, Dept. of Architecture, MIT, Cambridge, MA.
- [7] A. R. CURTIS AND J. K. REID, (1972), On the automatic scaling of matrices for Gaussian elimination, *J. Inst. Maths. Applics.*, 10, 118124.
- [8] I. S. DUFF, A. M. ERISMAN, AND J. K. REID, (1986), *Direct Methods for Sparse Matrices*, Oxford University Press, London.
- [9] I. S. DUFF AND J. KOSTER, (1999), On algorithms for permuting large entries to the diagonal of a sparse matrix, Tech. Rep. RAL-TR-1999-030, Rutherford Appleton Laboratory. *SIAM Journal on Matrix Analysis and Applications*.
- [10] Gero, J., and Radford, A. (1978), A Dynamic Programming Approach to the Optimum Lighting Problem, *Eng. Optimize.* 3(2), pp. 71-82.
- [11] HSL, (2000), A collection of Fortran codes for large scale scientific computation. <http://www.cse.clrc.ac.uk/Activity/HSL>.
- [12] D. Ruiz, A scaling algorithm to equilibrate both rows and columns norms in matrices, Tech. Rep. RAL-TR-2001-034, Rutherford Appleton Laboratory, 2001.
- [13] R. Rump, Optimal scaling for p-norms and componentwise distance to singularity, *IMA J. Numer. Anal.* 23 (2003), pp. 19.
- [14] M. H. SCHNEIDER AND S. ZENIOS, (1990), A comparative study of algorithms for matrix balancing, *Operations Research*, 38, 439455.
- [15] A. VAN DER SLUIS, (1969), Condition Numbers and Equilibration of Matrices, *Numer. Math.*, 14, 1423.
- [16] Watson, G. A. (1991), An Algorithm for Optimal ℓ_2 Scaling of Matrices. *IMA J. Numer. Anal.*, 11:481492.

BEHROUZ FATHI VAJARGAH

and

MOJTABA MORADI

Department of Statistics,
Faculty of Mathematical Sciences,
University of Guilan, Rasht,
Iran.

email: fathi@guilan.ac.ir; mmoradi@guilan.ac.ir.

Received October 2011