Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
Electronic Notes in Theoretical Computer Science 225 (2009) 195–200
www.elsevier.com/locate/entcs

An Asymptotic Approach for Testing P_0-Matrices

Lei Li[1,2]

Faculty of Engineering
Hosei University
Koganei, Tokyo 184-8584, Japan

Koya Hattori[1]

Faculty of Engineering
Hosei University
Koganei, Tokyo 184-8584, Japan

Abstract

A direct approach to the P-matrix or P_0-matrix problem is to evaluate all the principal minors of matrix A using standard numerical linear algebra techniques with $O(2^n n^3)$ computational time complexity. The computational time complexity of the P-matrix problem has been reduced from $O(2^n n^3)$ to $O(2^n)$ by applying recursively a criterion for P-matrices based on Schur complementation. But this algorithm can be not directly applied to test the P_0-matrices because the Schur complementation can be not computed when some zero diagonal elements appear.
This paper proposes an asymptotic approach for testing P_0-matrices with $O(2^n)$ computational time complexity. Some numerical examples show that the proposed algorithm is effective for testing P_0-matrices.

Keywords: P_0-matrix, complexity, principal minor, P-matrix.

1 Introduction

Recall that a matrix $A \in R^{n\times n}$ is called a P-matrix if all of its principal minors are positive, and A is called a P_0-matrix if all of its principal minors are nonnegative. P-matrices and P_0-matrices arise in a variety of mathematical contexts and applications (see, e.g., Berman and Plemmons [1]). The P-matrix or P_0-matrix problem, namely, the problem of testing whether a given matrix A is a P-matrix or P_0-matrix, is of importance in many of these applications, specifically in solving the linear complementarity problem. However the P-matrix or P_0-matrix problem

[1] This work was partially supported by the Research Grant of Hosei University.

[2] Email: `lilei@hosei.ac.jp`

1571-0661/
doi:10.1016/j.entcs.2008.12.074

seems inevitably of exponential time complexity. As is shown in Coxson [2], the P-matrix or P_0-matrix problem is co-NP-complete.

It is well known that the following Linear Complementarity Problem often appears in fields of the mathematical programming.

LCP(A, q) : Let $A \in R^{n\times n}$ and $q \in R^n$, finding one or all real vectors z with satisfying

$$Az + q \geq 0, \qquad z \geq 0, \qquad z^T(Az + q) = 0. \tag{1}$$

In fact, the P-matrix problem can be linked to a finite number of test LCP(A, q) having unique solution [4]. If A is a P_0-matrix, then LCP(A, q) has at least one solution [7].

A direct approach to the P-matrix or P_0-matrix problem is to evaluate all the principal minors of A using standard numerical linear algebra techniques with $O(2^n n^3)$ computational time complexity. In [3], the computational time complexity of the P-matrix problem has been reduced from $O(2^n n^3)$ to $O(2^n)$ by applying recursively a criterion for P-matrices based on Schur complementation. But this algorithm can be not directly applied to test the P_0-matrices because the Schur complementation can be not computed when some zero diagonal elements appear.

In this paper, we propose an asymptotic approach for testing the P_0-matrices by replacing the possible zero diagonal elements using an enough small positive number ε in the algorithm shown in [3]. Some numerical examples show that the proposed approach is effective for testing P_0-matrices.

2 An Asymptotic Approach for P_0-Matrices

Definition 2.1 Let matrix $A \in R^{n\times n}$, if all of its principal minors are nonnegative, then A is called a P_0-matrix.

Theorem 2.1[6] Let matrix $A \in R^{n\times n}$, then the following conditions are mutually equivalent.

(1) All principal minors of A are nonnegative.

(2) For any $x \in R^n$, $x \neq 0$, there exists i, $1 \leq i \leq n$ satisfying

$$x_i(Ax)_i \geq 0,$$

where $(Ax)_i$ is the ith element of Ax.

(3) For A and all principal square submatrices of A, their all real eigenvalues are nonnegative.

(4) For all $\varepsilon > 0$, $A + \varepsilon I_n$ is a P-matrix.

(5) For all positive diagonal matrix $D \in R^{n\times n}$, $A + D$ is a P-matrix.

(6) For all positive diagonal matrix $D \in R^{n\times n}$, $det(A + D) > 0$.

From the condition (1) and (4) of Theorem 2.1, it is easy to know, if introduce an enough small positive number ε, we can test P_0-matrix problem by the P- matrix algorithm shown in [3]. Of course, it is an asymptotic algorithm.

For the given matrix $A \in R^{n\times n}$, we block $A = (a_{ij})$ to the following form

$$A = \begin{pmatrix} a_{11} & b^T \\ c & B \end{pmatrix},$$

where

$$b^T = (a_{12}, a_{13}, \dots, a_{1n}), \qquad c^T = (a_{21}, a_{31}, \dots, a_{n1}),$$

$$B = \begin{pmatrix} a_{22} & a_{23} & \dots & a_{2n} \\ a_{32} & a_{33} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots \\ a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix}.$$

Take an enough small positive number ε, when $a_{11} \neq 0$, define the Schur complementation by

$$A/a_{11} = B - \frac{1}{a_{11}} cb^T.$$

Based on the P-matrix algorithm $P(A)$ shown in [3], it is easy to get the following P_0- matrix algorithm for testing P_0-matrices by replacing some possible zero diagonal elements with a small positive number ε.

But when $a_{11} = 0$, if we just replace a_{11} by ε, then this error ε will influence all other operations after this step in the algorithm. It will influence the precision of the testing algorithm. As a matter of fact, by exchanging some lines and rows of A (it is equivalent to multiply a permutation matrix P and consider matrix PAP^T), we can validly decrease this unnecessary precision down.

Consider the following simple example. Let

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 10001 \end{pmatrix}, \qquad \varepsilon = 0.0001$$

Because $det(A) = -1 < 0$, so A is not a P_0-matrix. But if we do not any matrix transformation, from $a_{11} = 0$, replace a_{11} by ε, we have

$$a_{11} = \varepsilon = 0.0001 > 0,$$

$$B = a_{22} = 10001 > 0,$$

$$A/a_{11} = B - a_{11}^{-1} cb^T = 10001 - 10000 \times 1 = 1 > 0.$$

So it is possible to misunderstand A is a P_0-matrix. Consider to do the following matrix transformation,

$$PAP^T = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 10001 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}^T$$

$$= \begin{pmatrix} 10001 & 1 \\ 1 & 0 \end{pmatrix} = \bar{A} = \begin{pmatrix} \bar{a}_{11} & \bar{b}^T \\ \bar{c} & \bar{B} \end{pmatrix}.$$

From

$$\bar{a}_{11} = 10001 > 0, \qquad \bar{B} = \bar{a}_{22} = 0 \geq 0,$$

$$\bar{A}/\bar{a}_{11} = \bar{B} - \bar{a}_{11}^{-1}\bar{c}\bar{b}^T$$

$$= 0 - \frac{1}{10001} \times 1 \times 1 = -\frac{1}{10001} < 0,$$

the above wrong judgment can be avoided.

Based on the above discussion, we propose the following algorithm.

P_0-matrix Algorithm $P_0(A)$

(1) Input $A = (a_{ij}) \in R^{n\times n}$, and an enough small positive number ε.

(2) If there exists i, $1 \leq i \leq n$, $a_{ii} < 0$, then output "A is not the P_0-matrix", Stop.

(3) If $a_{11} = 0$, then go to step (4), else go to step (5).

(4) If there exists k, $k > 1$ and $a_{kk} > 0$, then exchange the first row and kth row, the first column and kth column, else let $a_{11} = \varepsilon$.

(5) Call $P_0(B)$, Call $P_0(A/a_{11})$.

(6) Output "A is a P_0-matrix".

We show a simple example for using the above P_0-matrix algorithm. Let

$$A = \begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -2 & 1 \end{pmatrix},$$

because $a_{11} = 0$, in the first, we do exchange of the first line and the third line, the first row and the third row, and get

$$A \to \begin{pmatrix} 1 & -2 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{pmatrix}$$

so, zero elements are concentrated in the right down of the diagonal line

$$a_{11} = 1 > 0, \qquad B = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \qquad A/a_{11} = \begin{pmatrix} 2 & 0 \\ 3 & 1 \end{pmatrix}.$$

But it is easy to know $B \in P_0$ and $A/a_{11} \in P_0$, so we can conclude that A is a P_0-matrix.

3 Numerical Examples

The following four matrix examples are tested by using the above P_0-matrix algorithm when $n = 15, 20, 25, 30$. Used computer environment includes CPU Xeon (TM), 2.40GHz, the memory 1.5GB, Windows XPpro and Visual $C++6.0$. Example 3.1 and Example 3.2 are P_0-matries, and Example 3.3 and Example 3.4 are not P_0-matrices. Test results shown the algorithm is correct and practical. Running time (Second) are showed in the Table 1 where $\varepsilon = 0.0001$.

Example 3.1 Upper triangular matrix $A = (a_{ij}) \in R^{n\times n}$, $a_{ij} = k$, if $i \le j$, otherwise $a_{ij} = 0$. Where, k is a random integer number between $0 \sim 9$. It is obvious that A is a P_0-matrix.

Example 3.2

$$A = \begin{pmatrix} a & a & a & \cdots \\ b & b & b & \cdots \\ c & c & c & \cdots \\ \vdots & \vdots & \vdots & \end{pmatrix}$$

where $a, b, c, \ldots,$ are random integer numbers between $0 \sim 9$. It is obvious that A is a P_0-matrix.

Example 3.3

$$A = \left(\begin{array}{c|cccc} 0 & 1 & \cdots & \cdots & 1 \\ \hline -1 & & & & \\ \vdots & & & P & \\ -1 & & & & \\ 1 & & & & \end{array}\right)$$

where $P \in R^{(n-1)\times(n-1)}$ is a P-matrix. It is easy to know A is not a P_0-matrix.

Example 3.4

$$A = \left(\begin{array}{c|c} 0 & B \\ \hline B & 0 \end{array}\right)$$

where $B \in R^{\frac{n}{2}\times\frac{n}{2}}$ is a positive matrix (we assume n is an even number). It is easy to know A is not a P_0-matrix.

Table 1
Running Times (sec) of Testing the P_0-matrices

	n=15	n=20	n=25	n=30
Example 3.1	0.011	0.201	5.802	189.528
Example 3.2	0.011	0.206	5.983	190.256
Example 3.3	0.005	0.105	2.982	95.172
Example 3.4	-	0.001	-	0.001

4 Conclusions

This paper proposed an asymptotic approach for testing the P_0-matrices by replacing the possible zero diagonal elements in the algorithm shown in [3] by an enough small positive number ε. Some numerical examples shown that the proposed approach is effective and practical for testing P_0-matrices.

References

[1] A. Berman, R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, SIAM Press, Philadelphia, 1994.

[2] G. E. Coxson, *The P-matrix problem is co-NP-complete*, Math. Programming, 64 (1994), pp. 173–178.

[3] M. J. Tsatsomeros and Lei Li, *A Recursive Test for P-matrices*, BIT, 40 (2000), pp. 410–414.

[4] M. M. Kostreva, *Finite test sets and P-matrices*, Proc. Amer. Math. Soc., 84 (1982), pp. 104–105.

[5] S. Kodama and N. Suda, *Matrix Theory in the System Control*, The Society of Instrument and Control Engineers, Tokyo, 1978.

[6] K. Nagasaka and Y. Komaki, *Linear Algebra in Sciences and Technologies*, SHOKABO Pub. Co., Ltd, Tokyo, 1999.

[7] X. Chen and Y. Ye, *On Smoothing Methods for the P_0 Matrix LCP*, SIAM J. Optimization, 11(2000), pp.341-363.