

Electronic Notes in Theoretical Computer Science 225 (2009) 195–200

[www.elsevier.com/locate/entcs](http://www.elsevier.com/locate/entcs)

An Asymptotic Approach for Testing

*P*0-Matrices

## Lei Li[1](#_bookmark0)*,*[2](#_bookmark0)

*Faculty of Engineering Hosei University*

*Koganei, Tokyo 184-8584, Japan*

## Koya Hattori[1](#_bookmark0)

*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*nn*3) computational time complexity. The computational time complexity of the *P* -matrix problem has been reduced from *O*(2*nn*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 com- plexity. Some numerical examples show that the proposed algorithm is effective for testing *P*0-matrices.

*Keywords: P*0-matrix, complexity, principal minor, *P* -matrix.

# Introduction

Recall that a matrix *A ∈ Rn×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 nonneg- ative. *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](mailto:lilei@hosei.ac.jp)

1571-0661/Crown Copyright © 2008 Published by Elsevier B.V. Open access under [CC BY-NC-ND license.](http://creativecommons.org/licenses/by-nc-nd/3.0/)

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 ap- pears in fields of the mathematical programming.

LCP(*A*, *q*) : Let *A ∈ Rn×n* and *q ∈ Rn*, finding one or all real vectors *z* with satisfying

*Az* + *q ≥* 0*, z ≥* 0*, zT* (*Az* + *q*) = 0*.* (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*nn*3) computational time complexity. In [3], the computational time complex- ity of the *P* -matrix problem has been reduced from *O*(2*nn*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.

# An Asymptotic Approach for *P*0-Matrices

**Definition 2.1** Let matrix *A ∈ Rn×n*, if all of its principal minors are nonnegative, then *A* is called a *P*0-matrix.

**Theorem 2.1**[6] Let matrix *A ∈ Rn×n*, then the following conditions are mutu- ally equivalent.

* 1. All principal minors of *A* are nonnegative.
  2. For any *x ∈ Rn*, *x /*= 0, there exists *i*, 1 *≤ i ≤ n* satisfying

*xi*(*Ax*)*i ≥* 0*,* where (*Ax*)*i* is the *i*th element of *Ax*.

* 1. For *A* and all principal square submatrices of *A*, their all real eigenvalues are nonnegative.
  2. For all *ε >* 0, *A* + *εIn* is a *P* -matrix.
  3. For all positive diagonal matrix *D ∈ Rn×n*, *A* + *D* is a *P* -matrix.
  4. For all positive diagonal matrix *D ∈ Rn×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 ∈ Rn×n*, we block *A* = (*aij*) to the following form

*a*11 *bT*

where

*A* = *c B ,*

*bT* = (*a*12*, a*13*,..., a*1*n*)*, cT* = (*a*21*, a*31*,..., an*1)*,*

⎛ ⎞

*a*22 *a*23 *... a*2*n a*32 *a*33 *... a*3*n*

= ⎜⎝ ⎟⎠

*B ... ... ... . .. .*

*an*2 *an*3 *... ann*

Take an enough small positive number *ε*, when *a*11 */*= 0, define the Schur com- plementation by

*A/a*11

= *B −* 1

*a*11

*cbT .*

Based on the *P* -matrix algorithm *P* (*A*) shown in [3], it is easy to get the fol- lowing *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 *PAPT* ), we can validly decrease this unnecessary precision down.

Consider the following simple example. Let

*A* = ⎛ 0 1

⎞ *, ε* = 0*.*0001

⎝ 1 10001 ⎠

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 = *ε* = 0*.*0001 *>* 0*, B* = *a*22 = 10001 *>* 0*,*

*A/a*11 = *B − a−*1*cbT* = 10001 *−* 10000 *×* 1 = 1 *>* 0*.*

11

So it is possible to misunderstand *A* is a *P*0-matrix. Consider to do the following matrix transformation,

⎛ 0 1 ⎞ ⎛ 0 1

⎞ ⎛ 0 1 ⎞*T*

*PAPT* = ⎝

1 0 ⎠ ⎝

1 10001 ⎠ ⎝

1 0 ⎠

⎛ 10001 1 ⎞ ¯ ⎛ *a*¯11 ¯*bT* ⎞

= ⎝ 1 0 ⎠ = *A* = ⎝ ¯ ⎠ *.*

*B*

*c*¯

From

*a*¯11 = 10001 *>* 0*,*

*B*¯ = *a*¯22 = 0 *≥* 0*,*

*A*¯*/a*¯11 = *B*¯ *− a*¯*−*1*c*¯¯*bT*

11

= 0 *−* 1

10001

1

*×* 1 *×* 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* = (*aij*) *∈ Rn×n*, and an enough small positive number *ε*.
2. If there exists *i*, 1 *≤ i ≤ n*, *aii <* 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).

1. If there exists *k*, *k >* 1 and *akk >* 0, then exchange the first row and *k*th row, the first column and *k*th column, else let *a*11 = *ε*.
2. Call *P*0(*B*), Call *P*0(*A/a*11).
3. Output ”*A* is a *P*0-matrix”.

We show a simple example for using the above *P*0-matrix algorithm. Let

⎛ 0 1 1 ⎞

*A* = *,*

⎜⎝ *−*1 0 1 ⎟⎠

*−*1 *−*2 1

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

⎛ 1 *−*2 *−*1 ⎞

*A →* ⎜ 1 0 *−*1 ⎟

⎝ 1 1 0 ⎠

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

*a* = 1 *>* 0*, B* = ⎛ 0 *−*1 ⎞ *, A/a*

= ⎛ 2 0 ⎞ *.*

11 ⎝ 1 0 ⎠ 11 ⎝ 3 1 ⎠

But it is easy to know *B ∈ P*0 and *A/a*11 *∈ P*0, so we can conclude that *A* is a

*P*0-matrix.

# Numerical Examples

The following four matrix examples are tested by using the above *P*0-matrix algo- rithm 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. Exam- ple 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](#_bookmark1) where *ε* = 0*.*0001.

**Example 3.1** Upper triangular matrix *A* = (*aij*) *∈ Rn×n*, *aij* = *k*, if *i ≤ j*, otherwise *aij* = 0. Where, *k* is a random integer number between 0 *∼* 9. It is obvious that *A* is a *P*0-matrix.

### Example 3.2

⎛ *a a a ···* ⎞

*b b b ···*

⎜ ⎟

*A* =

⎜ *c c c ···* ⎟

⎝ .

⎠

.

.

where *a, b, c, . . .*, are random integer numbers between 0 *∼* 9. It is obvious that *A*

is a *P*0-matrix.

### Example 3.3

⎜ *−*1

⎛ 0

1 *··· · ··*

1 ⎞

⎜ . ⎟

⎜

⎜

*P*

⎝

*−*1

1

⎟

⎟⎟⎠

*A* =

where *P ∈ R*(*n−*1)*×*(*n−*1) is a *P* -matrix. It is easy to know *A* is not a *P*0-matrix.

### Example 3.4

*n × n*

⎛

⎜

0

*B*

*B*

0

*A* = ⎜

⎜

⎜⎜⎝

⎞

⎟

⎟

⎟

⎟⎟⎠

where *B ∈ R* 2 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 |

# Conclusions

This paper proposed an asymptotic approach for testing the *P*0-matrices by replac- ing 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 ap- proach 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.