

Title	Complexity of Graph and its J Program
Sub Title	
Author	竹中, 淑子(Takenaka, Yoshiko) 竹内, 寿一郎(Takeuchi, Juiciro) 松岡, 勝男(Matsuoka, Katsuo)
Publisher	慶應義塾大学日吉紀要刊行委員会
Publication year	2000
Jtitle	慶應義塾大学日吉紀要. 自然科学 No.28 (2000. 9) ,p.12- 19
JaLC DOI	
Abstract	We obtain a new approach to calculate the number of spanning trees for any simple undirected graphs on the basis of the singular value of the incidence matrix. By this result we obtain J program for computing the complexity of graphs.
Notes	
Genre	Departmental Bulletin Paper
URL	https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=AN10079809-20000930-0012

慶應義塾大学学術情報リポジトリ(KOARA)に掲載されているコンテンツの著作権は、それぞれの著作者、学会または出版社/発行者に帰属し、その権利は著作権法によって保護されています。引用にあたっては、著作権法を遵守してご利用ください。

The copyrights of content available on the KeiO Associated Repository of Academic resources (KOARA) belong to the respective authors, academic societies, or publishers/issuers, and these rights are protected by the Japanese Copyright Act. When quoting the content, please follow the Japanese copyright act.

Complexity of Graph and its J Program

Yoshiko TAKENAKA*
Juichiro TAKEUCHI**
Katsuo MATSUOKA***

Summary—We obtain a new approach to calculate the number of spanning trees for any simple undirected graphs on the basis of the singular value of the incidence matrix. By this result we obtain J program for computing the complexity of graphs.

Key words : J program , singular value , complexity of graph

1 INTRODUCTION

$G = (V, E)$ is a simple undirected graph with a vertex set $V = \{v_1, v_2, \dots, v_n\}$ and an edge set $E = \{e_1, e_2, \dots, e_m\}$. $K(G)$ is the complexity of graph G i.e. the number of spanning tree of G . By giving an arbitrary orientation to each edge of G , we define the

(0.1.-1) — incidence matrix $E = (e_{ij})$ of G in the following way

$$e_{ij} = \begin{cases} 1 & \text{if } v_i \text{ is the initial vertex of } e_j \\ -1 & \text{if } v_i \text{ is the terminal vertex of } e_j \\ 0 & \text{otherwise} \end{cases}$$

E_i is the matrix obtained from E by removing the i -th row. It is clear that

$$\text{rank } E = \text{rank } E_i = n - 1$$

for $i \in \{1, 2, \dots, n\}$.

*竹中淑子, 慶應義塾大学数学教室 (〒223-8521 横浜市港北区日吉4-1-1) : Dept. of Economics, Keio Univ., Hiyoshi, Kohoku-ku, Yokohama 223-8521, Japan, **竹内寿一郎, 慶應義塾大学工学部 : Dept. of Engineering Science, Keio Univ., Hiyoshi, Kohoku-ku, Yokohama 223-8521, Japan. e-mail: takeuchi@ae.keio.ac.jp, ***松岡勝男, 日本大学経済学部, Dept. of Economics, Nihon Univ., Tokyo, Japan. [Received Feb. 29, 2000]

Also we denote A_i the matrix obtained from the adjacent matrix A of G by removing the i -th row and i -th column. It means A_i is the adjacency matrix of $G - v_i$. By the same definition as A_i , D_i is the degree matrix $D = (\delta_{ij}d(v_i))$ of G , where $d(v_i)$ is the degree of vertex v_i . We put

$$L = D - A$$

and call L the combinatrial Laplacian of G . Matrix-tree Theorem states that

$$\kappa(G) = \det L_{ij}$$

where L_{ij} is the (i, j) cofactor of L and $i, j \in \{1, 2, \dots, n\}$.

Moreover between the $(0, 1, -1)$ -incidence matrix and the combinatrial Laplacian, there is the relation

$$E' E = L.$$

A, D is the adjacency matrix and the degree matrix of G respectively, then

$$E_i' E_i = L_{ii} = D_i - A_i$$

where L_{ii} is the (i, i) cofactor of L and $i \in \{1, 2, \dots, n\}$.

Consequently, $E' E$ and $E_i' E_i$ are independent of the orientation given to G and we have

$$\kappa(G) = \det E_i' E_i = \det(D_i - A_i)$$

2 SINGULAR VALUE

Now, we introduce the singular value $\text{Sing}(G)$ of graph G . Let E be a $(0, 1, -1)$ -incidence matrix of G , then $E_i' E_i$ is a regular and real symmetrix matrix. E_i is positive definite and $E_i' E_i$ is diagonalizable. We give the following definition in the same manner as the definition of the spectrum of a graph.

DEFINITION

E is the $(0,1,-1)$ -incidence matrix of G and $\lambda_1, \lambda_2, \dots, \lambda_s$ with $\lambda_1 > \lambda_2 > \dots > \lambda_s > 0$ are the eigenvalues of $E_i^t E_i$ and m_1, m_2, \dots, m_s are their multiplicities.

We call $u_1 = \sqrt{\lambda_1}, u_2 = \sqrt{\lambda_2}, \dots, u_s = \sqrt{\lambda_s}$ as the singular values of E_i and denote the singular values of G with respect to E_i by

$$Sing_i(G) = \begin{pmatrix} \mu_1 & \mu_2 & \dots & \mu_s \\ m_1 & m_2 & \dots & m_s \end{pmatrix}$$

In particular, if $Sing_i(G)$ is unique independently of i , we call it as the singular value of G and denote it by $Sing(G)$.

For a regular graph G and any vertex v of G , $G \cdot v$'s are isomorphic, therefore there exists $Sing(G)$. For the $(0,1,-1)$ -incidence matrix E of G , $E_i^t E_i$ is diagonalizable and

$$\det E_i^t E_i = \prod_{k=1}^s (u_k^2)^{m_k}$$

Then we obtain the following new method to calculate the complexity of a graph.

THEOREM Let G be a graph and let

$$Sing(G) = \begin{pmatrix} \mu_1 & \mu_2 & \dots & \mu_s \\ m_1 & m_2 & \dots & m_s \end{pmatrix}$$

then,

$$\kappa(G) = \prod_{k=1}^s (u_k^2)^{m_k}$$

3 EXAMPLES

By this theorem, we calculate the singular values of several special families of graphs and write down their complexities, which equal to the well known results.

1. For the complete graph with n vertices K_n ,

$$Sing(K_n) = \begin{pmatrix} \sqrt{n} & 1 \\ n-2 & 1 \end{pmatrix}$$

So we have

$$\kappa(K_n) = n^{n-2}$$

2. For the circuit graph of length n C_n ,

$$\text{Sing}(C_n) = \begin{pmatrix} \sqrt{2 - 2 \cos \frac{(n-1)\pi}{n}} & & & \\ & \sqrt{2 - 2 \cos \frac{(n-2)\pi}{n}} & & \\ & & \dots & \\ & & & \sqrt{2 - 2 \cos \frac{\pi}{n}} \end{pmatrix}.$$

So we have

$$\kappa(C_n) = \prod_{k=1}^{n-1} \left(2 - 2 \cos \frac{k\pi}{n} \right)$$

3. For the wheel graph with n vertices W_n ,

$$\text{Sing}(W_n) = \begin{cases} \begin{pmatrix} \sqrt{3 - 2 \cos \frac{2(l-1)\pi}{2l-1}} & & & \sqrt{3 - 2 \cos \frac{2\pi}{2l-1}} & 1 \\ & \sqrt{3 - 2 \cos \frac{2(l-2)\pi}{2l-1}} & & & \\ & & \dots & & \\ & & & \sqrt{3 - 2 \cos \frac{2\pi}{2l-1}} & \\ & & & & 1 \end{pmatrix} & \text{if } n = 2l, \\ \begin{pmatrix} 5 & \sqrt{3 - 2 \cos \frac{(l-1)\pi}{l}} & & & \\ 1 & & \sqrt{3 - 2 \cos \frac{(l-2)\pi}{l}} & & \\ & & & \dots & \\ & & & & \sqrt{3 - 2 \cos \frac{\pi}{l}} & 1 \\ & & & & & 1 \end{pmatrix} & \text{if } n = 2l + 1. \end{cases}$$

So we have

$$\kappa(W_n) = \begin{cases} \prod_{k=1}^{l-1} \left(3 - 2 \cos \frac{2k\pi}{2l-1} \right) & \text{if } n = 2l, \\ 5 \prod_{k=1}^{l-1} \left(3 - 2 \cos \frac{k\pi}{l} \right) & \text{if } n = 2l + 1. \end{cases}$$

4. For the complete bipartite graph $K_{r,s}$ ($r \geq s$)

$$\text{Sing}(K_{r,s}) = \begin{cases} \begin{pmatrix} 1 \\ s \end{pmatrix} & \text{if } r = 1, \\ \begin{pmatrix} \sqrt{\frac{r+s + \sqrt{(r+s)^2 - 4s}}{2}} & & & \sqrt{\frac{r+s - \sqrt{(r+s)^2 - 4s}}{2}} \\ & \sqrt{r} & & \\ & & \sqrt{s} & \\ & & & \sqrt{\frac{r+s - \sqrt{(r+s)^2 - 4s}}{2}} \end{pmatrix} & \text{if } r \geq 2. \end{cases}$$

So we have

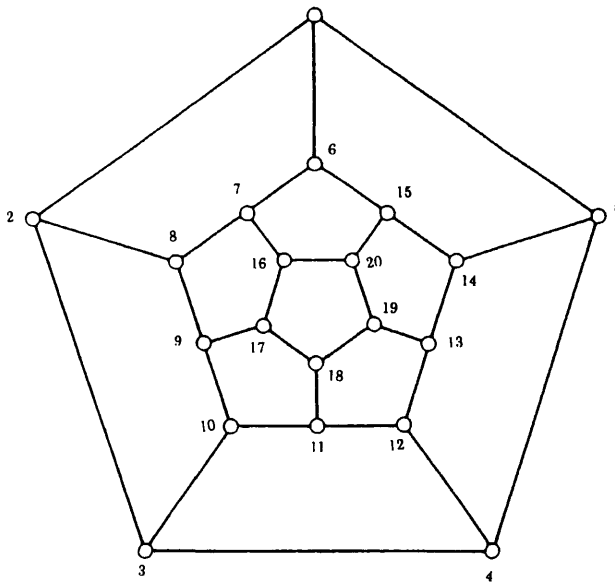
$$\kappa(K_{r,s}) = r^{s-1} s^{r-1}$$

4 COMPLEXITY of G(20,30)

By this theorem we obtain J program for computing the complexity of graphs.(cf.Appendix)

[Examples]

$$G = (V, E), |V| = 20, |E| = 30$$



We label the number 1,2,.....,20 to all vertices of graph G and indicate edges by two vertices number. We input the number series corresponding to all edges of G , no particular order observed. The important point of this program is that input is a simple number series and not a matrix. Then we obtain $Sing(G)$ and $\chi(G)$ of graph $G(20,30)$ as output.

```
X = . 1 2 2 3 3 4 4 5 1 5 1 6 2 8 3 10 4 12 5 14
      6 7 7 8 8 9 9 10 10 11 11 12 12 13 13 14 14 15 6 15 7 16
      9 17 11 18 13 19 15 20 16 20 16 17 17 18 18 19 19 20
 $\chi(G) = 5,184,000$ 
```


Appendix : J Program for Computing Complexity

J is a general-purpose programming language available as shareware on a wide variety of computers(<http://www.jsoftware.com>). The characteristics of J is that we can handle vectors, matrices, and other arrays as single entities like a pocket calculator. J has many kinds of convenient functions for instance, matrix operations including transpose or inverse matrix, multiple regression, outer products, operations with complex numbers, Taylor expansions, approximations by polynomials, decomposition into prime numbers and GCD and LCM. We can easily get J freeware from the following address <ftp://take.soc.ae.keio.ac.jp/take/j/exper/fd1/jfw.exe>.

Subroutines : (NB. Denotes Comments)

NB. Write to Screen

```
print=. (1!:2)&2
```

NB. Matrix Product or Inner Product

```
mp=. +/ .*
```

NB. Select eigen vectors corresponding to nonzero eigen values

```
nselect=. 1e_5 <@ (0&{)@{.#"1 }.
```

NB. Convert a vector of locations to a matrix form

```
mat=. 3 : '1 (<"1 (<.y),. (n#2)#i.n) (>./y), n.-.:#y.$0'
```

NB. Replace the first "1" in each column by "_1"

```
minus=. 3 : '_1 (<"1 t, i.#t=.( |:y.) i."1(1))y.'
```

NB. Jacobi's method for eigenvalues and vectors

NB. form: [tolerance] jacobi mat

NB. default tolerance 1e_5

NB. returns: eigenvalues;eigenvectors

```
jacobi=. 3 : 0
```

```
1e_5 jacobi y.
```

```
:
```

```
r=. y.
```

```
ir=. i.#r
```

```
q=. id=. =ir
```

```
diag=. (<0 1)&|:
```

```
len=. +/&.( *:"_)
```

```
imax=. i.>./
```

```
utm=. ,</~ir
```

```
mp=. +/ .*
```

```
ndx=. ,{:~ir
```

```
perm=. 0 0&{ ; ] ; |. ; 1 1&{
```

```
sign=. _1: ^ <&0
```

```
count=. 0
```

```
while.
```

```
big=. imax utm * |,r
```

```
ind=. big { ndx
```

```
p=. -ind { r
```

```
(x. < |p) *. big > 0
```



```

do.
  u=. -: /(>ind) {diag r
  v=. len p,u
  cos=. %:(v+|u) % +:v
  sin=. (sign u) * p % +: v * cos
  s=. ((cos,-sin),sin,cos) (perm >ind) } id
  r=. s mp r mp |:s
  q=. q mp |:s
  count=. >:countW
end.
r=. diag r
inx=. ⍶:r
(inx{r);inx{"1 q
)

```

Main Program for Calculating Complexity of a Graph

```

Complex=.3 : 0
a=}.minus print Mtrx=:mat y.
b=.a mp (|:a)
c=:>jacobi b
n.{$print Vec=:nset c
print*/print Eig=:n{.c
)

```

An Example :

NB. data

x=.1 2 2 3 3 4 1 4 1 3

NB. Execution form

Complex x

NB. Graph matrix

1 0 0 1 1

1 1 0 0 0

0 1 1 0 1

0 0 1 1 0

NB. Eigen vectors

_0.4082494 _0.7071062 0.5773503

0.8164966 _1.24831e_6 0.5773503

_0.4082472 0.7071074 0.5773503

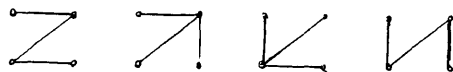
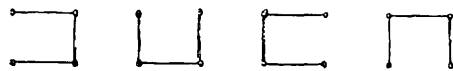
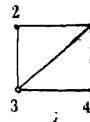
NB. Eigen values

4 2 1

NB. Complexity

8

(mailing address : takeuchi@ae.keio.ac.jp)



REFERENCES

- [1] N.Biggs, Algebraic Graph Theory, Cambridge University Press, London, 1974
- [2] D.M.Cvetković, M.Doob and H.Sachs, Spectra of Graph, Academic Press, New York. 1979
- [3] D.M.Cvetković, Graphs and their spectra, Univ.Beograd Publ, No.354-No.356,1971
- [4] Y.Takenaka and H.Handa, On the spectrum of a graph, Keio Engineering Report, Vol.31, No.9, 1978
- [5] 鈴木義一郎、北野利雄「J言語による数学計算～Windows版～」、森北出版 1996
- [6] 鈴木義一郎、「J言語による統計分析～Windows版～」、森北出版 1996
- [7] <http://www.jsoftware.com/>.
- [8] <http://www.soc.ae.keio.ac.jp/takeuchi/J.htm>