# Isomorphic Structure of Graphs with the Maximum Number of Trees

Hideshi Ido*, Yosuke Ohama*1, and Sigeru Omatu**
* Electrical Engineering and Computer Science, *1 Advanced Engineering Course,
Niihama National College of Technology,
7-1 Yagumo, Niihama, Ehime 792-8580, Japan
ido@ele.niihama-nct.ac.jp
** Computer and Systems Sciences, Graduate School of Engineering,
Osaka Prefecture University, Sakai, Osaka 599-8531, Japan
omatu@sig.cs.osakafu-u.ac.jp

**Abstract**

Two graphs $G_A$ and $G_B$ with the same number of nodes *n* and the same number of branches *b* are said to be isomorphic if these graphs have the same adjacency relations between nodes. We have got many graphs with the same maximum number of trees in the process of continuous branch additions. These graphs have same numbers of nodes *n*, branches *b* and trees *t*. We certified those graphs are isomorphic by a computer algorithm. This algorithm is based on permutations of nodes with the same degree in a sequence of node degrees. Examples of isomorphic graphs are presented in the process of adding branches one by one. If two graphs with same numbers of nodes *n* and branches *b* are isomorphic, these graphs have same numbers of trees. The further problem will be to prove the reverse condition.

*Keywords*: Isomorphic graph, Maximum number of trees, Continuous branch addition

## 1 Introduction

The number of trees of a graph would be an index to estimate the reliability of probabilistic communication networks with either high or low link probabilities when those probabilities are independent and equal [1].

Some analytical methods have been reported to calculate the number of trees of graphs obtained from the complete graph by deleting several types of subgraphs [2],[3]. Other ones are to maximize the number of trees when some branchs or subgraphs are deleted from the complete graph [4],[5]. Another one is to get the expressions of trees of a graph which is obtained by connecting graphs of special structures [6].

These analytical methods are based on obtaining the expressions derived from a node determinant of the target graphs with some special structures. But simple expressions haven't been known for graphs without any structural conditions so far.

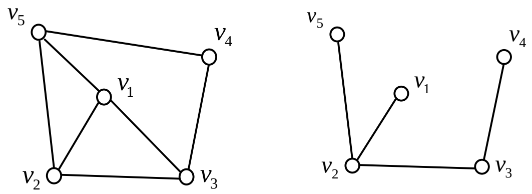
On the other hand, using numerical method, we constructed graphs with the maximum number of trees by adding a new branch continuously to the original graph[ 7].

In this paper, we have noticed that two or more graphs with the same maximum number of trees appeared many times in the process of branch additions. All of those graphs are made sure to be isomorphic for some examples which are started branch additions from a tree of series branches with nodes *n*=7 and a star-shaped tree of *n*=8.

In the following sections, we will show two algorithms for graph construction and isomorphism. Results of estimation by connectivity are also presented to examine strength of these graphs.

## 2 Preliminary results [7]

First, we introduce some graph theoretical terms. A graph denoted by $G=(V,E)$ is composed with a node set *V* and a branch set *E*. Each branch denoted by $(i,j)$ connects a pair of nodes $i$ and $j$. A graph is said to be simple if it has no parallel branches, no self-loops and no directed branches. In this paper, we treat only simple graphs as shown in Fig.1 (a). A tree is a subgraph which connects all nodes of a graph $G$ and includes no closed circuits, as shown in Fig.1 (b).




(a) (b)
Fig.1. (a) Simple graph, (b) Tree

The complete graph of *n* nodes has one branch $(i,j)$ between every node pair $i$, $j$, and includes $\frac{1}{2}n(n-1)$ branches. The matrix $A=(a_{ij})$ can be constructed from a graph *G* with *n* nodes, where for $1 \le i, j \le n-1$, each diagonal element $a_{ii}$ is the number of branches incident to node $i$, and each non-diagonal element $a_{ij}$ is –1 if there is a branch $(i,j)$ between two nodes $i$ and $j$, otherwise 0. Then the number of trees $T$ of the graph *G* is represented by the determinant $|A|$. Since the value of $T$ is always an integer for any graph, the determinant $T$ must be calculated by using double precision floating point number for matrix $A$ . Usual Gaussian elimination is available for this purpose.

*Algorithm*: *Max-trees*

*Step1*. Find the set $\overline{P}$ of position $(i, j)$ where there isn't an branch $(i, j)$ in the given original graph $G(V, E)$ . Now, set $G_0 \leftarrow G(V, E)$ and $k \leftarrow 0$.

*Step2*. Calculate the number of trees $T^*$ of each graph obtained by adding a branch $(i, j)$ to the graph $G_k$ , for each position $(i, j) \subset \overline{P}$ , respectively.

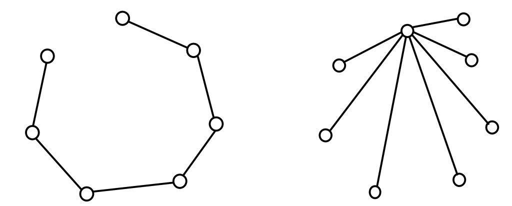
*Step3*. Find the branch $(i, j)$ that gives the maximum number of trees among $T^*$ 's in *Step2*. Replace $G_{k+1} \leftarrow G_k +$ branch $(i, j)$ , $\overline{P} \leftarrow \overline{P}$ - position $(i, j)$ , and $k \leftarrow k+1$.

*Step4*. If the set $\overline{P}$ is empty, then terminate. Otherwise go to *Step2*.

## 3 Isomorphic structure of graphs

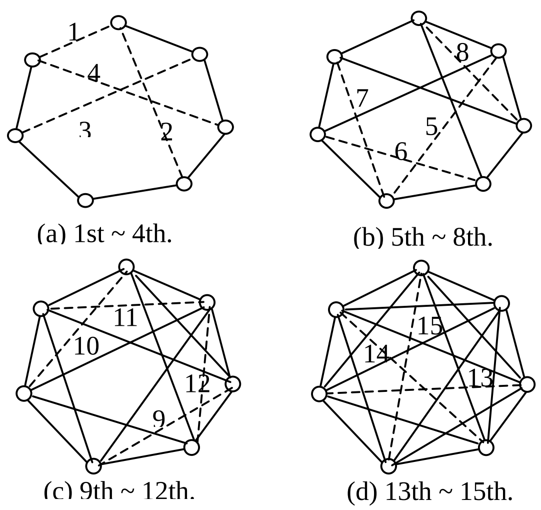
### 3.1 Isomorphic graphs

When the algorithm starts from a series-branch tree graph of Fig.2(a), we obtain the sequence of branches represented by numbers near dotted lines as shown in Figs.3.



(a) Series-branch tree. (b) Star-shaped tree.

Fig.2. Initial graphs.




(a) 1st ~ 4th. (b) 5th ~ 8th.

(c) 9th ~ 12th. (d) 13th ~ 15th.

Fig.3. Branch sequence for Series-branch tree of $n$=7.

In the process to obtain this branch sequence, there are many isomorphic graphs with the same maximum number of trees as shown in Table 1.

Table 1. Isomorphic graphs with the same maximum number of trees of Fig.3.

| Added branch | Total branches b | Max. number of trees | Isomorphic graphs* |
|---|---|---|---|
| 0 | 6 | 1 | 1 |
| 1 | 7 | 7 | 1 |
| 2 | 8 | 19 | 7 |
| 3 | 9 | 51 | 2 |
| 4 | 10 | 117 | 1 |
| 5 | 11 | 231 | 4 |
| 6 | 12 | 408 | 2 |
| 7 | 13 | 720 | 1 |
| 8 | 14 | 1200 | 1 |
| 9 | 15 | 1840 | 4 |
| 10 | 16 | 2800 | 1 |
| 11 | 17 | 4200 | 3 |
| 12 | 18 | 6125 | 2 |
| 13 | 19 | 8575 | 3 |
| 14 | 20 | 12005 | 2 |
| 15 | 21 | 16807 | 1 |

* Number of isomorphic graphs including itself

For example, on added branch 5 of the Table 1,there are 4 graphs with 11 branches and 231 trees as shown in Fig.4. In this figure, each of 4 branches represented by dotted lines A,B,C, and D is the candidate to be added next.

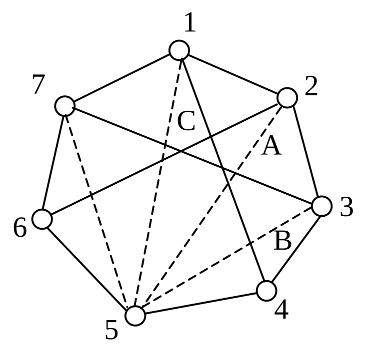



Fig.4. Four candidate branches

According to the criterion to select one branch that gives the maximum number of trees, any one among 4 branches can be selected. But its selection may cause different results in the later process of branch additions. Now, we denote graphs including branch A, including branch B, including branch C, and including branch D as $G_A$, $G_B$, $G_C$, and $G_D$, respectively in Fig.4.

$G_A$ and $G_C$ are isomorphic because of symmetric structure. Also $G_B$ and $G_D$ are isomorphic. To investigate isomorphism between $G_A$ and $G_B$, we have constructed the next algorithm.

### 3.2 Algorithm for Isomorphic Structure

A node degree d($v$) of a node $v$ of a graph G is a number of branches incident to node $v$ in G. A degree sequence S of a graph G is a sequence of node degrees of all nodes of a graph G. We assume that S is arranged in a decreasing order and separated into subsequences $S_1$, $S_2$,・・・,$S_m$, where $S_i$ includes the same degrees. In general, two graphs $G_A$ and $G_B$ must have the same degree sequences $S_A$ and $S_B$ to be isomorphic. Then we assume that degree sequences $S_A$ and $S_B$ corresponding to graphs $G_A$ and $G_B$ are composed with the same subsequences $S_A = S_B = (S_1, S_2, \cdots, S_m)$. The number of nodes for subsequence $S_k$ is denoted by $|S_k|$.

Outline of the algorithm to determine isomorphism between $G_A$ and $G_B$ is as follows.

*Algorithm Isomorphic*

*Step1*. Set k=1(Subsequence k is noticed).

*Step2*. Generate next permutation of $|S_k|$ nodes corresponding to $S_k$ of $S_B$.
If permutation exhausted, $G_A$ is not isomorphic to $G_B$ (terminate the algorithm).

*Step3*. Examine adjacency relations of graphs $G_A$ and permuted $G_B$ within the range of $S_1 \sim S_k$.
If this examination has succeeded and k<m, then set k $\leftarrow$ k+1 and go to *Step2*.
If this examination has succeeded and k=m, then $G_A$ is isomorphic to $G_B$ (terminate the algorithm).

*Step4*. Otherwise, set k $\leftarrow$ k-1 and return to *Step2*.

For programming of this algorithm, the generation of permutation in *Step2* is skipped for subsequence $S_k$ with $|S_k|=1$.

Using this algorithm, all graphs with the maximum number of trees in Table 1 have been found out to be isomorphic. Of course, cases of symmetric structures are omitted. For another initial graph of star-shaped tree shown in Fig.2(b) with nodes $n$=8, the number of isomorphic graphs are shown in Table 2.

Isomorphic graphs have appeared many times in the process of continuous branch additions. The rate of a number of cases that give two or more isomorphic graphs is 9/14=64% in Table 1, and 15/20=75% in Table 2.

### 3.2 Estimation of Graphs with the Maximum Number of trees by Connectivity

The number of trees of a graph G has the relation with the reliability of communication networks with the same branch probability $p$.
As another criterion to estimate communication networks, we introduce the connectivity of corresponding graph.

Table 2. Isomorphic graphs with the same maximum number of trees started from Fig.2(b).

| Added branch | Total branches b | Max. number of trees | Isomorphic graphs* |
|---|---|---|---|
| 0 | 7 | 1 | 1 |
| 1 | 8 | 3 | 21 |
| 2 | 9 | 9 | 10 |
| 3 | 10 | 27 | 3 |
| 4 | 11 | 72 | 6 |
| 5 | 12 | 168 | 4 |
| 6 | 13 | 377 | 4 |
| 7 | 14 | 841 | 1 |
| 8 | 15 | 1537 | 7 |
| 9 | 16 | 2800 | 2 |
| 10 | 17 | 4928 | 1 |
| 11 | 18 | 8056 | 4 |
| 12 | 19 | 12440 | 2 |
| 13 | 20 | 19200 | 1 |
| 14 | 21 | 28800 | 1 |
| 15 | 22 | 40800 | 4 |
| 16 | 23 | 57600 | 1 |
| 17 | 24 | 80640 | 3 |
| 18 | 25 | 110592 | 2 |
| 19 | 26 | 147456 | 3 |
| 20 | 27 | 196608 | 2 |
| 21 | 28 | 262144 | 1 |

* Number of isomorphic graphs including itself

A local connectivity k($i$,$j$) between distinct two nodes $i$ and $j$ of a graph G=(V,E) is the minimum number of nodes removed to separate G. Then the connectivity of

Table 3. Connectivity k(G) of the graph G appeared in Table 1.

| Branches b | Ideal k | Real k(G) |
|---|---|---|
| 6 | 1 | 1 |
| 7 | 2 | 2 |
| 8 | 2 | 2 |
| 9 | 2 | 2 |
| 10 | 2 | 2 |
| 11 | 3 | 3 |
| 12 | 3 | 3 |
| 13 | 3 | 3 |
| 14 | 4 | 3* |
| 15 | 4 | 4 |
| 16 | 4 | 4 |
| 17 | 4 | 4 |
| 18 | 5 | 5 |
| 19 | 5 | 5 |
| 20 | 5 | 5 |
| 21 | 6 | 6 |

Table 4. Connectivity k(G) of graphs started from series branch tree of nodes $n$=8.

| Branches b | Ideal k | Real k(G) |
|---|---|---|
| 7 | 1 | 1 |
| 8 | 2 | 2 |
| 9 | 2 | 2 |
| 10 | 2 | 2 |
| 11 | 2 | 2 |
| 12 | 3 | 3 |
| 13 | 3 | 3 |
| 14 | 3 | 3 |
| 15 | 3 | 3 |
| 16 | 4 | 4 |
| 17 | 4 | 4 |
| 18 | 4 | 4 |
| 19 | 4 | 4 |
| 20 | 5 | 4* |
| 21 | 5 | 4* |
| 22 | 5 | 5 |
| 23 | 5 | 5 |
| 24 | 6 | 6 |
| 25 | 6 | 6 |
| 26 | 6 | 6 |
| 27 | 6 | 6 |
| 28 | 7 | 7 |

graph G denoted by k(G) is the minimum value of local connectivity k($i$,$j$). Connectivity k(G) represents the strength of the network against damages of nodes and branches.

Table 3 shows a comparison between ideal connectivity k and real k(G) for the graphs in Fig. 2. Ideal connectivity k is given by [2$b$/$n$], where [ ] denotes Gaussian notation[1]. Table 4 shows for graphs started from series branch trees with $n$=8.

From Tables 3 and 4, graphs with the maximum number of trees constructed by algorithm *Max-trees* give good results in most cases except for cases marked with *.

Graphs shown in Tables 3 and 4 started from series branch trees tend to balance node degrees into two consecutive integers.

## 5. Conclusion

We demonstrated that graphs with maximum number of trees constructed by simple algorithm *Max-trees* have isomorphic structures in. many cases as more than 60%. Graphs with the maximum number of trees are needed to be the models of reliable networks.

Connectivity k(G) of a graph G is adapted as an another criterion to estimate the strength of networks against damages of nodes and branches. Except for few cases graphs with maximum number of trees give good performance.

We used computer algorithm to examine isomorphism for graphs with the same number of nodes n, branches b, and trees. If the same numbers of trees means isomorphic of graphs, it is very easy to examine isomorphism. This is the future problem.

## References


[1] Leggett J.D.,Bedrosian S.D., “Synthesis of Reliable Networks”,IEEE Trans. Circuit Theory, Vol.CT-16, No.8, pp.384-385, 1969.

[2] Kasai T.,Kusaka H.,Yoneda S. and Taki I., “Total Number of Trees in Four Kinds of Incomplete Graphs Derived from a Complete Graph (in Japanese) ”,Jour. Inst. Elect. Comm. Engrs. Japan.Vol.49, No.1,pp.69-77, 1966.

[3] Ishikawa H., Ido H.,and Sasaki T., “Number of Trees in a Certain Incomplete Graph (in Japanese)”, Trans. IEICE, Vol.57-A,No.5, pp.423-424, 1974.

[4] Shier D.R., “Maximizing the Number of Spanning Trees in a Graph With n Nodes and m Branchs”,Jour. Res. of Nat. Bur. of Stds.-B. Math. Sci.,Vol.78B, No.4, pp.193-196, 1974.

[5] Ido H., Ishikawa H., and Sasaki T., “Number of Trees in a Specified Incomplete Graph and its Min-Max Problem (in Japanese)”, Trans. IEICE, Vol.58-A,No.1, pp.81-82, 1975.

[6]Hatakeyama M., Sato K., “Construction Metod for a Connection Graph of Cycles from a General Graph (in Japanese)”, Proc. IEICE Gen Conf. A-1-6, 2003.

[7] Ido H., Ohama Y., and Takahashi T.., “Graph Construction with the Maximum Number of Trees by Continuous Edge Addition”, Artificial Life and Robotics, Vol.10, No.2, pp.185-188, 2006.