TECHNICAL RESEARCH REPORT

Institute for Systems Research

An Efficient Algorithm for the Automatic Enumeration of the Topological Structures of Mechanisms

by H-I. Hsieh and L.W. Tsai

The Institute for Systems Research is supported by the National Science Foundation Engineering Research Center Program (NSFD CD 8803012), Industry and the University

TR 93-39

An Efficient Algorithm for the Automatic Enumeration of the Topological Structures of Mechanisms

Hsin-I Hsieh, Gradute Research Assistant and
L. W. Tsai, Professor
Department of Mechanical Enigineering and
Institute for Systems Rearch
University of Maryland
College Park, Maryland

March 23, 1993

Abstract

In this paper, a new algorithm for the automatic enumeration of the topological structures of mechanisms is developed. The algorithm permits the deviation of conventional graphs from contracted graphs by the partition of binary vertices. A row vector is formed to present the binary-vertex chains. The partition of binary vertices is transformed into the solution of one linear equation in several unknown integers. No permutation of binary-vertex chains is necessary. This leads to the development of a highly efficient computer algorithm for the enumeration of conventional graphs.

Two tables of conventional graphs with seven and eight vertices, and with up to five loops have been developed. We believe that the results of these conventional graphs are new. Mechanisms of higher pair joints up to five loops and with eight links can now be synthesized from these tables.

1 Introduction

Various techniques for the enumeration and classification of the kinematic structures of mechanisms can be found in the literature. (Crossley, 1964a , 1964b and 1965; Buchsbaum and Freudenstein, 1970; Freudenstein and Maki, 1979; Mruthyunjaya, 1979; Yan, 1980; Tuttle and Peterson, 1987; Tuttle et al, 1989a and 1989b; Hwang et al, 1991). Some of the methods employed for the enumeration are based on graph theory (Freudenstein, 1967; Woo, 1967; Huang and Soni, 1973; Tsai, 1987; Alhakim and Shrivastava, 1991). The advantages of using graphs are: (1) the network propertities of graphs are directly applied to automate the kinematic and dynamic analysis of mechanisms, and (2) a single atlas of graphs can be used to enumerate a large class of mechanisms.

Woo (1967) applied contracted graphs for the enumeration of conventional graphs with ten vertices and thirteen edges. Mruthyunjaya (1984a, 1984b, and 1984c) developed a computer program, which is based on the transformation of binary chains, for the structural synthesis of kinematic chains. Sohn and Freudenstein (1986) introduced the concept of dual graphs for the enumeration of the kinematic structures of mechanisms. Planar linkage with up to three degrees of freedom and five loops were developed. Mayourian and Freudenstein (1984) developed an atlas of graphs with up to six vertices for the kinematic structures of arbitrary joint types. An atlas of contracted graphs was recently developed by Hsu (1989).

In this paper, a new algorithm for the derivation of conventional graphs of general nature. All permissible conventional graphs with seven and eight vertices and with up to five loops are developed and tabulated.

2 Definitions of Graphs

2.1 Conventional Graph

A graph is a collection of vertices and edges. In the conventional graph representation of a mechanism, vertices represent links, edges represent joints, and the edge-connection between vertices corresponds to the pair connection between links. The relationship between the number of loops (including the peripheral loop), L, the number of vertices, v, and the number of edges, e, is given by Euler's theorem (Harary, 1969):

$$L = e - v + 2 \tag{1}$$

The *degree* of a vertex in a graph is defined as the number of edges incident with that vertex (Harary, 1969). Let v_i be the number of vertices with degree i. Then v_i is related to v and e as (Woo, 1967):

$$\sum_{i=2}^{\beta} v_i = v \tag{2}$$

$$\sum_{i=2}^{\beta} i \; v_i = 2e \tag{3}$$

where β denotes the upper bound on the degree of a vertex. The value of β can be derived from the realization that the number of loops of which the vertex is a part is equal to its degree, i.e. $\beta \leq L$.

The lower and upper bounds on the number of edges in a loop is given by Mayourian and Freudenstein (1984),

$$3 \leq e \leq 2v - 3 \tag{4}$$

For example, Fig. 1 is a mechanism with ten links and thirteen joints. Fig. 2 shows the graph representation of the mechanism. This conventional graph has ten vertices, thirteen edges and five loops. In this paper, we shall concern ourselves with only those graphs which are planar with no articulation points or cut edges.

2.2 Contracted Graph

A *binary-vertex chain* in a conventional graph is defined as the maximum possible sequence of alternating edges and vertices of degree two, starting and ending with an edge. A *contracted graph* is a graph derived from a conventional graph by replacing each of the binary-vertex chains with a single edge. By contraction, parallel edges may exist in a contracted graph.

From the above definition, we conclude that the numbers of vertices, edges and loops, v^c, e^c, and L^c, in a contracted graph are related to that of the corresponding conventional graph by the following equations.

$$v_2 = \sum_{i=1}^{m} i \; b_i \tag{5}$$

$$v^c = v - v_2 \tag{6}$$

$$e^c = e - v_2 \tag{7}$$

$$L^c = e^c - v^c + 2 = L \tag{8}$$

where b_i denotes the number of binary-vertex chains with "i" vertices and "$i+1$" edges, and m denotes the upper bound on the number of vertices in a binary-vertex chain. According to the study of Sohn and Freudenstein (1986),

$$m \leq F + \lambda - 1 \tag{9}$$

where F denotes the degrees of freedom and λ denotes the motion parameter. For planar mechanisms, $\lambda = 3$; and for spatial mechanisms, $\lambda = 6$.

The contracted graph of Fig. 2 has four vertices and seven edges as shown in Fig. 3. An atlas of contracted graphs was developed by Hsu (1989) recently. This atlas of contracted graphs will be used for the enumeration of conventional graphs to be described in the

following section.

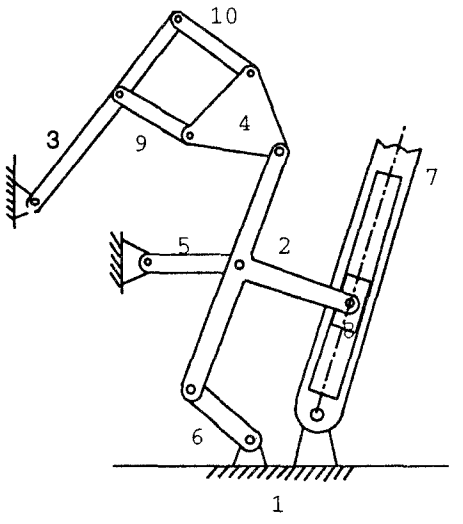


Figure 1: A mechanism with ten links and thirteen joints.

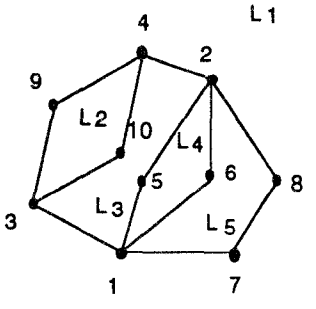


Figure 2: Conventional graph representation of Fig. 1

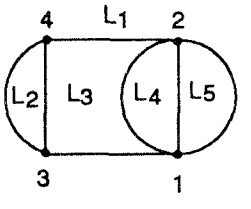


Figure 3: Contracted graph of Fig. 2

3 Enumeration of Conventional Graphs from Contracted Graphs

Conventional graphs can be derived from contracted graphs by using a technique called "expansion" first introduced by Woo (1967). Given v and e for the desired conventional graphs, Woo's approach starts with solving Eqs. (2) and (3) for all the possible solutions of the degree listings v_i, $i = 2, 3, \ldots, m$. For each of the degree listings, the corresponding contracted graphs are identified. Then, the binary vertices are partitioned into all possible partitions satisfying the kinematic requirement on partial subgraphs. Finally, the permutation group for each contracted graph is found, and conventional graphs are enumerated for each partition.

This procedure is systematic. However, it is not easy to be implemented by a computer program. In particular, the algorithm for finding the permutation group for a contracted graph can be very complicated. In what follows, we present a more straightforward procedure which can be easily implemented on a computer.

3.1 Procedure

The procedure permits the insertion of binary vertices on the edges of contracted graphs in a systematic manner. The edges of a contracted graph are labeled and a row vector is used to represent the edges. A $(2 \times v^c)$ matrix is used to store the end vertices of the edges. Then the problem is transferred into solving one linear equation in several unknown integers. This makes efficient computation possible. Using this procedure, graph isomorphism due to symmetry in parallel binary-vertex chains can be avoided before they are generated.

Given v and e for the desired conventional graphs, the enumeration proceeds in the following sequence:

Step 1. Identification of the admissible contracted graphs.

Step 2. Expansion of the edges into binary-vertex chains.

Step 3. Identification of the admissible graphs and graph isomorphism.

Step 1:

The first step is the identification of admissible contracted graphs. Euler's formula is used to determine the number of loops required of a contracted graph. That is

$$L = e - v + 2 \tag{10}$$

For example, L=5 for (10,13) graphs. Therefore, only those contracted graphs with five loops are permissible for the enumeration of (10,13) graphs. The atlas of contracted graphs developed by Hsu(1989) is used as the data base for the enumeration.

Step 2:

The second step is the insertion of binary vertices onto the edges of a contracted graph. The number of binary vertices is obtained by solving Eq. (6), i.e. $v_2 = v - v^c$. Edges in a contracted graph are labeled and arranged in a row vector as shown below:

$$K = (k_1, k_2, \ldots, k_p) \tag{11}$$

where p=e^c is equal to the number of edges and the subscript of k denotes the edge label in a contracted graph. The value of k_i represents the number of binary vertices to be inserted onto the ith edge of the contracted graph. Hence k_i, $i = 1, 2, \ldots, m$, intially take the values of zeros, and the end vertices of all edges in a contracted graph are arranged in a $(2 \times e^c)$ matrix, called the "E" matrix, to establish a data base for the enumeration of conventional graphs. Each row in the E matrix represents an edge in a contracted graph. The two elements in a row represent the node numbers associated with the end vertices of an edge. The row number corresponds to the edge label. For example, Fig. 4 shows a contracted graph with the edges labeled from 1 to 5 enclosed in small circles. The E matrix for the contracted graph shown in Fig. 4 is given by

$$E = \begin{bmatrix} 1 & 2 \\ 1 & 2 \\ 1 & 2 \\ 1 & 3 \\ 1 & 3 \\ 2 & 3 \end{bmatrix} \begin{array}{l} \\ \\ \\ \\ \\ \end{array} \begin{array}{l} Group\ 1 \\ \\ \\ Group\ 2 \\ \\ Group\ 3 \end{array} \tag{12}$$

For convenience, the end vertices in each row of the E matrix are arranged in an ascending order. Parallel edges are labeled in sequence such that they form a group, and the various groups of parallel edges are arranged in a descending order according to the number of edges in its group.

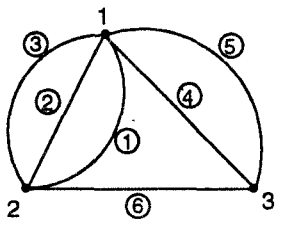


Figure 4: A (3,6) contracted graph

For example, the edges of Fig. 4 are arranged into three groups. Group 1 consists of three parallel edges connecting vertices 1 and 2; group 2 consists of two parallel edges connecting vertices 1 and 3; and group 3 is a single edge as shown in Eq. (12).

During the process of expansion, some of the values of k_i's are replaced by positive integers. The value of k_i represents the number of vertices in a binary-vertex chain. Thus, if k_i=0, the ith edge in a contracted graph is not altered; if k_i=1, the ith edge in a contracted graph is replaced by a binary-vertex chain with one vertex and two incident edges, etc. Hence, the summation of k_i is equal to the number of binary vertices, i.e.

$$k_1 + k_2 + \ldots + k_p = v_2 \tag{13}$$

where $k_{max} \geq k_i \geq 0$, i=1,2,...,p, are all integers, and k_{max} is the largest number of vertices in a binary-vertex chain.

We note that each group of n parallel edges in a contracted graph requires the insersion of at least $n-1$ vertices, one on each of the $n-1$ out of n parallel edges, to prevent loops formed by two edges only. Let r denote the least number of binary vertices to be inserted onto the parallel edges of a contracted graph to avoid any two-edged loop. Then, r is equal to the total number of parallel edges minus the number of parallel-edge groups.

Once the r vertices are inserted, the remaining vertices can be inserted arbitrarily. A longest binary-vertex chain is obtained when all the remaining vertices are inserted onto one of the parallel edges which already contains a binary vertex. Therefore, the upper bound on k_i is

$$k_{max} = v_2 - r + 1 \tag{14}$$

The expansion of a contracted graph is equivalent to solving Eq. (13) for all possible positive integer solutions (including zero). Fortunately, this can be easily accomplished by either the solution method outlined in Appendix A or by a computer program to vary the values of k_i from 0 to k_{max}.

If $k_s, k_{s+1}, \ldots, k_{s+r}$ belong to a group of parallel edges, then the constraints, $k_s \leq k_{s+1} \leq \ldots \leq k_{s+r}$, are imposed on k_i. This reduces greatly the chance of producing isomorphic graphs and the number of "*do-loops*" in the computer program. A nested *do-loop* for the enumeration of conventional graphs from contracted graphs can be written as shown in Fig. 5.

```
Do  i1   =  0, km
    K(1) =  i1;
                                                    Group 1
Do  i2   =  i1, km
    K(2) =  i2;

Do  i3   =  i2, km
    K(3) =  i3;
-------------------------------------------------------------
        Do  i4   =  0,   km
            K(4) =  i4;                             Group 2
        Do  i5   =  i4,  km
            K(5) =  i5;
-------------------------------------------------------------
                Do  i6   =  0,      km              Group 3
                    K(6) =  i6;

                . . . . . . . . . .                . . . . . . . . . .

                                                    Group n
                Do  ip   =  ip-1,   km
                    K(p) =  v-v^c-K(1)-K(2)-K(3)-K(4)-,...,-K(p-1)
                If  K(p) < 0 or K(p) > km  ,  try next K(p-1)
                                              discard the solution
                else save K(i), i=1,...,p
                Continue
                . . . . . . . . . .

                Continue
        Continue
        Continue
Continue
Continue
Continue
```

Figure 5: A computer program for solving k_i

Step 3:

In Step 2 the partition of binary vertices onto the edges of a contracted graph is accomplished, which results in a family of conventional graphs. However, detection of graph isomorphism is still necessary. In this paper, the random number technique of the characteristic polynomial is used to check graph isomorphism (Tsai, 1987).

The characteristic polynomial is only a necessary, but not sufficient condition for checking graph isomorphism. It has been shown that when the number of vertices becomes sufficiently large, two non-isomorphic graphs may possess identical characteristic polynomials (Sohn, and Freudenstein, 1986). However, the probability for two non-isomorphic graphs that are derived from the same contracted graph and, at the same time, possess identical characteristic polynomials is negligibly small. Hence, the problem can be avoided by classifying conventional graphs into different contracted graph families before the values of characteristic polynomials are compared. Only those graphs that are derived from the same contracted graph are checked against one another for graph isomorphism. Using the method developed by Tsai (1987), we have not found any two non-isomorphic graphs that are derived from the same contracted graph, possess the same characteristic polynomial.

In order to check for graph isomorphism, the vertex-to-vertex (V-V) incident matrix of a conventional graph needs to be constructed. In what follows, we describe a method of generating the V-V incident matrix from the E matrix.

Once the values of k_i, i=1,2,...,p, are found, the V-V incident matrix of the corresponding conventional graph can be constructed. Recall that the pair of end vertices of an edge in a contracted graph is contained in a row of the E matrix. The insertion of k_i binary vertices onto each edge of a contracted graph will increase each of the number of edges and the number of vertices by k_i. The number of rows in the expended E matrix will also increase by k_i. In the following procedure, the vertex numbers in the original E matrix will not be altered, and the newly added vertices will be numbered sequentially following the last vertex number of the contracted graph. The insertion of k_i binary vertices starts from the first edge

of the contracted graph. The procedure is outlined as follows:

We start with the first edge. If the first element is $k_1 = 0$ and the first row of the E matrix is (a, b), then let

$$A_{a,b} = 1.$$

If the first element $k_1 = x$, then x vertices are inserted onto the first edge as follows:

$$A_{a,v^c+1} = 1,\ A_{v^c+1,v^c+2} = 1,\ \ldots\ A_{v^c+x-1,v^c+x} = 1,\ and\ A_{v^c+x,b} = 1.$$

where v^c denotes the number of vertices in the contracted graph.

If the second element is k_2=0 and the second row of E matrix is (c, d), then let

$$A_{c,d} = 1.$$

If the second element $k_2 = y$, then y vertices are inserted onto the second edge as follows:

$$A_{c,v^c+x+1} = 1,\ A_{v^c+x+1,v^c+x+2} = 1,\ \ldots,\ A_{v^c+x+y-1,v^c+x+y} = 1,\ and\ A_{v^c+x+y,d} = 1.$$

This process is repeated until all the binary vertices are added to the contracted graph. Finally, we let $A_{i,j} = A_{j,i}$, for $i \neq j$, and $A_{i,j} = 0$, for $i = j$ for all i and j needed to complete the matrix.

3.2 Example

In this section, the above mentioned procedure will be applied to derive some of the (10,13) conventional graphs.

From Eq. (10) we obtain $L = 13 - 10 + 2 = 5$. Hence, all the contracted graphs with five loops are candidates. There are 14 contracted graphs with five loops (Hsu, 1989). For the purpose of demenstration, the (3,6) contracted graph with its edges labeled from 1 to 5 as shown in Fig. 4 is chosen as an example. The matrix E for this contracted graph is given

by Eq. (12). The number of binary vertices to be inserted is found by solving Eq. (6), i.e. $v_2 = v - v^c = 10 - 3 = 7$. Hence, a family of (10,13) conventional graphs can be derived by solving Eq. (13), i.e.

$$k_1 + k_2 + \ldots + k_6 = 10 - 3 = 7 \tag{15}$$

Since there are two groups of parallel edges as shown in Fig. 4. The smallest number of vertices to be inserted onto the parallel edges to prevent any loop formed by just two edges is

$$r = 5 - 2 = 3 \tag{16}$$

The upper bound on k_i is

$$k_{max} = 7 - 3 + 1 = 5 \tag{17}$$

The lower bound on k_i is

$$k_{min} = 0. \tag{18}$$

Solving Eq. (15) subject to the constraints imposed by Eqs. (17) and (18), yields all possible partitions of the binary vertices. Using the computer algorithm outlined in Fig. 5, we found sixty solutions of $K(k_1, k_2, \ldots, k_6)$.

Let us use one of the solutions to Eq. (15), $K = (1,1,1,1,1,2)$, to illustrate the concept of constructing the V-V incident matrix.

Since K(1)=1 and the first row of E is (1,2), let $A_{1,4} = 1$ and $A_{4,2} = 1$.

Since K(2)=1 and the second row of E is (1,2), let $A_{1,5} = 1$ and $A_{5,2} = 1$.

Since K(3)=1 and the third row of E is (1,2), let $A_{1,6} = 1$ and $A_{6,2} = 1$.

Since K(4)=1 and the fourth row of E is (1,3), let $A_{1,7} = 1$ and $A_{7,3} = 1$.

Since K(5)=1 and the fifth row of E is (1,3), let $A_{1,8} = 1$ and $A_{8,3} = 1$.

Since K(6)=2 and the sixth row of E is (2,3), let $A_{2,9} = 1$, $A_{9,10} = 1$ and $A_{10,3} = 1$.

The complete V-V incident matrix A is shown in Eq. (19), and the corresponding conventional graph is shown in Fig. 6.

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{19}$$

Solution No.	Edge label					
	①	②	③	④	⑤	⑥
1	0	2	2	0	2	1
2	0	2	2	1	1	1
3	0	2	2	1	2	0
4	0	2	3	1	1	0
5	1	1	1	1	1	2

Table1. Some admissible partitions of binary vertices for (10,13) conventional graphs

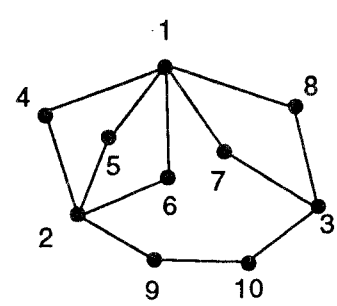


Figure 6: The resulting (10,13) conventional graph

Number of vertices	Number of edges	Number of loops	Number of graphs
7	8	3	4
	9	4	20
	10	5	49
	11	6	76
8	9	3	6
	10	4	40
	11	5	156
	12	6	394

Table 2: Numbers of conventional graphs enumerated with seven and eight vertices

4 Development of an Atlas of Conventional Graphs

A comprehensive set of 230 planar, five-loop, single degree-of-freedom mechanisms with simple joints was developed by Woo (1967). Planar mechanisms with up to five loops and three degrees-of-freedom were determined by Sohn and Freudenstein (1986). An atlas of conventional graphs with up to six vertices, which can be used to create mechanisms of any type, was initially developed by Buchsbaum and Freudenstein (1970) and then further simplified by Mayourian and Freudenstein (1984).

In this paper, using the above procedure, more than 700 new conventional graphs with seven and eight vertices and with up to six loops have been enumerated by using the above procedure. Table 2 lists the number of graphs enumerated as a function of various combinations of the numbers of vertices, edges, and loops. Tables 3 to 8 contain 275 of these new conventional graphs with seven and eight vertices, and with up to five loops.

Graphs with eight vertices, twelve edges and six loops are not included in this paper due to space limitation. Note that, for planar mechanisms, $k_{max} = F + \lambda - 1 = F + 2$ (Sohn, and Freudenstein, 1986). Hence, only those graphs with their binary-vertex chain satisfying $k \leq 3$ are admissible for planar one-degree-of-freedom mechanisms. Furthermore, for planar

mechanisms with simple joints, the minimum number of edges in a loop is four. Thus those graphs with one or more triangular loops are not applicable for planar linkages.

5 Summary

In this paper, an efficient algorithm for the enumeration of conventional graphs from contracted graphs is presented. The algorithm permits the deviation of conventional graphs from contracted graphs by the insertion of binary vertices on the edges of contracted graphs. Two data bases are necessary for automated enumeration of the topological structure of mechanisms. One data base contains contracted graphs classified by the number of loops. The other contains the end vertices of each edge in the contracted graphs. The advantage of this algorithm is that no permutations of binary chains are necessary. The partition of binary vertices is transformed into solving one linear equation in several unknown integers. An efficient method for solving one equation in several integers is described. Based on this method, a computer program has been written for the enumeration of conventional graphs with up to eight vertices and six loops. The enumerated graphs with four to six vertices are in complete agreement with the findings by Bushsbum and Freudenstein (1970). The atlas of graphs with seven and eight vertices are believed to be new. Mechanisms of higher pair joints with seven and eight links can be synthesized by using this atlas.

The random number technique of characteristic polynomial is used to detect graph isomorphism. Since different contracted graphs have different structural topologies, conventional graphs can be isomorphic with another only if they are derived from the same contracted graph. Hence, it is only necessary to check graph isomorphism for those conventional graphs which are derived from the same contracted graph. The existing problem of two non-isomorphic graphs yielding identical characteristic polynomials is avoided.

The method is very effective, and is recommended for further development, including automatic sketching of graphs and their corresponding mechanisms.

Acknowledgement

The authors are grateful to the U.S. Department of Energy for the support of this research under Grant No. DEFG 05-88ER 13977 to the Institute of Systems Research at the University of Marylard.

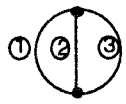
Contracted graphs	Conventional Graphs			
		Partitions of binary vertices		
	No.	①	②	③
	1	0	1	4
	2	0	2	3
	3	1	1	3
	4	1	2	2

Table 3: (7,8) conventional graph

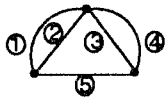
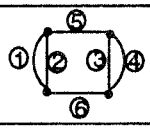
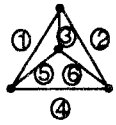
Contracted graphs	Conventional Graphs					
		Partitions of binary vertices				
	No.	①	②	③	④	⑤	⑥
	1	0	1	1	3		
	2	0	1	2	2		
	3	1	1	1	2		
	4	0	1	0	1	2	
	5	0	1	0	2	1	
	6	0	1	0	3	0	
	7	0	1	1	1	1	
	8	0	1	1	2	0	
	9	0	2	0	2	0	
	10	0	2	1	1	0	
	11	1	1	1	1	0	
	12	0	1	0	1	0	1
	13	0	1	0	2	0	0
	14	0	1	1	1	0	0
	15	0	0	0	0	0	3
	16	0	0	0	0	1	2
	17	0	0	0	1	1	1
	18	0	0	1	0	1	1
	19	0	0	1	1	0	1
	20	0	0	1	2	0	0

Table 4: (7,9) conventional graphs

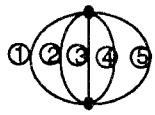
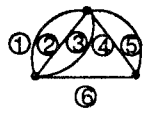
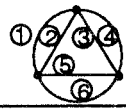
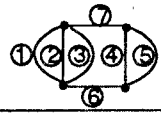
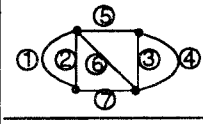
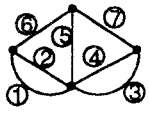
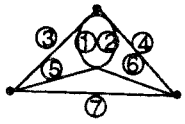
Contracted graphs	Conventional Graphs						
		Partitions of binary vertices					
	No.	①	②	③	④	⑤	⑥	⑦
	1	0	1	1	1	2		
	2	1	1	1	1	1		
	3	0	1	1	0	1	1	
	4	0	1	1	0	2	0	
	5	0	1	1	1	1	0	
	6	0	1	2	0	1	0	
	7	1	1	1	0	1	1	
	8	0	1	0	1	0	2	
	9	0	1	0	1	1	1	
	10	0	1	1	0	1	0	0
	11	0	1	0	1	0	0	1
	12	0	1	0	1	0	1	0
	13	0	1	0	2	0	0	0
	14	0	1	1	1	0	0	0
	15	0	1	0	1	0	0	1
	16	0	1	0	1	1	0	0
	17	0	1	0	2	0	0	0
	18	0	1	1	1	0	0	0
	19	0	1	0	0	0	0	2
	20	0	1	0	0	0	1	1
	21	0	1	0	0	0	2	0
	22	0	1	0	0	1	1	0
	23	0	1	0	1	0	1	0
	24	0	1	0	1	1	0	0
	25	0	2	0	0	0	0	1
	26	0	2	0	0	0	1	0
	27	0	3	0	0	0	0	0
	28	1	1	0	0	0	0	1
	29	1	1	0	0	0	1	0
	30	1	2	0	0	0	0	0

Table 5: (7,10) conventional graphs

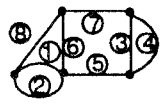
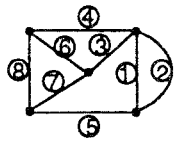
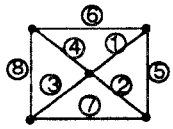
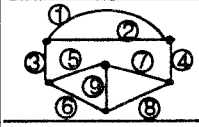
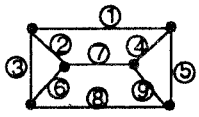
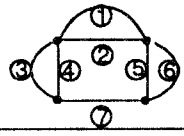
Contracted graphs	Conventional Graphs: Partitions of binary vertices									
	No.	①	②	③	④	⑤	⑥	⑦	⑧	⑨
	31	0	1	0	1	0	0	0	0	
	32	0	1	0	0	0	0	0	1	
	33	0	1	0	0	0	1	0	0	
	34	0	1	0	0	1	0	0	0	
	35	0	1	0	1	0	0	0	0	
	36	0	2	0	0	0	0	0	0	
	37	1	1	0	0	0	0	0	0	
	38	0	0	0	0	0	0	0	2	
	39	0	0	0	0	0	0	1	1	
	40	0	0	0	0	0	1	1	0	
	41	0	0	0	1	0	0	0	1	
	42	0	0	0	1	0	0	1	0	
	43	0	0	0	2	0	0	0	0	
	44	0	0	1	1	0	0	0	0	
	45	0	1	0	1	0	0	0	0	
	46	0	1	0	0	0	0	0	0	
	47	0	0	0	0	0	0	0	0	1
	48	0	0	0	0	0	0	0	1	0
	49	0	1	0	1	0	1	0		

Table 5: (7,10) conventional graph (continued)

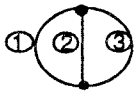
Contracted graphs	Conventional Graphs: Partitions of binary vertices			
	No.	①	②	③
	1	0	1	5
	2	0	2	4
	3	0	3	3
	4	1	1	4
	5	1	2	3
	6	2	2	2

Table 6: (8,9) conventional graph

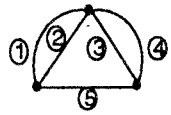
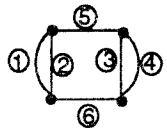
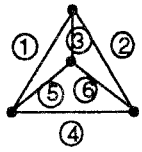
Contracted graphs	Conventional Graphs: Partitions of binary vertices					
	No.	①	②	③	④	⑤	⑥
	1	0	1	1	4	
	2	0	1	2	3	
	3	0	2	2	2	
	4	1	1	1	3	
	5	1	1	2	2	
	6	0	1	0	1	3
	7	0	1	0	2	2
	8	0	1	0	3	1
	9	0	1	0	4	0
	10	0	1	1	1	2
	11	0	1	1	2	1
	12	0	1	1	3	0
	13	0	1	2	2	0
	14	0	2	0	2	1
	15	0	2	1	1	1
	16	0	2	0	3	0
	17	0	2	1	2	0
	18	0	3	1	1	0
	19	1	1	1	1	1
	20	1	1	1	2	0
	21	0	1	0	1	0	2
	22	0	1	0	1	1	1
	23	0	1	0	2	0	1
	24	0	1	0	3	0	0
	25	0	1	1	1	0	1
	26	0	1	1	2	0	0
	27	0	2	0	2	0	0
	28	0	2	1	1	0	0
	29	1	1	1	1	0	0
	30	0	0	0	0	0	4
	31	0	0	0	0	1	3
	32	0	0	0	0	2	2
	33	0	0	0	1	1	2
	34	0	0	1	0	1	2
	35	0	0	1	1	0	2
	36	0	0	1	1	1	1
	37	0	0	1	2	0	1
	38	0	0	1	3	0	0
	39	0	0	2	2	0	0
	40	0	1	1	1	1	0

Table 7: (8,10) conventional graphs

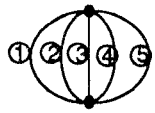
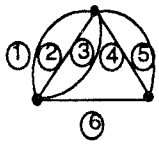
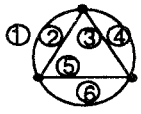
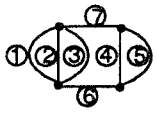
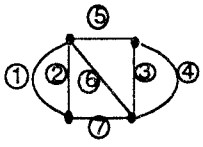
Contracted graphs	Conventional Graphs: Partitions of binary vertices						
	No.	①	②	③	④	⑤	⑥	⑦
	1	0	1	1	1	3		
	2	0	1	1	2	2		
	3	1	1	1	2	2		
	4	0	1	1	0	1	2	
	5	0	1	1	0	2	1	
	6	0	1	1	0	3	0	
	7	0	1	1	1	1	1	
	8	0	1	1	1	2	0	
	9	0	1	2	0	1	1	
	10	0	1	2	0	2	0	
	11	0	1	2	1	1	0	
	12	0	1	3	0	1	0	
	13	0	2	2	0	1	0	
	14	1	1	1	0	1	1	
	15	1	1	1	0	2	0	
	16	1	1	1	1	1	0	
	17	1	1	2	0	1	0	
	18	0	1	0	1	0	3	
	19	0	1	0	1	1	2	
	20	0	1	0	2	0	2	
	21	0	1	0	2	0	0	
	22	0	1	1	1	1	1	
	23	0	1	1	0	1	0	1
	24	0	1	1	0	2	0	0
	25	0	1	1	1	1	0	0
	26	0	1	2	0	1	0	0
	27	1	1	1	0	1	0	0
	28	0	1	0	1	0	0	2
	29	0	1	0	1	0	1	1
	30	0	1	0	1	0	2	0
	31	0	1	0	1	1	0	1
	32	0	1	0	2	0	0	1
	33	0	1	0	2	0	1	0
	34	0	1	0	2	1	0	0
	35	0	1	0	3	0	0	0
	36	0	1	1	1	0	0	1
	37	0	1	1	1	0	1	0
	38	0	1	1	1	1	0	0
	39	0	1	1	2	0	0	0
	40	0	2	0	2	0	0	0
	41	0	2	1	1	0	0	0
	42	1	1	1	1	0	0	0

Table 8: (8,11) conventional graphs

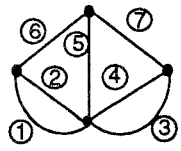
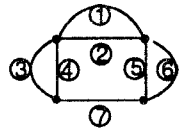
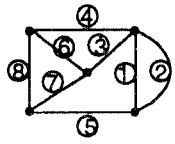
Contracted graphs	Conventional Graphs: Partitions of binary vertices								
	No.	①	②	③	④	⑤	⑥	⑦	⑧
	43	0	1	0	1	0	0	2	
	44	0	1	0	1	0	1	1	
	45	0	1	0	1	1	0	1	
	46	0	1	0	1	2	0	0	
	47	0	1	0	2	0	0	1	
	48	0	1	0	2	0	1	0	
	49	0	1	0	2	1	0	0	
	50	0	1	0	3	0	0	0	
	51	0	1	1	1	0	0	1	
	52	0	1	1	1	0	1	0	
	53	0	1	1	1	1	0	0	
	54	0	1	1	2	0	0	0	
	55	0	2	0	2	0	0	0	
	56	0	2	1	1	0	0	0	
	57	1	1	1	1	0	0	0	
	58	0	1	0	1	0	1	1	
	59	0	1	0	1	0	2	0	
	60	0	1	0	1	1	1	0	
	61	0	2	0	1	0	1	0	
	62	1	1	0	1	0	1	0	
	63	0	1	0	0	0	0	0	2
	64	0	1	0	0	0	0	1	1
	65	0	1	0	0	0	1	0	1
	66	0	1	0	0	0	2	0	0
	67	0	1	0	0	1	0	0	1
	68	0	1	0	0	1	1	0	0
	69	0	1	0	0	2	0	0	0
	70	0	1	0	1	0	0	0	1
	71	0	1	0	1	0	0	1	0
	72	0	1	0	1	0	1	0	0
	73	0	1	0	1	1	0	0	0
	74	0	1	1	1	0	0	0	0
	75	0	2	0	0	0	0	0	1
	76	0	2	0	0	0	1	0	0
	77	0	2	0	0	1	0	0	0
	78	0	2	0	1	0	0	0	0
	79	0	3	0	0	0	0	0	0
	80	1	1	0	0	0	0	0	1
	81	1	1	0	0	0	1	0	0
	82	1	1	0	0	1	0	0	0
	83	1	1	0	1	0	0	0	0
	84	1	2	0	0	0	0	0	0

Table 8: (8,11) conventional graphs (continued)

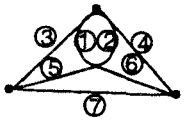
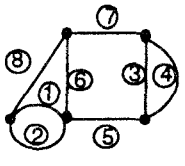
Contracted graphs	Conventional Graphs: Partitions of binary vertices							
	No.	①	②	③	④	⑤	⑥	⑦	⑧
	85	0	1	0	0	0	0	3
	86	0	1	0	0	0	1	2
	87	0	1	0	0	0	2	1
	88	0	1	0	0	0	3	0
	89	0	1	0	0	1	1	1
	90	0	1	0	0	1	2	0
	91	0	1	0	1	0	1	1
	92	0	1	0	1	0	2	0
	93	0	1	0	1	1	0	1
	94	0	1	0	1	1	1	0
	95	0	1	0	1	2	0	0
	96	0	2	0	0	0	0	2
	97	0	2	0	0	0	1	1
	98	0	2	0	0	0	2	0
	99	0	2	0	0	1	1	0
	100	0	2	0	1	0	1	0
	101	0	2	0	1	1	0	0
	102	0	3	0	0	0	0	1
	103	0	3	0	0	0	1	0
	104	0	4	0	0	0	0	0
	105	1	1	0	0	0	0	1
	106	1	1	0	0	0	1	1
	107	1	1	0	0	0	2	0
	108	1	1	0	1	1	0	0
	109	1	1	0	1	0	1	0
	110	1	1	0	1	1	0	0
	111	1	2	0	0	0	0	1
	112	1	2	0	0	0	1	0
	113	1	3	0	0	0	0	0
	114	2	2	0	0	0	0	0
	115	0	1	0	1	0	0	0	1
	116	0	1	0	1	0	0	1	0
	117	0	1	0	1	0	1	0	0
	118	0	1	0	1	1	0	0	0
	119	0	1	0	2	0	0	0	0
	120	0	1	1	1	0	0	0	0
	121	0	2	0	1	0	0	0	0
	122	1	1	0	1	0	0	0	0

Table 8: (8,11) conventional graphs (continued)

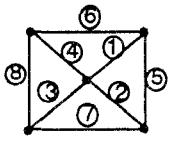
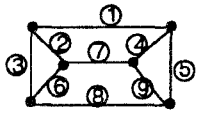
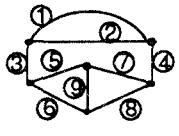
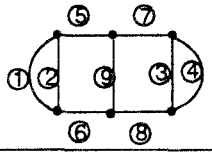
Contracted graphs	Conventional Graphs: Partitions of binary vertices								
	No.	①	②	③	④	⑤	⑥	⑦	⑧	⑨
	123	0	0	0	0	0	0	0	3
	124	0	0	0	0	0	0	1	2
	125	0	0	0	0	0	1	1	1
	126	0	0	0	0	0	1	2	0
	127	0	0	0	1	0	0	0	2
	128	0	0	0	1	0	0	1	1
	129	0	0	0	1	0	0	2	0
	130	0	0	0	1	0	1	0	1
	131	0	0	0	1	0	1	1	0
	132	0	0	0	1	1	0	1	0
	133	0	0	0	2	0	0	0	1
	134	0	0	0	2	0	0	1	0
	135	0	0	0	3	0	0	0	0
	136	0	0	1	1	0	0	0	1
	137	0	0	1	1	0	0	1	0
	138	0	0	1	1	1	0	0	0
	139	0	0	1	2	0	0	0	0
	140	0	1	0	1	0	0	0	1
	141	0	1	0	2	0	0	0	0
	142	0	1	1	1	0	0	0	0
	143	0	0	0	0	0	0	0	0	2
	144	0	0	0	0	0	0	0	1	1
	145	0	0	0	0	0	0	0	2	0
	146	0	0	0	0	0	0	1	1	0
	147	0	0	0	0	0	1	0	0	1
	148	0	0	0	0	1	0	0	0	1
	149	0	0	0	0	1	0	1	0	0
	150	0	0	0	0	1	1	0	0	0
	151	0	1	0	0	0	0	0	0	1
	152	0	1	0	0	0	0	0	1	0
	153	0	1	0	1	0	0	0	0	0
	154	0	2	0	0	0	0	0	0	0
	155	1	1	0	0	0	0	0	0	0
	156	0	1	0	1	0	0	0	0	0

Table 8: (8,11) conventional graphs

References

Alhakim, L. A., and Shrivastava, A., 1991, "Application of Graph Theory for Structual Enumeration and Presentation of Mechanisms," *Eighth World Congress on the Theory of Machines and Mechanisms*, Prague, Czechoslovakia.

Buchsbaum, F., and Freudenstein, F., 1970, "Synthesis of Kinematic Structure of Geared Kinematic Chains and Other Mechanism," Journal of Mechanism, Vol. 5, pp. 357–392.

Crossley, F. R. E., 1964a, "A Contribution to Grubler's Theory in the Number Synthesis of Plane Mechanisms," ASME Transactions, Journal of Engineering for Industry, Vol. 86B, pp. 1–8.

Crossley, F. R. E., 1964b, "The Permutation of Ten-Link Plane Kinematic Chains," Antriebstechnik, Vol. 3(5), pp. 181–185.

Crossley, F. R. E., 1965, "The Permutations of Kinematic Chains of Eight Members or Less from the Graph-Theoretic Viewpoint," Developments in Theoretical and Applied Mechanics, Vol. 2, pp. 467–486.

Freudenstein, F., 1967, "The Basic Concepts of Polya's Theory of Enumeration with Application to the Structural Classification of Mechanisms," Journal of Mechanism, Vol. 3, No. 3, pp. 275–290.

Freudenstein, F., and Maki, E. R., 1979, "The Creation of Mechanisms According to Kinematic Structure and Function," Journal of Environment and Planning, Vol. 4, pp. 357–391.

Gelfond, A. O., 1981, *Solving Eqautions in Integers*, Mir Publisher, Moscow, Translated from the Russian by O.B. Sheinin.

Harary, F., 1969, *Graph Theory*, Addison Wesley. Reading, MA.

Hsu, C. H., 1989, "Atlas of Contracted Graphs for Basic Kinematic Chains with Simple Joints," *Proceedings of the 5th IFTOMM International Symposium on Linkage and Computer Design Methods-Theory and Practice of Mechanisms*, Bucharest, Romania.

Huang, M., and Soni, A. H., 1973, "Application of Linear and Nonlinear Graphs in Structural Synthesis of Kinematic Chains," ASME Transactions, Journal of Engineering for Industry, Vol. 95, No. 6, pp. 525–532.

Hwang, W. M., Hwang, Y. W., and Liu, Y. Y., 1991, "A Method for the Number Synthesis of Planar Kinematic Chains with Simple Joints," Journal of the Chinese Society of Mechanical Enigineering, Vol. 12, No. 2, pp. 166–176.

Mayourian, M., and Freudenstein, F., 1984, "The Development of an Atlas of the Kinematic Structures of Mechanisms," ASME Transactions, Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 106, pp. 458–461.

Mruthyunjaya, T. S., 1979, "Structual Synthesis by Transformation of Binary Chains," Mechanism and Machine Theory, Vol. 14, No. 4, pp. 357–391.

Mruthyunjaya, T. S., 1984a, "A Computerized Methodology for Structuaral Synthesis of Kinematic Chains: Part 1-Formulation," Mechanism and Machine Theory, Vol. 19, No. 6,

pp. 487–495.

Mruthyunjaya, T. S., 1984b, "A Computerized Methodology for Structuaral Synthesis of Kinematic Chains: Part 2-Application to Several Fully or Partially Known Cases," Mechanism and Machine Theory, Vol. 19, No. 6, pp. 497–505.

Mruthyunjaya, T. S., 1984c, "A Computerized Methodology for Structuaral Synthesis of Kinematic Chains: Part 3-Application to the New Case of 10-Link, Three Freedom Chains," Mechanism and Machine Theory, Vol. 19, No. 6, pp. 507–530.

Sohn, W. J., and Freudenstein, F., 1986, "An Application of Dual Graphs to the Automatic Generation of the Kinematic Structure of Mechanisms," ASME Transactions, Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 108, No. 3, pp. 392–398.

Tsai, L. W., 1987, "An Application of the Linkage Characteristic Polynomials to the Topological Synthesis of Epicyclic Gear Trains," ASME Transactions, Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 109, No. 3, pp. 329–336.

Tuttle, E. R., and Peterson, S. W., 1987, "Symmetric Group Theory Applied to Kinematic Chain Enumeration," *Proceedings of the 10th Applied Mechanisms Conference*, Vol. 2, No.4B, New Orleans, Louisiana.

Tuttle, E. R., Peterson, S. W., and Titus, J. E., 1989a, "Enumeration of Basic Kinematic Chains Using the Theory of Finite Groups," ASME Transactions, Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 111, No. 4, pp. 498–503.

Tuttle, E. R., Peterson, S. W., and Titus, J. E., 1989b, "Further Applications of Group to the Enumeration and Structural Analysis of Basic Kinematic Chains," ASME Transactions, Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 111, No. 4, pp. 494–497.

Woo, L. S., 1967, "Type Synthesis of Plane Linkages," ASME Transactions, Journal of Engineering for Industry, Vol. 89B, pp. 159–172.

Yan, H. S., 1980, "The Characteristic Polynomial of Kinematic Chains," PhD thesis, University of Purdue, West LaFayette, Indiana.

Appendix A

Solving one linear equation in m unknown integers

In this section a simple procedure for solving one linear equation in more than one unknown integers is described. This procedure can be applied for the enumeration of conventional graphs.

Equation (A.1) shows a linear equation in m integers, k_i, $i = 1, \ldots, m$, where p is a known positive integer. Solving Eq. (A.1) for these m integers, all of which are non-negative, is called m partitions of an integer p.

$$k_1 + k_2 + \ldots + k_m = p \tag{A.1}$$

Since there are m unknowns in one equation, we may choose $m-1$ unknowns arbitrarily and then solve Eq. (A.1) for the remaining unknown, provided all the k_i's are non-negative. In order to systematically choose these $m-1$ integers, we let

$$l_1 = \sum_{i=2}^{m} k_i \tag{A.2}$$

Substituting Eq. (A.2) into Eq. (A.1), yields

$$k_1 + l_1 = p \tag{A.3}$$

Equation (A.3) contains only two unknowns. Hence, we may choose one of the two variables arbitrarily and solve for the other variable. This can be easily accomplished by letting k_1 to assume the values of 0 to p, and then solving Eq.(A.3) for l_1. The results are as shown below:

Variables	Solutions				
k_1	0	1	2	$\cdots$	p
l_1	p	p-1	p-2	$\cdots$	0

For each set of (k_1, l_1), let

$$k_2 + l_2 = l_1 \tag{A.4}$$

where $l_2 = \sum_{i=3}^{m} k_i$.

Again, we have one equation in two variables. We may let k_2 to assume the values of 0 to l_1 and solve Eq. (A.4) for l_2. The results are as shown below:

Variables	Solutions				
k_2	0	1	2	$\cdots$	l_1
l_2	l_1	l_1-1	l_1-2	$\cdots$	0

The above procedure can be repeated until all sets of partitions are obtained. The procedure is very simple and can be easily implemented on a digital computer. A more rigorus solution procedure for solving one linear equation in two unknowns can be found in (Gelfond, 1981). The following shows a simple nested *do-loop* program to solve for all the solutions.

```
Do  i1  = 0,  p
     k1  = i1;
     l1  =  p - k 1
    Do  i2   =  0,  l1
         k2  =  i2;
         l2  =  l1 - k 2
    ..........
         Do  im-1 =  0,  lm-2
              km-1 =  im-1 ;
              km  =  lm-2  - k m-1
         Continue
    ..........
    Continue
Continue
```

Figure 7: The do-loop for the computer program