Spring 1991

Kinematic Synthesis of Deployable-Foldable Truss Structures Using Graph Theory

Dirk B. Warnaar
Old Dominion University

KINEMATIC SYNTHESIS OF DEPLOYABLE-FOLDABLE TRUSS STRUCTURES USING GRAPH THEORY

by

Dirk B. Warnaar
M.S. May 1985, Delft University of Technology
Delft, The Netherlands,1985

A Dissertation Submitted to the Faculty
of Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy

Department of Mechanical Engineering and Mechanics
Old Dominion University
Norfolk, VA

April 2, 1991

Approved by :

Dr. Meng-Sang Chew (Director)	Dr. Gene W. Hou
Dr. Melvin S. Anderson	Dr. Jer-Nan Juang
Dr. Stephan G. Cupschalk	Dr. John P. Raney

ABSTRACT

KINEMATIC SYNTHESIS OF DEPLOYABLE-FOLDABLE TRUSS STRUCTURES USING GRAPH THEORY

Dirk B. Warnaar

Old Dominion University

Director: Dr. Meng-Sang Chew

A graph theoretic approach is applied to the conceptual design of deployable truss structures. The characteristics that relate to the inter-connectivity of the elements of a deployable truss structure can be captured in a schematic representation, called a graph. A procedure is presented that enables the exhaustive generation of these graphs for structures of any given number of nodes and links and which are foldable onto a plane or onto a line.

A special type of truss structures, called truss modules, is presented. Graphs of this class of structures form a subset of the graphs of truss structures. Two procedures are presented that are applied to recognize these graphs among graphs of truss structures. The procedures also generate information on the relative lengths of the links in a truss module by examining the graph it represents. This enables the generation of numerous novel (deployable) truss modules as well as those that have been reported in the literature.

A procedure is presented for the generation of all possible folded configurations of deployable truss structures. By applying this procedure to deployable truss modules, truss modules are identified that exhibit special geometrical properties which allow the module to fold using fewer joints than dictated in the initial phase of the conceptual design process. Using an alternate definition of graphs, procedures are presented for the specification of the joint types and joint inter-connectivity that accommodates the folding and/or deployment of a deployable truss structure. These procedures are applied to generate all possible joint assignments for deployable truss modules.

Procedures for the conceptual design of deployable truss structures result in the generation of innumerable design concepts. An expert system is developed to aid the designer of deployable truss structures in the evaluation of such designs. Incorporated in this expert system are selection criteria that are developed to assist a designer in selecting the best candidates for any given application. Employing this approach, many promising novel designs, as well as those that have been reported in the literature, are identified.

Dedicated to my wife, Kathy,

and my daughters, Susan and Kristina.

ACKNOWLEDGEMENTS

It is my great pleasure to express sincere gratitude to my advisor Dr. Meng-Sang Chew. Throughout the Ph.D. Program, Dr. Chew has provided me with invaluable guidance in relation to my research as well as my professional career. Furthermore, he has continually provided me with challenges related to this research and the encouragement to meet them. I am convinced that his support and his outstanding knowledge in the area of mechanism design were the two most important factors in meeting the objectives of this research. Also, I will fondly remember all the discussions I have had with Dr. Chew in the past three and a half years as well as the numerous occasions at which I and my family were his geusts.

I also wish to recognize the support, encouragement, and technical guidance from the additional members of my Advisory Committee; Drs. M. S. Anderson, S. G. Cupschalk, G. W. Hou, J. Juang and J. P. Raney. I appreciate the time and effort each member expended for my research.

I would like to thank Dr. Mel Anderson also for his very special friendship. Dr. Anderson's support enabled me to enter this country and recieve the opportunity to perform this research. He and Mrs. Anderson made my first months here easier with warm and greatly appreciated acts of friendship.

CONTENTS

TABLES

TABLE

FIGURES

FIGURE

CHAPTER 1

INTRODUCTION

1.1 Motivation

A deployable structure distinguishes itself from other types of structures in that it can be folded into or deployed from a very compact package to facilitate its transportation. Structures of this type are commonly used in applications ranging from baby carriages and tie racks to exhibition poster boards and solar energy panels. Deployable truss structures for space applications have received tremendous attention because storage space on the space shuttle or any other type of rocket booster is a premium [1].

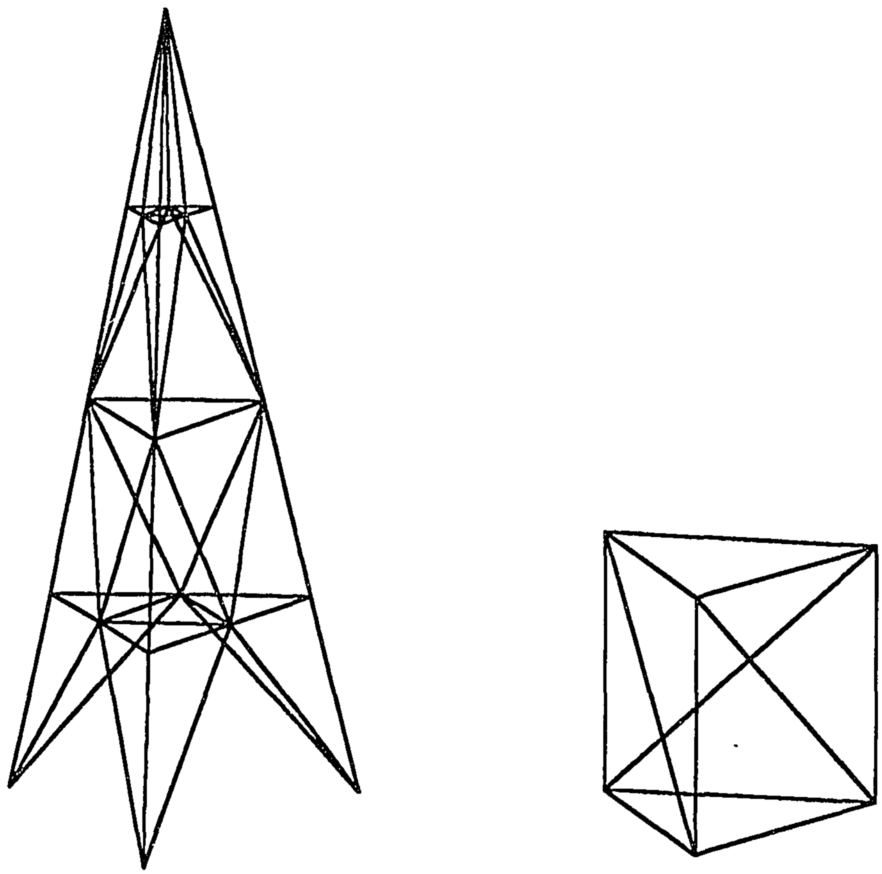
The necessity to store a truss structure in a compact form has led to many creative designs of deployable truss structures. In some previous work [1-15], such designs have been introduced and discussed from the viewpoint of their usefulness and performance. However, the experience and intuitive abilities that led to the generation of these previous designs are not generally available or discussed. Hence, to date, designs of deployable structures have only been created by trial and error. The disadvantages of this approach is that a considerable demand in time and effort is placed on the designer and that this effort has to be repeated for each new design. Furthermore, despite the effort involved in this approach, there is no guarantee that the best design is found. Thus, a need exists for a systematic design procedure that does not have these shortcomings.

The first attempts to advance the understanding of deployable truss structures have been made by Stoll [16] and Gvamitchava [17], who independently developed systematic design procedures for this type of structures. However, their procedures have very limited capabilities and are impractical for use on spatial deployable structures. Also, little attention has been given to enhancing the procedures so as to minimize the need for human judgement. The objective of the research reported in the following chapters, is to advance the understanding of the design process of (spatial) deployable truss structures through the development of systematic design procedures for the generation of all conceptual design of this type of structures.

To allow an in-depth investigation of deployable truss structures, this research concentrates on an important class of truss structures, called truss modules. Modules are the commonly used building blocks of large truss structures such as domes, booms and space-based platforms (see Fig. 1.1). Hence, large deployable truss structures can be formed by combining a large number of deployable truss modules. It should be pointed out that, in such applications, most truss modules have special properties in addition to those of truss structures, such as the presence of an interface to an adjacent module. However, not all truss structures have these additional properties. Truss modules may therefore be considered a subset of truss structures. As an example, shows two truss structures of which only the truss structure shown in Fig. 1.2b belongs to the set of truss modules since this second structure may be used as a building block for larger structures.

1.2 Conceptual Design of Deployable-Foldable Truss Structures

The development of techniques for the conceptual design of new mechanisms has received considerable interest in the past two decades. Much literature can be found that relates to the *kinematic structure* (i.e. the kinematic composition) of mechanisms and their enumeration with the aid of graph theory [18-26]. These



(a) A truss structure

(b) A truss structure that is also a truss module

Fig. 1.2 Illustration of truss modules being a subset of truss structures

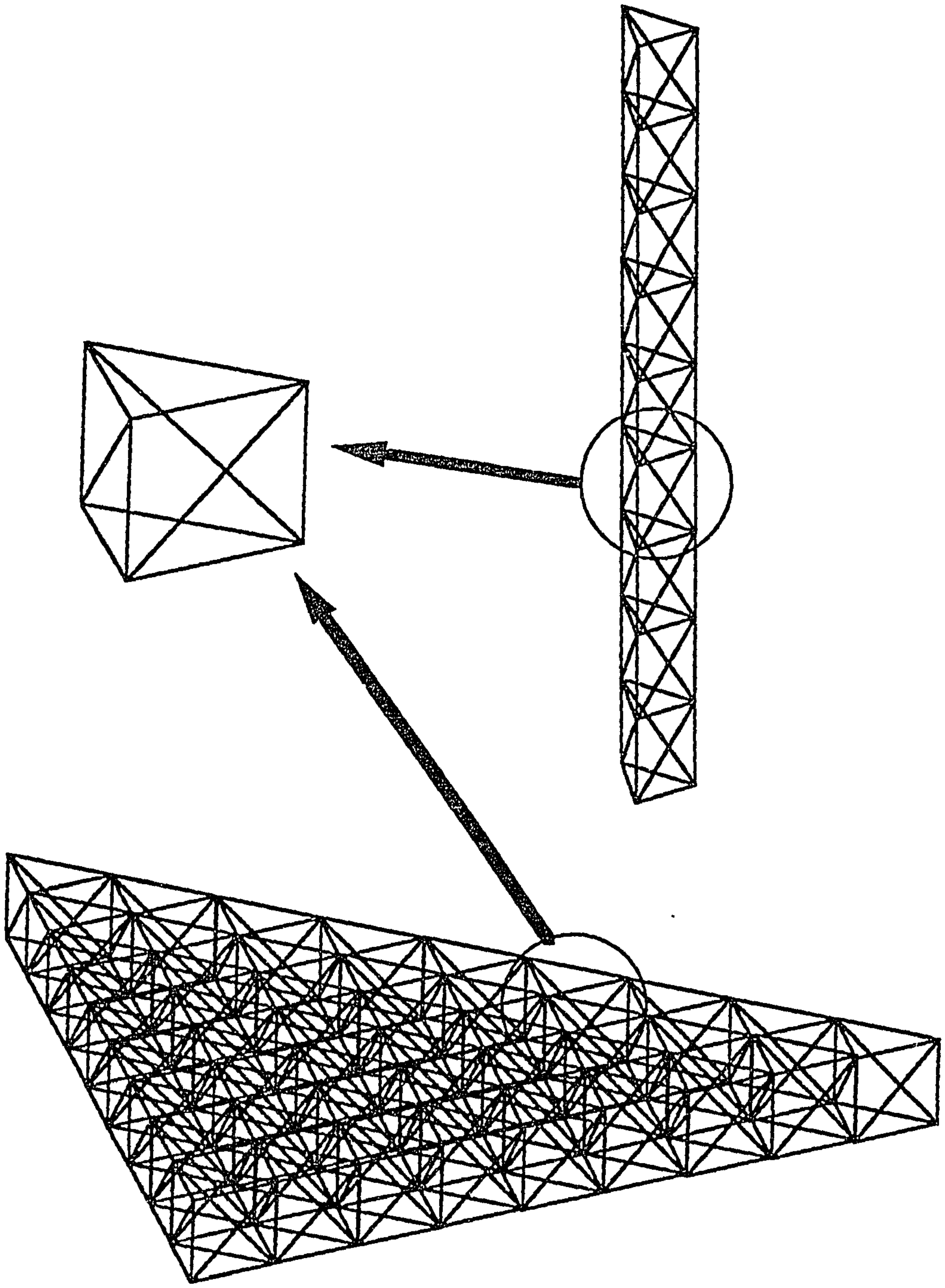


Fig. 1.1 Examples of large truss structures that are built out of a large number of identical modules.

techniques are commonly referred to as *kinematic structural synthesis*. Applications have extended the basic techniques to aid the designer in arriving at new mechanisms ranging from shaft couplings and gear trains to robot arms and window mechanisms.

A cursory examination would show that most deployable truss structures are, in essence, mechanisms during the deployment and retraction phases. For this reason, the kinematic structural synthesis technique seems suitable for application into this new arena. However this technique, as it is presently applied to mechanisms, has the drawback that tremendous resources are required for enumeration and evaluation of mechanisms with ten or more links as a result of the extremely large number of possible combinations in the topological inter-connectivity of the links. This limitation is severe when the technique is applied directly to deployable truss structures since even the most basic cell of a spatial structure contains twelve or more links. That, in combination with different types of joints that connect the various links together, results in a problem quite unmanageable even with current computational resources. Thus, a modification of the kinematic structural synthesis technique is needed before it can be applied to deployable truss structures. The modified technique, based on a structures perspective, will be presented in Chapter 2.

Application of the techniques presented in Chapter 2 results in the generation of abstract representations, called graphs, each revealing the connectivity of a deployable truss structure. However, the majority of these graphs represent the same inter-connectivity of the elements and therefore represent the same deployable truss structure. To avoid unnecessary computation in subsequent steps of the conceptual design process, graphs representing the same inter-connectivity must be eliminated. These so called isomorphic graphs can be detected by exploring all possible renumbering sequences of the vertices of a graph until one is found which transforms this graph into a previously identified graph. Although this procedure is accurate, it is

very inefficient. Applications of graphs in mechanisms design [11-19] often involve a large number of checks for isomorphism. For this reason, much effort has been made in an attempt to develop efficient procedures for isomorphism detection [19,33-39]. However, to date, the available techniques have proven to be either inadequate in detecting all non-isomorphic graphs [19,38] or are only efficient for certain types of graphs [34,36] or graphs of limited size [33,35,37,39]. Therefore a need exists for an efficient procedure that finds conclusive evidence for isomorphism for any size graphs, in particular for graphs of large sizes, such as those for deployable truss structures. Chapter 3 introduces such a procedure.

In Chapter 4 the conceptual design process for deployable truss structures, initiated in Chapter 2, is further refined for truss modules by making use of the additional properties of this class of truss structures. These properties make it possible not only to identify graphs of truss modules, but also to generate data on the relative lengths of the links of a truss module by investigating the topology of its graph. This is a surprising development because it is generally accepted that graphs do not yield information on the relative dimensions of the physical systems they represent. Given this information, it is then also possible to construct a three-dimensional model of the corresponding truss module.

The relative lengths of the links in a deployable truss structure are very important and directly impact the deployed and folded configurations as well as the joints that can be used to enable the transition between the two configurations. Special dimensions for the links of a deployable truss structure may therefore result in deployed and folded configurations that have very desirable characteristics (such as a repetitive geometry and efficient packaging) and which may also allow a transition between the two configurations with fewer joints used and/or joints providing fewer degrees of freedom. In order to determine whether a deployable truss structure has favorable packaging and/or joint characteristics it is necessary to generate its deployed configuration and all its folded derivatives. The objective of Chapter

5 is to accommodate this evaluation through generating all folded configurations of a deployable truss structure of which the deployed configuration or the relative lengths of its links are known.

During the transition between the folded and the deployed configuration, a deployable truss structure behaves like a mechanism. A frequently used medium for representing the kinematic structure of a mechanism is a graph. This approach, which was first introduced by Freudenstein and Dobrjanskyj [18], has opened the door to a wide variety of applications of graph theory in areas such as the enumeration of mechanisms [23,26]; the generation of mechanisms [19,25,26]; and the analysis of mechanisms [20-22,24]. The complexity of spatial mechanisms, due to the availabilty of a wide variety of spatial joints permitting one or more degrees of freedom and the complexity of spatial trajectories, have limited most of the applications of graphs to planar mechanisms. However, it will be shown in Chapter 6 that graphs can also be instrumental in specifying joint types of (spatial) deployable truss structures.

1.3 Design Selection

Chapters 2-6 give a complete description of procedures for the conceptual design of deployable truss structures that can be used as building blocks of much larger deployable truss structures. These techniques lead to the creation of an enormous number of conceptual designs. Although the availability of a large number of alternative designs is generally considered an asset, a designer is only interested in at most a handful of designs that will best serve the application at hand. Selecting such a small set of promising designs from among a large number of designs can be cumbersome, since it involves the evaluation and comparison of each available design.

A designer can be relieved from the burden of design selection by automating the selection process. Recent developments in mechanism design [27-31] suggest

that this can best be achieved by using computer programs that are referred to as *expert systems*. The flexible architecture of expert systems allows a designer to easily implement and alter a set of criteria used in the design selection. This has the advantage that the designer can concentrate on defining these criteria and leave the design selection to the expert system.

The formulation of an expert system for the selection of designs of deployable truss structures is severely complicated by the fact that little attention has been given in the literature to considerations that have led to the development of existing designs of deployable truss structures. Hence, criteria for the evaluation and comparison of novel designs of deployable truss modules (e.g. the designs generated by procedures introduced in Chapters 2-6) are not readily available. The objective of Chapter 7 is to discuss criteria for the selection of conceptual designs and to implement them in an expert system for the design selection of deployable truss structures. Included in the discussion are examples of the application of the expert system in the selection of a truss module that is to be used as a building block of a much larger truss structure.

The following chapter discusses the first step in the conceptual design process for deployable truss structures : the generation of graphs of deployable-foldable truss structures.

CHAPTER 2

GENERATION OF GRAPHS OF DEPLOYABLE-FOLDABLE TRUSS STRUCTURES

2.1 Outline

In the next section, definitions of a truss structure and its graph will be presented. These definitions will be used to establish relationships between the characteristics of a physical structure and the characteristics of its graphical representation. A subsequent section covers a similar discussion, but concentrates on deployable truss structures. A comparison will then be made between the characteristics of truss structures and deployable truss structures.

From the development of the following two sections (2.2 and 2.3) two procedures will be presented for the systematic design of deployable structures. The first procedure generates all graphs of truss structures that contain a given number of nodes and links. The second procedure derives all the graphs of deployable structures by operating on each graph produced by the first procedure. A detailed discussion of the procedures will be provided in Sections 2.4 and 2.5 respectively.

2.2 Graph Properties of Truss Structures

2.2.1 Definition

Figure 2.1a shows a six node spatial truss structure. The graph of this structure shown in Fig. 2.1b, has six vertices and twelve edges. The vertices in the graph correspond to the nodes of the structure and the edges to its links. A graph can be represented by an adjacency matrix [32], which has the property that when element a_{ij} is non-zero, vertices i and j are inter-connected. This property implies that the adjacency matrix is symmetric. The number of non-zero entries in column (or row) i of the matrix is equivalent to the number of edges incident to vertex i. This number is called the degree of vertex i. It is noted that the graph merely contains information on the connectivity of the nodes and does not contain any information on the length of the links nor on the locations of the nodes.

2.2.2 Properties of Truss Structures

For the purpose of identifying the characteristics of truss structures it is necessary to establish a definition for the term *truss structure*. The following definition has been adopted for the purposes of this investigation:

> A truss structure is a collection of nodes and links, where the links form the connections between the nodes such that the distances between the nodes cannot be changed without deformation of the links.

This definition leads to the conclusion that a truss structure can be viewed as a mechanism that is completely constrained. Although the definition is also valid when a subset of the links are cables, links will be regarded as elements that can carry both tension and compression. The following observations can then be made on the characteristics of spatial truss structures :

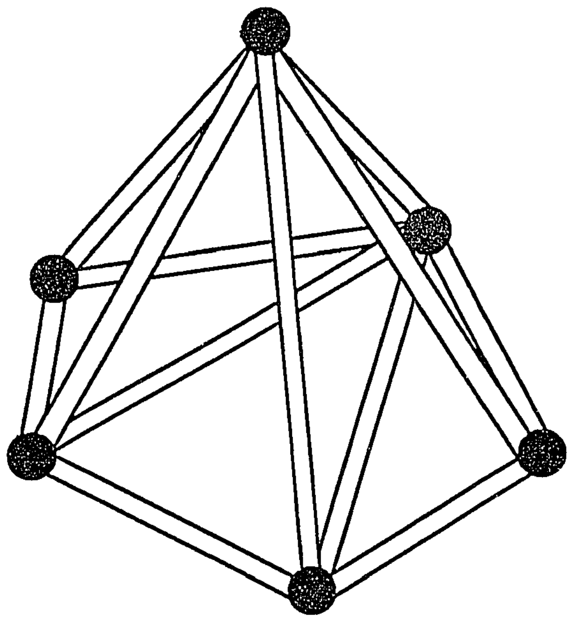


Fig. 2.1a, A six node truss structure

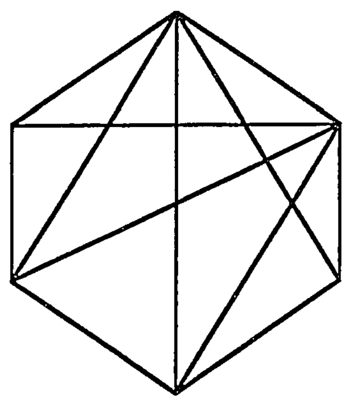


Fig. 2.1b, Graph of the structure in Fig. 2.1a

1. The translation of a node can only be restricted by employing a minimum of three links, one for each direction of translation.
2. When two nodes of degree three are adjacent to each other, then only five of the six links are available to restrict the translation of both nodes with respect to the remaining structure. This situation should be avoided. However, an exception exists for the smallest possible spatial truss structure (i.e. a structure with four nodes).
3. There is at least one loop wherein the number of links is equal to the number of nodes. In such a loop each node is visited exactly once in traversing that loop. Using the fact that each node is connected to at least three nodes, it can be proven that such a loop always exists.
4. The number of nodes n and links l in a structure must satisfy the inequality $l \geq 3n - 6$. This relation is found by requiring that at least three links are incident to each node in the structure and allowing the total structure to have six rigid-body degrees-of-freedom.
5. A special case exists when $l = 3n-6$. Such a structure is statically determinate. The nodes of this type of structure are completely constrained by a minimum number of links. Any additional link leads to an over-constrained structure.

Implicit in the observations on the characteristics of truss structures, presented above, is the assumption that necessary and sufficient number of degrees-of-freedom at the joints have been specified so that statical determinacy is not due to insufficient degrees-of-freedom at the joints. The five observations can be translated into a corresponding number of properties of graphs of spatial truss structures. A summary of these properties is presented in Table 2.1. Similar observations can be made regarding planar and linear truss structures. The graph properties of these types of structure are also provided in this table.

1 The degree of each vertex in the graph is at least equal to the dimension parameter α.

2 The graph contains no vertex of degree α that is adjacent to another vertex of degree α, unless the total graph contains exactly $\alpha + 1$ vertices.

3 The graph has at least one loop consisting of a number of edges equal to the number of vertices, such that each vertex is visited once.

4 The number of vertices v and edges e in the graph must satisfy : $e \geq \alpha v - \beta$, where β is the freedom parameter.

5 The graph of a statically determinate structure satisfies the relation : $e = \alpha v - \beta$.

For spatial structures : $\alpha = 3$, $\beta = 6$; planar structures : $\alpha = 2$, $\beta = 3$; linear structures : $\alpha = 1$, $\beta = 1$.

Table 2.1, Summary of properties of graphs of truss structures.

2.3 Graph Properties of Deployable Truss Structures

2.3.1 Definition

Figure 2.2a shows a six node deployable spatial truss structure. Note that this structure differs from the structure of Fig. 2.1a only by the three additional nodes that are located along three of the five vertical links. The graph of this structure, Fig. 2.2b, has six vertices and twelve edges, similar to the graph of the structure of Fig. 2.1a, but with the difference that three edges of the graph in Fig. 2.2b are drawn as dashed lines. These lines are used to indicate that the particular edge represents a set of two edges and one vertex, as illustrated in Fig. 2.3. The combination of edges in Fig. 2.3a permits the vertices A and C (the end points, or nodes, of the combination) to fold towards each other. It can be noted that the two edges have the unique property that they are incident to one vertex that is adjacent to these two edges only. By this definition, the dashed lines in Fig. 2.2b represent the links of the structure in Fig. 2.2a that contain an additional node that represents a joint with the necessary and sufficient degrees of freedom for folding. In general, dashed lines will be used to indicate links that have the property that the distance between its end points is variable. This concept will be discussed in greater detail in the next section.

2.3.2 Properties of Deployable Truss Structures

A deployable truss structure may be defined as follows :

A deployable truss structure is a truss structure in which additional degrees-of-freedom can be activated so that it can be folded to a configuration of lower dimension.

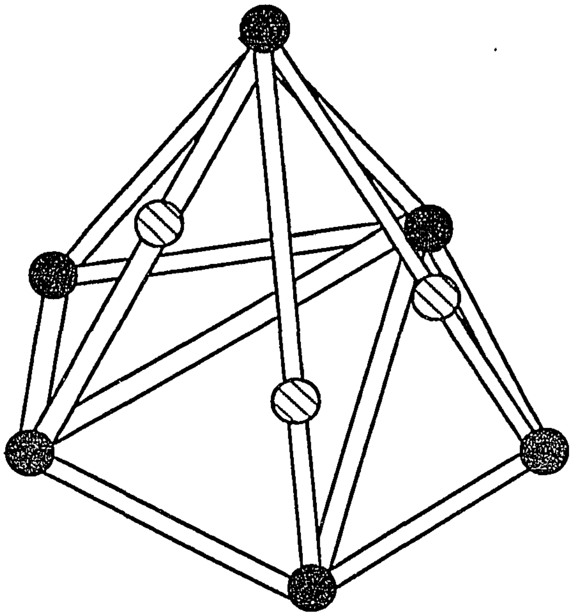


Fig. 2.2a. A six node deployable truss structure.

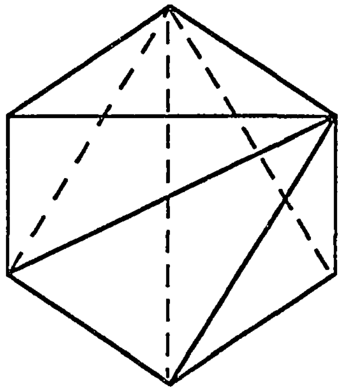


Fig. 2.2b, Graph of the structure in Fig. 2.2a.

Fig. 2.3a Combination of two edges that are incident to a vertex of degree two

Fig. 2.3b Graph equivalent presentation of the combination of edges shown in Fig. 2.3a

The definition implies that a truss structure is deployable when it can be changed into a mechanism. In general, this change takes place by removing obstructions from particular joints, thus activating additional degrees-of-freedom. Examples are : removing the lock-ups, activating a motor of a screw joint or buckling of certain members within the structure. Removing an obstruction only changes the length of a link that contains this obstruction. In this process, this link is changed such that the distance between the end points of this link is variable. At this stage of the conceptual design process it will be assumed that there is no limitation to the change in length of the member. Since this type of member enables the transition (folding or deploying) of the structure it shall be referred to as a *transition link*. It is emphasized that a transition link does not put a constraint on the folded configuration of a structure since its length, measured from end point to end point, is variable. Thus, the folded configuration is determined only by the links that are of fixed length.

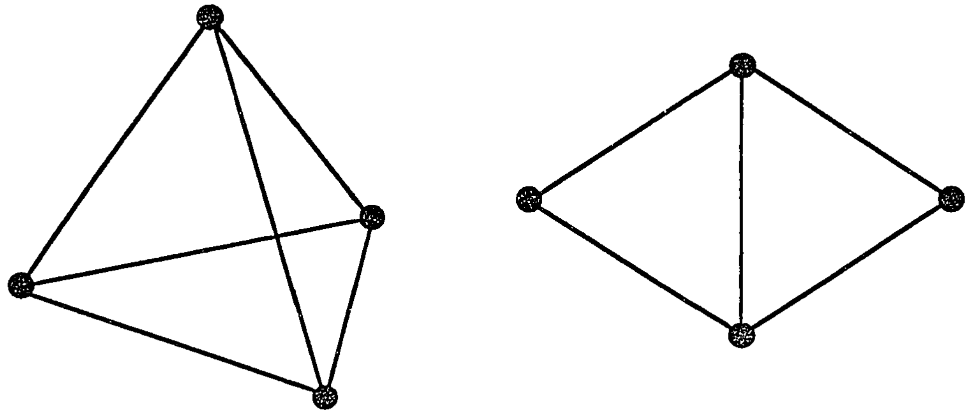
A deployable structure in its folded configuration should always occupy less volume than in its deployed state. This is to imply that a spatial truss structure is either foldable onto a plane or onto a line, while a planar structure is foldable onto a line. The transition of the structure from either a spatial or planar structure to a planar or linear structure must be continuous. This means that the structure, in its transition phase, is not prevented from reaching the fully folded configuration. Obstructions can occur due to collision of links or when a situation is reached where the mechanism is over-constrained (i.e. at bifurcation or branch points). The collision of links can be avoided by a proper choice of joint types and joint orientations. The second situation can be avoided by requiring that the folded configuration not be over-constrained, which can be satisfied by selecting the appropriate number and location of transition links.

The minimum number of transition links, required to enable folding of the structure, is equal to the total number of links in the structure minus the maximum

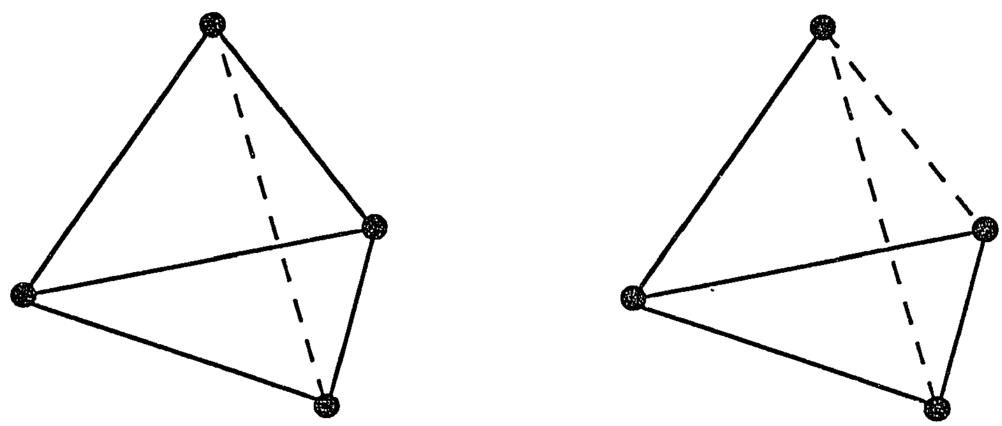
number of links of fixed length that can be present in the folded configuration. This maximum number is equivalent to the minimum number of links required to form a planar structure, when the mechanism is folded onto a plane; or a linear structure, when the mechanism is folded onto a line, since any additional link results in an over-constrained configuration. As an example, consider the spatial truss structure shown in Fig. 2.4a. This structure has four nodes and six links. To form a planar structure of four nodes a minimum of five links is needed, as shown in Fig. 2.4b. Since this number is equal to the maximum number of links of fixed length that can be present in the folded configuration, the structure of Fig. 2.4a must have at least $6 - 5 = 1$ transition links to be able to fold onto a plane. This can be verified by selecting any of the links in this structure as transition link. An example is shown in Fig. 2.4c (with the dashed lines indicating the transition link). It is noted that it is also possible to have more then one transition link in the structure of Fig. 2.4a. A deployable structure resulting from an arbitrary selection of two links as transition links is illustrated in Fig. 2.4d.

The previous discussion can be summarized as follows :

1. Transition links are links of variable length and therefore do not constrain the folded configuration.
2. A mechanism cannot become completely folded when the folded configuration is over-constrained.
3. The minimum number of transition links is determined by the maximum number of links of fixed length that can be present in the folded configuration.
4. The number of fixed length links in the folded configuration is a maximum when these links form a planar statically determinate structure ($l = 2n - 3$), when the mechanism is folded onto a plane; or a linear statically determinate structure ($l = n - 1$), when the mechanism is folded onto a line.



(a) A four-node spatial truss structure (b) A four-node planar truss structure



(c) A four-node deployable truss structure that s foldable onto a plane and has a minimum number of transition links

(d) A four-node deployable truss structure with two transition links

Fig. 2.4 Illustration of the selection of transition links in a truss structure

5. A structure is also foldable when it has transition links in addition to those that are required to satisfy the previous statement.

These five observations can be translated into a corresponding number of properties of graphs of foldable truss structures. Table 2.2 summarizes these properties necessary for folding onto a plane as well as onto a line. At this point, the properties of graphs of truss structures, as well as of graphs of deployable truss structures, have been presented. In the following two sections a discussion will be presented on the generation of graphs for each of these two categories of truss structures.

2.4 Generation of Graphs of Truss Structures

A graph of a truss structure containing v nodes and e links can be obtained by choosing an appropriate set of e connections from among a maximum number of $v(v-1)/2$ connections between the nodes. The choice of connections can be narrowed down. One of the properties of truss structures is the presence of a loop consisting of all nodes in the structure. Since node numbering is arbitrary, it can be assumed that Node 1 is connected to Node 2, 2 to 3, and so on, until the loop is closed by connecting the last node to the first. This has the advantage that only $e - v$ of the remaining $v(v-3)/2$ possible connections need to be selected.

In the following algorithm, connections between nodes of a truss structure are selected by picking arbitrary locations for non-zero entries in the adjacency matrix. The resulting graph is then checked to determine whether the graph possesses all the properties required for graphs of truss structures, as previously discussed in Section 2.2.

This procedure has been enhanced to minimize the effort involved in the detection of non-unique graphs. These, so called, isomorphic graphs are characterized by being identical to other graphs after renumbering their vertices. However, two

1 The total number of edges e, the number of regular edges e_f (edges representing links of fixed length) and the number of transition edges e_t (edges representing links of variable length) are related as $e = e_f + e_t$.

2 $e_f \leq \gamma n - \tau$, where γ is the stowing parameter and τ is the reduced freedom parameter.

3 $(e_t)_{min} = e - (e_f)_{max}$

4 If $e_t = (e_t)_{min}$ and $e_f = (e_f)_{max}$ then, after eliminating the transition edges, the remaining graph satisfies the requirements as listed in Table 1 for $\alpha = \gamma$ and $\beta = \tau$.

5 If $e_t > (e_t)_{min}$ then a number of transition edges can be changed to regular edges such that property (4) is satisfied.

For folding onto a plane : $\gamma = 2$, $\tau = 3$; folding onto a line : $\gamma = 1$, $\tau = 1$.

Table 2.2, Summary of properties for graphs of deployable truss structures.

vertices of two separate graphs can only be given identical numbers when they are of the same degree. This implies that two graphs can only be isomorphic when they have identical arrangement of degree numbers n_k (the number of vertices of degree k, where k ranges from the dimension parameter α to the number of vertices v in the graph). The following algorithm employs this characteristic by generating all graphs corresponding to a particular arrangement of degree numbers n_k before proceeding with the generation of all graphs corresponding to another arrangement. This has the advantage that isomorphism detection only need to be carried out for graphs that are based on the same arrangement. A detailed discussion on the detection of isomorphic graphs is presented in Chapter 3.

2.4.1 Algorithm A

1 Select dimension parameter α ($\alpha = 3$ for spatial structures , $\alpha = 2$ for planar structures, $\alpha = 1$ for linear structures) and determine the corresponding freedom parameter β ($\beta = 6$, 3 or 1, for $\alpha = 3$, 2 or 1, respectively).

2 Select the number of nodes v and links e in the structure, such that $e \geq \alpha v - \beta$.

3 Based on the selections made in Steps 1 and 2, obtain an arrangement of degree numbers n_k that satisfies :

$$\sum_{k=\alpha}^{v-1} k n_k = 2e \quad and \quad \sum_{k=\alpha}^{v-1} n_k = v \tag{1}$$

If all arrangements have been investigated, then terminate the procedure.

4 Fill in the elements of the adjacency matrix that are associated with the loop in the graph containing v vertices and v edges. Since the numbering of the nodes is arbitrary, it can be assumed that the nodes can be numbered such that Node 1 is connected to Node 2, 2 to 3, etc. until the loop is closed by connecting Node v to Node 1.

5 Pick $e - v$ of the available $v(v-3)/2$ locations in the upper triangle of the adjacency matrix. If all combinations have been investigated then go to Step

3. Make the adjacency matrix symmetric by reflecting the upper triangle about the diagonal.

6 If the degree numbers n_k of the graph resulting from Step 5 do not correspond with the arrangement picked in Step 3, then pick another combination in Step 5.

7 If the graph has more than $\alpha + 1$ vertices and the graph contains a vertex of degree α that is adjacent to another vertex of degree α, then go to Step 4.

8 Accept the graph, go to Step 4.

2.4.2 Example

The following example explains the generation of all graphs of truss structures containing six nodes and twelve links. The example follows the algorithm in step-by-step format.

1, 2 Select $\alpha = 3$, $v = 6$ and $e = 12$. It is noted that, although the selection of v and e corresponds to statically determinate structures, this has no impact on the generality of this example.

3 Based on the selection in Steps 1 and 2, the degree numbers have to satisfy the equalities $2n_5 + n_4 = 6$ and $n_5 + n_4 + n_3 = 6$. Choose $n_5 = 0$, which leads to $n_4 = 6$ and $n_3 = 0$. This means that all graphs following the selection of $n_5 = 0$ must have six vertices of degree four.

4 The loop in the graph is associated with the non-zero positions in the adjacency matrix shown in Fig. 2.5a.

5 Now, six out of the nine available positions in the upper triangle of the adjacency matrix have to be filled with ones. The first choice could result in the adjacency matrix displayed in Fig. 2.5b (the added ones are indicated with circles).

$$\begin{matrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{matrix}$$

(a)

$$\begin{matrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{matrix}$$

(b)

$$\begin{matrix} 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \end{matrix}$$

(c)

$$\begin{matrix} 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \end{matrix}$$

(d)

Fig. 2.5 Adjacency matrices

6 The adjacency matrix of this graph does not correspond to the arrangement determined in Step 3, thus the graph is rejected. Successive operations of Steps 4 and 5 lead to the graph of which the adjacency matrix is displayed in Fig. 2.5c (the circles indicate ones that are added in step 5).

7,8 The graph defined in Fig. 2.5c proves to be acceptable in Step 7, which results in the acceptance of the graph.

7,8 After successive operations of Step 4-7, the graph presented in Fig. 2.5d is encountered. Note that the vertices of this graph can be renumbered such that the graph has the same appearance as the graph found previously (Check : renumber the vertices 1 through 6 as 1, 3, 5, 6, 4, 2).

3 After all possible combinations of non-zero entries in the adjacency matrix have been investigated in successive operations of Steps 4-8 (resulting in additional graphs), the algorithm resumes with Step 3. Now, choose $n_5 = 1$, which leads to $n_4 = 4$ and $n_3 = 1$. The algorithm proceeds as discussed above until all arrangements produced in Step 3 are investigated and the algorithm terminates. Figure 2.6 displays all graphs of truss structures containing six nodes and twelve links.

In this section an algorithm has been presented for the generation of all graphs of truss structures given the number of nodes and links in the structure. It will be shown in the next section that, for each of these graphs, it is possible to obtain a set of graphs of *deployable* truss structures.

2.5 Generation of Graphs of Deployable Truss Structures

As discussed in Section 2.3, a truss structure can be transformed into a deployable truss structure by designating some of its links as transition links. The equivalent operation for a graph is to choose some of the edges of the graph of a

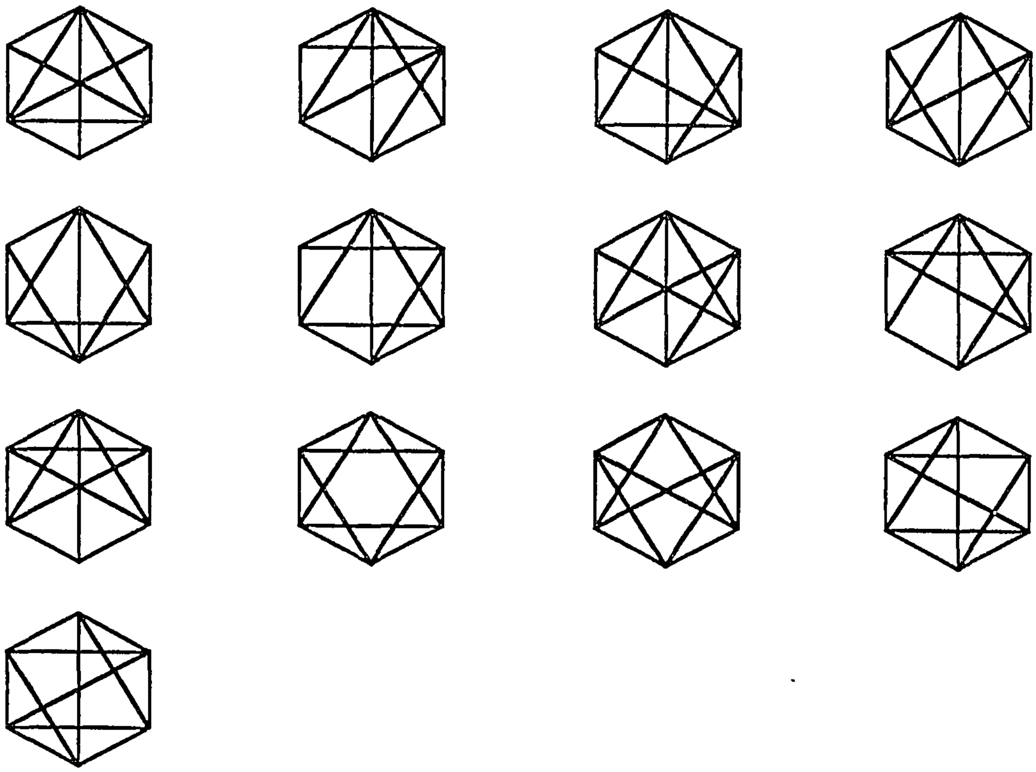


Fig. 2.6 Graphs representing all six-node structures with twelve links

truss structure as edges that represent transition links by displaying them as dashed lines. This choice is not arbitrary but is confined by the requirements set forth in Table 2.2, which summarizes the necessary properties for folding onto a plane and onto a line.

Algorithm B, presented in the following subsection, generates all the deployable truss structures given the number of nodes, total number of links and the number of transition links in the structure. This is achieved by first selecting a minimum number of edges as transition links so that Requirement 4 of Table 2.2 is satisfied, i.e. after the transition links are removed, the graph of the remaining connectivity resembles that of a planar structure when the original structure is folded onto a plane or a linear structure, when the original structure is folded onto a line. Based on this requirement, the minimum number of transition edges is given by:

$$(e_t)_{min} = e + \gamma(2 - v) - 1 \tag{2}$$

where the stowing parameter γ is 2 when the structure is folded onto a plane and γ is 1 when it is folded onto a line. When the conditions of Table 2.2 have been satisfied, the remaining edges representing transition links can then be selected arbitrarily from the edges that have not yet been so designated.

2.5.1 Algorithm B

1 Select the number of nodes v and links e for the structure.

2 Select the stowing parameter γ ($\gamma = 1$: structures are folded onto a line; $\gamma = 2$: structures are folded onto a plane) and determine the reduced-freedom parameter τ (τ =3 or 1 for $\gamma = 2$ or 1 respectively).

3 Determine the minimum number of transition links $(e_t)_{min}$ using the relation :

$$(e_t)_{min} = e + \gamma(2 - v) - 1$$

4 Select the desired number of transition links such that $e_t \geq (e_t)_{min}$.

5 Generate all graphs with v vertices and e edges using Algorithm A (Section 2.4.1).

6 Pick a graph that has been generated in Step 5. If all graphs have been picked, terminate the execution of this algorithm.

7 Pick $(e_t)_{min}$ of the e edges in the graph and designate them to be transition links. If all combinations have been picked return to Step 6, otherwise proceed with Step 8.

8 If the edges that are picked in Step 7 are removed, check that the resulting graph satisfies the requirements in Table 2.1 for $\alpha = \gamma$ and $\beta = \tau$. If these requirements are not satisfied, then return to Step 7, otherwise proceed with Step 9.

9 Pick the remaining $e_t - (e_t)_{min}$ transition links in the graph produced in Step 7. If all combinations are picked, return to Step 7, otherwise accept the graph and repeat Step 9.

2.5.2 Example

The following example is an extension of the example provided in Section 2.4.2 The example follows the algorithm in a step-by-step format. The numbers correspond to the steps in the algorithm.

1,2 Select $v = 6$, $e = 12$ and $\gamma = 2$.

3 The selection dictates that $(e_t)_{min} = 3$.

4 The choice of e_t is arbitrary, select $e_t = (e_t)_{min} = 3$.

5 The generation of graphs of truss structures has been covered by the example in the previous section. The results are shown in Fig. 2.6.

6 Pick the first graph in Fig. 2.6.

7,8 Now, three of the twelve edges have to be designated as transition links. The first choice could result in the graph shown in Fig. 2.7a. However, this graph

is rejected since it does not represent a planar truss structure. Successive operations of Steps 7-8 lead to the acceptable graph shown in Fig. 2.7b.

9 Since $e_t - (e_t)_{min} = 0$, no additional edges have to be picked to represent transition links. The graph shown in Fig. 2.7b remains therefore unchanged.

9 After successive operations of Steps 7-8 the graph is generated that is shown in Fig. 2.7c. Note that the vertices of this graph can be renumbered such that the graph looks identical to the graph in Fig. 2.7b (check : renumber nodes one through six as 1, 2, 5, 4, 3, 6).

6 Successive operations produce all graphs of deployable structures based on the structure picked in Step 6. Figure 2.8 displays all the graphs of deployable truss structures with six nodes and twelve links that are produced by this algorithm for the first graph of Fig. 2.6. The algorithm proceeds with the generation of graphs of deployable structures based on the next graph shown in Fig. 2.6. This goes on until all graphs in Fig. 2.6. have been treated (giving a total of 1962 graphs).

2.6 Results

The algorithms A and B, described in this paper, have been incorporated in a computer program, which was written in FORTRAN. For a given number of nodes, number of links and number of transition links, each separate run produces all the corresponding graphs of deployable structures. Table 2.3 summarizes some of the results produced by this code as well as a listing of existing designs that correspond to the results obtained.

A comparison of the results with graphs of existing designs, shows that the technique provided here is successful in generating all possible graphs of truss structures

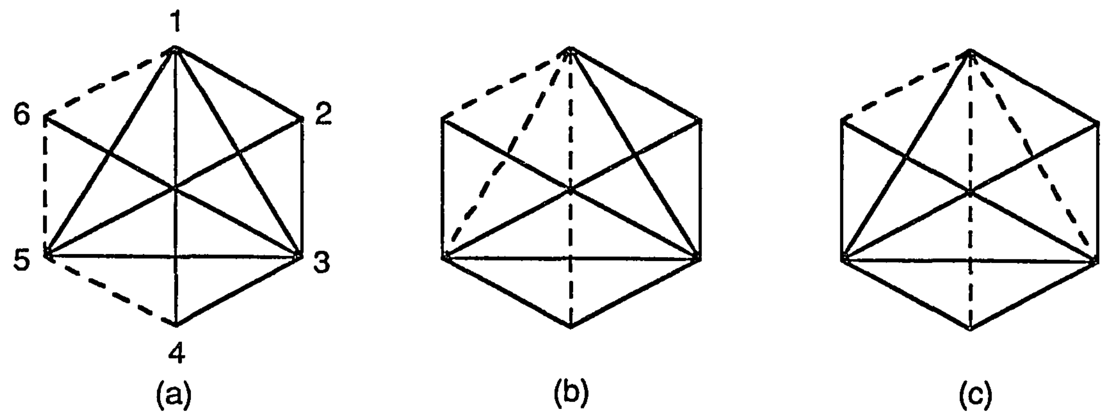


Fig. 2.7 Selection of edges, representing transition links, in a graph of a truss structure

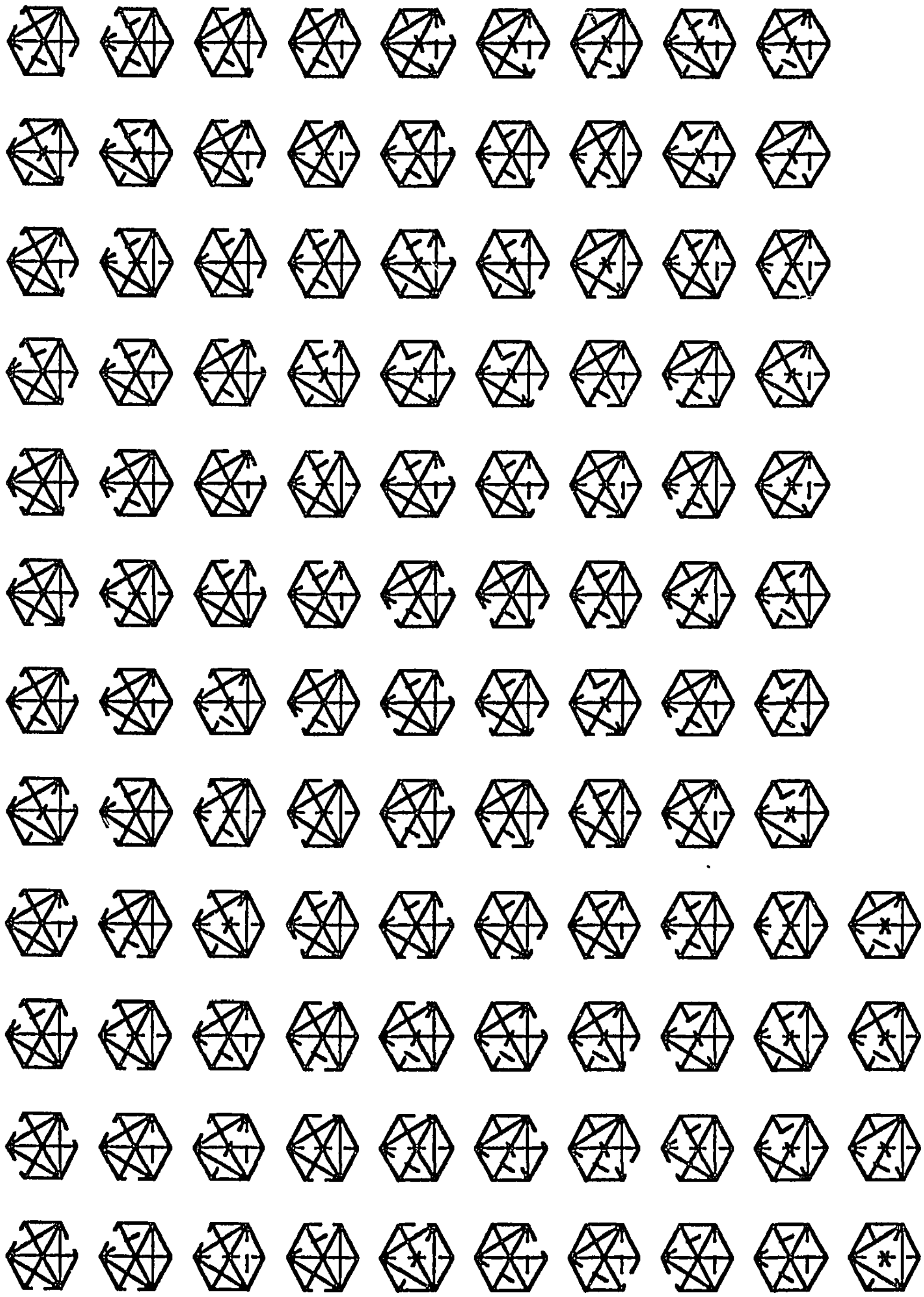


Fig. 2.8 Graphs representing all deployable truss structures that are based on the first graph in Fig. 2.6.

Number of nodes	Number of links	Number of graphs of truss structures	Number of graphs of deployable structures		Existing designs of deployable structures that are foldable to a plane and which are identified among the generated graphs
			Foldable to a plane	Foldable to a line	
4	6	1	6	16	
5	9	1	30	75	
5	10	1	100	125	
6	12	13	1962	4698	(1)
6	13	6	2718	3419	
6	14	2	2524	1744	
6	15	1	3035	1308	(2)
7	15	215	164,209	372,799	
7	16	165	367,339	441,364	
7	17	87	525,738	353,189	
7	18	34	521,444	204,403	
7	19	10	365,910	87,389	
7	20	2	160,610	24,848	
7	21	1	162,599	17,382	

(1) Expandable Truss Module [2]; Controllable Geom. Mod.[10,11]; Atriculated Astromast [14]

(2) Deployable Cell Module [5]; Deployable Mast Module [7]

Table 2.3 Generation of graphs of deployable truss structures, summary

and deployable truss structures. As an illustration, Table 2.3 lists some of the existing designs that have been identified among the graphs produced by the approach. The code also produces graphs of deployable truss structures that are not found in existing designs. At this point it is not yet possible to make a full comparison between the existing designs and the graphs of alternative structures created by the program, since the dimensions of the structure they represent are not yet known. What can be said at this stage is that some of these graphs may indeed result in useful deployable structures that have not been identified. To identify these structures, some relative dimensions of the various links are needed. These relative dimensions will be generated in Chapter 4 for a particular class of truss structures called truss modules. Such additional information will then enable the identification of possible alternatives to existing designs.

It should be noted that many of the graphs shown in Figs. 2.6 and 2.8 are isomorphic, i.e. represent the same inter-connectivity of the elements. Since each isomorphic graph will result in a duplicate conceptual design produced in Chapters 3-6, such graphs must be eliminated before the conceptual design process is continued. This topic will be discussed in the following chapter.

CHAPTER 3

DETECTING ISOMORPHISM IN GRAPHS OF DEPLOYABLE-FOLDABLE TRUSS STRUCTURES

3.1 Outline

The most efficient method for isomorphism detection to date uses matrices, each revealing the interconnectivity of components of a graph, and determines the characteristic equation for these matrices [38]. Two graphs are then considered isomorphic when the characteristic equations of their matrices are identical. A variant to this approach is to pick a random number, add that random number to the diagonal terms of each matrix and calculate the determinants of these matrices. Two graphs are then considered isomorphic when the determinants of their matrices are identical (this approach is equivalent to substituting the random number into the characteristic equation, but results can be achieved with much greater efficiency). Although this "determinant method" does not guarantee that all non-isomorphic graphs are found, it has been applied in one form or another in previous investigations [38,41-43].

This chapter introduces a procedure to test pairs of graphs for isomorphism that is more accurate than the determinant method. The procedure is based on vertex degree correspondence, a technique that was introduced by Unger [33]. Others, such as Bohm and Santolini [34] and, Corneil and Gotlieb [36], have also presented

variations on this technique to the detection of isomorphic graphs, but did not improve the inefficient techniques used in the original approach. The present procedure, however, extends the degree correspondence of vertices to sets of vertices, thereby achieving a substantial increase in efficiency, which makes the method suitable for treating large numbers of graphs. The basic technique of the procedure presented here is discussed in the following sections.

3.2 Isomorphic Graphs

A graph is defined as a set of points, called vertices, v, that are connected by a set of lines, called edges, e. An example of a graph is given in Fig. 3.1. A graph representation of an object can be obtained by establishing relationships between certain elements of the object corresponding to the vertices and edges of its graph. In mechanisms design, for example, links are often displayed as vertices and joints as edges. Figure 3.2 illustrates how this definition can be used to represent the kinematic structure of a variable-stroke engine mechanism by a graph.

One way to describe a graph is to assign distinct numbers to its vertices. For an arbitrary graph with n vertices there are a total of $(n!)$ different numberings possible, which theoretically can result in as many different looking graphs. Given any graph of this set, a renumbering scheme can be found for every other graph in this set which makes the graph identical in appearance to the first graph. The presence of such a scheme is absolute proof that two graphs are isomorphic. However, the process for finding such a scheme can be cumbersome. As an illustration, consider the three graphs shown in Fig. 3.3. Only two of these graphs, A and C, are isomorphic. Even when this information is given, it is not trivial to determine the renumbering scheme that shows the graphs to be identical.

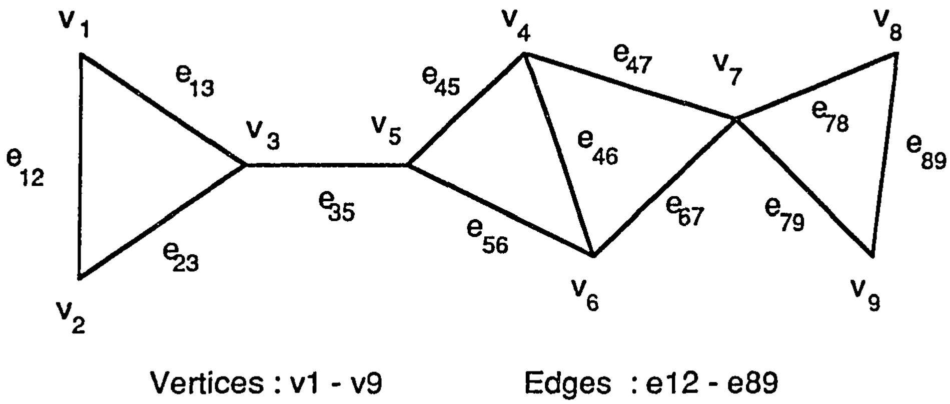


Fig. 3.1 Example of a graph

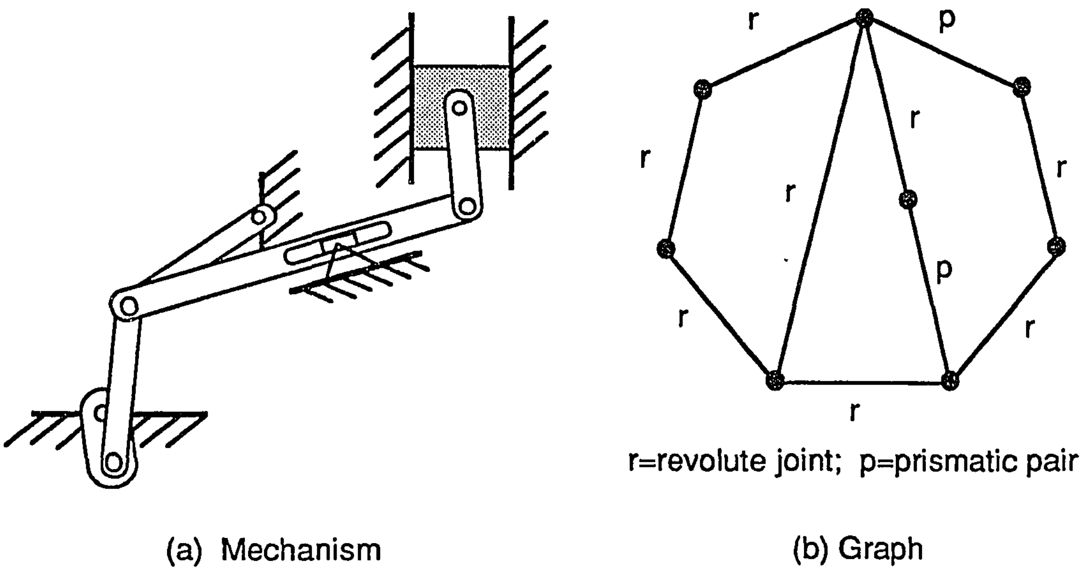
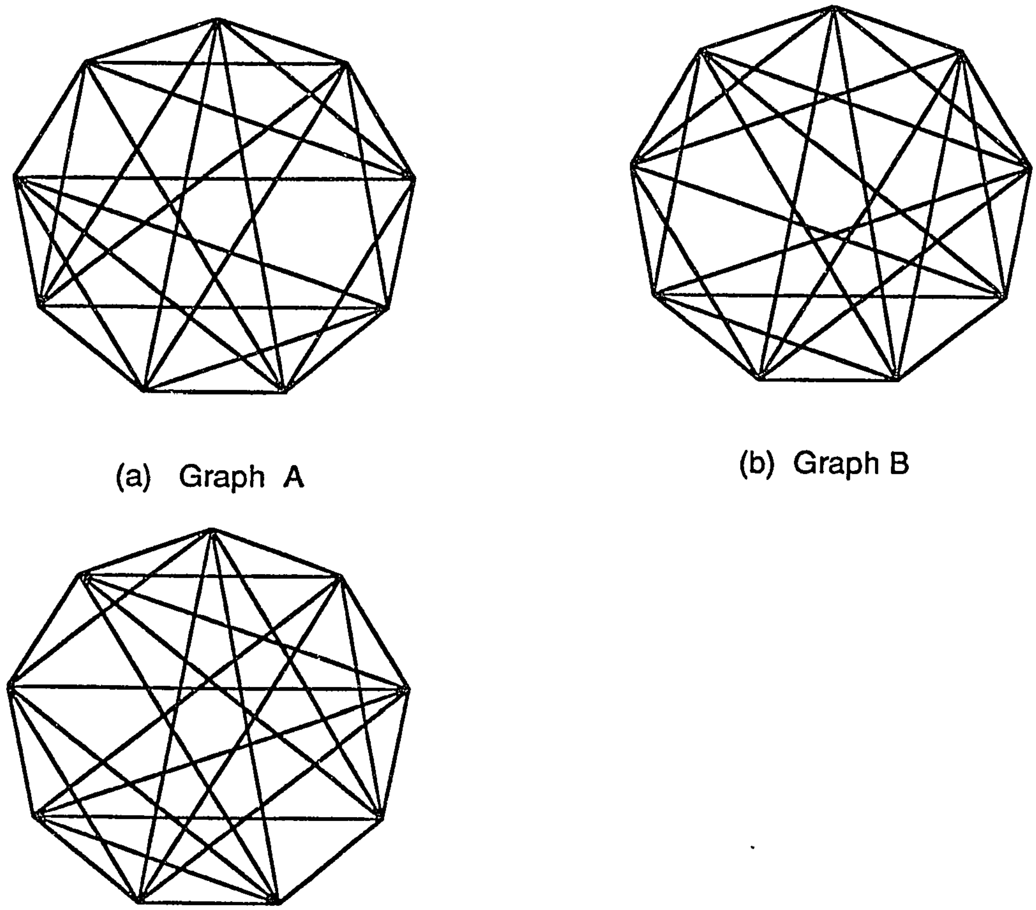


Fig. 3.2 A mechanism and the graph of its kinematic structure



(a) Graph A

(b) Graph B

(c) Graph C

Fig. 3.3 Three graphs of which all vertices are of degree six

Before a discussion of the properties of isomorphic graphs is given, some definitions are in order. Let the total number of vertices in the graph be denoted by n. An edge connecting Vertex v_i to Vertex v_j is indicated as e_{ij} or (v_i, v_j). Vertices v_i and v_j are said to be adjacent when there exists an edge (v_i, v_j). An adjacency list for a vertex is defined as a list of vertices that are adjacent to it. The degree of a vertex is equal to the number of vertices adjacent to this vertex. The degree number n_k is defined as the number of vertices in the graph of degree k. The Set s_a is defined as a set of vertices of Graph A, so that the Graph $(A\text{-}s_a)$ is defined as a graph formed by eliminating the vertices in Set s_a from Graph A, including all edges from Graph A that are incident to the vertices in this set. Using this notation, isomorphism may be defined as [40]:

> Two graphs, A and B, are isomorphic if and only if there is a one-to-one correspondence between the vertices of Graph A and the vertices of Graph B such that the number of edges joining any two vertices in Graph A is equal to the number of edges joining the corresponding two vertices in Graph B.

This definition will be used in the following section to derive the basic techniques for a procedure that tests graphs for isomorphism.

3.3 The Approach

Close examination of the definition of graph isomorphism leads to the following observations:

Observation 1 :

If Graphs A and B are isomorphic, then a vertex of Graph A can only correspond to a vertex of Graph B that is of the same degree.

Observation 2 :

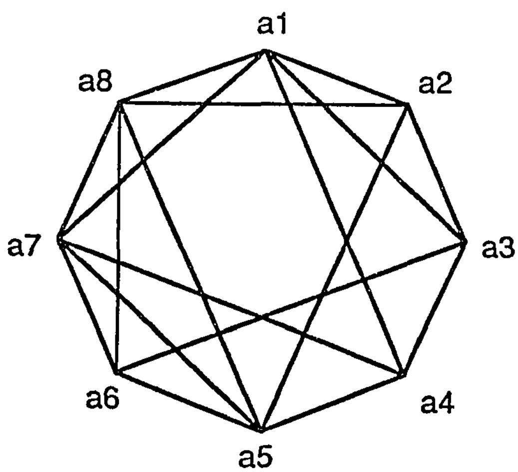
A necessary (but not sufficient) condition for isomorphism is that both graphs have the same number of vertices of a particular degree. This means that two graphs can only be isomorphic when the degree numbers n_k, with $k = 1, 2, ..., n-1$, are identical for both graphs.

Observation 3 :

If Graphs A and B are isomorphic and Set s_a is a set of all vertices of Graph A of degree k and Set s_b is the corresponding set of vertices of Graph B, then the two subgraphs induced by Set s_a and Set s_b are isomorphic, the Subgraphs (A-s_a) and (B-s_b) are isomorphic; and the subgraphs formed by the edges between the vertices of Graph (A-s_a) and Set s_a; and the subgraph formed by the edges between the vertices of Graph (B-s_b) and Set s_b are isomorphic.

These observations can now be applied to determine whether a pair of graphs is isomorphic. Consider two arbitrary graphs, A and B, which are shown in Fig. 3.4. Since both graphs have four vertices of degree four and four vertices of degree five (i.e. the graphs have identical degree numbers), it is possible that Graphs A and B are isomorphic (see Observation 2). Assuming that Graph A is isomorphic to Graph B, then it must be possible to renumber the vertices of Graph A so that their appearance is identical. Since the graph has eight vertices, there are a total of 8!=40320 possible renumbering schemes. However, according to Observation 1, a vertex of Graph A can only correspond to a vertex of Graph B of the same degree. Hence, the number of possible renumbering schemes reduces to 4!*4!=576, since both graphs have two sets of four vertices of the same degree.

Observation 3 can be used to reduce the number of possible renumbering schemes even further by using the following technique. Let Set s_a be the set of all vertices of Graph A of degree five and Set s_b the set of all vertices of Graph B of degree five. Since each vertex in Set s_a must correspond to one vertex in Set



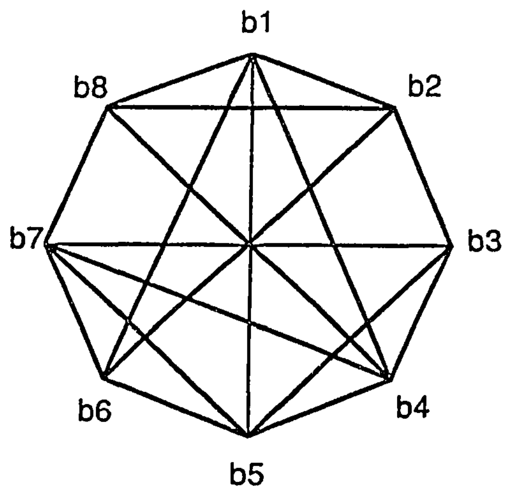


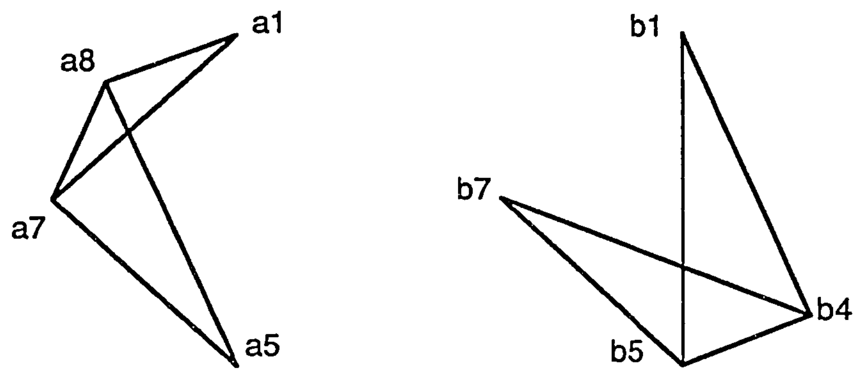
Fig. 3.4 Graphs treated in the example of Sections 3.3 and 3.4.2

s_b (Observation 1), the subgraphs of Graphs A and B induced by the vertices of Sets s_a and s_b respectively, see Fig. 3.5a, must be isomorphic. Also, each vertex of Graph A that is not in Set s_a must correspond to a vertex in Graph B that is not in Set s_b, i.e. the Subgraphs (A-s_a) and (B-s_b), see Fig. 3.5b, must be isomorphic. Furthermore, the subgraphs formed by the edges in Fig. 3.4 that are not represented in Figs. 3.5a and 3.5b (see Fig. 3.5c), must also be isomorphic.

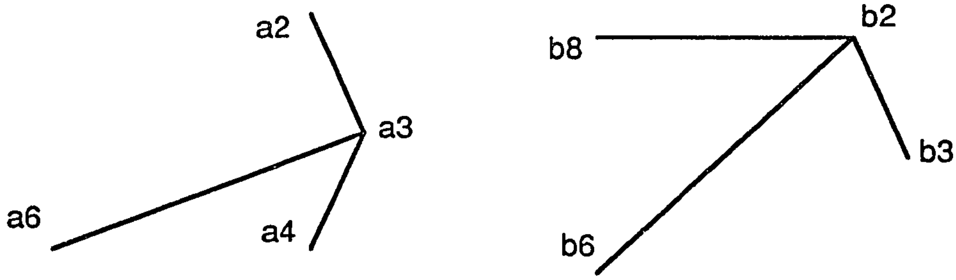
Observations 1 and 2 can now be applied to determine whether the subgraphs in Figs. 3.5a, 3.5b and 3.5c could be isomorphic, i.e. the n_k numbers must be identical for each pair of subgraphs. To obtain the degree of each vertex in each of the subgraphs (and thereby the n_k numbers), it is not necessary to construct the graphs in Figs. 3.5a, 3.5b and 3.5c. For example, Vertex a_1 in Graph A is connected to two vertices of the same degree (degree five), which means that Vertex a_1 is of degree two in Fig. 3.5a and of degree three in Fig. 3.5c. Similarly, Vertex a_2 in Graph A (of degree four) is connected to three vertices of degree five, so that Vertex a_2 is of degree one in Fig. 3.5b and of degree three in Fig. 3.5c.

A comparison of the degree of each vertex in each of the subgraphs, resulting from the separation of Sets s_a and s_b from Graphs A and B respectively, reveals possible correspondences between the vertices of Graphs A and B, e.g. Vertex a_1 can only correspond to Vertex b_1 or b_7 (see Fig. 3.5a), whereas Vertex a_3 can only correspond to Vertex b_2 (see Fig. 3.5b). All possible correspondences for each vertex of Graph A are listed in Fig. 3.6b, which shows that the number of possible renumbering schemes has reduced to 24.

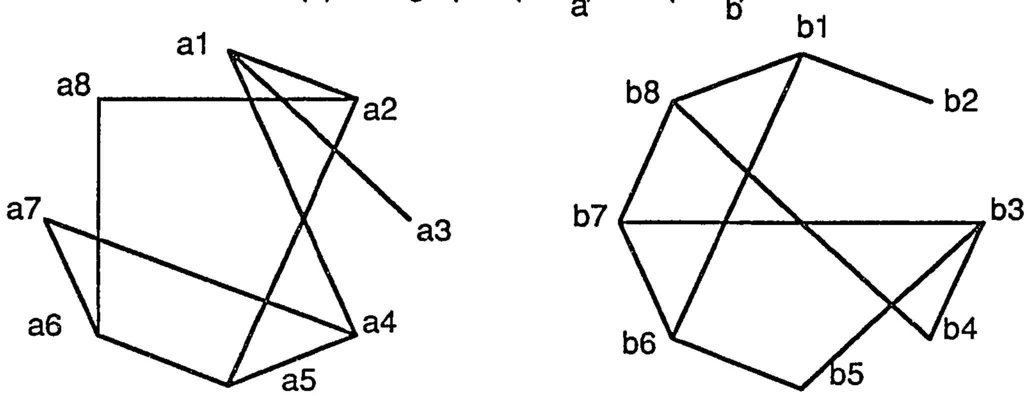
The remaining possible renumbering schemes must be tested to find absolute proof that one of the schemes transforms Graph A into Graph B. This can be achieved very efficiently, i.e. there is no need to investigate all possible renumbering schemes. The details of the procedure that searches for a valid renumbering scheme are discussed as part of a procedure for isomorphism detecting, given in the next section.



(a) Subgraphs induced by Set s_a and Set s_b



(b) Subgraphs $(A-s_a)$ and $(B-s_b)$



(c) Subgraphs involving edges of Graphs A and B in Fig. 4 that are not represented in (a) or (b).

Fig. 3.5 Subgraphs resulting from the separation of all vertices of degree five from Graphs A and B in Fig. 3.4.

Vertex		List
a1	→	b1, b4, b5, b7
a2	→	b2, b3, b6, b8
a3	→	b2, b3, b6, b8
a4	→	b2, b3, b6, b8
a5	→	b1, b4, b5, b7
a6	→	b2, b3, b6, b8
a7	→	b1, b4, b5, b7
a8	→	b1, b4, b5, b7

(a) C-lists before a comparison is made between sets of vertices

Vertex		List
a1	→	b1, b7
a2	→	b3, b6, b8
a3	→	b2
a4	→	b3, b6, b8
a5	→	b1, b7
a6	→	b3, b6, b8
a7	→	b4, b5
a8	→	b4, b5

(b) C-lists after a comparison is made between sets of vertices

Fig. 3.6 Correspondence lists for vertices of Graph A in Fig. 3.4

It is noted that the example described above concerns two isomorphic graphs. However, the conclusion that two graphs are not isomorphic is often reached without checking all the possible renumbering schemes. For example, if one graph has four vertices of degree five and another graph has only two vertices of degree five, then the two graphs cannot be isomorphic (Observation 2). Similarly, if the subgraph induced by vertices in Set s_a contains three vertices of degree one and the subgraph induced by vertices in Set s_b contains two vertices of degree one, then the two graphs cannot be isomorphic either.

The approach described in this section is the basis of a procedure for the detection of graph isomorphism, which is discussed in detail in the next section.

3.4 Procedure for the Detection of Isomorphic Graphs

The techniques, described in Section 3.3, have been incorporated in the following algorithm (DIG). The input into the algorithm consists of a given pair of graphs, A and B. The first step of the procedure checks whether the two graphs have the same number of vertices of a particular degree (a condition for isomorphism, see Observation 2 in Section 3.3). In Step 2 of the procedure, a list is formed for each vertex of Graph A which contains all the vertices of Graph B that are of the same degree. Since each vertex in such a list constitutes a possible one-to-one correspondence with the vertex of Graph A for which the list is formed, it will be referred to as a *correspondence list*. In Steps 3 and 4, sets of vertices of the same degree are separated from Graphs A and B and the resulting subgraphs are compared for isomorphism, which in most cases results in a reduction of the correspondence lists. In the last step of the procedure, Step 5, an attempt is made to find a valid renumbering scheme, based on the correspondence lists, that transforms Graph A into Graph B.

Isomorphic graphs go through all steps of the procedure, whereas the conclusion that the graphs are not isomorphic can be reached at intermediate steps. However, there is one class of graphs that always reaches the last step of the procedure. These graphs have the characteristic that all vertices are of the same degree. Steps 3 and 4 are skipped for this type of graph, since they do not reduce the correspondence lists, no matter what the connectivity of the graph is.

The complete algorithm is listed in the following subsection. An example that illustrates the algorithm is provided in a subsequent subsection.

3.4.1 Algorithm DIG

Given two graphs, A and B, this algorithm checks whether the graphs are isomorphic using the vertex correspondence technique.

1. Check that both graphs have the same degree numbers n_k, where $k = 1, 2, ..., n-1$ (see definitions in Section 3.2). If this is not true, then report that the graphs are not isomorphic and halt the execution. Otherwise proceed with Step 2.
2. For every vertex of Graph A produce a list of all the vertices of Graph B that have the same degree. This list is called a correspondence list or C-list. Proceed with Step 5 if all vertices of A and B are of the same degree, otherwise continue with Step 3.
3. Pick the smallest set of all vertices of Graph A of the same degree : Set s_a, and the corresponding set of Graph B : Set s_b. Compute the degree of each vertex in the subgraphs induced by the vertices in Set s_a, and the subgraph induced by the vertices in Set s_b, as well as the degree of each vertex in the Subgraphs (A-s_b) and (B-s_b). Use the following technique : for each edge (v_a,w_a), where Vertex v_a is part of Set s_a and Vertex w_a is part of Subgraph (A-s_a), reduce the degree of Vertex v_a and the degree

of Vertex w_a by one. As a result, the degree of each vertex reflects the degree of that vertex in either Subgraph (A-s_a) or the subgraph induced by the vertices in Set s_a. Repeat this operation for the vertices in Graph B. Proceed with Step 4.

4. For each Vertex v_a in graph A, eliminate all vertices of Graph B from the C-list of Vertex v_a that are not of the same degree as Vertex v_a in the subgraphs created in Step 3. If the C-list for Vertex v_a becomes empty, then report that the graphs are not isomorphic and halt the execution. Otherwise, let Graph A^r stand for Graph (A-s_a), let Graph B^r stand for Graph (B-s_b) and repeat Step 3 for Graphs A^r and B^r. If Graphs A^r and B^r are empty, continue with Step 5.

5. Select a possible renumbering scheme as follows : pick a Vertex v_a of Graph A and pick a Vertex v_b of Graph B from the C-list of Vertex v_a. Assume that Vertex v_a corresponds to Vertex v_b and check that if there is an edge between Vertex v_a and vertices of Graph A that have been picked prior to Vertex v_a, that this edge also exists between the corresponding vertices of Graph B. In the event that none of the vertices picked for Vertex v_a are accepptable, one must back up and repeat Step 5 while picking another correspondence for the vertex of Graph A picked prior to Vertex v_a. When an acceptable correspondence for Vertex v_a is found, then the search for a valid renumbering scheme continues with Step 5 for the next vertex of Graph A. This process continues until a valid correspondence is found for the last vertex of Graph A (i.e. the graphs are isomorphic) or until no new correspondence can be picked for the first vertex picked of Graph A (i.e. the graphs are not isomorphic).

3.4.2 Example

As an example, the previous algorithm is applied to test the graphs, shown in Fig. 3.4. These graphs were chosen because they challenge the algorithm and therefore provide a good illustration of its effectiveness.

1-2 Graph A and Graph B each have four vertices of degree five and four vertices of degree four, so that all degree numbers are zero except that $n_4 = 4$ and $n_5 = 4$. Figure 3.6a shows the C-lists produced in Step 3.

3 Let Set s_a be all vertices of Graph A that are of degree five and let Set s_b be all vertices of Graph B of degree five. The degree of each vertex in the subgraph induced by the Set s_a and the Subgraph (A-s_a), is obtained as follows. Vertex a_1, of degree five, is adjacent to only two vertices that are not of degree five, namely Vertices a_2 and a_3. Thus, the degree of Vertex a_1 is reduced with two and the degrees of Vertices a_2 and a_3 is reduced with one. Similarly, Vertex a_5, of degree five, is adjacent to two vertices that are not of degree five, Vertices a_2 and a_4. Therefore, the degree of Vertex a_5 is reduced to three, whereas the degrees of Vertices a_2 and a_4 is reduced with one. A similar operation is carried out for Vertices a_7 and a_8. As a result, the degree of each vertex reflects the degree of that vertex in Fig. 3.5a or 3.5b. Subsequently, the previous operation is repeated to determine the degree of each vertex in Subgraph (B-s_b) and the subgraph induced by Set s_b. Figure 3.7 summarizes the results produced by Step 3. Note that the difference between the degree of a vertex before and after this operation (Lines 1 and 2) is the degree of that vertex in Fig. 3.5c.

4 It can be concluded from Fig. 3.7a that Vertex a_1 has a "5" on Line 1, a "2" on Line 2 and a "3" on Line 3, i.e. Vertex a_1 is of degree five in Graph A; degree two in one of the subgraphs; and degree three in the subgraph containing all the edges "eliminated" in Step 3. The only two

Vertices	a1	a2	a3	a4	a5	a6	a7	a8	Line
Degree in Graph A	5	4	4	4	5	4	5	5	1
Degree in Subgraph (A-s_a) or in the subgraph induced by Set s_a	2	1	3	1	2	1	3	3	2
Degree in Graph A^r	0	1	3	1	0	1	0	0	3
Degree in Subgraph (A^r-s_a) or in the subgraph induced by Set s_a	0	0	0	0	0	0	0	0	4

(a) The degree of each vertex of Graph A and subgraphs of Graph A

Vertices	b1	b2	b3	b4	b5	b6	b7	b8	Line
Degree in Graph B	5	4	4	5	5	4	5		1
Degree in Subgraph (B-s_b) or in the subgraph induced by Set s_b	2	3	1	3	3	1	2	1	2
Degree in Graph B^r	0	3	1	0	0	1	0	1	3
Degree in Subgraph (B^r-s_b) or in the subgraph induced by Set s_b	0	0	0	0	0	0	0	0	4

(b) The degree of each vertex of Graph B and subgraphs of Graph B

Fig. 3.7 Illustration of the example discussed in Section 3.4.2

vertices of Graph B that have the same numbers on Lines 1-2 in Fig. 3.7b (and can therefore correspond to Vertex a_1) are the Vertices b_2 and b_7. Thus, Vertices b_4 and b_5 must be eliminated from the C-list of Vertex a_1. Carrying out the previous operation for each vertex of Graph A results in the C-lists listed in Fig. 3.6b.

3 Line 3 in Fig. 3.7a reflects the degree of each vertex in Graph A^r. Note that the numbers on Line 3 are obtained from Line 2 and that the degree of each vertex of Set s_a picked in the previous execution of Step 3 is set to zero. The same operation was carried out to obtain the degree for each vertex of Graph B^r, displayed on Line 3 in Fig. 3.7b. The execution of Step 3 results in the selection of Vertex a_3 as Set s_a and Vertex b_2 as Set s_b. The degree of each vertex of Graphs A^r and B^r in the subgraphs formed by Set s_a, Set s_b; the Subgraphs (A^r-s_a) and (B^r-s_b) are displayed on Lines 4 in Fig. 3.7.

4 Comparison of the numbers on Lines 4 in Fig. 3.7 does not alter the C-lists in Fig. 3.6b and the procedure continues with Step 5 since Graphs A^r and B^r are empty (Lines 5 in Fig. 3.7 contain only zero entries).

5 Pick a possible correspondence for the first vertex of Graph A, e.g. assume that a_1 corresponds to b_1 (i.e. the first entry of the list for Vertex a_1 in Fig. 3.6b). Next, pick a possible correspondence for the second vertex of Graph A, e.g. assume that Vertex a_2 corresponds to Vertex b_3 (see Fig. 3.6b). Now, check whether Vertex a_2 is connected to any vertices of Graph A picked earlier, and if so, check whether the corresponding vertices are also connected. For example, Vertices a_2 and a_1 are connected. However, since b_3 and b_1 are not connected, either the correspondence between Vertex a_1 and Vertex b_1 is invalid or the correspondence between Vertex a_2 and Vertex b_3 is invalid. Assume that the latter is the case, pick another correspondence for Vertex a_2 say, Vertex b_6. Since Vertices b_6 and b_1

are connected, this correspondence is accepted. Step 5 continues with picking a correspondence for Vertex a_3, e.g. Vertex b_2 (the only possible choice). Since Vertex a_3 is connected to both Vertices a_1 and a_2 (treated prior to Vertex a_3), Vertex b_2 must also connect to Vertices b_1 and b_6. This is indeed the case, so that Step 5 can be continued with picking a correspondence for Vertex a_4, e.g. Vertex b_3. However, this choice must be rejected since Vertex a_4 is connected to Vertex a_1 while Vertices b_3 and b_1 are not connected. The next choice, Vertex b_6, has been picked previously so that Vertex a_4 must correspond with Vertex b_8. This choice proves to be correct and the Step 5 continues. After several executions of Step 5, the following renumbering scheme is established : $a_1 \rightarrow b_1$, $a_2 \rightarrow b_6$, $a_3 \rightarrow b_2$, $a_4 \rightarrow b_8$, $a_5 \rightarrow b_7$, $a_6 \rightarrow b_3$, $a_7 \rightarrow b_4$ and $a_8 \rightarrow b_5$.

The previous example has shown that the algorithm has arrived at the conclusion that the graphs are isomorphic without much computation, while the renumbering scheme that transforms Graph B into A comes as a by-product. The following section provides a general discussion on the performance of this algorithm.

3.5 Discussion and Results

Close examination of the algorithm provided here leads to the conclusion that it is most likely that the algorithm will be least efficient for graphs of which all vertices are of the same degree. An example of such graphs is shown in Fig. 3.3. The algorithm has determined that Graphs A and C are isomorphic and that Graph B is not isomorphic to any of the other graphs. It should be noted, however, that graphs as in Fig. 3.3 are a minority among the complete set of graphs with nine vertices and 27 edges, as created by the procedure listed above. Furthermore, a set of graphs does not contain any graphs of which all vertices have the same degree if

the result of $2n_e/n_v$ is not an integer. It should be noted, however, that this result is only an integer for two-, three- and six-node statically determinate (deployable) truss structures. Since the number of graphs of these classes of structures, using the techniques presented in Chapter 2, is relatively small, no special attention will be given to the extra CPU time required for graphs of which all vertices are of the same degree.

The DIG-algorithm (listed in Section 3.4.1) has been applied to identify all unique graphs among sets of graphs of deployable truss structures with four, five, six and seven nodes that were generated using the algorithms listed in Chapter 2. Figures 3.8 and 3.9 show all unique graphs of six-node statically determinate truss structures that are foldable onto a plane, which were identified by the DIG-algorithm among the graphs shown in Figs. 2.6 and all deployable derivatives of the graphs in Fig. 2.6, a subset of which is shown in Fig. 2.8. Results of the application of the DIG-algorithm are summarized in Table 3.1.

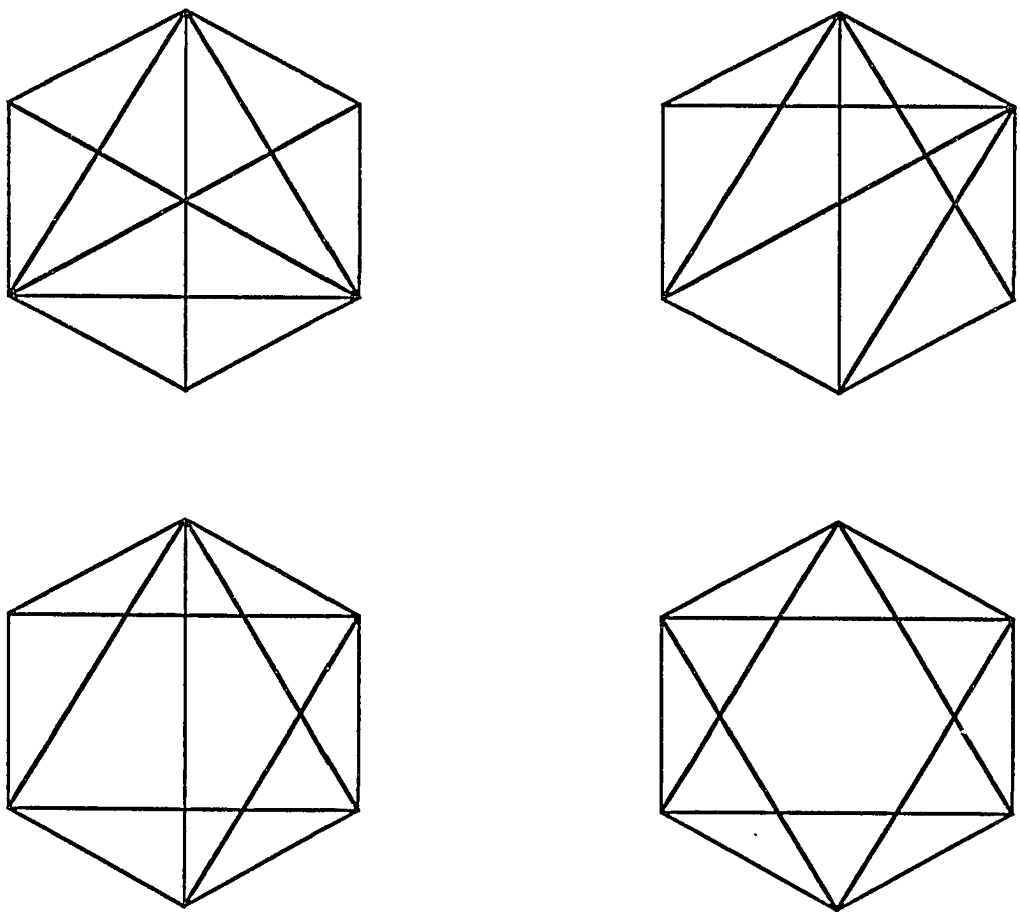


Fig. 3.8 Graphs representing all six-node structures with twelve links

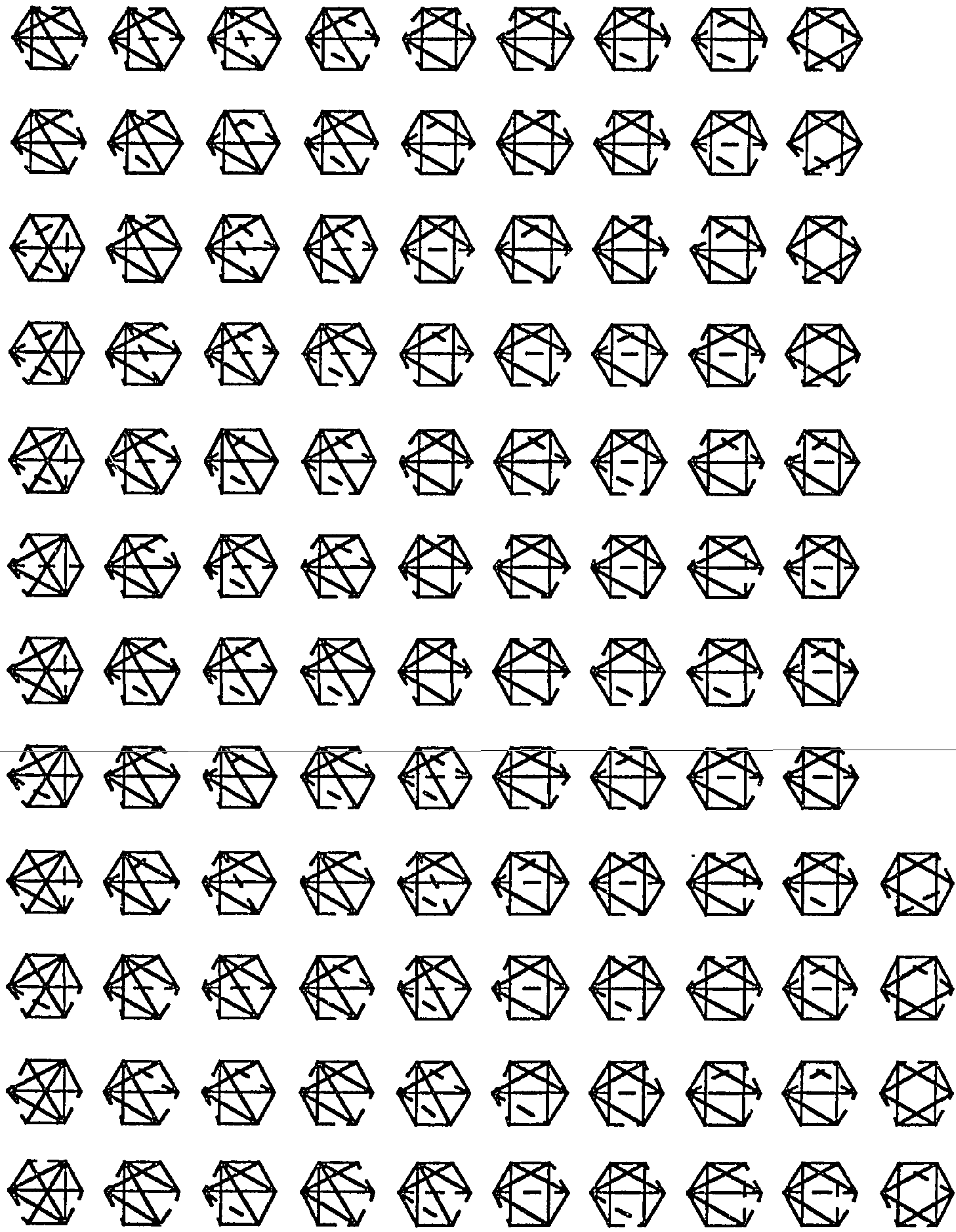


Fig. 3.9 All graphs of deployable structures with six nodes, twelve links and three (the minimum number) transition links.

Number of nodes	Number of links	Number of graphs of truss structures	Number of graphs of deployable structures		Number of unique graphs of truss structures	Number of unique graphs of deployable structures	
			Foldable to a plane	Foldable to a line		Foldable to a plane	Foldable to a line
4	6	1	6	16	1	1	2
5	9	1	30	75	1	5	9
5	10	1	100	125	1	3	3
6	12	13	1962	4698	4	112	219
6	13	6	2718	3419	2	101	96
6	14	2	2524	1744	1	27	43
6	15	1	3035	1308	1	6	13
7	15	215	164,209	372,799	25	7132	35467
7	16	165	367,339	441,364	17	9837	52785
7	17	87	525,738	353,189	9	6547	23893
7	18	34	521,444	204,403	5	3678	11674
7	19	10	365,910	87,389	2	1073	3289
7	20	2	160,610	24,848	1	316	712
7	21	1	162,599	17,382	1	29	53

Table 3.1 Identification of unique graphs of deployable truss structures, summary

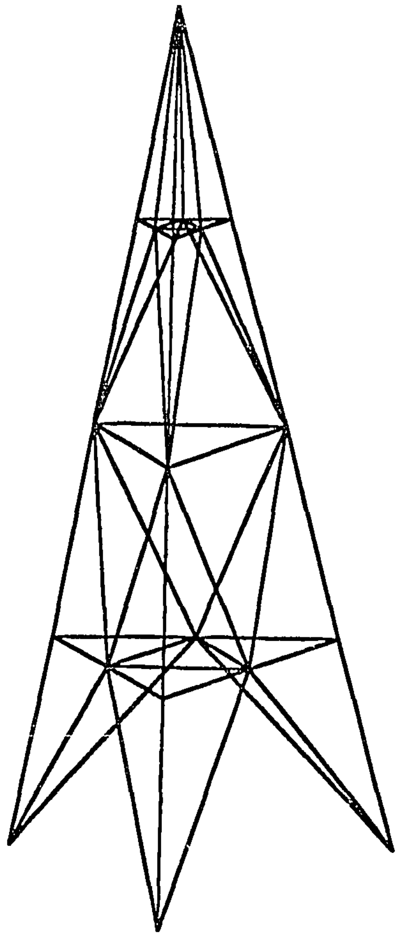
CHAPTER 4

GENERATION OF DEPLOYED CONFIGURATIONS OF DEPLOYABLE-FOLDABLE TRUSS MODULES

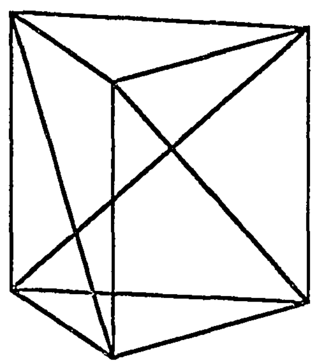
4.1 Outline

A graph theoretical approach, presented in Chapters 2 and 3, was applied to the generation of all graphs representing deployable truss structures, each graph representing a unique deployable truss structure. In this chapter, the conceptual design process is further refined for an important class of spatial truss structures, called truss modules. Modules are the commonly used building blocks of large truss structures such as domes, booms and space-based platforms. In such applications, most truss modules have special properties, such as the presence of an interface to an adjacent module, in addition to those of truss structures. However, not all truss structures have these additional properties. Truss modules may therefore be considered a subset of truss structures. As an example, Fig. 4.1 shows two truss structures of which only the truss structure shown in Fig. 4.1b belongs to the set of truss modules since this second structure may be used as a building block for larger structures.

It will be shown in this chapter that graphs of truss modules can be recognized among graphs of truss structures by checking each graph for characteristics corresponding to graphs of truss modules. These characteristics will be identified and



(a) A truss structure



(b) A truss structure that is also a truss module

Fig. 4.1 Illustration of truss modules being a subset of truss structures

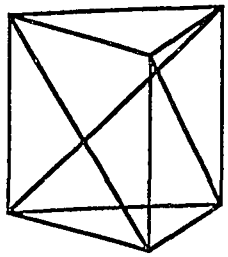
discussed in Sections 4.2 and 4.3. They lead to the formulation of two procedures. The first procedure checks for characteristics related to the functions of a module. The second procedure checks for characteristics associated with the assembly of a module and then generates plots of all possible topologies of truss modules that are based on a particular graph. Both of these procedures are applicable to graphs of truss structures and to graphs of deployable truss structures. The properties of truss modules are presented in the following section.

4.2 Properties of Truss Modules

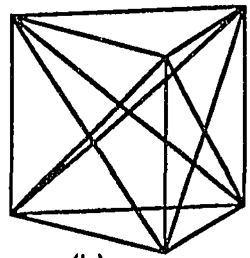
Figure 4.2 shows a selection of truss modules that were found in prior literature [1-15]. At least one deployable concept has been published for each of the structures shown in this figure. References to the source of these concepts are indicated. In this section a discussion will be presented on each of the properties common to all truss modules. This will provide the necessary information to establish the relationship between properties of this type of structure and the properties of their corresponding graphs. It should be emphasized that, in compliance with the restriction made in Chapter 2 regarding truss structures, this discussion will be limited to truss modules that do not use cables.

Truss modules can be used as building blocks for larger spatial truss structures. Therefore, an obvious characteristic of truss modules is that they have all the properties that correspond to spatial truss structures. However, in the discussion that follows, it will be shown that not all spatial truss structures have all the properties that correspond to truss modules, which means that truss modules form a subset of the set of spatial truss structures.

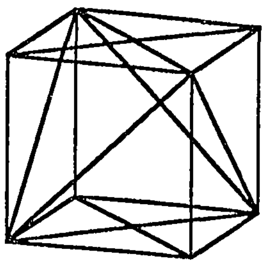
Truss modules are generally combined to form either one-dimensional structures (e.g. mast or arc) or two-dimensional structures (e.g. platform or dome). The



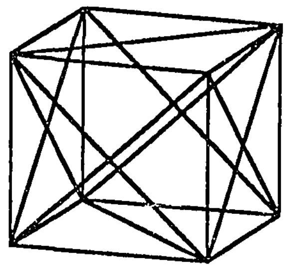
(a)
Coilable Astromast [1]
Expandable Truss Module [2]
Stacbeam [8]
Controllable Geometry Module [10]
Variable Geometry Truss Module [11]
Articulated Astromast [14]



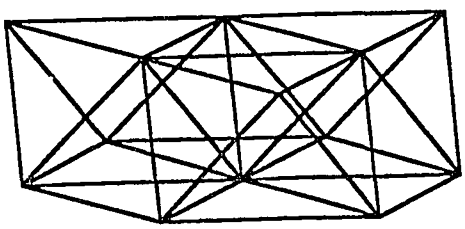
(b)
Deployable Cell Module [5]
Deployable Mast Module [7]



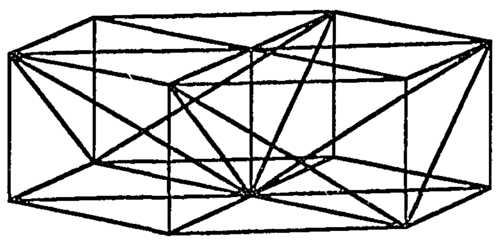
(c)
Box Truss Structure [6]
Deployable Truss Boom [12]



(d)
Single Hinge Double Fold [13]



(e)
Tetrahedral Truss Module [3]



(f)
Triangular Cell [15]

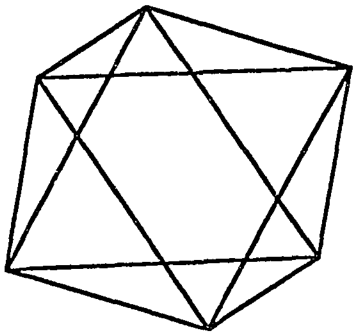
Fig. 4.2 Truss modules reported in the literature

nodes of a truss module lie in two parallel planes, each plane containing roughly half the number of nodes in the structure. These planes form either interfaces between adjacent modules (to form one-dimensional structures) or desired surfaces (to form two-dimensional structures). This maximizes the contribution of each module to either the total area of a two-dimensional structure or the total length of a one-dimensional structure.

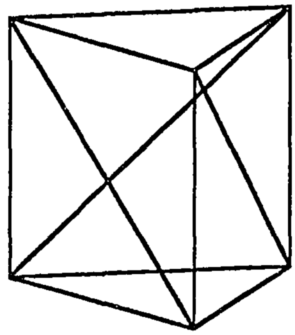
Each plane of a truss module forms a planar truss structure, which means that the n nodes and the l links of <u>each</u> plane satisfy the relationship $l \geq 2n - 3$. A derivation of this relationship has been provided in Section 2.2 of Chapter 2. To minimize the total length of the links in each plane, the links are arranged such that each plane is composed of triangles. Also, each node must be connected to at least one node that is not within the same plane. This last requirement results from the definition of a truss structure, which states that each node must be restricted in all three components of translation.

To limit the number of different structural components, truss modules are typically constructed of links of *at most* two different lengths. This restriction will be adopted for the purposes of this investigation. The presence of this characteristic is generally an indication that the connectivity of a structure is such that it can be built out of links of three or more different lengths without dramatically changing the appearance and therefore the design of the structure. An example is illustrated in Fig. 4.3, where slightly different truss modules are obtained when they are constructed of single, two or three link-lengths.

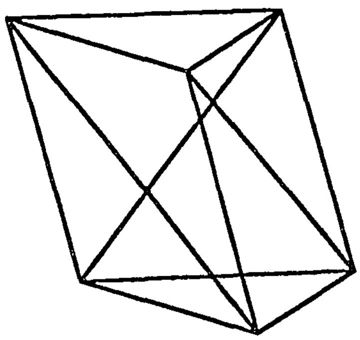
Truss modules do not have crossing links. These are links that cross each other on the same plane. To accommodate crossing links in a structure, the links must be either curved or offset from the nodes. This not only complicates the design of this structure, but may also reduce the load carrying capacity in its deployed configuration. Crossing links can also impose severe limitations on the way the structure is deployed, since any interference of the links during the deployment phase must be



(a) A truss module constructed using links of one given length



(b) The truss module in (a) constructed using two link lengths



(c) The truss module in (a) constructed using three link lengths

Fig. 4.3 Truss modules based on a particular graph, using : (a) a single link length, (b) two link lengths and (c) three link lengths.

avoided. Therefore, there is greater freedom in the choice of deployment method as well as in joint selection when no crossing links are present in the structure.

The properties of truss modules that are considered in this research may be summarized as follows :

1. Truss modules form a subset of the set of *spatial* truss structures.
2. Truss modules have two parallel planes. Each plane forms a planar truss structure; is composed of triangles; and contains approximately half the number of nodes in the module.
3. Each node in a truss module connects to at least one node in another plane.
4. Truss modules considered, will contain links of at most two different lengths.
5. Truss modules do not have crossing links.

These five properties can be subdivided into two categories. The first category contains Properties 1 through 3, which are associated with the function of a module. The second category is comprised of Properties 4 and 5, which result from the requirements for assembling a module. All these properties should also relate directly to the graph properties of a truss module. However, the derivation of these properties is not straight forward, but requires insight and experience in the construction of this type of structures. This subject will be addressed in the following section.

4.3 Properties of Graphs of Truss Modules

The corresponding graph of each of the structures in Fig. 4.2 is shown in Fig. 4.4. Each graph consists of vertices, representing the nodes, and edges, representing the links. Recall that the degree of a vertex is equivalent to the number of edges incident to a vertex and that a graph only reflects the inter-connectivity of the nodes

of the structure it represents. The following two subsections will be concerned with the properties of graphs of truss modules according to function and assembly of the modules.

4.3.1 Function of a Module

Properties 1-3, presented in Section 4.2, relate to the function or intended use of a truss module. The presence of the first property can be demonstrated by checking that the graph of the truss module has all the properties that correspond to the graph of a spatial truss structure, which have been presented in Section 2.2 of Chapter 2. The second property indicates that a graph of a plane within a structure is triangulated, which means that the graph is composed of triangles. Algorithms that check whether a graph is triangulated are readily available and have been presented by Rose and Tarjan [44,45]. Property 2 can thus be checked by looking for two triangulated subgraphs within the graph of a truss structure in which the number of vertices and edges in each of the subgraphs correspond to that of a planar truss structure. The presence of the third property can be determined when the two subgraphs that represent the two parallel planes, have been identified.

The properties of graphs of a truss module, that correspond to the function of the module, can be summarized as follows :

1. The graph exhibits the properties specified in Table 3.1 of Chapter 3.
2. The graph contains two triangulated subgraphs, where each vertex in the graph belongs to one of these subgraphs and each subgraph contains about half the number of vertices in the graph.
3. The number of vertices v and the number of edges e in each subgraph satisfy the relationship : $e \geq 2v - 3$.
4. Each vertex is connected to at least one vertex in a different subgraph

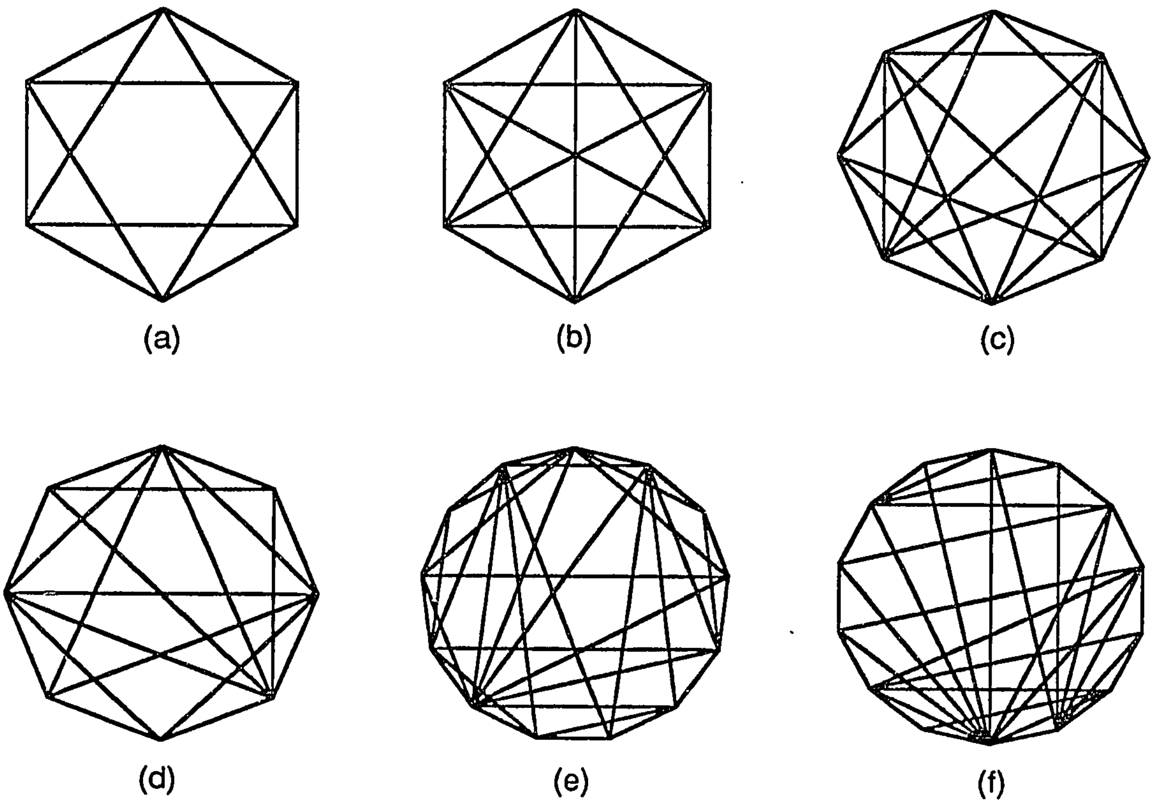


Fig. 4.4 Graphs corresponding to the structures in Fig. 4.2

A procedure that checks a graph for Properties 2-4 will be discussed in Section 4.5.

4.3.2 Assembly of a Module

Properties 4 and 5, listed in Section 2.2, relate to constraints on the assembly of a truss module. Specifically, Property 4 dictates that the module can be constructed by using links of no more than two different lengths. The detection of this characteristic is not straight forward since a graph does not reveal any information concerning the length of the links of the structure it represents. However, close examination of truss structures that have been constructed with links of no more than two different lengths, has led to the formulation of four theorems that will be discussed in this subsection. These theorems together describe the graph properties associated with Property 4 of Section 2.2, i.e. it is possible to construct truss modules using at most two link-lengths. The theorems will also describe how these different lengths may be detected from the graphs of the truss module.

Theorem 4.1

> Consider a graph that has all the properties as described in Section 4.3.1. The structure associated with this graph can only be built with links of two different lengths, when for each Vertex v there is at least one Vertex w in a different subgraph, such that all vertices, that are connected to but do not lie within the same subgraph as Vertex v, are connected to Vertex w.

Proof: Consider an arbitrary structure that has Properties 1-3 in Section 4.2 so that there exists two parallel truss planes A and B. Assume that Node 1 (corresponding to Vertex v in the theorem) in Plane A, lies on top of a Node 2 (corresponding to Vertex w in the theorem) in Plane B. Recall that each node is connected to one other node in a different plane (Property 3, Section 4.3.1). Let

there be a link between Node 1 in Plane A and Node 3 in Plane B (see Fig. 4.5). If Node 3 is also connected to Node 2 (in Plane B), then the triangle formed by Node 1, Node 2 and Node 3 can be constructed with links of two different lengths. If every node in Plane B that is connected to Node 1 is also connected to Node 2, then all triangles that are so formed, can be constructed using no more then two different length links. However, this can only be possible when Plane B is composed of triangles and does not have any crossing links. This allows all links between Node 2 and the nodes in Plane B that connect to both Node 1 and Node 2, to be of the same length. Extending this observation to every node in the structure leads to the proof for Theorem 4.1. It may be noted that there can be more than one node in Plane B that is adjacent to all nodes in Plane B that are connected to Node 1. The question that arises then is how to determine which node is to lie on top of which other node in the structure, when transposing from the graph to a structure. This subject will be addressed in the next theorem.

Theorem 4.2

Consider a graph that satisfies the conditions given in Theorem 4.1. Each node in the structure lies on top of or below one other node. Then, for each Vertex v in the graph, it is possible to select a unique Vertex w that does not lie in the same subgraph as Vertex v. This choice is restricted by the condition that Vertex w must be adjacent to all vertices (in the same subgraph as Vertex w) that are adjacent to Vertex v, while Vertex v must be adjacent to all vertices (in the same subgraph as Vertex v) that are adjacent to Vertex w.

Proof: The first part focuses on the question of whether two nodes of an arbitrary structure can lie on top of each other. The basis for this discussion can be found in the proof for Theorem 4.1. Consider the situation as illustrated in Fig. 4.5. When Node 2 is not adjacent to all nodes in Plane B that are adjacent to

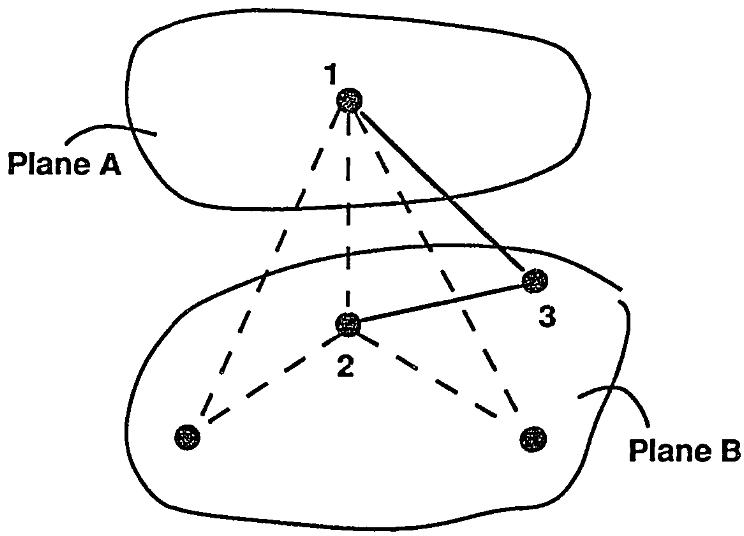


Fig. 4.5 Positions of Nodes 1, 2 and 3

Node 1, then Node 1 cannot lie on top of Node 2 since this would require the links originating in Node 1 to be constructed with more than two different link lengths. Similarly, when Node 1 is not adjacent to all nodes in Plane A that are adjacent to Node 2, then Node 2 cannot lie underneath Node 1. Thus, Node 1 and Node 2 can only lie on top of each other when Node 2 is adjacent to all vertices in Plane B that are adjacent to Node 1, and Node 1 is adjacent to all vertices in Plane A that are adjacent to Node 2. The graph equivalent of this requirement is stated in the theorem.

The second part of the proof focuses on the extension of the requirements for two nodes to every node in the structure. Consider three nodes, Node 1, Node 2 and Node 3. Assume that Node 1 can only lie on top of Node 2 and also that Node 3 can only lie on top of Node 2. This results in a conflict, since there can be one, and only one node, lying on top of any other node. Therefore, nodes that lie on top of each other must be pairs of unique nodes. This observation has been implied in Theorem 4.2.

The selection of vertices, described in Theorem 4.2, is further restricted by the requirement that each plane of the truss module must be composed of triangles, and without crossing links. This additional restriction is expressed in the next theorem.

Theorem **4.3**

Given a graph that satisfies the conditions given in Theorem 4.1. Consider pairs of vertices, v_i, w_i, such that the node represented by Vertex v_i lies on top of or below the node represented by Vertex w_i, as described in Theorem 4.2. The structure can be constructed using two different link-lengths when for every two vertices, v_1 and v_2, that are connected and lie within the same subgraph, there is an edge (w_1,w_2) that is part of the graph. There is only one exception : when there are two vertices, v_3 and v_4, within the same subgraph

as, and adjacent to, both v_1 and v_2; and if the edge (w_3,w_4) exists, then none of the edges (v_1,w_2), (v_2,w_1), (v_3,w_4) and (v_4,w_3) may exist.

Proof: Recall that when a structure has a graph that has the properties as set forth in Section 4.3.1, each plane of this structure can be constructed with triangles. Assume that the planes can also be constructed without crossing links. When Plane A is constructed in this manner, and the nodes on Plane B are placed above or below the nodes on Plane A (in accordance to Theorem 4.2) there is still no guarantee that Plane B will have no crossing links. Assume that two nodes, Node 1 and Node 2, lie within the same plane and are connected. Assume that Node 3 and Node 4 are connected and lie on top of or below Node 1 and Node 2 respectively. When this can be said of any two connected vertices that lie within the same plane, then if one plane can be constructed of triangles with no crossing links, so can the other. Now assume that Node 3 and Node 4 are not connected. Figure 4.6 illustrates the only situation when this is acceptable. Due to the fact that there is no link between Node 3 and Node 4, there must be a link between Node 7 and Node 8 to ensure that Plane A is still composed of triangles. However, in that case, there cannot be any diagonals in the frame (1,2,4,3) nor in the frame (5,6,8,7) , since this would mean that the structure will have to be constructed with at least three different length links. The theorem describes all allowable situations and the corresponding necessary conditions.

The fourth theorem follows from the observation that it may be possible for a node in the structure not to lie on top of (or below) any other node in the same structure.

Theorem 4.4

Consider a graph, that satisfies the conditions laid out in Theorem 4.1. Let there be a vertex, v, that represents a node that does not lie on top of any other

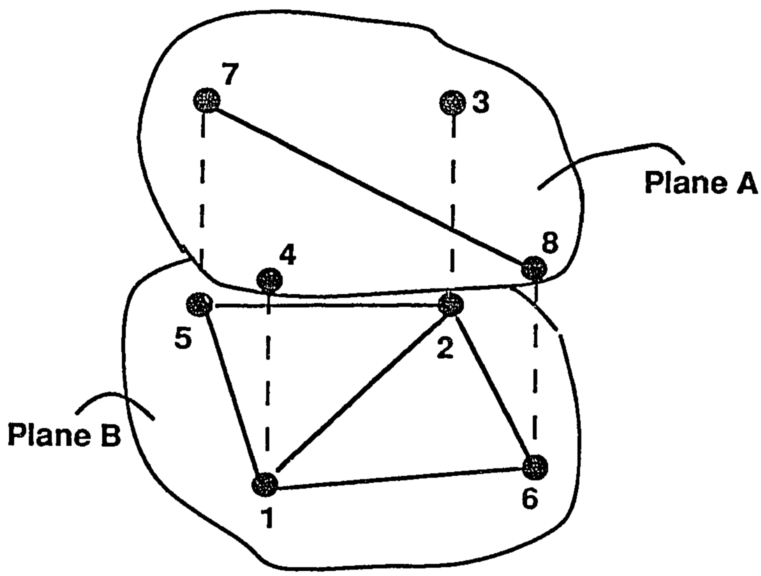


Fig. 4.6 Positions of Nodes 1 through 8

node. The structure can then only be constructed with two different length links when it is possible to add one vertex and two edges to the subgraph that does not contain Vertex v. This addition is restricted by the condition that the subgraph remains triangulated and the added vertex is adjacent to all vertices in this subgraph that are adjacent to Vertex v.

Proof: Assume that a truss module has one node, Node 1, that does not lie on top of any other node in the structure. Add a new node to the structure, Node 2, that lies below Node 1. Connect Node 2 to Node 1. At this point, Node 1 and Node 2 satisfy the relationships as specified in Theorem 4.2, except that Node 2 is not part of a plane composed of triangles. However, this limitation can be eliminated by adding two more links that connect Node 2 with the other nodes on the plane in which it lies, as illustrated in Fig. 4.7. This operation will not compromise any of the characteristics of the truss module. Therefore, if this new structure can be built using links of two different lengths (i.e. its graph satisfies Theorem 4.3), so can the original structure. However, it is not required that Node 2 be connected to Node 1, since this only serves to satisfy Property 3 of Section 4.2. The required verification procedure for a graph is described in Theorem 4.4.

Theorems 4.1-4.4 form the basis from which all possible arrangements of nodes that lie on top of the nodes in the bottom plane can be generated or identified. Given this information, the detection of crossing links in a truss module is readily achieved. A procedure that accomplishes these tasks will be presented in Section 4.6.

The properties of truss modules that are given in this section can be employed to find graphs of truss modules from among the graphs of truss structures, and to find graphs of deployable truss modules from among the graphs of deployable truss structures. In the following sections it will be shown that this leads to the

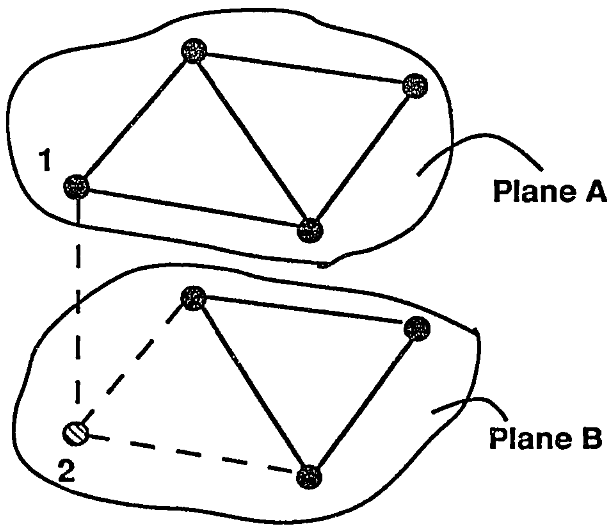


Fig. 4.7 Addition of Node 2 and the links connecting to Node 2

conceptual design of truss modules as well as the conceptual design of deployable truss modules.

4.4 Generation of Graphs of Truss Modules

Graphs representing truss modules form a subset of the graphs representing truss structures, since not all graphs of truss structures have the properties as listed in Section 4.3. Graphs of truss modules can therefore be found by first generating graphs of truss structures and subsequently checking each graph for the additional properties of graphs of truss modules. The following procedure not only accomplishes this task, but also generates a three-dimensional model of each truss module.

1 Generate graphs representing spatial (deployable) truss structures.

2 Check that each graph represents a truss structure with two separate planes, each being composed of triangles and without crossing links. Eliminate all graphs that do not satisfy this characteristic.

3 Check that each graph, that represents a truss structure, can be built using links of two different lengths while avoiding crossing links. Eliminate graphs that do not have this property.

4 Construct a three-dimensional representation of the truss module that reveals the relative lengths of the links in the structure.

Step 1 of this procedure can be carried out using the algorithms provided in Chapter 2. Separate algorithms for Steps 2 and 3 of the graph generation procedure will be presented in the following two sections. Each algorithm will be preceded by a brief discussion relating to the basic idea behind its development. Each section will also include an example that demonstrates the application of the algorithm in a

step-by-step manner. It will be shown that Step 4 of the generation procedure can readily be carried out once Step 3 has been completed. The discussion of Step 4 will therefore be included in the discussion of Step 3. In the next section, a procedure will be discussed relating to Step 2 - the identification of two separate planes, each being triangulated.

4.5 Procedure to Determine Two Triangulated Subgraphs within a Graph

One of the characteristics of truss modules is the existence of two separate planes. Generally, each plane forms a statically determinate planar structure. Another important characteristic of a plane is that it is composed of triangles. This means that it is not possible for four or more nodes in a plane to be connected such that their links form only one loop.

Based on the properties of the planes, the existence of two separate planes can be verified by checking if the graph of a truss structure can be split into two triangulated subgraphs. Further, each subgraph should contain approximately half the number of vertices of the total graph. The number of edges e in each subgraph is also bounded by the relation : $e \geq 2v - 3$, where v is the number of vertices in the subgraph.

The following algorithm checks for the existence of two triangulated subgraphs in the graph of a truss structure. The algorithm is based on a Lexicographic Breadth First Search (LBFS), an efficient algorithm for checking triangulated graphs, developed by Rose and Tarjan [44]. The LBFS algorithm first produces a particular ordering of the vertices. It then checks that if the vertices (and their incident edges) are eliminated based on this ordering, then only triangles are eliminated from each reduced graph. If this test fails, then the graph is not triangulated. The original LBFS algorithm will be modified to accommodate the search for all triangulated

subgraphs within a graph. After the first triangulated subgraph (Plane B) is found, the algorithm then checks whether the subgraph containing the remaining vertices (Plane A) is also triangulated.

Algorithm C is an algorithm for determining all combinations of triangulated subgraphs within a graph. This process is crucial, since the results of this algorithmic search serve as input to the algorithm, discussed in Section 4.6, which determines the relative link-lengths for the truss modules. The following algorithm has also been enhanced to perform one of the tasks of the algorithm in Section 4.6, namely to check whether the planes of a truss module has crossing links. This check can be readily performed by checking that if the vertices of each subgraph are eliminated in the order produced by the LBFS, then no more than two triangles are eliminated at a time.

4.5.1 Algorithm C

Given a graph with v vertices that represents a spatial truss structure, this algorithm determines all combinations of two triangulated subgraphs and checks for crossing links within the two parallel planes of the truss module.

1 Select the number of nodes s in Subgraph A so that $s \approx v/2$. Note : Subgraph B must then contain $v - s$ nodes.

2 Label all vertices 0; Set index $k = v - 1$; Select Vertex 1 to be part of Subgraph B. The vertices corresponding to Subgraphs A and B will be stored subsequently in the array P. The vertices are stored beginning at the last available position in this array, thus $P(v) = 1$.

3 Perform Steps 4-7 beginning with $i = v$ and decrease i by 1 for each subsequent loop provided that $i \geq s + 1$.

4 If $i > k$ continue with Step 7. If $i < k$ pick the first available vertex with a non-zero label and continue with Step 5. If $i = k$ pick the next available

vertex with a non-zero label and continue with Step 5. If no vertex is available then $k = k + 1$. If $k < v$ repeat Step 4. If $k = v$ then all possibilities have been investigated for s nodes in Subgraph A. When $s \neq v - s$, then there are remaining possibilities which can be investigated by repeating the procedure starting at Step 1 and assuming that Subgraph A contains $v - s$ nodes (thus $s = v - s$).

5 $P(i) =$ vertex found in Step 4. If $i \neq k$ then $k = k - 1$.

6 Add all vertices that are not represented in P and that have a larger label than the vertex found in Step 4 to Subgraph A.

7 Increase the labels of all the vertices adjacent to the vertex stored in $P(i)$ with the value i. Continue with Step 3 if the loop of Steps 3-7 is not completed.

8 Check whether the vertices stored in $P(i), i = s + 1, ..., v - 2$ are arranged such that $P(a)$ is connected to $P(b)$ and $P(c)$; and $P(b)$ is connected to $P(c)$, where $a < b < c$. The vertex in $P(a)$ can be connected to only one more vertex $P(d)$, where $d > c$, provided that one only one of the vertices $P(b)$ and $P(c)$ is connected to $P(d)$. If these conditions are not met, then the sub-graph is not triangulated and/or the corresponding plane contains crossing links. In that case reset index k to $k = s + 1$, label all vertices 0 and return to Step 3.

9 Label all vertices 0.

10 Perform Step 11-12 beginning with $i = s$ and decrease i by 1 for each subsequent loop provided that $i \geq 1$.

11 Store the first available vertex with the highest non-zero label in $P(i)$.

12 Increase the labels of all vertices adjacent to the vertex found in Step 11 by the value of i.

13 Check whether the vertices stored in $P(i), i = 1, ..., s - 2$ are arranged such that they meet the conditions as described in Step 8. If this result is negative, then reset index k to $k = s + 1$, label all vertices 0 and return to Step 3.

14 Check for vertices of degree 3 that are only connected to vertices of the same plane. If such a vertex exists, reject the solution and continue with Step 4.

15 Check whether the solution is identical to a previous solution. Do this by looking for a renumbering scheme for the graph that makes two solutions identical. If no such scheme can be found, then proceed with Step 16, otherwise go to Step 17.

16 List Subgraph A as vertices contained in $P(1), P(2), ..., P(s)$ and list Subgraph B as vertices contained in $P(s+1), P(s+2), ..., P(v)$.

17 Find alternative solutions. Thus, reset index k to $k = s + 1$, label all vertices 0 and return to Step 3.

4.5.2 Example

The following example concerns the execution of the algorithm for the graph of a six node truss structure, shown in Fig. 4.8. The example follows the previous algorithm in chronological order.

1 Select $s = 3$, since it is required that $s \approx v/2$.

2 Label all vertices 0; Set $k = 6 - 1 = 5$; Set $P(6) = 1$.

3,4,7 Since $i = 6$ and $i > k$, Steps 5-6 are skipped. Increase the labels of vertices 2,3,4,5 and 6 by the value "6" since they are adjacent to the vertex stored in $P(6)$, which is Vertex 1.

3,4,5 Since $i = 5$ and $i = k$, the next available vertex with a non-zero label is picked that has a higher number than stored in $P(5)$. Note that at this point P(5)=0. Pick Vertex 2. Thus, $P(5) = 2$.

6 This step requires no action at this point, since there is no available vertex with a smaller label than Vertex 2.

7 Vertices 1,3,4,5 and 6 are adjacent to Vertex 2. Increase the labels of these vertices by the value "5".

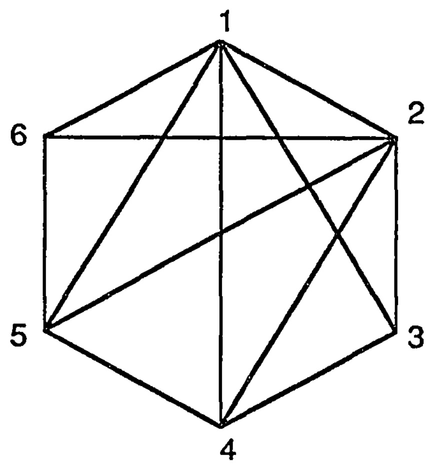


Fig. 4.8 Graph of a six-node truss structure

3,4,5 Now $i = 4$ and $i < k$. The first available vertex is Vertex 3. Thus $P(4) = 3$ and $k = 5 - 1 = 4$.

6,7 Again, Step 6 requires no action. Step 7 leads to increasing the labels of Vertices 1,2 and 4 by the value ”4”.

8 The vertex stored in $P(4)$ is Vertex 3. Since this vertex is indeed adjacent to the vertices stored in $P(5)$ and $P(6)$ and there is no third vertex in this subgraph that is adjacent to it, the vertices stored in $P(4)$,..., $P(6)$ form a triangulated graph.

9,10 Reset all labels to 0 and set $i = 3$.

11,12 Pick one of the remaining vertices, e.g. Vertex 4. Set $P(3) = 4$ and increase the labels of Vertices 1,2,3 and 5 by the value ”3”.

10-12 Now $i = 2$ and $P(2) = 5$. The labels of Vertices 1,2,4 and 6 are increased by ”2”.

10-12 The last step of this loop gives : $P(1) = 6$.

13 Vertex 6, stored in $P(1)$ is not adjacent to the vertices stored in $P(2)$ and $P(3)$, thus the vertices stored in $P(1)$ through $P(3)$ do not form a triangulated graph and the solution is rejected. Label all vertices 0 and set $k = 4$.

3,4,7 $i = 6$; Since $i > k$ and $P(6) = 1$, increase the labels of vertices adjacent to Vertex 1 by ”6”.

3,4,7 $i = 5$; Since $i > k$ and $P(5) = 2$, increase the labels of vertices adjacent to Vertex 2 by ”5”.

3,4 $i = 4$; Since $i = k$, a vertex needs to be picked with a number higher than the one presently stored in $P(4)$. Pick Vertex 4, thus $P(4) = 4$.

14 Steps 8-13 lead to the conclusion that the vertices not stored in $P(4)$ through $P(6)$ do not comprise a triangulated graph. The procedure continues with the repeated operations of the loops contained in Steps 3-8 and 9-13 until the contents of $P(1)$ through $P(6)$ is : 6,5,2,4,3,1. At this point, the set formed

by $P(6)$ through $P(4)$ and the set $P(3)$ through $P(1)$ each form a triangulated graph. Neither graph fails the check made in Step 14.

15-17 Since this is the first solution and therefore unique, no isomorphism check needs to be made. The planes are : Vertices 2,5,6 (Plane A) and Vertices 1,3,4 (Plane B). Set k=4, label all vertices 0 and find an alternative solution.

15 The next solution produced by the algorithm is : 6,5,2,3,4,1 (stored in $P(1)$ through $P(6)$). However, this is not an acceptable solution since it results in the same planes as the first solution. The algorithm proceeds until $P(1)$ through $P(6)$ has the following contents : 4,3,2,6,5,1. This solution is unique since no renumbering scheme can be found such that it is identical to the one found previously.

4 The algorithm continues until k becomes equal to v (i.e. $k = 6$). At this point all solutions have been found and the execution of the algorithm is terminated.

4.6 Procedure for Selection of Relative Link-Lengths

A truss module has the property that it can be built out of links of two different lengths. Theorems 4.1-4.4, discussed in Section 4.3.2, provide the basis for a procedure to verify that this property exists within a given graph. Within this procedure it is possible to determine which node lies on top of (or below) which other node, given the graph of a truss module. This information can then be used not only to check for crossing links in the structure, but also to construct a three-dimensional presentation of the truss module. As a result of this process the length of each link can be determined, given the length of the shortest member. However, since this length is unknown in the conceptual design stage, the length of each link will be determined relative to the length of the shortest member.

The following algorithm, Algorithm D, will determine the relative link-lengths for each graph that represents a truss module. This is carried out in three stages. The first stage examines the graph for the properties described in Theorems 4.1, 4.2 and 4.4 of Section 4.3.2. The execution of this stage begins with generating a list for each vertex of the graph. Each list contains all the vertices that satisfy Theorem 4.1 of Section 4.3.2 for the vertex associated with the list. For example, given a graph and its two subgraphs A and B, the list for Vertex v in Subgraph A contains Vertex w in Subgraph B when Vertex w is adjacent to all vertices in Subgraph B that are adjacent to Vertex v. This means that, if the list for any given vertex is empty, then the graph does not comply with Theorem 4.1 and therefore does not represent a truss module.

Each list created in the first stage of the procedure contains the vertices that represent nodes that can lie on top of or below the node represented by the vertex that is associated with the list. In compliance with Theorem 4.2, two nodes, represented by Vertices v and w, can only lie on top of each other when the list for Vertex v contains Vertex w and the list for Vertex w contains Vertex v. The algorithm removes inconsistencies to Theorem 4.2, by eliminating Vertex v in the list for Vertex w, if Vertex w does not appear in the list for Vertex v. Subsequently, a dummy vertex, Vertex "0", is added to the list for each vertex that complies with Theorem 4.4. The graph then complies with Theorem 4.2, when it is possible to pick one vertex in the list for each of the vertices of one of the subgraphs, such that no vertex is picked more then once. All possible selections are then passed on to the next stage of the algorithm.

In the second stage Algorithm D checks whether each selection produced by the first stage complies with Theorem 4.3, which states the necessary conditions for preventing crossing links in each of the planes. Recall that each selection represents the position of the nodes in one plane with respect to the nodes in the other plane. Thus, at this point a check can be performed as to whether a selection would

result in a structure with crossing links. Since Algorithm C already checks for the occurrence of crossing links in each of the planes, Algorithm D only needs to check for the occurrence of situations as shown in Fig. 4.9

In the third and final stage, the algorithm constructs a three-dimensional presentation of the truss module. First, the relative link lengths are determined based on the information obtained in the previous two stages. Both planes of the module are then constructed. The top plane is then maneuvered into the correct position on top of the bottom plane. This construction is completed by adding all the links that connect the top and bottom plane.

All possible selections of relative link-lengths for a given graph are generated by the algorithm. This is achieved by generating all possible arrangements of vertices that are picked in the first stage of the execution of the procedure. The two subsequent stages are then executed for each arrangement separately. The complete algorithm is listed in the following subsection.

4.6.1 Algorithm D

Given a graph of a spatial truss structure and two of its triangulated subgraphs, which represent parallel planes in the structure, this algorithm determines the relative link-lengths for the corresponding truss module.

1 Pick a Vertex w. Proceed with Step 5 if all vertices have been picked in this step.

2 Pick a Vertex v that is not in the same subgraph as Vertex w. If all possible vertices have been picked then go to Step 4, otherwise proceed with Step 3.

3 If Vertex v is adjacent to all vertices in the same subgraph as Vertex v, that are adjacent to Vertex w, then add Vertex v to a list for Vertex w. Return to Step 2.

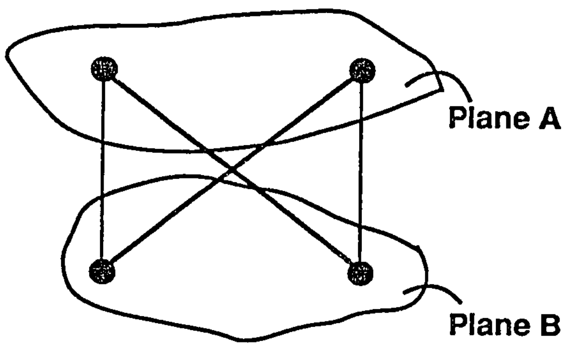


Fig. 4.9 Situation where crossing links are present

4 If the list for Vertex w is not empty, then return to Step 1, otherwise reject the graph, since the resulting structure will consist of more than two different lengths.

5 For each list do the following : if there is a Vertex v in the list for Vertex w such that Vertex w is not in the list for Vertex v then remove Vertex v from the list for Vertex w.

6 For each vertex of the graph do the following. If a Vertex v connects to no more then two Vertices x and y, which lie in another subgraph, then, if Vertices x and y are adjacent and there is no more then one vertex that is adjacent to both Vertices x and y within the same subgraph, then add Vertex "0" to the list for Vertex v.

7 Pick the largest subgraph and call it Subgraph A. Call the other, Subgraph B.

8 For each vertex in Subgraph A, select one vertex in the list for this vertex without picking any non-zero vertex more then once. Terminate the execution of this algorithm when all possible selections have been investigated and reject the graph if none of them were acceptable.

9 Check that every vertex in Subgraph B that does not appear in the selection of Step 8 has a zero in its list. If the result is negative, then reject this selection and return to Step 8, otherwise continue with Step 10.

10 When there exists a vertex in Subgraph A for which vertex "0" is picked in Step 8, while this vertex is connected to a vertex in Subgraph B that is not selected in Step 8, return to Step 8. In that case a node that does not lie on top of any other node is connected to a node (not within the plane of in which the first node lies), that does not lie below any other node. This is not acceptable, since Theorem 4.4 cannot be satisfied. If such a situation does not occur, continue with Step 11.

11 Adopt the following notation for Steps 11-15. Let Vertex w_i be the vertex that was picked in Step 8 for Vertex v_i, where i is any integer in the range 1 through

the size of Subgraph A. Pick a set of two vertices in Subgraph A, Vertices v_1 and v_2, for which the corresponding vertices, Vertices w_1 and w_2, are non-zero. If all possible sets of two vertices have been picked, then accept the selection made in Step 8 and go to Step 15, otherwise continue with Step 12.

12 If $(v_1, w2)$ and (v_2, w_1) are edges, then proceed with Step 8. The corresponding structure has crossing links.

13 If only one of the edges (v_1, v_2) and (w_1, w_2) exists, then there must be a vertex, Vertex v_3, within the same subgraph as and adjacent to both Vertex v_1 and Vertex v_2; and a vertex, Vertex w_4, within the same subgraph as and adjacent to both Vertex w_1 and Vertex w_2, while none of the edges (v_1, w_2), (v_2, w_1), (v_3, w_4) and (v_4, w_3) exist (in compliance with Theorem 4.3). If this is not satisfied then return to Step 8, otherwise proceed with Step 14.

14 If there are two vertices, Vertices v_3 and v_4, that are adjacent to both Vertex v_1 and Vertex v_2, then there cannot be a vertex, Vertex w_5, that is adjacent to both Vertex w_1 and Vertex w_2, while Vertex $w_5 \neq$ Vertex w_3 (i.e. Vertex w_5 is not the same as Vertex w_3) and Vertex $w_5 \neq$ Vertex w_4. Similarly, if there are two vertices, Vertices w_3 and w_4, that are adjacent to both Vertex w_1 and Vertex w_2, then there can not be a vertex, Vertex v_5, that is adjacent to both Vertex v_1 and Vertex v_2, while Vertex $v_5 \neq$ Vertex v_3 and Vertex $v_5 \neq$ Vertex v_4. Continue with Step 11 if both requirements are met. Otherwise, return to Step 8 since the structure, built according to the selection made in Step 8, has a node(s) that do not agree with Theorem 4.4.

15 Let $i \neq j$. Select all edges (v_i, w_i) to be of unit length. Also, select all edges (v_i, v_j) and (w_i, w_j) to be of unit length, provided that (v_i, w_j) or (w_i, v_j) exists. Let all edges (v_i, w_j) be of length $\sqrt{2}$ $*$ (unit length), as well as the edges (v_1, v_2), (v_3, v_4), (w_1, w_2) and (w_3, w_4), in situations as described in Step 13. The lengths of the remaining edges can be chosen either of unit length or

$\sqrt{2} *$ (unit length), while the length of the edges (v_i, v_j) and (w_i, w_j) must be identical.

16 Build Plane A (represented by Subgraph A) and Plane B (represented by Subgraph B).

17 Pick a node of Plane A and translate Plane A in a direction that places this node on top of the node in Plane B that corresponds to the selection made in Step 8.

18 Pick another node of Plane A and rotate Plane A about the node picked in Step 17 until the node picked in this step lies above its corresponding node, as determined in Step 8.

19 Pick yet another node of Plane A. If this node already lies on top of its corresponding node in Plane B, then proceed with Step 20. Otherwise, rotate Plane A over 180 degrees about the line between the two nodes of Plane A that were picked in Steps 17 and 18.

20 Place the two planes a unit length apart and draw the links connecting them. Proceed with Step 8 to find alternative solutions.

4.6.2 Example

The following example investigates the graph shown in Fig. 4.8, which was also used as an example in Section 4.5.2. The leading numbers correspond to the steps in the algorithm.

1 Pick Vertex 1.

2 Two sets of vertices of triangulated subgraphs are : 2,3,4 (Subgraph 1) and 1,5,6 (Subgraph 2). Since Vertex 1 is in Subgraph 2, pick a vertex in Subgraph 1 : Vertex 2.

3 The vertices that are adjacent to Vertex 1 and do not lie in the same subgraph as Vertex 1 are the Vertices 2,3 and 4. Vertex 2 is adjacent (or incident) to

Vertices 2, 3 and 4 and is therefore adjacent to all vertices in the same subgraph as Vertex 2 that are adjacent to Vertex 1. Thus, vertex 2 is added to the list for Vertex 1.

4 After successive operation of Steps 2 and 3, the list for Vertex 1 contains the Vertices 2,3 and 4. Since this list is not empty, the algorithm returns to Step 1.

5 The algorithm reaches this Step 5 after Steps 2-4 have been executed for all vertices in the graph. At this point, the list for each vertex contains the vertices as shown in Fig. 4.10a. Step 5 does not alter this list since there is no Vertex w that appears in the list for Vertex v, such that Vertex v is not in the list of Vertex w (i.e. both $1 \rightarrow 2$ and $2 \rightarrow 1$ are present).

6 Pick Vertex 1. This vertex has more then two connections with another subgraph so that vertex "0" is not added to its list. This also applies to Vertex 2. Vertex 3 however, has only one connection to another subgraph and therefore has a vertex "0" added to its list. Similarly, vertex zero is added to the lists for Vertices 4, 5 and 6. The revised lists are shown in Fig. 4.10b.

7 Since both subgraphs contain the same amount of vertices, any of the two subgraphs can be chosen as Subgraph A. Choose Subgraph 1.

8 One of the selections that can be made is the following : $2 \rightarrow 1$, $3 \rightarrow 5$, $4 \rightarrow 6$ (Node 2 lies on top of Node 1, Node 3 lies on top of Node 5 and Node 4 lies on top of Node 6).

9 All vertices of Subgraph B are present in the selection made in Step 8, so that the algorithm proceeds with step 11.

11 Let $v1 = 2$ and $v2 = 3$. Thus, $w1 = 1$ and $w2 = 5$.

12 Since (2,5) and (3,1) are edges of the graph, the selection made in Step 7 leads to a structure with crossing links. Reject this selection and try to find another by executing Step 7.

Vertex		List
1	→	2, 3, 4
2	→	1, 5, 6
3	→	1, 5, 6
4	→	1, 5, 6
5	→	2, 3, 4
6	→	2, 3, 4

(a) basic List

Vertex		List
1	→	2, 3, 4
2	→	1, 5, 6
3	→	1, 5, 6, 0
4	→	1, 5, 6, 0
5	→	2, 3, 4, 0
6	→	2, 3, 4, 0

(b) extended list

Fig. 4.10 List for each vertex of the graph in Fig. 4.4

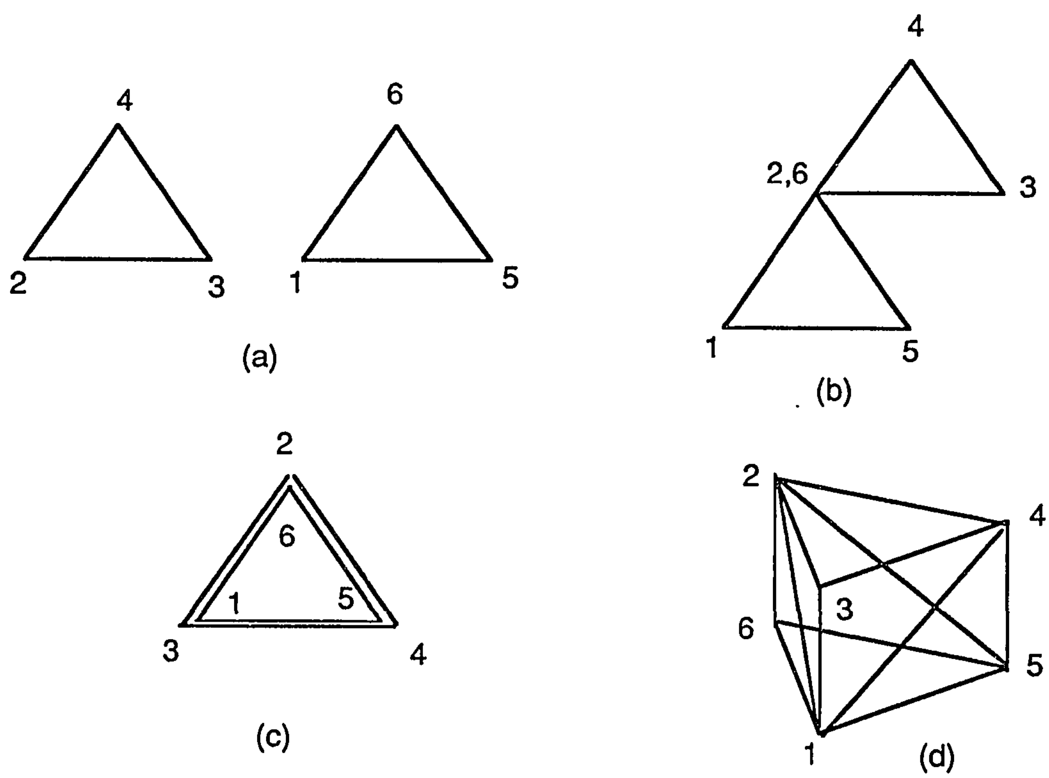


Fig. 4.11 Construction of a truss module

15 After successive operation of Steps 8-11, the following selection is encountered : $2 \rightarrow 6$, $3 \rightarrow 1$ and $4 \rightarrow 5$. This selection passes the tests of Steps 9-14, which means that the graph is acceptable since it represents a truss module. In Step 15 all links in Plane A and Plane B are selected to be of unit length, as well as the links between 2 and 6, 3 and 1, 4 and 5. The remaining links are of $\sqrt{2}$ * (unit length).

16 The construction of the truss module begins by building Plane A and Plane B of the module using links of unit length. The result is shown in Fig. 4.11a.

17 Pick Node 2 of Plane A and move Plane A so that Node 2 lies on top of Node 6 (see Fig. 4.11b).

18 Pick Node 3 and rotate Plane A about Node 2 until Node 3 lies on top of Node 1 (see Fig. 4.11c).

19 Pick the last node of Plane A : Node 4. Step 19 requires no further action, since this node already lies on top of Node 5.

20 Place the two planes a unit length apart and draw the links that connect them. The result is shown in Fig. 4.11d.

8 The algorithm is terminated after all selections, derived in Step 8, are investigated. At that point, the algorithm has found two acceptable configurations of truss modules that are based on the graph shown in Fig. 4.8. Figure 4.12 shows these truss modules among all truss modules that are determined using this algorithm from all unique graphs of statically determinate truss structures, containing six nodes, which were identified in Section 3.5 of Chapter 3

4.7 Results

The procedure to generate truss modules, described in Section 4.6, has been incorporated in a computer program, which was written in FORTRAN. This code can be used to either generate truss modules or deployable truss modules, given

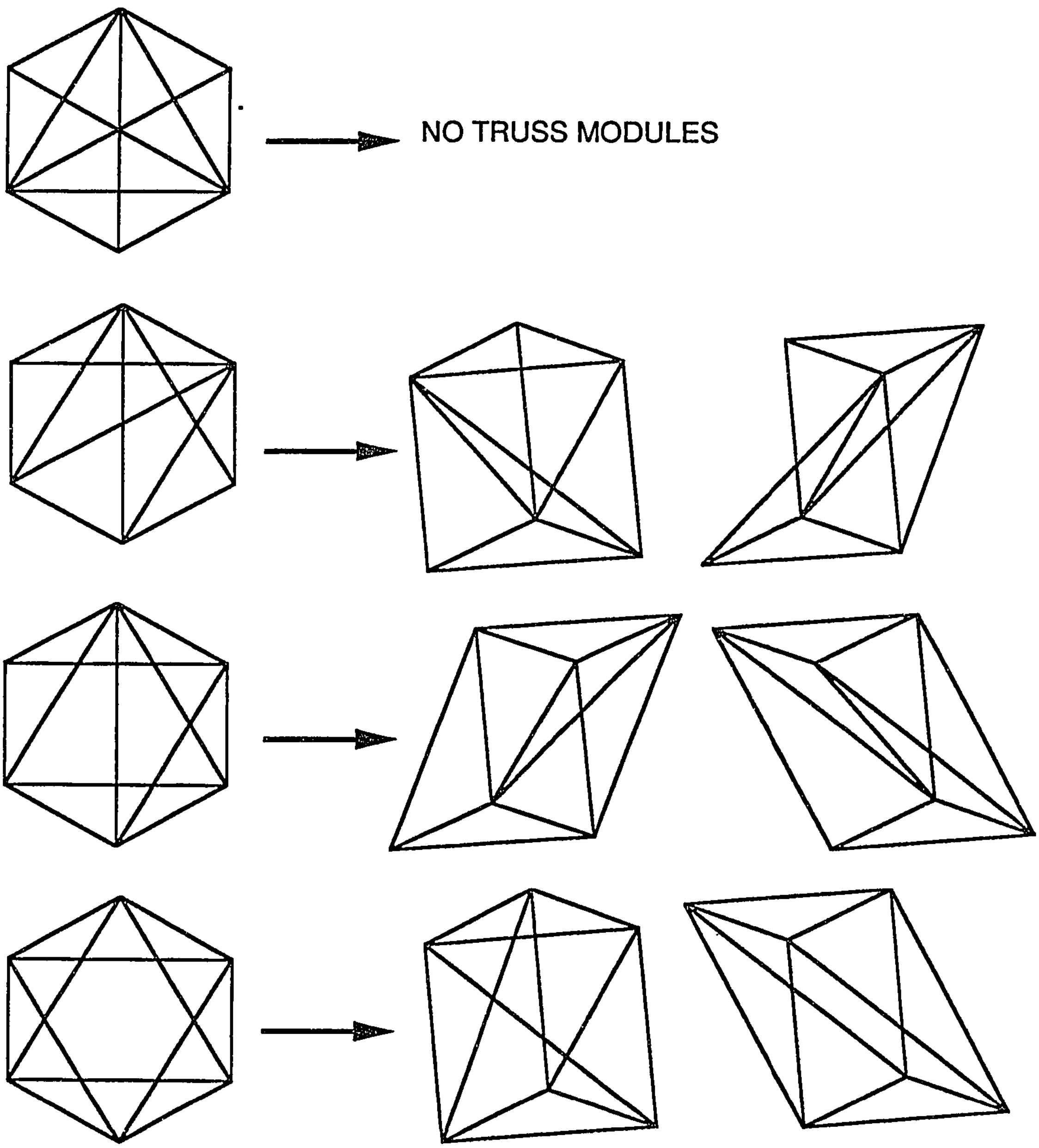


Fig. 4.12 All graphs of statically determinate truss structures and the truss modules they represent.

the desired number of nodes in the structure. The number of links and the number of transition links required to fold the structure are then determined based on the facts that truss modules are generally statically determinate and contain only the minimum number of transition links. However, truss modules and deployable truss modules that do not have these properties can also be generated by the program by supplying it with the desired number of links and transition links in the module.

The results have been compared with existing designs such as those shown in Fig. 4.2. It has been shown to be possible to generate not only all known existing designs that have the properties as listed in Section 4.2, but also to generate alternatives to each design. As an example, Fig. 4.13 shows all possible statically determinate six-node truss modules that have three transition links, which is the minimum number required to fold the structure. Although one of these 709 structures is the (well-known) truss module shown in Fig. 4.14, a vast majority of these structures have not been reported in the literature.

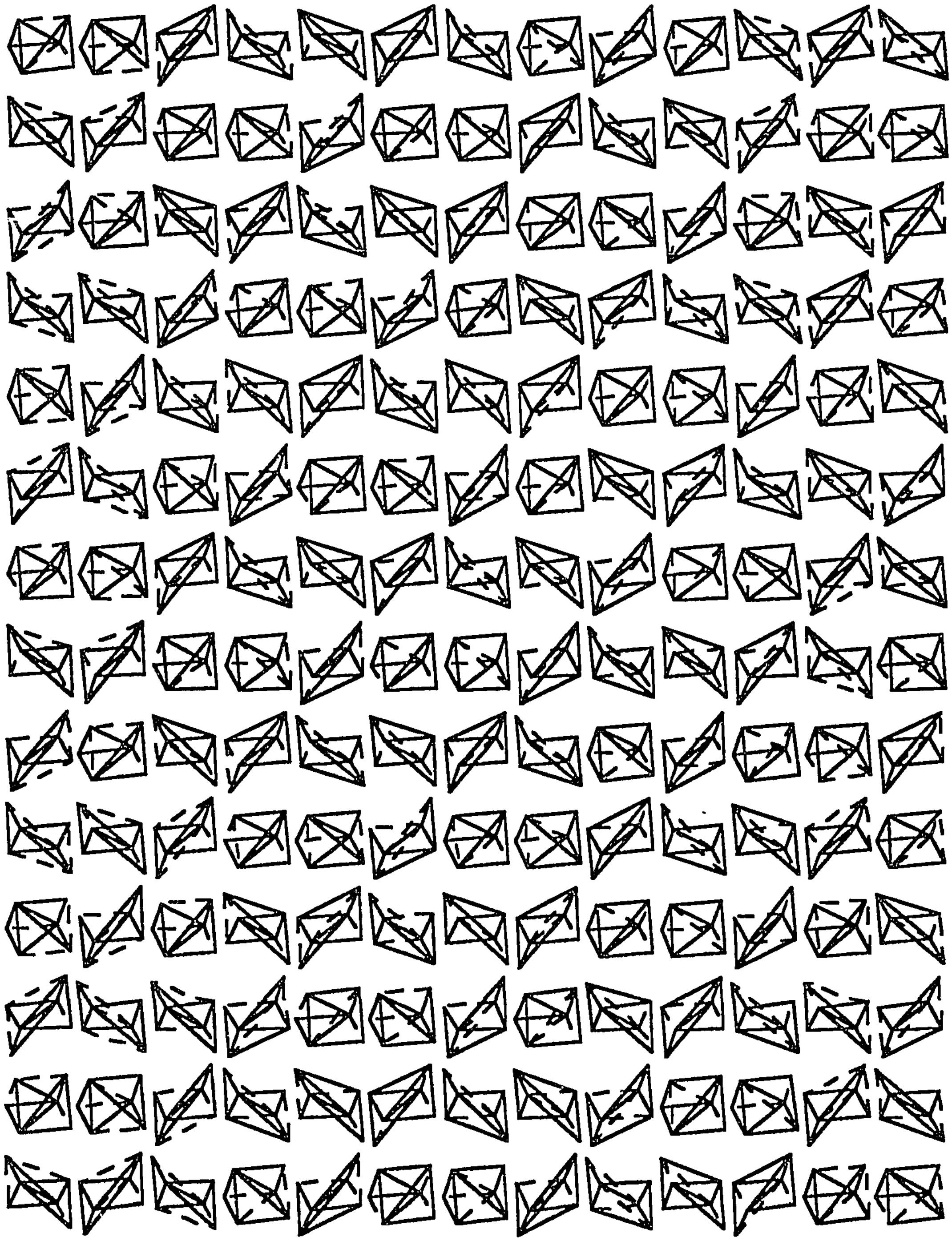


Fig. 4.13 All possible deployable truss modules with six nodes and twelve links that have a minimum number of transition links.

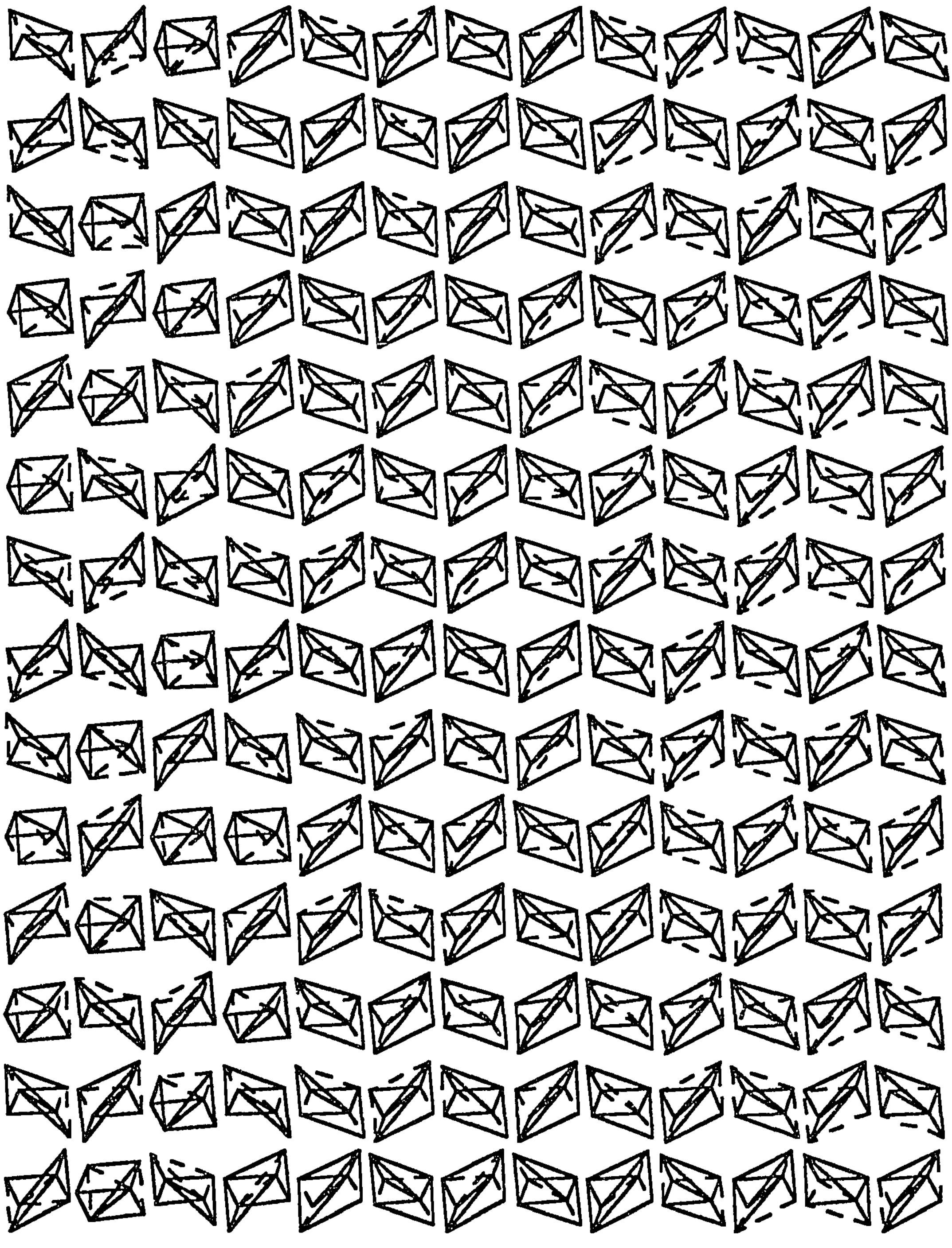


Fig. 4.13 Continued

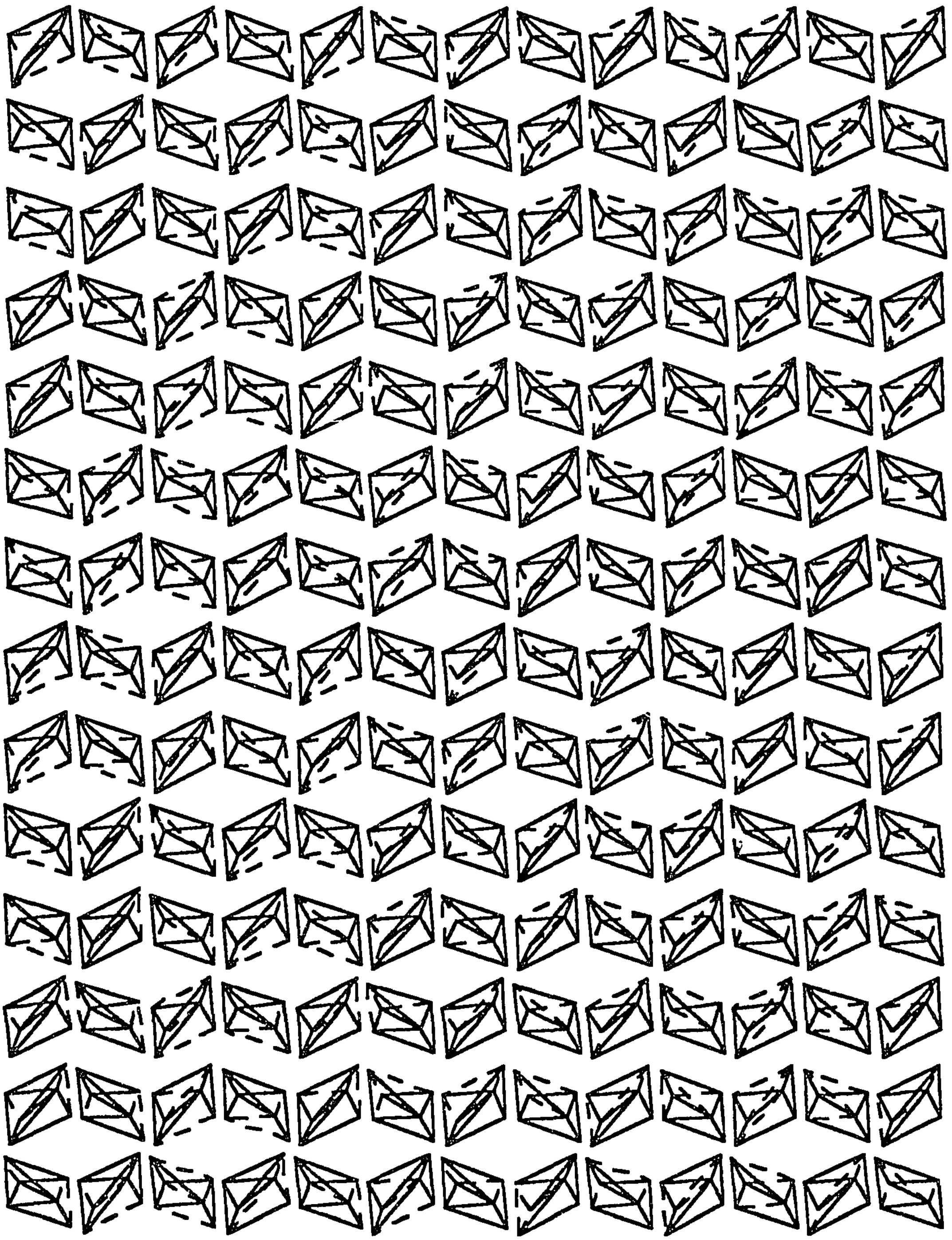


Fig. 4.13 Continued

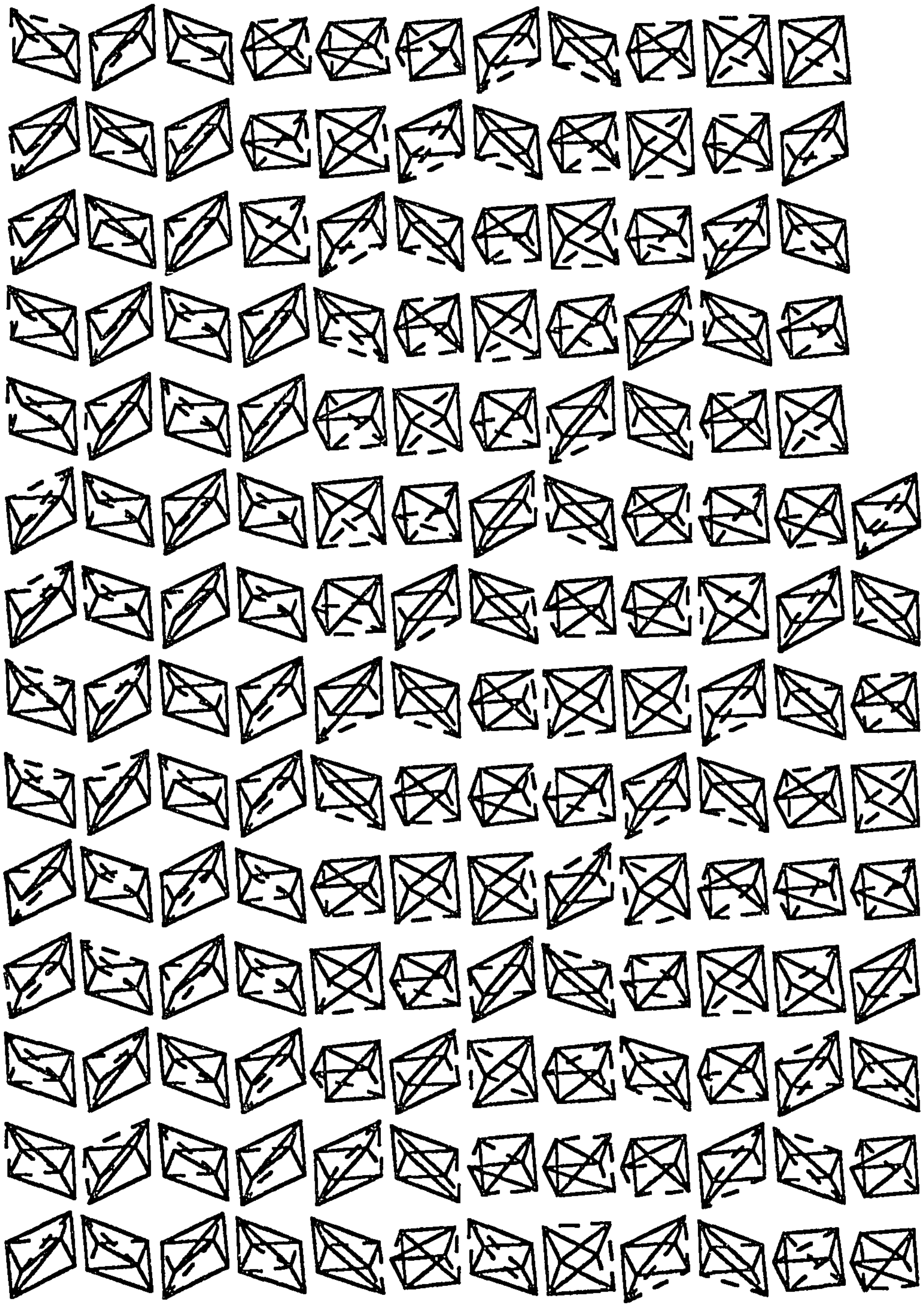


Fig. 4.13 Continued

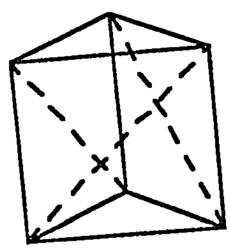


Fig. 4.14 Articulated Astromast [14]

CHAPTER 5

GENERATION OF ALL FOLDED CONFIGURATIONS OF DEPLOYABLE-FOLDABLE TRUSS STRUCTURES

5.1 Outline

A deployable truss structure differs from other mechanisms in that it contains joints that are locked in the deployed configuration of the structure but allow the structure to move to a stowed configuration when they are unlocked. The number and locations of these joints can be determined with the techniques presented in Chapter 2, which guarantees that the deployable truss structures are able to fold to a configuration of lower dimension (a plane or a line). These techniques are based on the assumption that the structure is not prevented from attaining the stowed configuration due to an inappropriate choice of link lengths and on the assumption that up to three degrees of freedom (d.o.f.) are available per joint. This chapter concentrates determining whether a structure with given link lengths is able to fold to a stowed configuration.

The relative lengths of the links in a deployable truss structure not only impacts the deployed and folded configurations, but also the joints that can be used to enable the transition between the two configurations. Special dimensions for the links of a deployable truss structure may therefore result in deployed and folded configurations that have very desirable characteristics, such as a repetitive geometry

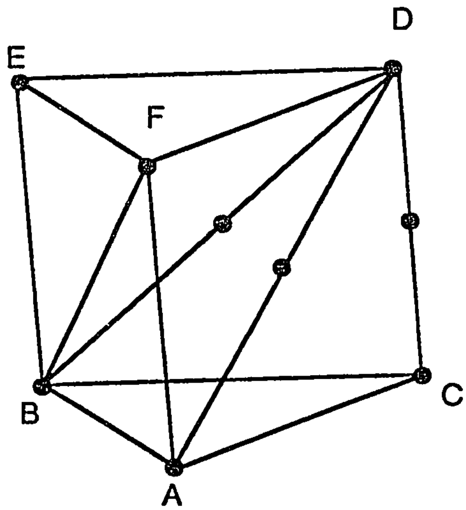
and efficient packaging, and which may also allow a transition between the two configurations with fewer joints used and/or joints supplying fewer degrees of freedom. To identify the existence of these characteristics, all possible folded configurations of a deployable truss structure must first be determined.

The following two sections discuss how all folded configurations of a truss structure can be determined when the graph of the structure and the (relative) lengths of the links are given. To this effect, Section 5.2 first discusses the properties of deployable truss structures and how they are related to the properties of folded configurations. By applying these properties, a procedure for the generation of all possible folded configurations of a given deployable truss structure can then be formulated, and this is discussed in Section 5.3. Once all folded configurations are available it is then possible to determine whether a given deployable structure has certain special properties that may allow a reduction in its number of joints. This final topic will be addressed in Section 5.4.

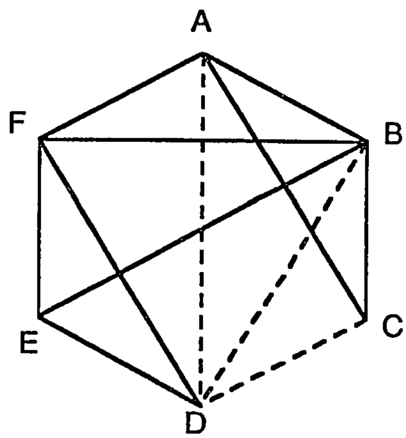
5.2 Properties of Folded Configurations of Truss Structures

Figure 5.1 shows a deployable truss structure and its graph. Note that the graph in this figure has two types of edges : one type is displayed as a solid line and the other as a dashed line. A solid line represents a link that is of a fixed length. The dashed lines represent the *transition links* of the structure, links that contain joints that are locked in the deployed position but otherwise provide the necessary degrees of freedom to the structure to accommodate the transition to a folded configuration. In the truss structure of Fig. 5.1 there are three such joints, located along two diagonal members and one longitudinal member of the truss.

Figure 5.2 shows one of the folded configurations of the deployable truss structure shown in Fig. 5.1, when it is folded to a plane. The transition from the



(a) Structure



(b) Graph

Fig. 5.1 A deployable truss structure and its graph

deployed to the folded configuration (or vice versa) is due solely to the ability of each transition link to vary in length, i.e. to change the distance between the two end points of the transition link. This means that a transition link could be a folding link (a combination of two links connected by one joint) such as in Fig. 5.1a; a telescoping link (two links with a sliding joint) or any other device that allows variation of the distances between the two end points of the link. It also means that the transition links do not constrain the folded configuration so that the folded configuration is dictated only by the (fixed) lengths of the remaining links. Implicit in this observation is the assumption that there is no limitation on the range of length variation permitted by the transition links. The effects of this assumption will be addressed in Section 5.4.

A transition link contains at least one joint that must be locked-up in the deployed position of the structure to ensure the structural stability of the deployed configuration. Each such joint requires a separate device or mechanism (called deployer) that deactivates the joint in coordination with similar joints in the other transition links of the structure. An increase in the number of transition links therefore results in a significant increase in the complexity of the deployer mechanism for the deployable truss structure. Furthermore, each additional joint reduces the load carrying capacity of the deployed configuration. Therefore it is desirable to keep the number of transition links in a deployable structure to a minimum.

To accommodate the transition of a spatial truss structure to a configuration of lower dimension, such as a plane, the selection of transition links must be such that the deployable structure, in the absence of transition links, is not a spatial truss structure and is not over-constrained in the folded configuration. In satisfying these requirements, while minimizing the number of transition links, this leads to the conclusion that the number of links of fixed length in the structure is equal to the minimum number of links required to completely constrain the nodes in the

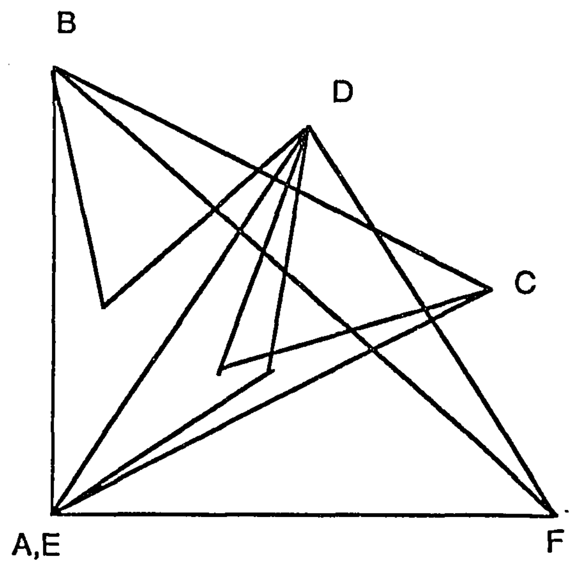


Fig. 5.2 One of the folded configurations of the deployable truss structure shown in Fig. 5.1a.

folded configuration (see Chapter 2). A structure with this characteristic is called a *statically determinate* structure.

In summary, folded configurations of deployable truss structures have the following properties :

1. Transition links do not constrain the folded configuration.
2. Removal of the transition links leads to a 2-dimensional statically determinate truss structure when folded to a plane and a 1-dimensional statically determinate truss structure when folded to a line.

The following section incorporates the properties of folded configurations of truss structures in a procedure for the generation of these configurations.

5.3 Generation of Folded Configurations of Truss Structures

Based on the discussion presented in the previous section, the folded configuration of a deployable truss structure can be constructed by first determining the packaged configuration of the structure without the transition links and subsequently adding the transition links to this configuration. The first step of this procedure can be accomplished by constructing the truss structure according to a given assembly sequence for the structure. This sequence is determined in the following sub-section.

5.3.1 Assembly Sequence

Assuming that the links of a structure are pin-connected, exactly one link is needed to constrain a given node on a line, two links to constrain the node on a plane and three links to constrain the node in a three-dimensional space. It is noted that, although the assumption is generally not valid for deployable truss structures, it allows us to first construct the deployed and folded structure and than

to determine the joint orientations that accommodates the transition between the two configurations. Now consider the planar truss structure in Fig. 5.3, which is obtained from the folded configuration in Fig. 5.2 by eliminating the transition links. Node C in this structure is connected to two other nodes. Thus, the position of Node C can be determined if the locations of the two nodes to which it connects are known. A similar observation can be made for each of the nodes to which Node C is connected, and so on. Applying this to one, two and three-dimensional structures, results in the following theorem :

Theorem 5.1

> An α-dimensional truss structure can be constructed consecutively only if it is possible to eliminate from this structure all but α of its nodes one by one, such that each node connects to α other nodes at the time of elimination.

Theorem 5.1 indicates that the truss structure constructed in this manner is not overconstrained and is therefore a statically determinate truss structure. Furthermore, the theorem implies that the sequence in which the nodes are eliminated can be followed in reverse order to construct the structure. This sequence can be determined using the following algorithm, which operates on the graph of a truss structure.

1. Eliminate all edges of transition links.
2. Extract the vertex of lowest degree as well as the edges incident to this vertex (note, the degree of a vertex is equal to the number of edges incident to it).
3. Add the vertex extracted in Step 2 to the front of the list of previously extracted vertices.
4. Continue with Step 2 until all vertices are eliminated.

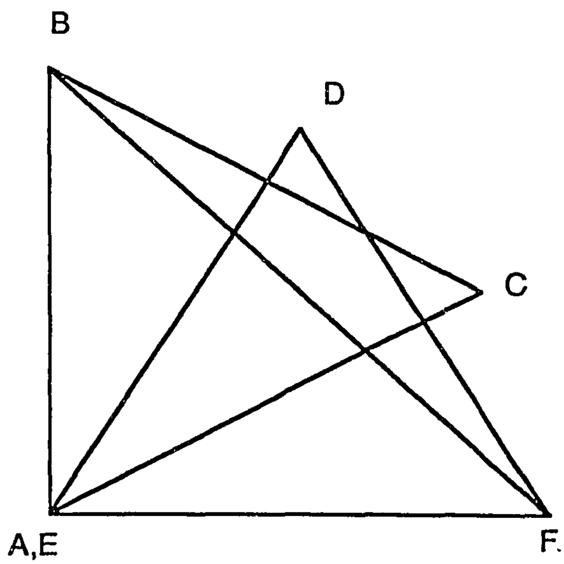


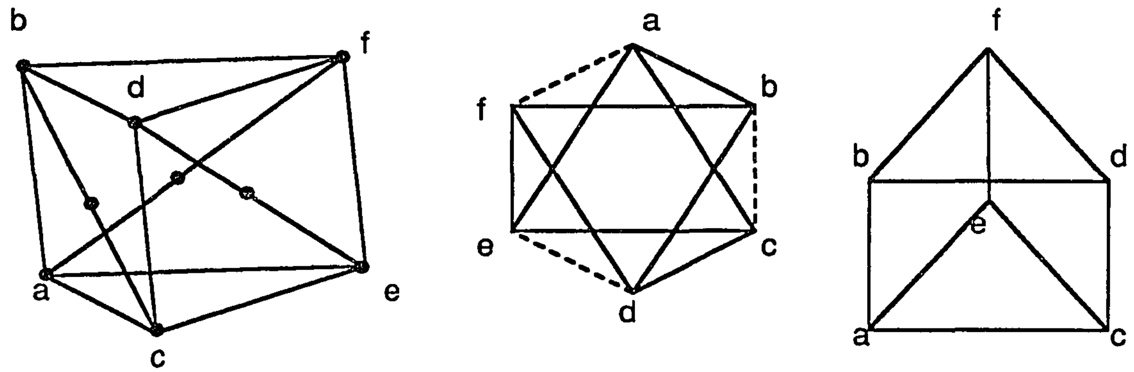
Fig. 5.3 Folded configuration of the deployable truss structure shown in Fig. 5.1a, with its transition links omitted.

As an example, application of the algorithm to the structure in Fig. 5.1 results in the assembly sequence $ABCFED$, although other sequences are also possible since there is generally more than one option when executing Step 2 of this algorithm.

A given truss structure only complies with Theorem 5.1 when each vertex picked in Step 2 is of degree α. Although most truss structures satisfy this criterion, there are some statically determinate truss structures that do not. This means that Theorem 5.1 presents a necessary but insufficient condition for a truss structure to be statically determinate. An example of such a structure is the folded configuration of the structure in Fig. 5.4. In this case, the configuration of the structure can only be found by simultaneously satisfying the distance requirements between the nodes. An exception can be made for the class of truss structures that are built out of at most two different link lengths. It will be shown in the next section that the folded configurations of structures of this type can be constructed according to the assembly sequence determined by the algorithm listed above, even when the structure does not comply with Theorem 5.1.

5.3.2 Construction of a Folded Configuration

The algorithm listed in Section 5.3.1 always leads to an assembly sequence in which the first two nodes are connected to each other. The position and orientation of the link connecting these nodes can be chosen arbitrarily, since this does not effect the final folded configuration. Assume that the third node in the sequence is connected to the previous two nodes. The possible positions of the third node can then be constructed using the technique demonstrated in Fig. 5.5, which indicates that for every additional node there are two possible positions on a plane. Figure 5.6 shows how repetitive application of this technique to the structure shown in Fig. 5.3, for which the assembly sequence was given in Section 5.3.1, leads to all folded configurations of this structure.



(a) Structure

(b) Graph

(c) Folded configuration (transition links not displayed)

Fig. 5.4 A deployable truss structure of which its folded configuration does not comply with Theorem 5.1.

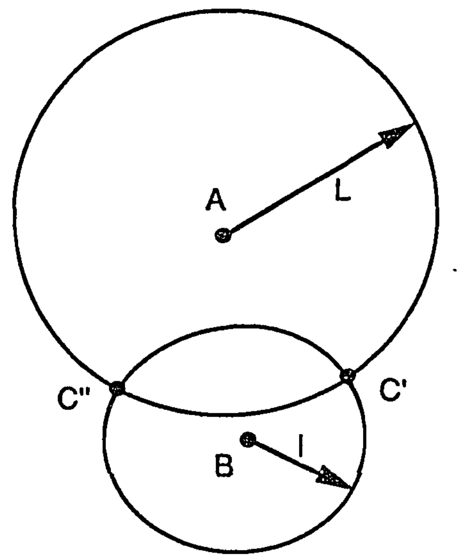


Fig. 5.5 Construction of the possible locations of Node C (C' and C"), which is connected to Node A by a link of length L and to Node B by a link of length l.

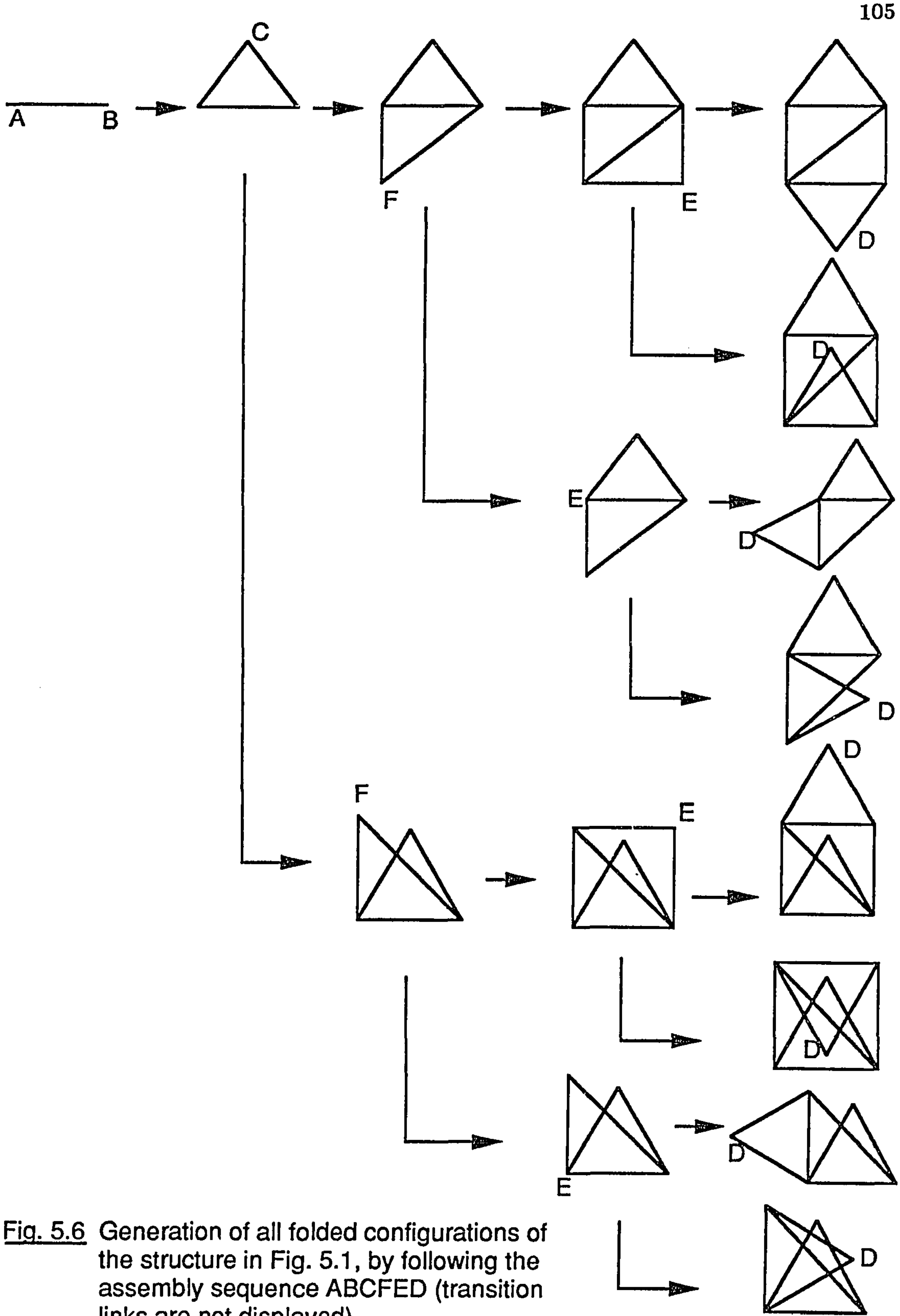


Fig. 5.6 Generation of all folded configurations of the structure in Fig. 5.1, by following the assembly sequence ABCFED (transition links are not displayed)

The technique demonstrated in Fig. 5.6 may encounter the following undesirable conditions when adding a node to the existing sub-structure :

1. The added node is connected by more than two links.
2. The added node is connected by two links to nodes that are further apart than the combined length of the two links.
3. The added node is connected by two links of different lengths to two nodes that lie on top of each other.
4. The added node is connected by two links of the same length to two nodes that lie on top of each other.
5. The added node is attached to only one link.

In the first condition, the possible position of a node can be constructed using any two of the links. The position is than acceptable only when the remaining link(s) fit between the existing nodes (including the added node). If no such acceptable position can be found then the configuration must be eliminated. It is noted that the latter also occurs when the second or third condition is encountered. Conditions 4 and 5, however, are more difficult since there are theoretically an infinite number of options available in positioning the particular node. This problem can be overcome for truss modules, which are characterized by the fact that they can be built with links of unit-length and $\sqrt{2}*$ unit-length. As a result, a certain regularity can be detected in the features of this type of truss structure. Therefore, a reasonable assumption is that the folded configurations of deployable truss modules show a similar regularity. This assumption has lead to the following guideline for choosing the location of a node of this type of truss structure when either Condition 4 or Condition 5 is encountered :

> Given a link that connects a fixed node to the node that encounters Condition 4 or Condition 5, all possible locations of the added node can be found by

positioning the link such that the angle between this link and the link positioned at the very beginning of the construction process is a multiple of $\pi/6$ radians.

Figure 5.7 demonstrates how the previous guideline leads to the folded configurations of the deployable truss structure shown in Fig. 5.4, with its transition links omitted.

5.3.3 Positioning of Transition Links

As discussed in Section 5.2, the locations of the end-points of transition links are determined by the lengths and inter-connectivity of the remaining links in the truss structure. Procedures to determine these locations were discussed in the previous two sub-sections. These locations are independent of whether the transition link is a folding link (combination of two links connected by one revolute joint) or a telescoping link (combination of two links connected by one sliding or one screw joint) or any other combination of links and joints.

Assuming that the transition link consists of a combination of two links and one joint, the location of the transition joint can be determined using the technique demonstrated in Fig. 5.5. An exception is made when the transition link is a telescoping link in which case the joint is located along the line between the end points of the link. In particular cases, such as when the end-points of the transition link coincides, the techniques presented in Section 5.3.2 can then be used to determine the position of the transition joint.

The location of the transition joint can only be found when the transition link is of a known type and when the relative lengths of the two links that form the transition link are known. In the conceptual design stage, this information is often not available or is incomplete. The following section discusses how the availability of all folded configurations of a deployable truss structure could be used to select the types of transition links in a structure.

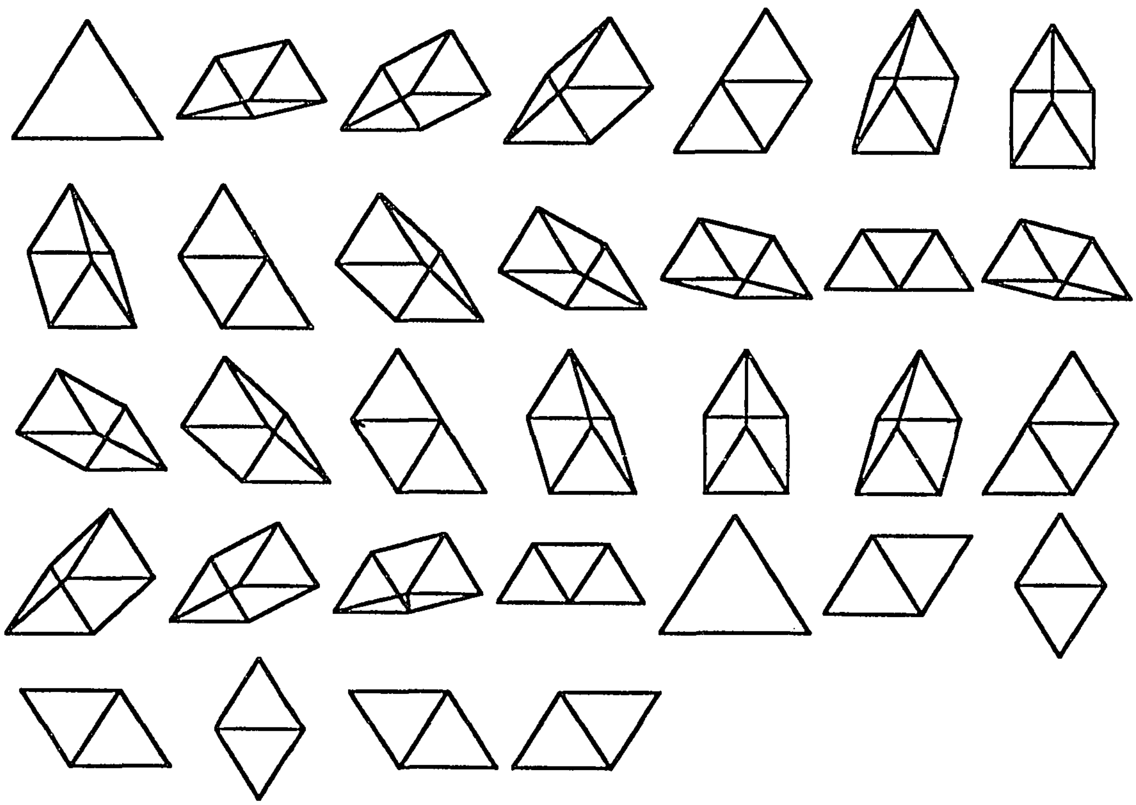


Fig. 5.7 All folded configurations of the structure in Fig. 5.4 (transition links are not displayed)

5.4 Type Selection of Transition Links

Each type of transition link has its own limitations in length variation. For example, the length from end-point to end-point of a folding link must be smaller in the folded position than in the deployed position and the length of a single telescoping link in its folded position should be between half and twice its length in its deployed position. Given the folded and deployed configuration of a truss structure, the relative change in length of the transition link can be determined and matched with the allowable ranges of all possible types of transition links. In the event that there is no such match, it can be concluded that the appropriate type of transition link is not available and that the folded configuration is therefore infeasible. However, if there are multiple matches, then the designer has the option to choose among them. In this way, the designer can influence the number of feasible folded configurations through choosing the number of transition link types.

A special situation occurs when the length of a transition link in the folded configuration is equal to the length of the link in its deployed configuration. In this case, the transition link could first increase and then decrease in length (or vice versa) during the deployment or folding of the structure. This case is, however, not at all likely since it would mean that the motion of this particular link is reversed during the deployment or folding phase. Therefore, the transition link is most likely unnecessary and may be replaced by a link of fixed length, unless it is the only transition link in the structure. Although this is strictly an empirical observation, extensive application of this rule has not lead to any contradicting results.

5.5 Results

The procedures to generate all folded configurations of a deployable truss structure, described in Section 5.3, have been incorporated into a computer program, which was written in FORTRAN. This code has been used to generate all folded configurations of all six-node and selected eight-node truss modules that were created using the techniques described in Chapters 2 and 4. Tables 5.1 and 5.2 summarize some of the results produced by this code.

Table 5.1 shows that the procedures described in Section 5.3 generates folded configurations for all truss modules that fold onto a line. However, no folded configurations were found for three of the 709 six-node truss modules that were created with the techniques described in Chapters 2 and 4. Careful examination of these modules, shown in Fig. 5.8, reveals that the module in Fig. 5.8a cannot be folded onto a plane due to an inappropriate choice of relative link lengths, whereas the remaining modules can only be folded onto a plane by going through a very complex folding sequence, resulting in highly irregular configurations such as the one shown in Fig. 5.8d. Such truss modules are not likely to be candidates in a design selection process, and it is therefore concluded that modules for which no folded configuration can be found form an insignificant category.

Table 5.2 reveals that the number of folded configurations found for a particular module reduces considerably when a constraint is placed on the length variation of its transition links, e.g. when the transition links of the module are either all folding links or all single telescoping links. This is an important observation since the introduction of such a constraint in the conceptual design process can significantly reduce the effort involved in the selection of the most desirable design from among all available designs. This topic is discussed in detail in Chapter 7.

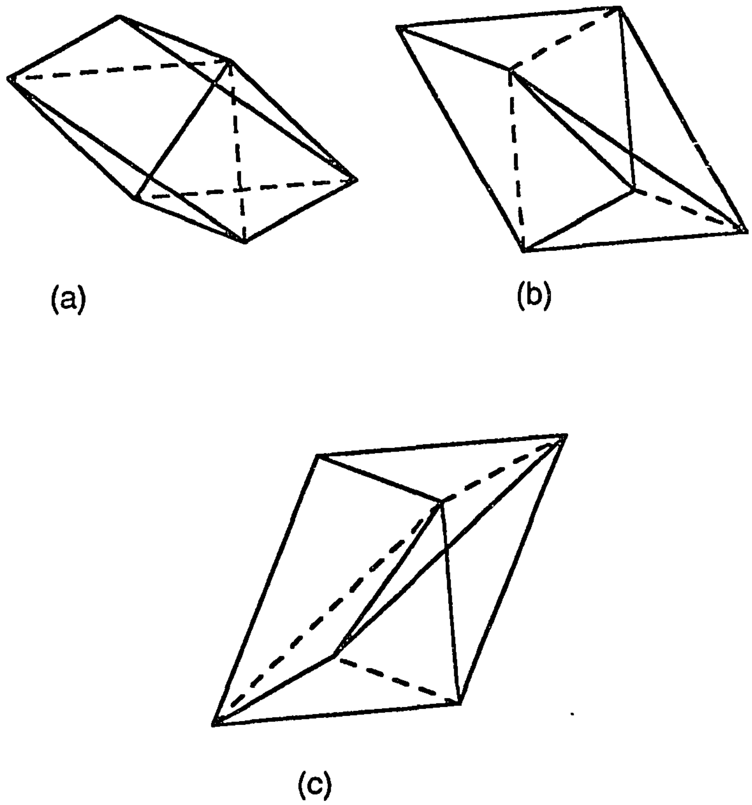


Fig. 5.8 Deployable truss modules for which the techniques introduced in Section 5.4 did not produce folded configurations (transition links are displayed as dashed lines).

Type of module	Number of modules						
	Total	0 transition links eliminated (1)	1 transition links eliminated (1)	2 transition links eliminated (1)	3 transition links eliminated (1)	4 transition links eliminated (1)	> 4 transition links eliminated (1)
6 nodes; 12 links; 3 transition links; foldable onto a plane	709	620	86	0	0	0	0
6 nodes; 12 links; 7 transition links; foldable onto a line	1474	1233	139	82	20	0	0
8 nodes; 18 links; 5 transition links; foldable onto a plane	30,329	14,577	6,500	4,305	3,255	1692	0
8 nodes; 18 links; 11 transition links; foldable onto a line	53,093	28,514	14,790	7,079	2,477	202	31

(1) Number of modules in which could be folded even when the indicated number of transition links were eliminated from the original concept.

Table 5.1 Statistics on the elimination of transition links of deployable truss modules.

Type of module	Number of modules	Number of folded configurations			
		Total	all transition links folding pairs (1)	all transition links telescoping pairs (2)	all transition links folding or telescoping pairs (3)
6 nodes; 12 links; 3 transition links; foldable onto a plane	709	5151	665	1467	3999
6 nodes; 12 links; 7 transition links; foldable onto a line	1474	23616	736	26	6547
8 nodes; 18 links; 5 transition links; foldable onto a plane	30,329	857,971	97,113	79,465	370,486
8 nodes; 18 links; 11 transition links; foldable onto a line	53,093	2,990,912	33,092	100	276,465

(1) All transition links have a length ratio between 0 and 1 (length ratio = ratio of length in deployed and folded configuration).
(2) All transition links have a length ratio between .5 and 2.
(3) All transition links have a length ratio between 0 and 2.

Table 5.2 Statistics on the number of folded configurations of deployable truss modules.

CHAPTER 6

THEORY ON JOINT TYPE SPECIFICATIONS IN THE DESIGN OF DEPLOYABLE TRUSS STRUCTURES

6.1 Outline

The purpose of a deployable truss structure is to serve as a structure in its deployed configuration and can yet be folded compactly onto a plane or even onto a line. The conceptual design of a deployable truss structure must therefore focus on the structural as well as the kinematic aspects of this class of structures. In Chapters 2-5 some of the structural aspects of deployable truss structures have been discussed and procedures for the generation of conceptual designs of such structures have been presented. Inherent to the approach taken, each design produced by the procedures includes information on the geometry of the deployable truss structures in the deployed and folded configuration and the locations of the joints that accommodate the transition between the two configurations. However, the designs do not include information on the joint types nor the inter-connectivity of the joints that allow such a transition. The specification of these joints is the topic of this chapter.

During the transition between the folded and the deployed configuration, a deployable truss structure behaves like a mechanism. A frequently used medium for representing the kinematic structure of a mechanism is a graph which indicates which components in a mechanism are connected to each other with what type

of joint. Therefore, the kinematic structure of a mechanism does not refer to the structural aspects of a mechanism but instead is a description of its kinematic composition.

Instrumental in the derivation of the graph connectivity of the kinematic structure of a deployable truss structure is the introduction of a graph that reveals all possible joint connections. The construction of this type of graph, referred to as a *fundamental graph*, will be discussed in Section 6.2. Since the fundamental graph generally reflects a larger number of joints than is needed for a deployable truss structure, it can be used as a basis for the graph of the kinematic structure. Guidelines for the derivation of the graph of the kinematic structure from the fundamental graph are discussed in Section 6.3.

The specification of joint types in a deployable truss structure, a process known as *coloring* of the graph of the kinematic structure, will be discussed in Section 6.4. This section first reviews the available joint types and the restrictions in their application. The information is than used to formulate a systematic procedure for the coloring of the graph of the kinematic structure of a deployable truss structure.

6.2 Fundamental Graphs of Deployable Truss Structures

6.2.1 Definitions

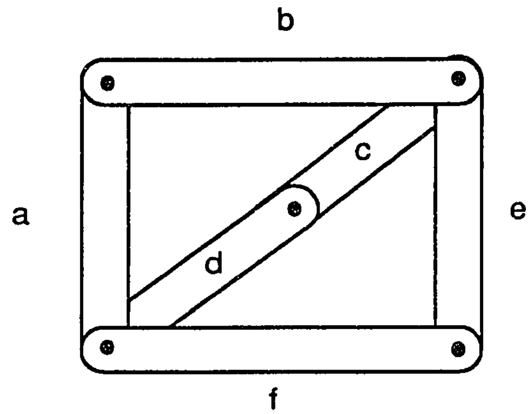
Figure 6.1a shows an example of a (planar) deployable truss structure. The points of intersection of the links of this structure (a,b,c,d,e and f) are called the *nodes* of the structure and are assumed to be the locations of the joints that connect the individual links. Figure 6.1b shows the graph of the kinematic structure of the deployable truss structure given in Fig. 6.1a. The vertices and edges of this graph correspond to the links and joints of the mechanism respectively. It should be pointed out that the definition of the graph of the kinematic structure is exactly

the opposite of the definition of links and joints in the graphs considered in the previous chapters in which the nodes and links were represented by vertices and edges.

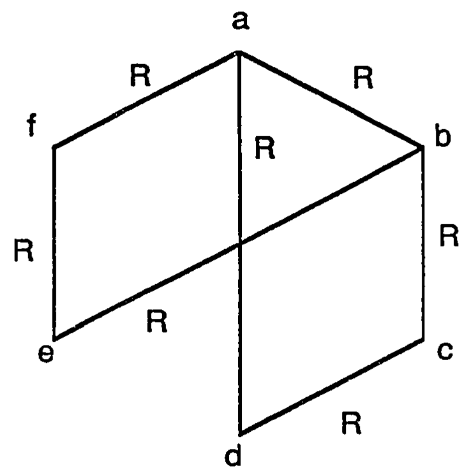
The graph in Fig. 6.1b indicates that, in this particular design, links d and f; and links c and e of the structure in Fig. 6.1a are not connected by a joint. This conclusion could not have been made from Fig. 6.1a, which means that the graph of the kinematic structure contains more information than can be obtained by examining the schematic of a deployable truss structure. To facilitate the transition between the schematic of a deployable truss structure and the graph of its kinematic structure, a graph is introduced in which *all possible* joint connections are represented. Such a graph will be referred to as the *fundamental graph* of the deployable truss structure and can be constructed by placing an edge between any two vertices that represent links that have a common node. As an example, Fig. 6.2 shows the fundamental graph of the deployable truss structure in Fig. 6.1a. Note that this graph does not reveal what type of joint an edge represents. The reason for this is that the joint selections can only be established once the fundamental graph is reduced to the graph of the kinematic structure.

6.2.2 Construction of Fundamental Graphs

Construction of a fundamental graph of a deployable truss structure according to the definition given in Section 6.2.1 generally results in a graph with a large number of vertices and edges. For example, Fig. 6.3 shows that a relatively simple (spatial) deployable truss structure results in a very complex fundamental graph. The complexity of a fundamental graph can be dramatically reduced by making use of the characteristics of deployable truss structures. This leads to the following simplifications :



(a) Planar deployable truss structure



(b) Graph of the mechanism in (a).
(R = Revolute joint)

Fig. 6.1 A deployable truss structure and the graph of its kinematic structure

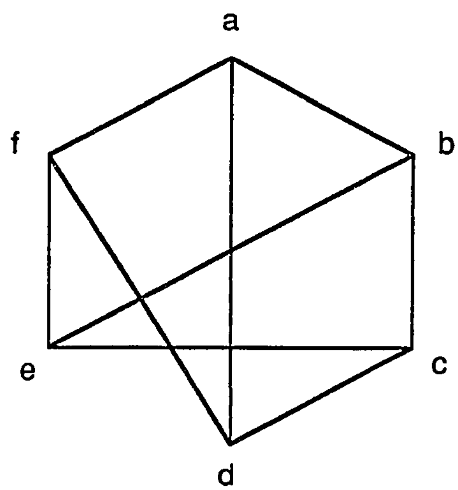
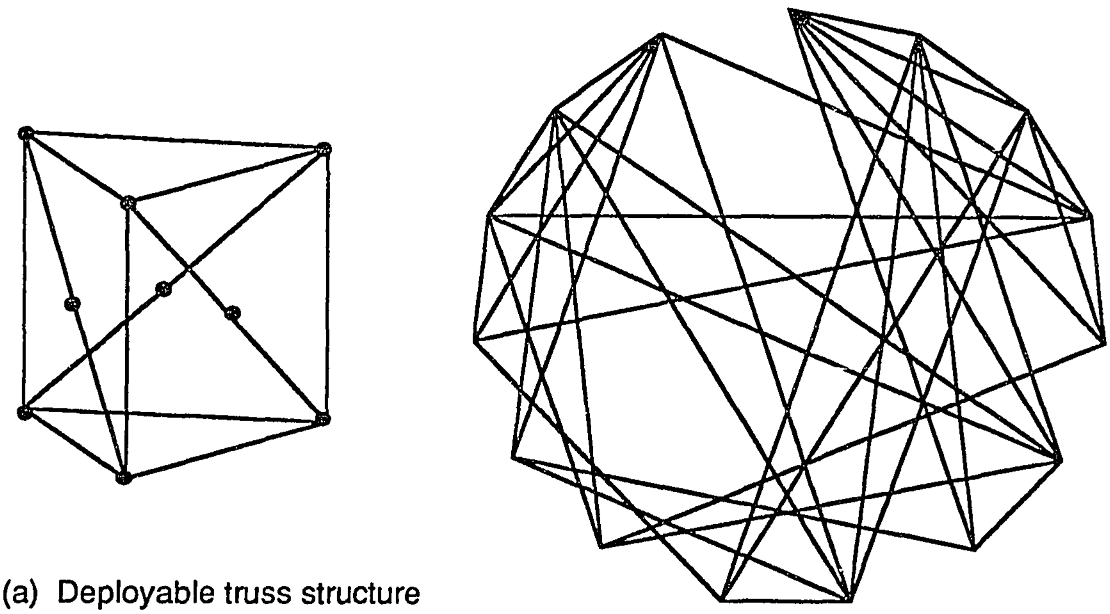


Fig. 6.2 The fundamental graph of the deployable truss structure in Fig. 6.1a.



(a) Deployable truss structure

(b) Fundamental graph of the deployable truss structure in (a)

Fig. 6.3 A deployable truss structure and its fundamental graph

1. Deployable truss structures often contain combinations of links that are unable to move relative to each other. A combinations of links with this characteristic, referred to as a *rigid frame*, is comprised of three links that form a triangle. Joints connecting the links of a rigid frame are meaningless since they do not add to the mobility of the mechanism as a whole. Thus, a rigid frame can be regarded as a single component of the mechanism and represented by a single vertex in the fundamental graph.
2. Deployable truss structures contain nodes that are incident to only two links. However, to ensure that a deployed truss structure is structurally stable, each node of the structure must be incident to at least three links (see Chapter 2). Therefore, a deployable truss structure can only contain a node that is incident to two links when the joint connecting the two links is locked in the deployed configuration, thereby making the two links act as a single link. This joint and the two links it connects, together referred to as a *transition link*, can be represented by one edge and two vertices in the fundamental graph. However, to simplify the graph, the transition link will be represented by a single vertex. To indicate the special features of this link, all edges incident to the vertex associated with this link will be displayed as dashed lines.

By representing each rigid frame and each transition link of the deployable truss structure as a single vertex, the fundamental graph shown in Fig. 6.3b can be reduced to the graph in Fig. 6.4, showing immense simplification in the fundamental graph.

By definition, the connectivity of the graph of the kinematic structure of a deployable truss structure can be derived from its fundamental graph by eliminating an appropriate number of edges of the fundamental graph. Rules for the construction of graphs of the kinematic structure from fundamental graphs are described in the following section.

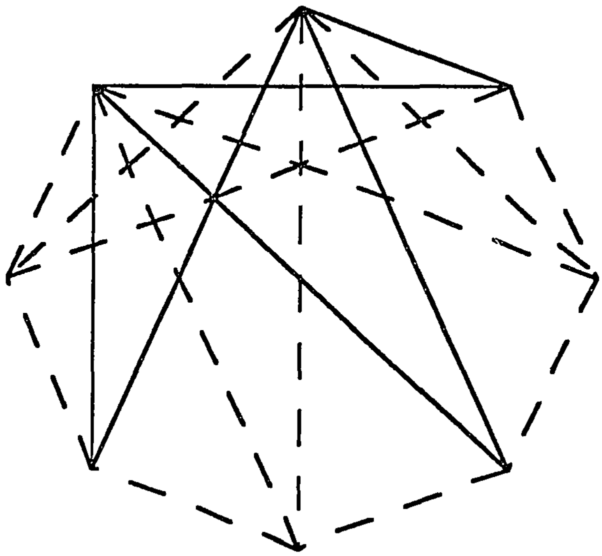


Fig. 6.4 Fundamental graph of the deployable truss structure in Fig. 6.3a (simplified representation).

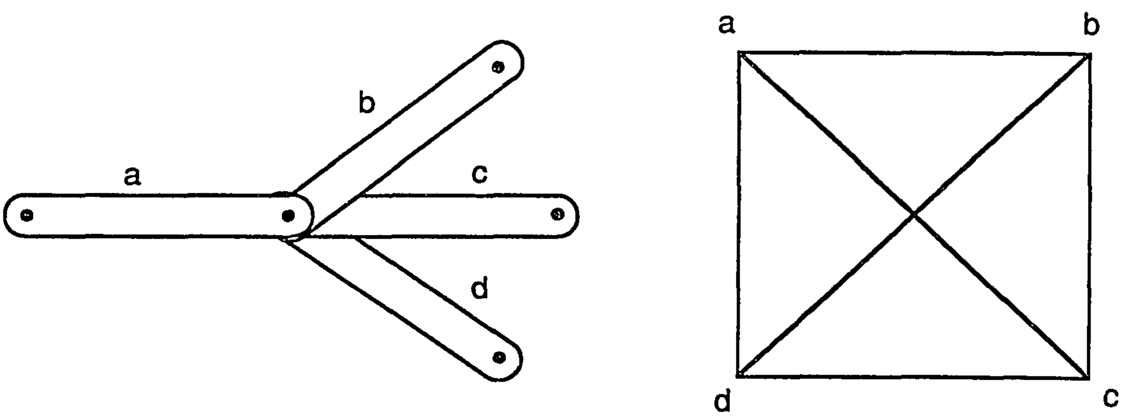
6.3 Construction of Graphs of the Kinematic Structure

6.3.1 General Rule

Figure 6.5a shows a situation where four links come together at a given node. Constructing the fundamental graph of this open-loop mechanism, using the approach described in Section 6.2, results in a graph in which each vertex in the graph is connected to each other vertex in the graph, as shown in Fig. 6.5b. In graph theory, a collection of vertices with this property is commonly referred to as a *clique* [32]. Note that the number of joints needed to connect n links is equal to $n-1$ and that each link must be connected to one of the other links. The graph theoretical term for such a graph is a *tree* [32], the main characteristic of which is that it contains no loops. Thus, if a fundamental graph is a clique, then the associate graph of the kinematic structure can be obtained by eliminating all loops of the clique while ensuring that each vertex is incident to at least one edge.

According to the definition of a fundamental graph, each set of links and/or rigid frames that have one common node produces a clique. For the purpose of this investigation, a set of vertices will only be referred to as a clique when each frame or link represented by a vertex in the set has a node in common with all other frames and/or links represented by vertices in the set. With this definition, a fundamental graph of a deployable truss structure contains as many cliques as there are nodes in the structure. Furthermore, the fundamental graph of a deployable truss structure can be reduced to the associated graph of the kinematic structure by taking out sufficient edges to eliminate all loops in each clique of the fundamental graph. This observation can be stated as follows :

Rule 6.1 **A fundamental graph can be reduced to the graph of the kinematic structure by reducing each clique in the fundamental graph to a tree.**



(a) Mechanism

(b) Fundamental graph of the mechanism in (a)

Fig. 6.5 A mechanism consisting of four links that come together at one node, and its fundamental graph, a clique.

In general, the choices in eliminating edges from a fundamental graph to conform to this rule are arbitrary and will depend on the preference of the designer. For example, reducing the fundamental graph of Fig. 6.5b could result in any one of eight valid alternatives for the graph of the kinematic structure. Unfortunately, there are no general guidelines to determine which of these choices is preferable for all classes of mechanisms. On the other hand, for particular classes such as the class of deployable structures, it is possible to formulate some guidelines for reducing a fundamental graph to the graph of the kinematic structure of the corresponding mechanism. These guidelines will be discussed in the next sub-section.

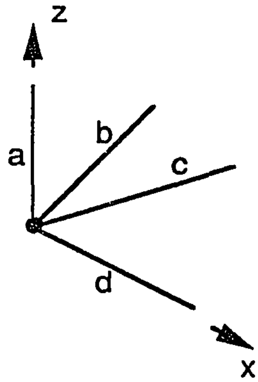
6.3.2 Reduction Rules for Deployable Truss Structures

A deployable truss structure can be regarded as a collection of rigid frames and single links. In general, the connections between these components can be divided into the following two categories :

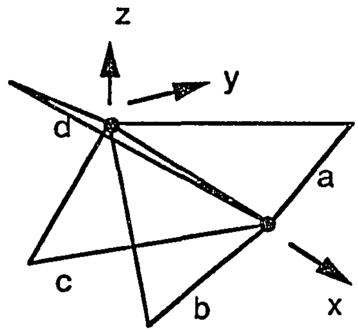
1. Single-point connections (i.e. connections between two links; a link and a rigid frame; and two rigid frames that have one node in common).
2. Two-point connections (i.e. connections between two rigid frames that have two nodes in common)

Figures 6.6a and 6.6c show two situations of multiple connections of the first category; Fig. 6.6b shows a situation of multiple connections of the second category; and Fig. 6.6d shows a situation of multiple connections of the first and second category. Each situation can be represented by the fundamental graph in Fig. 6.6e. The reduction of the fundamental graph to a tree (the graph of the kinematic structure) is the objective of the following discussions.

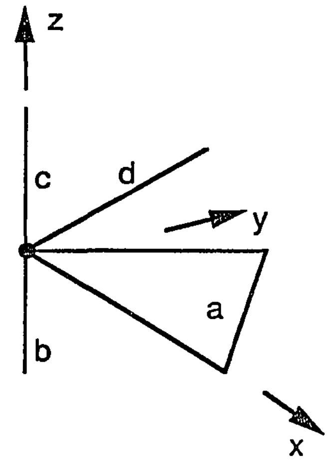
Consider joining four links together as shown in Fig. 6.6a. A well known fact in structural mechanics is that structural weight can be minimized by establishing the shortest load path for the forces in a structure. For a truss structure this means



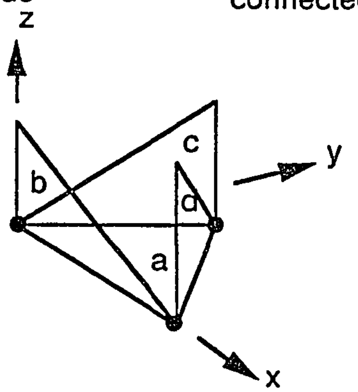
(a) Four links connected at one node



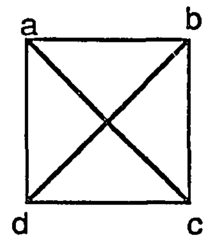
(b) Four rigid frames connected at two nodes



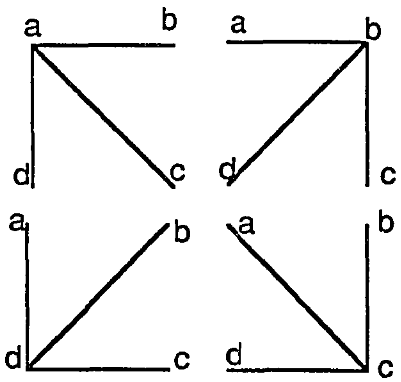
(c) Three links connected at one node on a rigid frame



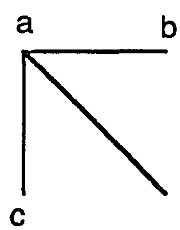
(d) Four rigid frames connected at three nodes



(e) Fundamental graph of the mechanisms in (a), (b), (c) and (d)



(f) All four valid reductions of the graph in (e) for the situations in (a) and (b)



(g) Reduction of the graph in (e) for the situations in (c) and (d).

Fig. 6.6 Connections of links and/or rigid frames, their fundamental graphs and their graphs of the kinematic structure.

that the load paths of all links that come together at one node must intersect at one single point. It also means that structural weight can be minimized by placing joints between two links along the load path of either link. The most obvious location for these joints is the point of intersection of the load paths, e.g. the center of the node in Fig. 6.6a. However, the physical dimensions of links and joints prevents the placement of multiple joints at these points. In that case, the joints must be offset from the node, i.e. placed at a small distance from the node but along the link connecting to the node. Placing a joint along each link that is incident to a given node will result in one joint too many. For example the situation in Fig. 6.6a would lead to four joints whereas only three are needed. By eliminating any one of these joints, a situation arises where only one of the links is connected to all other links. Thus, the graph of the kinematic structure of the situation in Fig. 6.6a must be one of the graphs shown in Fig. 6.6f.

Although the previous discussion focuses on multiple connections of the first category, a similar observation can be made regarding combinations of rigid frames that all share two nodes. This means that the fundamental graph in Fig. 6.6e of the situation in Fig. 6.6b must also reduce to one of the four graphs in Fig. 6.6f. The technique used in both cases to reduce the fundamental graph to the graph of the kinematic structure can be formulated in a rule as follows.

Rule 6.2 **If all edges of a clique in a fundamental graph represent one-point connections (or all edges represent two-point connections), then this clique reduces to a tree in the graph of the kinematic structure, such that one vertex in the tree is connected to all other vertices in this tree.**

Rule 6.2 does not indicate the preference in selecting which vertex is to be chosen to connect to all the other vertices in the clique. Rules indicating such

a preference can only be established when the vertices in the clique have distinguishable properties, e.g. when some vertices represent rigid frames while others represent individual links. Consider the situation in which three single links are connected to a rigid frame, as illustrated in Fig. 6.6c. Examination of Fig. 6.6c leads to the conclusion that the preferable way to offset three joints from the node is to place them along the single links, thereby creating a situation where the single links each connect to the frame but not to each other (as indicated in Fig. 6.6g). This observation is formulated in the following rule.

Rule 6.3 **If a clique contains vertices representing rigid frames, then all edges in this clique must be eliminated that represent joints between two links.**

The third and last rule for the reduction of a fundamental graph to the graph of the kinematic structure focuses on situations involving two-point connections, such as that shown in Fig. 6.6d. Based on the discussion in the previous section it can be concluded that three edges of the fundamental graph in Fig. 6.6e need to be eliminated to arrive at the graph of the kinematic structure. Since the graph of the kinematic structure of the mechanism in Fig. 6.6d must be a tree, at least one of the edges must be eliminated from the loop formed by Vertices a, b and c in the fundamental graph of Fig. 6.6d. Suppose the edge between Vertices a and b in Fig. 6.6d is eliminated. This means that there is no two-point connection between the rigid frame b and the remaining mechanism, a situation that does not reflect that the rigid frames a and b do indeed have a common link and are therefore necessarily connected. A similar observation can be made for the edge between the Vertices a and c, so that the only edge that can be eliminated in the loop formed by the Vertices a, b and c, in Fig. 6.6b, is the edge between vertices b and c. Repeated application of this technique to the fundamental graph in Fig. 6.6d leads to the tree

in Fig. 6.6g. The following rule can be derived by extending the previous discussion to loops consisting of more than three rigid frames.

Rule 6.4 **If a clique in the fundamental graph contains a loop in which only one edge reflects a single-point connection then this edge must be eliminated when reducing the fundamental graph to the graph of the kinematic structure.**

Rules 6.2-6.4 cover all situations that occur in a fundamental graph. However, there are still certain options available to the designer when reducing the fundamental graph to the graph of the kinematic structure. For instance, Rules 6.2 and 6.3 indicate that it is left to the designer to choose the vertex to which all others are connected. Similarly, Rule 6.4 indicates that there is more than one option in reducing a clique if the clique contains several rigid frames that are connected to each other by two-point connections. Nevertheless, Rules 6.2,6.3 and 6.4 minimize the number of choices a designer has to make when reducing the fundamental graph to the graph of the kinematic structure according to Rule 6.1.

6.3.3 Systematic Procedure for the Construction of Graphs of the Kinematic Structure

This sub-section provides a systematic procedure for the construction of the graph of the kinematic structure of a deployable truss structure for which the schematic (and hence, the geometry) is known. The procedure is based on the discussions presented in Section 6.2 and the Sub-sections 6.3.1 and 6.3.2, and is formulated to readily facilitate implementation into an automated procedure. This procedure is listed below.

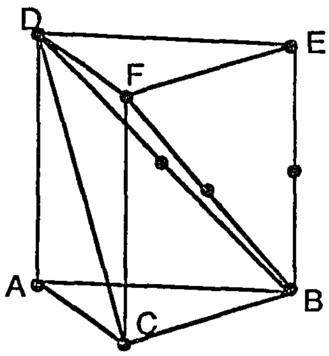
1. List the sets of vertices that represent frames and/or links that have a common node. Note that each set forms a clique in the fundamental graph.

2. Apply Rules 6.3 and 6.4 to eliminate edges from cliques identified in the previous step.
3. If Step 2 reduces a clique to a sub-graph that is not a tree, then apply Rules 6.1 and 6.2 to each clique within this sub-graph.
4. Apply Rules 6.1 and 6.2 to each clique, identified in Step 1, that was not affected by Step 2.
5. Combine all vertices and the remaining edges into one graph. This graph is the graph of the kinematic structure.

To illustrate its execution, the procedure is applied to the construction of the graph of the kinematic structure of the deployable truss structure illustrated in Fig. 6.7a. This structure consists of four rigid frames (formed by the triangles ABC, ACD, CDF and DEF in Fig. 6.7a) and three transition links (formed by the lines BD, BE and BF). The notation for the vertices representing each frame and each link is given in Fig. 6.7b.

The first step of the procedure, identification of cliques in the fundamental graph of a structure, can be carried out by identifying all links and rigid frames that have a given node in common. For example, Node D is part of the frames represented by Vertices b, c and d as well as the link represented by Vertex e. Thus, the Vertices b, c, d and e form a clique in the fundamental graph. Repeating this exercise for every node in the structure leads to the cliques listed in Fig. 6.7c.

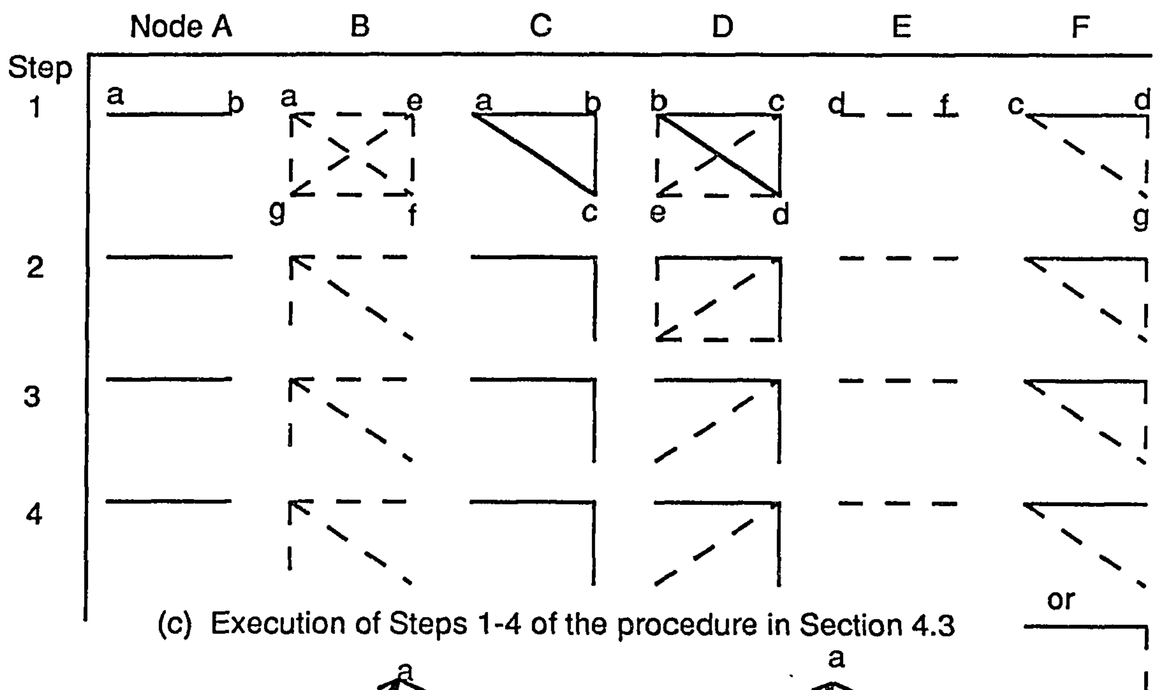
The cliques in Fig. 6.7c can be reduced to trees by executing Steps 2-4 of the procedure. In Step 2, the application of Rule 6.3 leads to the conclusion that all edges within the clique at Node B that are not incident to Vertex a, must be eliminated since Vertex a represents a rigid frame. As a result, the clique at Node B reduces to a tree. The application of Rule 6.4 to Step 2, leads to the elimination of the edge between Vertices a and c in the clique at Node C as well as the elimination of the edge between Vertices b and d in the clique at Node D. The clique at Node



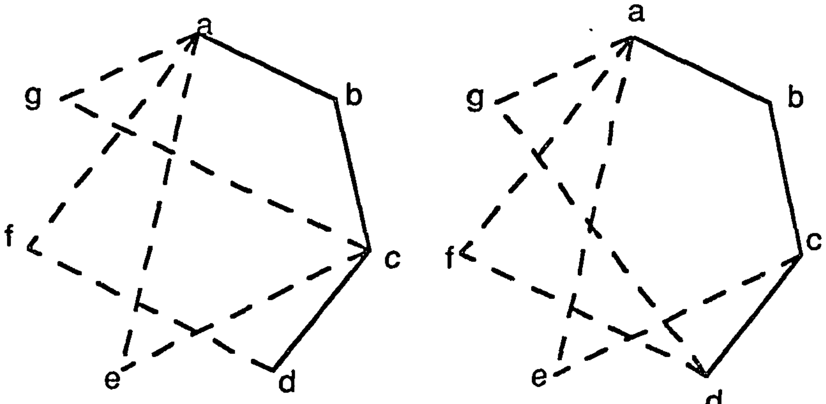
(a) Deployable truss structure

Frame or link in (a)	Vertex representing the frame or link
ABC	a
ACD	b
CDF	c
DEF	d
BD	e
BE	f
BF	g

(b) Vertices representing the links and frames of the structure in (a)



(c) Execution of Steps 1-4 of the procedure in Section 4.3



(d) All possible graphs of the kinematic structure of the deployable truss structure in (a)

Fig. 6.7 Construction of graphs of the kinematic structure of a deployable truss structure

C thereby reduces to a tree, while the clique at Node D does not. The end result of Step 2 is displayed in Fig. 6.7c.

The clique at Node D is the only clique that was affected in Step 2, but that has not yet been reduced to a tree. Thus, the clique at Node D is the only clique that is subjected to Step 3 of the procedure. Figure 6.7c indicates that at the end of Step 2 this clique has been reduced to a sub-graph that contains two smaller cliques, i.e. a clique formed by Vertices b, c and e; and a clique formed by Vertices c, d and e. Application of Rule 6.3 to each of these cliques leads to the observation that exactly one edge of each clique must be eliminated. This can only be accomplished by eliminating the edge between Vertices b and e and the edge between Vertices d and e. Elimination of these edges reduces the subgraph at Node D to a tree, as indicated in Fig. 6.7c.

The only clique that has not yet been reduced to a tree is the clique at Node F. The only rule in Step 4 of the procedure that is applicable to this clique is Rule 6.1. Application of this rule leads to the observation that one of the two edges incident to Vertex g must be eliminated from this clique. Since no other rules apply to this situation, two options remain for the reduction of this clique, as indicated in Fig. 6.7c.

The fundamental graph can be constructed by combining the vertices and edges of all cliques that were identified in Step 1 of the procedure. Similarly, the graph of the kinematic structure can be put together by combining all trees that remain at the end of Step 4. When there is more than one option in reducing a clique to a tree, then all possible graphs of the kinematic structure can be found by combining all unique permutations of trees. Application of this technique leads to two possible graphs of the kinematic structure of the structure in Fig. 6.7a. These graphs are shown in Fig. 6.7d. It is noted that these graphs are incomplete in the sense that they do not reveal what types of joints the edges represent. Specification of these joints, i.e. graph coloring, is the topic of the following section.

6.4 Joint Type Specification

6.4.1 Available Joint Types

A deployable spatial truss structure can become a spatial mechanism by activating (or unlocking) certain joints in the structure. In general, the motion of the resulting mechanism will describe a three dimensional trajectory when going from a deployed to a folded configuration or vice versa. Joint types that allow this kind of motion (excluding those that allow separation of links) are presented in Fig. 6.8.

Revolute and *prismatic* (*slider*) joints are among the most commonly used joints and they allow a single degree-of-freedom (d.o.f.) of motion between the links they connect. Another single degree-of-freedom joint is the *helix* (or *screw*) pair which establishes a linear relationship between the axial translation and the screw rotation. This coupling is absent in the *cylindrical* pair so that this joint permits two degrees of freedom. The *Hooke's* joint (or *universal* joint) allows rotations in two independent directions between the connecting links, thereby creating two d.o.f. Three d.o.f. are permited by *spherical* joints, which allow three independent rotations; and *plane* joints, allowing one rotation and two independent translations.

The joints in a deployable truss structure can be divided into two categories based on the intended use of the joints. These categories are :

1. Joints which can be activated (unlocked) and deactivated (locked).
2. Joints which are always activated (unlocked).

As defined in Section 6.2.2, joints of the first category only occur along transition links. These joints generally provide only one d.o.f. because the mechanism required to lock up a joint with one d.o.f. tends to be less complex than a mechanism that locks a joint with two or three d.o.f. Joints of the second category are not limited by the complexity of a locking mechanism and can therefore permit any number of d.o.f. of motion. However, as discussed earlier, it is desirable that joints

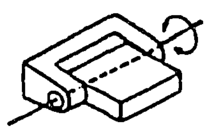
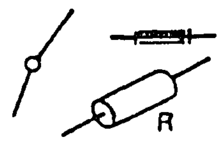
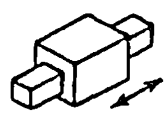
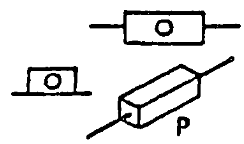
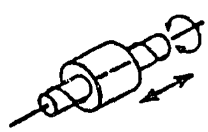
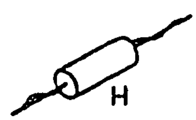
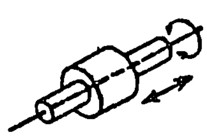
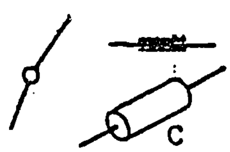
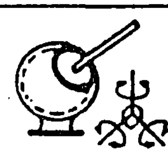
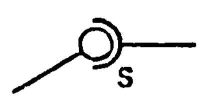
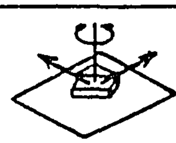
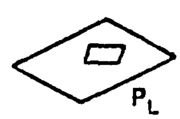
Name of pair	Geometric form	Schematic representations	Relative degrees of freedom between elements of pair
Revolute (*R*)			1
Prismatic (*P*) (slide)			1
Helical (*H*) (screw)			1
Cylinder (*C*)			2
Sphere (*S*)			3
Plane (P_L)			3

Fig. 6.8 Joint types that allow 1, 2 or 3 degrees of freedom

in this category maintain a given load path in the structure and therefore cannot permit motion in the longitudinal direction of the link. This criterion excludes the prismatic, helical, cylindrical and plane joints from the second category, leaving only the revolute, universal and spherical joints as alternatives.

6.4.2 Restriction on Joint Type Selection

Given the number of links and/or rigid frames and the joint types connecting them, the number of independent d.o.f. of a mechanism can be determined using the following "degree-of-freedom equation" [46]:

$$F = \lambda(l - j - 1) + \sum_{i=1}^{j} f_i \qquad (1)$$

where

F = number of d.o.f. of the mechanism

λ = mobility number (3 for planar, 6 for spatial mechanisms)

l = number of links (including ground)

j = number of joints

f_i = number of d.o.f. permited by the i^{th} joint

To enable the application of this equation, known as *Gruebler's equation*, to the graph of the kinematic structure of a deployable truss structure, the equation is simplified as follows. Recall that each rigid frame and each remaining link is represented by a vertex in the graph of the kinematic structure. Furthermore, each transition link is also represented by a single vertex in the graph so that the joint along a transition link is not represented by an edge. As a result, all edges in the graph of the kinematic structure of a deployable truss structure represent either revolute joints, universal joints or spherical joints, which supply 1,2 and 3 d.o.f. respectively. Equation 1 can then be written as follows :

$$5e_r + 4e_u + 3e_s = 6v + v_t - F - 6 \qquad (2)$$

where

e_r = number of edges representing revolute joints

e_u = number of edges representing universal joints

e_s = number of edges representing spherical joints

v = total number of vertices in the graph

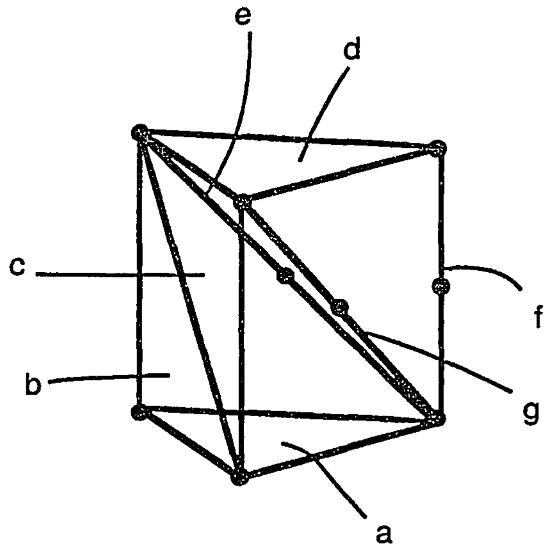
v_t = number of vertices representing transition links

F = number of d.o.f. of the mechanism

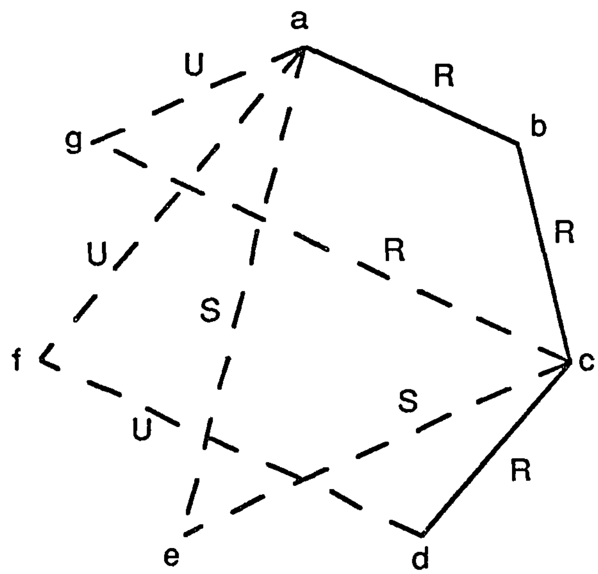
Consider the deployable truss structure in Fig. 6.9a. The graph of the kinematic structure, shown in Fig. 6.9b, indicates that this structure has four revolute joints, three universal joints and two spherical joints. The graph also shows that the deployable truss structure has three transition links since there are three vertices (Vertices e, f and g) that are incident to dashed lines only. Substitution of $e_r = 4$, $e_u = 3$, $e_s = 2$, $v_t = 3$ and $v = 7$ leads to the conclusion that the mechanism in Fig. 6.9a has one d.o.f. Inspection of this mechanism reveals that the loop consisting of Vertices a, b, c and g represents a four-bar with three revolute joints and one universal joint. According to Eq. 2, this four-bar has zero d.o.f. and is therefore rigid unless the four-bar is either a Bennett mechanism [47] or moves within a plane and can therefore be treated as planar. Excluding these cases, the inability of this four-bar to move prevents the total mechanism from deploying or folding completely. Therefore, a deployable structure can only fold and deploy completely when every closed chain that is part of the mechanism has at least one d.o.f. This leads to the following requirement :

Rule 6.5 **A deployable truss structure can only fold and deploy completely when every loop in its graph of the kinematic structure represents a mechanism which has *at least* one d.o.f.**

The number of d.o.f. of each chain in a mechanism can be determined using either Gruebler's equation, Eq. 1, or its derivative, Eq. 2. However, the drawback



(a) Deployable truss structure



(b) Graph of the kinematic structure of the mechanism in (a)

Fig. 6.9 Schematic and graph of a deployable truss structure

of these equations is that they do not reveal the existence of d.o.f. that result from particular combinations of joints. For example, a pair of spherical joints (one at each end of a link) results in one additional d.o.f. about the link's longitudinal axis. In general, this extra d.o.f. is unnecessary and therefore needs to be avoided in deployable truss structures. A similar observation can be made for a combination of two links and one joint, which has a spherical joint at each end of the combination, e.g. the combination formed by the edges incident to Vertex e in Fig. 6.9b. Again this results in one extra d.o.f. since locking up the joint in the middle reduces the combination to a single link with a spherical joint at each end. Hence the following requirement must be satisfied to avoid unnecessary d.o.f.'s in a mechanism:

Rule 6.6 **A branchless section of a loop (i.e. only the vertices at the end points of the section are incident to more than two edges) in the graph of the kinematic structure of a mechanism, must not contain more than one edge that represents a spherical joint.**

The number of d.o.f. of each joint in a deployable truss structure can be chosen arbitrarily as long as the total mechanism has at least one d.o.f. and Rules 6.5 and 6.6 are satisfied. The following sub-section provides a systematic procedure to aid the designer in making such selections.

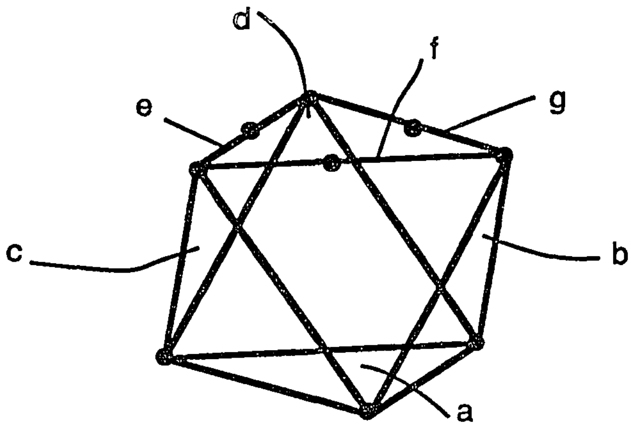
6.4.3 Systematic Procedure for Joint Selection

The primary purpose of the joints in a deployable truss structure is to accommodate deployment and/or folding of the structure. Hence, there is no need to design a deployable truss structure with more d.o.f. than the minimum number required to meet this objective. The minimum number of d.o.f. of a deployable truss structure is equal to one, unless in this case Rules 6.5 and 6.6 cannot be satisfied. In general there are several combinations of joints of different types that

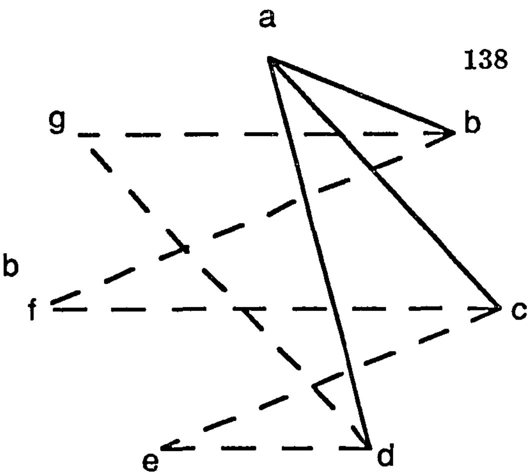
result in a deployable structure with a minimum number of d.o.f. However, practical considerations, such as the complexity of joints of d.o.f. greater than one and their inherent production cost, dictate that the parameters e_r, e_u and e_s in Eq. 2 be chosen such that e_r is maximized (and e_s minimized). The selection of joints in a deployable truss structure is also limited by the requirement that a revolute joint must be present between two rigid frames that are connected at two points, since this type of connection will only permit one d.o.f. The following procedure colors the graph of the kinematic structure so that it meets all these requirements in addition to those given by Rules 6.5 and 6.6.

1. Select parameters e_r, e_u and e_s such that $e_r+e_u+e_s$ is equal to the total number of edges in the graph; Eq. 2 results in $F = 1$; and e_r is at a maximum. Check that e_r is larger than the number of two-point connections in the structure. Otherwise increase F until a valid solution is found.
2. Assign a revolute joint to every edge that reflects a two-point connection.
3. Distribute e_u universal joints, e_s spherical joints and the remaining number of revolute joints over the graph.
4. Check that Rules 6.5 and 6.6 are satisfied. If not, repeat the previous step. If the joint distributions tried in Step 3 are exhausted, select a different set of parameters in Step 1 by reducing e_r.

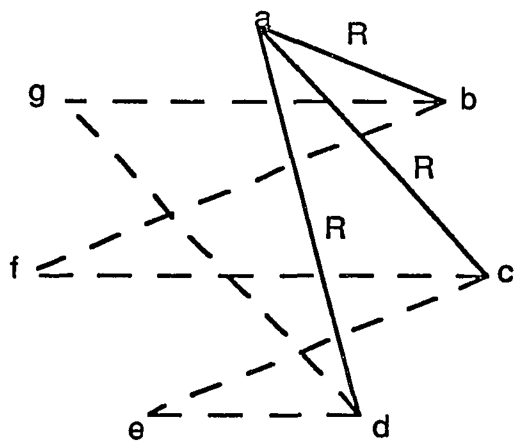
The procedure is employed to specify the joint types of the deployable truss structure in Fig. 6.10a. Note that the (uncolored) graph of the kinematic structure, shown in Fig. 6.10b, has seven vertices, of which three represent transition links; and nine edges, of which three represent two-point connections. This means that a selection of e_r, e_u and e_s must satisfy $5e_r + 4e_u + 3e_s = 38$; $e_r + e_u + e_s = 9$ and $e_r \geq 3$. Maximizing e_r, while satisfying these requirements leads to the selection of $e_r = 4$, $e_u = 3$ and $e_s = 2$ in Step 1 of the procedure.



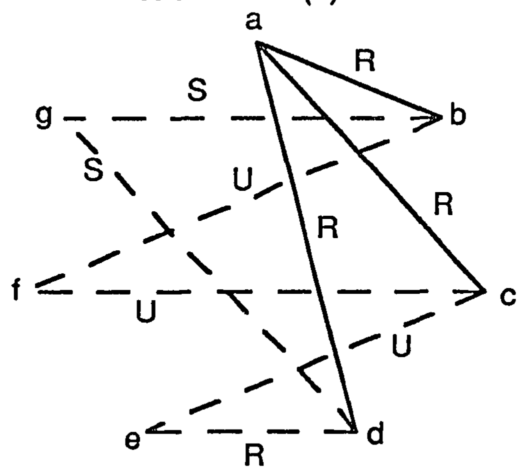
(a) Deployable truss structure



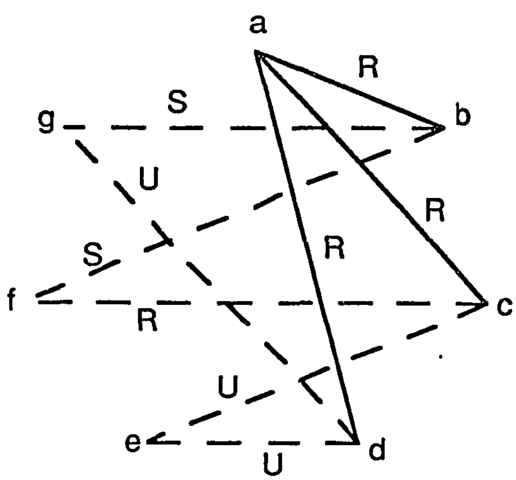
(b) Uncolored graph of the structure in (a)



(c) Graph in (b) after the execution of Step 2 of the procedure in Section 5.3



(d) A possible coloring of the graph in (b).



(e) A valid coloring of the graph in (b).

Fig. 6.10 Coloring of a graph of the kinematic structure of a deployable truss structure

Assigning revolute joints to edges representing two-point connections (Step 2 of the procedure) leads to the graph shown in Fig. 6.10c. One of the possible assignments of the remaining joint types to the edges of the graph (Step 3) is shown in Fig. 6.10d. However, this graph does not comply with Rule 6.5 (the loop comprising of Vertices a, c, d and e represents a chain with less than one d.o.f.) and does also not comply with Rule 6.6 (the two edges connecting Vertices b, d and g both represent spherical joints). Successive execution of Steps 3 and 4 leads to all 72 valid solutions for the selection picked in Step 1. One of these solutions is indicated in Fig. 6.10e.

6.5 Results

The procedures for the construction and coloring of graphs of the kinematic structure of deployable truss structures, described in Sections 6.3 and 6.4, have been incorporated in a computer program which was written in FORTRAN. Tables 6.1 and 6.2 summarize some of the results, produced by this code, for six- and eight-node deployable truss structures that were created using the techniques presented in Chapters 2-4. These tables indicate that it is always possible to produce a joint assignment (i.e. a colored graph of the kinematic structure) for a deployable structure such that it has one degree of freedom during deployment and retraction. Furthermore, the number of unique joint assignments, from which a designer can choose, proves to be enormous. Hence, there is a need for a systematic approach to determine the joint assignments that are most favorable. This topic is addressed in the following chapter.

Type of module	Number of modules	Number of graphs of the kinematic structure.	
		Uncolored	Colored
6 nodes; 12 links; 3 transition links; foldable onto a plane	709	13,710	688,109
8 nodes; 18 links; 5 transition links; foldable onto a plane	30,329	969,821	48,675,586

Table 6.1 Statistics on the number of graphs of the kinematic structure of deployable truss modules.

Type of module	Number of modules						
	Total	with 0 spherical joints (F=1)	with 1 spherical joint (F=1)	with 2 spherical joints (F=1)	with 3 spherical joints (F=1)	with 4 spherical joints (F=1)	with > 4 spherical joints (F=1)
6 nodes; 12 links; 3 transition links; foldable onto a plane	709	8	246	275	159	18	3
8 nodes; 18 links; 5 transition links; foldable onto a plane	1474	49	244	7,504	15,253	6,859	420

Table 6.2 Statistics on the number of spherical joints in deployable truss modules.

CHAPTER 7

AN EXPERT SYSTEM APPROACH TO THE SELECTION OF CONCEPTUAL DESIGNS OF DEPLOYABLE-FOLDABLE TRUSS STRUCTURES

7.1 Outline

Techniques for the generation of conceptual designs of deployable-foldable truss structures, introduced in Chapters 2-6, lead to the creation of an enormous number of conceptual designs. As an example, the deployable truss structures shown in Fig. 7.1 are only two of the 18,543 conceptual designs with six nodes and twelve links that were created using these techniques. Although the availability of a large number of alternative designs is generally considered an asset, a designer is only interested in the design that best serves the application at hand. Selecting such a design among a large number of designs can be cumbersome, since it involves the evaluation and comparison of each available design. However, a designer can be alleviated from the burden of design selection by automating the selection process. Recent developments in mechanism design [27-31] suggest that this can best be achieved using computer programs that are referred to as *expert systems*. The flexible architecture of expert systems allows a designer to easily implement and alter a set of criteria used in the design selection. This has the advantage that the

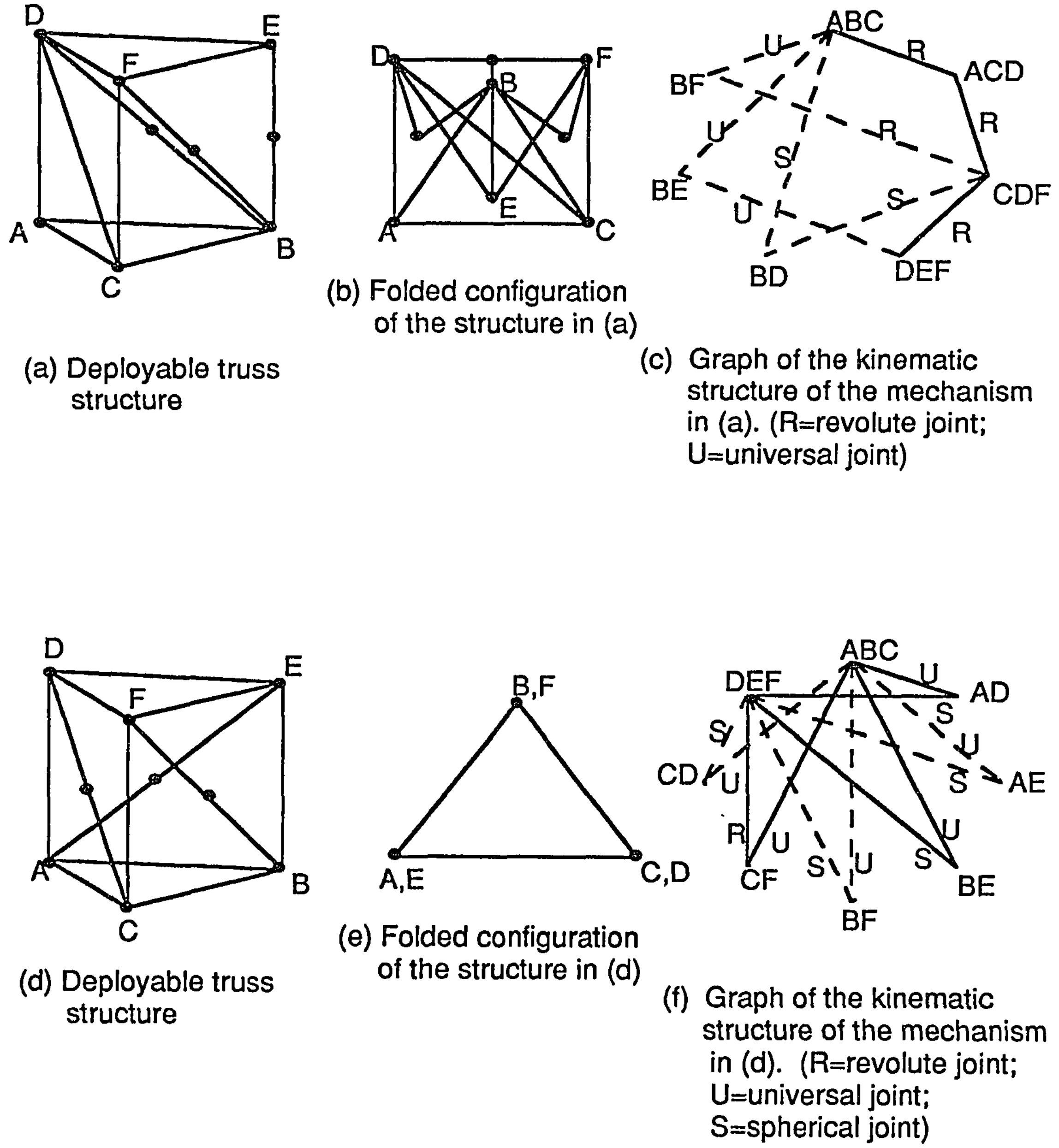
designer can concentrate on defining these criteria and leave the design selection to the expert system.

The following section gives a general introduction of expert systems and describes the steps involved in the formulation of an expert system used for the design selection of deployable truss structures. This section is followed by Sections 7.3 and 7.4, which discuss selection criteria for deployable truss modules and how these criteria can be implemented in a selection process as carried out by an expert system. This article is then concluded with examples concerning the application of an expert system and a general discussion of the results achieved.

7.2 Design Selection Using Expert Systems

An expert system is a construction of computer programs that are utilized to make decisions based on available data. These decisions are inspired by the knowledge of an expert system on a subject concerning the application for which the expert system is used (e.g. the selection of conceptual designs of deployable truss structures). The knowledge of an expert system, generally referred to as a *knowledge base*, must be provided by an expert to the developer of an expert system in the form of a set of *rules*, i.e. a collection of conditions and the actions to be taken if the conditions are met. An expert system that does not contain any knowledge is referred to as a *shell*. Since shells are commercially available, the developer of an expert system only needs to concentrate on gathering knowledge on a given application of the expert system and on defining the rules that will embody this knowledge.

Figure 7.2 shows a schematic of the data flow in an expert system suitable for design selection. In this case, the knowledge base of the expert system consists of rules for the selection of conceptual designs. These rules determine whether a



(a) Deployable truss structure

(b) Folded configuration of the structure in (a)

(c) Graph of the kinematic structure of the mechanism in (a). (R=revolute joint; U=universal joint)

(d) Deployable truss structure

(e) Folded configuration of the structure in (d)

(f) Graph of the kinematic structure of the mechanism in (d). (R=revolute joint; U=universal joint; S=spherical joint)

Fig. 7.1 The conceptual designs of two deployable truss structures

given design meets certain criteria that are imposed by a designer in an attempt to solve a particular design problem. If the design is acceptable, additional rules are applied to satisfy criteria that demand that the best solution to the design problem be found. Hence, a given design could be subjected to evaluation more than once during the execution of an expert system, as is indicated in Fig. 7.2. An important advantage of expert systems is that the design selection can be easily influenced by either adding or subtracting rules, or by prioritizing certain rules. Thus, rules can be tailored to suit a given application. Furthermore, a designer can manipulate the rules in order to determine the effect of a particular criterion on the design selection.

The schematic shown in Fig. 7.2 will be used as the basis of an expert system, developed in this chapter, for the selection of the most favorable conceptual design of a deployable truss structure for a given application. The expert system strictly deals with applications where the deployable truss structure serves as one of many identical modules that together form a much larger deployable truss structure, as illustrated in Fig. 7.3. The selection criteria for these types of structures are discussed in the following section. These criteria, which form the basis of the rules for the design selection, will be discussed in Section 7.4.

7.3 Selection Criteria for Deployable Truss Structures

The selection of a conceptual design involves the comparison of available designs and the rejection of those designs that have less favorable characteristics. The decision to reject a design must be based on certain rules that establish whether a particular characteristic of a conceptual design of a deployable truss structure is less favorable than a characteristic of another design. Although the set of rules used in the selection process is different for each application of large deployable truss structures, each such set focusses on satisfying one or more of the following criteria:

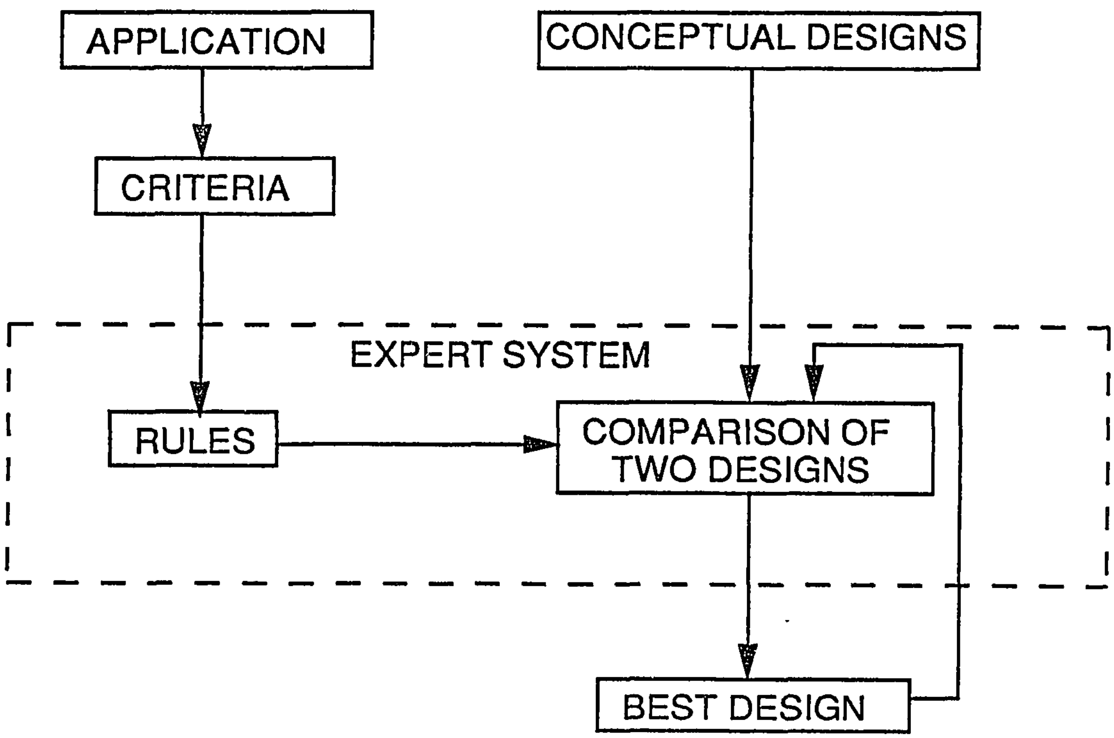


Fig. 7.2 Schematic of an expert system used for design selection

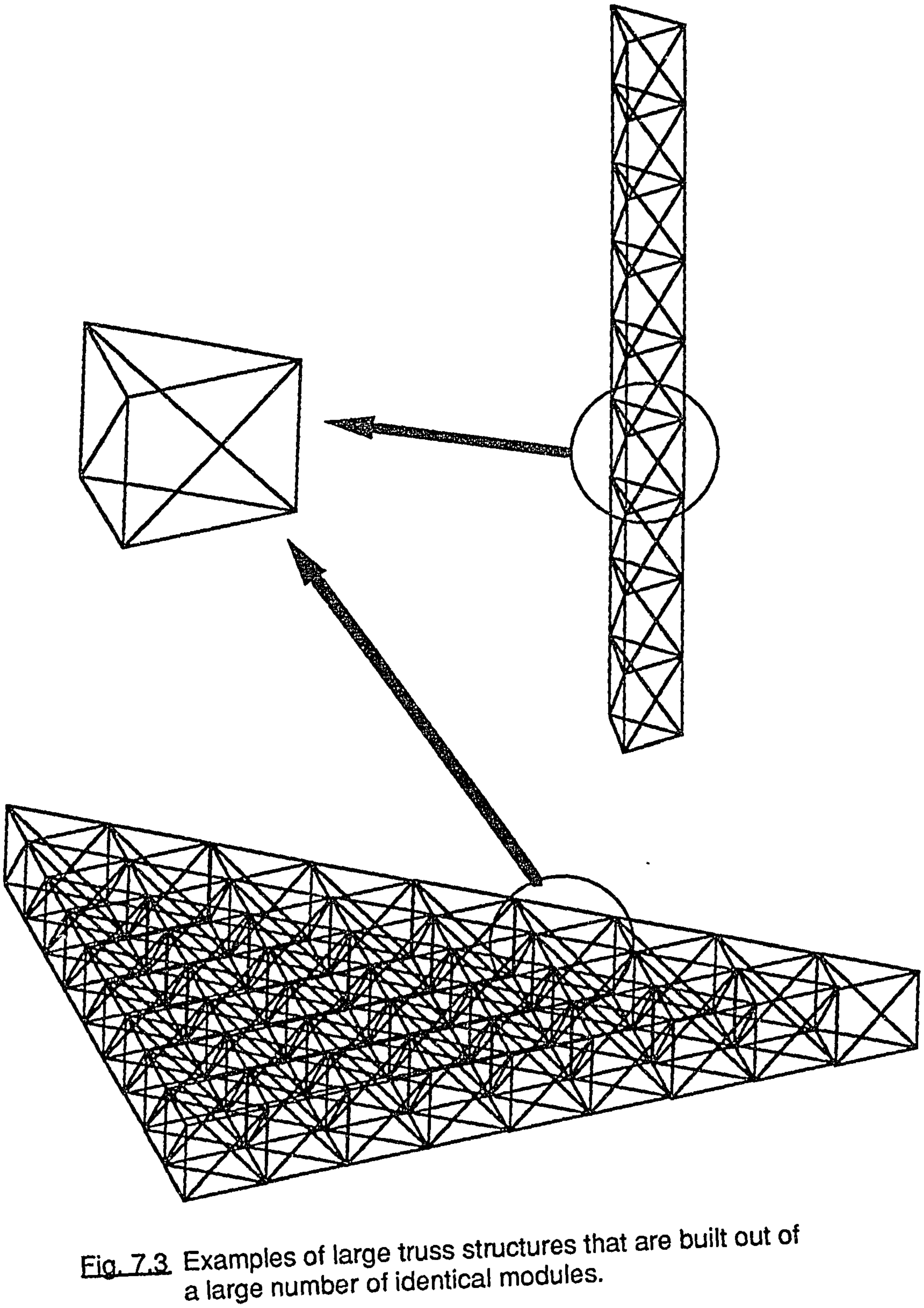


Fig. 7.3 Examples of large truss structures that are built out of a large number of identical modules.

(1) Minimum storage space

(2) Favorable deployment characteristics

(3) Minimum weight

(4) Maximum rigidity in the deployed configuration

It should be noted that it is not always possible to find a conceptual design that meets all the criteria imposed by a given application. In such a case, compromises must be made to obtain the best candidate for a given application. For example, a designer could allow an additional ten percent of weight in return for more favorable deployment characteristics.

The objectives, presented above, form the basis of rules for the selection of designs of deployable truss modules. However, these rules can only be formulated when the relationships between each objective and the characteristics of a deployable truss module are known. These characteristics must be determined from the conceptual design of a deployable truss structure, which merely includes a description of the deployed and folded configuration as well as the joints and joint connectivity. Hence, no information is available on characteristics that only can be obtained from detailed analyses of each individual truss structure (e.g. of the deployment dynamics, static deformations and/or vibrations), since these analyses are very time consuming and are therefore not carried out as part of a selection process involving a large number of design concepts. Nevertheless, an attempt can be made to derive these characteristics through close examination of the remaining characteristics of each conceptual design.

The following section discusses the relationships between the characteristics of deployable truss structures and the criteria of the selection process for this type of structures. Through these relationships, rules for the selection of conceptual designs are formulated.

7.4 Rules for the Selection of Conceptual Designs

Rules for the selection of conceptual designs of deployable truss modules, formulated in the following sub-sections, lead to the rejection or acceptance of a conceptual design based on an evaluation of its characteristics. A designer can control the design selection process by prioritizing and or omitting certain rules, thereby tailoring the selection process to suit a particular application. Hence, the decision to apply a given rule lies with the designer, who must decide whether the rule is applicable to the design problem under consideration.

For simplicity, it is assumed that only conceptual designs generated using the techniques reported in Chapter 2-6 (e.g. the modules shown in Fig. 7.1), are available. It is further noted that each rule concentrates on a certain aspect associated with a one of the criteria listed in the previous section. The rules listed below are not unique, but they have proven to be sufficient in selecting designs that meet one or more of the criteria listed in Section 7.3

7.4.1 Minimum Storage Space

A deployable truss structure is either foldable to a plane or, at times, to a line. Due to the absence of information on the cross-sectional dimensions of the links, the storage space required for a deployable truss structure can only be expressed in terms of the area or length occupied by the folded configuration on the plane or line to which it is folded (see Fig. 7.4). Nevertheless, observations concerning the height of the package in Fig. 7.4a can be made by determining the maximum number of links that cross each other in the folded configuration, thereby indicating the number of links that are stacked on top of each other. Similarly, observations can be made concerning the radius of the package in Fig. 7.4b by determining the

maximum number of links that lie on top of each other at any given point along the length of the package.

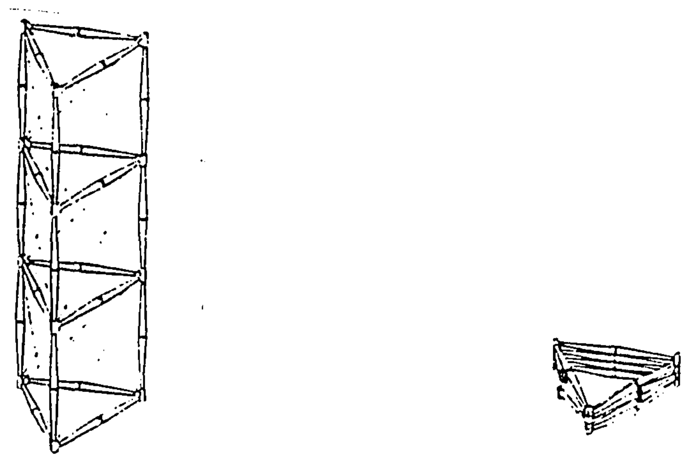
The previous discussion leads to the formulation of the following rules which can be applied to determine which of the available conceptual designs requires the least storage space:

Rule 7.1 Given two designs, reject the design of which the folded configuration requires a larger area to store it on a plane or a larger length to store it on a line.

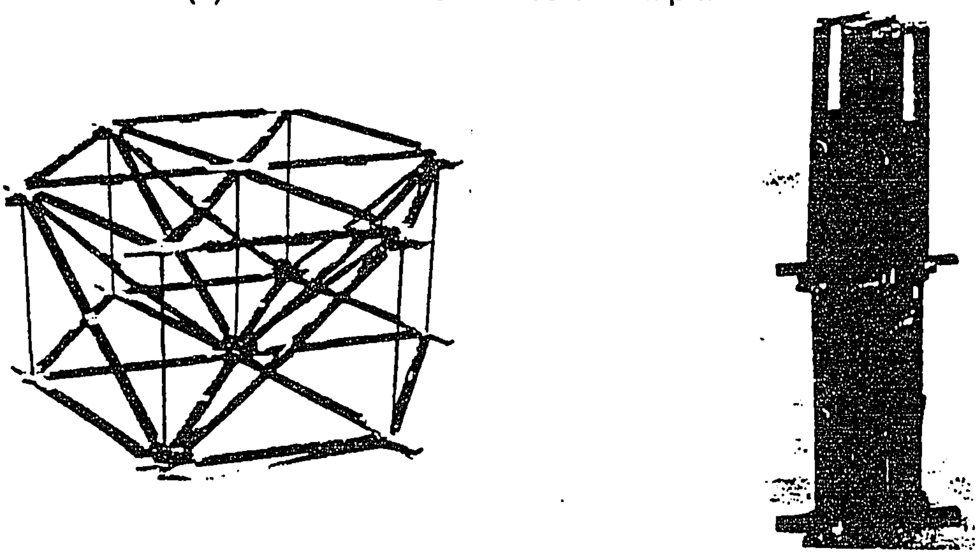
Rule 7.2 Given two designs, reject the design which, in its folded configuration, has the largest number of links crossing any one link of the module.

It should be noted that Rule 7.2 must be applied with caution since there is no guarantee that the application of this rule does not lead to the elimination of valuable designs. Furthermore, when calculating the area of a structure that is folded onto a plane, consideration must be given to requirements concerning the transportation of the folded configuration. For example, consider a truss structure for which the folded configuration is shown in Fig. 7.5a. To facilitate its transportation, the folded truss could be wrapped in plastic foil, placed in a cylinder or box, or placed in any other type of storage medium. When the configuration is wrapped in foil, the storage area should be calculated by determining the area enclosed by the lines describing the shortest circumference around the configuration (Fig. 7.5b). Similarly, for a folded truss structure that is packed in a cylinder (or box), the storage area is equivalent to the smallest circle (or rectangle) that encloses the folded configuration (Figs. 7.5c and 7.5d).

Implicit in the formulation of Rules 7.1 and 7.2 is the assumption that each deployable truss module has comparable overall dimensions. This is true for all modules generated with the techniques introduced in Chapters 2-6, since each module consists of links of unit length and $\sqrt{2} *$ unit length. However, depending on the

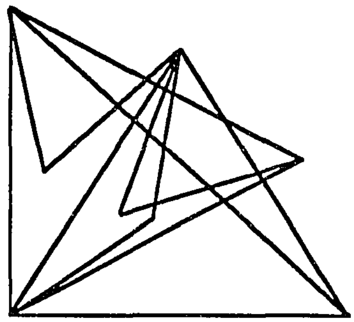


(a) Truss module that folds onto a plane

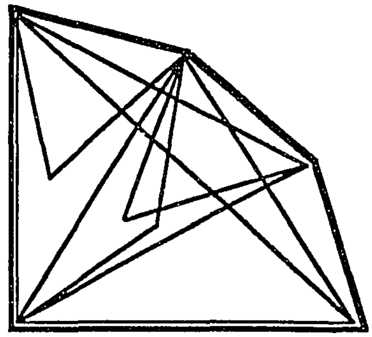


(b) Truss module that folds onto a line

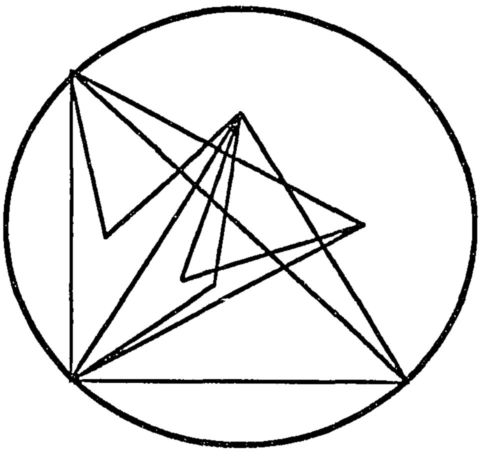
Fig. 7.4 Two truss modules and their packaged configurations



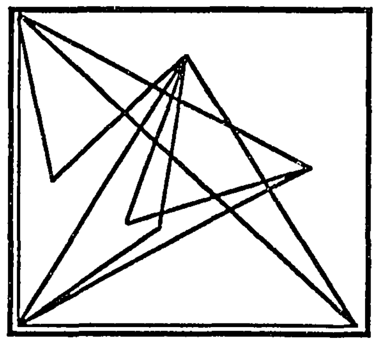
(a) Folded configuration of a deployable truss structure



(b) Lines decribing the shortest distance around the configuration in (a).



(c) Smallest circle enclosing the configuration in (a).



(d) Smallest box enclosing the configuration in (a).

Fig. 7.5 Three different methods of determining the area of the folded configuration of a deployable truss structure

relative placement of nodes, not all modules have the same height as expressed in terms of unit length. This difference has implications on the storage space and/or weight of a deployable truss structure. For example, by leaving the height unchanged, the storage area and weight of a deployable truss structure increases if it uses modules of less height since more modules are needed to create a mast as in Fig. 7.3a. By increasing the height to that of other modules, the storage area increases quadraticly when it is folded to a plane while the weight of the structure remains almost unchanged (see Section 7.3.3). Hence, extreme caution must be taken when comparing two modules of different height.

7.4.2 Favorable Deployment Characteristics

Characteristics related to the deployment of a deployable truss structure focus on the motion of the individual components during the deployment (or folding). Observations can be made concerning the motion of a component by determining the change in position and orientation between the folded and deployed configuration of the structure. For example, consider the deployable truss structure in Fig. 7.1a. Close examination of this figure reveals that, during deployment, the top plane (made up by Nodes A, B and C) of the structure must rotate 0 degrees about the z-axis and 180 degrees about an axis perpendicular to the z-axis. Implicit in this observation is the assumption that the rotation angles show a continuous increase during deployment, i.e. the angles do not increase to more than 0 and 180 degrees respectively and then decrease to assume their final values. This assumption will be maintained for all deployable truss structures due to the lack of a complete deployment analysis for each of these structures.

The energy needed to deploy a structure is assumed to be proportional to the rotation angles of the top plane of a module and the translations of the top plane in the direction of deployment. In these cases, the energy could be estimated using

the following expression, which is a linear combination for the energies needed for each direction in which the top plane of a module is translated or rotated.

$$E = (1 + c_e)(a_1 d_z + a_2 d_{xy} + a_3 \phi_z + a_4 \phi_{xy}) \tag{1}$$

where

E : Energy needed for deployemnt

d_z : translation in z-direction

d_{xy} : translation in a direction perpendicular to the z-axis

ϕ_z : rotation angle about z-axis

ϕ_{xy} : rotation angle about an axis perpendicular to the z-axis

c_e : correction factor for elastic deformation (≥ 0)

a_i : empirically obtained parameters, $i = 1, ..4$

It is noted that, although Eq. 1 is useful in the comparison of designs of deployable truss structures, conclusions can only be based on large differences in the outcome of the equation. However, the accuracy of the equation improves when comparing two structures that have approximately the same values for d_z and d_{xy} and that only require rotation about the z-axis ($\phi_{xy} = 0$) or only require rotation about an axis perpendicular to the z-axis ($\phi_z = 0$), since a conclusion is less sensitive to inaccuracies of the empirically determined parameters.

Another discriminating factor in the comparison of conceptual designs of deployable truss structures is the number of links in a structure that contain joints along their lengths. These links, referred to as transition links, can be recognized by the fact that the joints located along their links causes the structure to be unstable in its deployed configuration unless the joints are equipped with locking mechanisms that secure the joints when the structure is deployed. The number of transition links in a structure is an indication of the complexity of the supplemental mechanism that

is required for folding the structure, since each lock must be released (activated) before folding can take place. Therefore, the fewer the number of transition links, the easier it is to initiate folding of a structure, and the less complex is the folder or deployer mechanism.

In cases where the driving force of the deployment of a deployable truss structure is applied only in one direction, transition links in the direction perpendicular to the direction of deployment will only indirectly receive the energy that is needed for deployment. For example, it can be assumed that the module in Fig. 7.1a is deployed by a force (or moment) along the longitudinal axis of the mast. Therefore, the force acting on a transition link situated in one of the two parallel planes of the module is brought about by the tendency of the longitudinal links of the module to straighten. This force decreases when each longitudinal link comes closer to its final position, so that the force on the longitudinal link must increase to ensure that the transition link deploys fully. To reduce the chance that a structure does not completely deploy, transition links in directions perpendicular to the deployment must be limited.

Summarizing, the following rules can be applied in the selection process to ensure that a chosen design has favorable deployment characteristics :

<u>Rule 7.3</u> Given two designs, reject the design that needs more energy to deploy.

<u>Rule 7.4</u> Given two designs, reject the design that uses more transition links.

<u>Rule 7.5</u> Given two designs, reject the design that contains more transition links in a direction perpendicular to the direction of deployment.

7.4.3 Structural Weight

The structural weight of a deployable truss structure is comprised of the weight of the links and nodes in the structure. Nodes that are located at the intersections of

the links will be aggregated to include the joints between the individual links. Hence, the weight of each individual node is made up of the weight of rigid connections, joint connections and, if applicable, locking mechanisms located at the node. The contribution of each of these components to the total weight can be established empirically, thereby allowing a rough estimate of the total weight of the structure using the following equation:

$$W = \alpha l + \beta_1 r + \beta_2 j + \beta_3 j_l \qquad (2)$$

where

W : total weight

l : # of links

r : # of rigid connections

j : # of joints

j_l : # of joints equipped with locking mechanisms

α,β_i : empirically obtained parameters

It is noted that the nodes of a structure generally represent the majority of the total weight of the structure. Among the components of the nodes, the joints are in most cases heavier than rigid connections and locking mechanisms are even heavier than joints. In these cases the parameters in Eq. 2 are related as follows : $\alpha < \beta_1 < \beta_2 < \beta_3$. It should also be noted that Eq. 2 is not effected by the lengths of the individual links of a deployable truss module, nor the dimensions of the module itself. The effects are ignored since the dimensions of each module are considered comparable (see Section 7.3.1).

Summarizing, the objective of finding a deployable truss structure which has a minimum weight can be realized using the following rule:

Rule 7.6 Given two designs, reject the heavier design.

7.4.4 Structural Rigidity

The ability of a deployable truss structure to withstand external loads is not only affected by the material and cross-sectional properties of the individual members, but also by the configuration of the truss structure, i.e. the inter-connectivity and lengths of the individual links; and the number, type and location of joints in the structure. The effect of the configuration on the rigidity of a deployable truss structure depends on the type of loading that is expected when the truss structure is deployed. Given the type of loading, observations can be made regarding the effectiveness of the structure in carrying the load.

Joints, even those that are locked when the structure is deployed, have a tendency to weaken the deployed configuration, since each joint interrupts the load path of an external load. To minimize this effect, an attempt must be made to reduce the number of joints, the number of degrees of freedom provided by the joints and the number of degrees of freedom provided by the joints to any one link of the deployable truss structure. For example, of the two links shown in Fig. 7.6, Link A will have the lowest buckling load.

Summarizing, the objective of finding a deployable truss structure which has a maximum rigidity can be realized using the following rules:

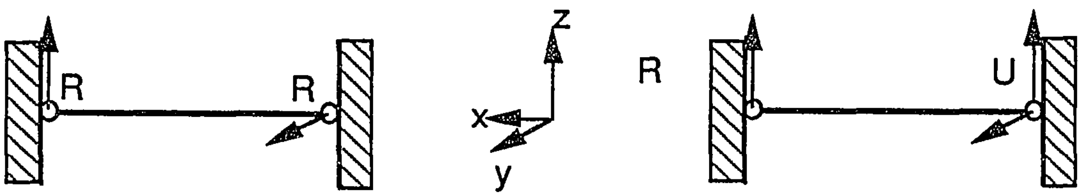
Rule 7.7 Given two designs, reject the design which contains more joints.

Rule 7.8 Given two designs, reject the design which contains a larger number of degrees of freedom provided by the joints.

Rule 7.9 Given two designs, reject the design which contains a link that has more degrees of freedom available than any link of the other design.

Rule 7.10 Given two designs, reject the design that has more transition links that are subject to compression.

Rule 7.11 Reject a design if it contains prismatic joints.



(a) Link A (b) Link B

Fig. 7.6 Schematic of two links that are embedded in a deployable truss structure (R=revolute joint; U=universal joint; arrows indicate hinge angles).

The discussion in the previous sub-sections has concentrated on the formulation of rules for the selection of conceptual designs of deployable truss modules. The implementation of these rules in an expert system is the topic of the next section.

7.5 Expert System Formulation

For this research, a CLIPS-shell [48,49] was chosen as the basis of an expert system for the selection of conceptual design of deployable truss structures. It should be pointed out, however, that any other shell would have been adequate provided that the schematic shown in Fig. 7.2 could have been implemented. As an illustration of the data preparation and knowledge base representation required for this particular expert system environment, consider the two designs in Fig. 7.1 of which the characteristics are determined in accordance with the discussions in Section 7.4. Let their characteristics be defined as follows:

(Design 1 : nodes 6, links 12, area 1.5)

(Design 2 : nodes 6, links 12, area 1.2)

The following lines define Rule 7.1 in this particular expert system environment:

```
(defrule minimum-area
    ?fdat1 ← (Design ?number1 ?? area ?area1)
    ?fdat2 ← (Design ?number2 ?? area ?area2)
    test(≠ ?number1 ?number2)
    test(> ?area1 ?area2)
    ⇒
    retract ?fdat1)
```

The rule defined above carries the name "minimum-area" and assigns the character strings of two designs (stating its characteristics) as variables "?fdat1" and "?fdat2" (e.g. "?fdat1" is equivalent to the character string "(Design 1 : nodes 6, links 12, area 1.5)"). By recognizing variables identified with one question mark as single field variables and variables identified with two question marks as multiple field variables, this rule assigns the value "1" to ?number1, "2" to ?number2, "1.5" to ?area1 and "1.2" to ?area2, when applied to the strings listed above. The rule then tests whether two distinct designs are being compared and then determines if ?area1 is greater than ?area2. Since this is true, the rule proceeds with eliminating the string assigned to ?fdat1, which means that Design 1 is rejected.

The previous two sections have concentrated on the formulation and implementation of rules in an expert system that is used for design selection. It should be pointed out, however, that whether a given rule is used in the design selection depends on the application in which the design is to be used. Examples are provided in the following section.

7.6 Design Selection Using Expert Systems, Examples

In the following two sub-sections, examples are given for the application of an expert system in the design selection of deployable truss modules. The modules that are subjected to evaluation are those that were generated using the techniques described in Chapters 2-6, which is our source for such structures. Each example involves the selection of a set of rules deemed appropriate to the application of the deployable truss module. However, no claim is made that another selection of rules would have been inappropriate.

7.5.1 Example 1

A deployable truss structure is to be designed to serve as an erectable communications tower. Since this structure is meant to function under emergency conditions, the structure must be capable of withstanding significant bending moments as caused by wind and earthquakes. To allow a wide range of applications, the structure must be easy to deploy and light-weight. The storage area of this structure is considered of minor importance.

A designer could make the assumption that the mast is built out of modules that each have three links in longitudinal direction (e.g. the modules shown in Fig. 7.1). A first attempt to find the best solution to the design problem could be made by first applying Rules 7.7-7.11 and then Rules 7.3-7.6 to conceptual designs of which the module folds onto a plane. Selecting $c_e = 0$ (no deformation of the links during deployment), $a_1 = a_3 = 10$ and $a_2 = a_4 = 1$ in Eq. 1 (thereby penalizing rotations about and translations in the direction of an axis perpendicular to the vertical direction of the mast); and $\alpha = \beta_1 = 1$, $\beta_2 = 2$ and $\beta_3 = 4$ in Eq. 2, the expert system produces the deployable truss module in Fig. 7.7 as the best solution. Applying the same rules to conceptual designs of modules that fold onto a line leads to the selection of the design illustrated in Fig. 7.8. This figure shows that a considerable savings in storage space can be achieved by allowing three more transition links and an increase in weight caused by more joints.

7.5.2 Example 2

An attempt is made to find a deployable alternative for the truss-frame of the proposed U.S. Space Station, which consists of seven mast-segments (see Fig. 7.9). Each segment consists of modules with eight nodes and eighteen links. Due to the limited capacity of launch vehicles, both weight and storage space are considered the main criteria in the design selection.

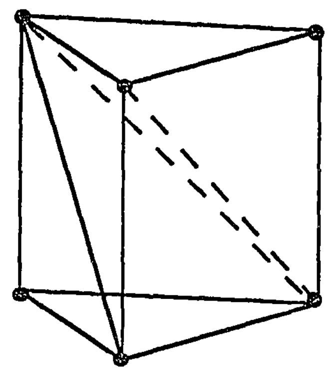


Fig. 7.7 A truss module that is foldable onto a plane and which was selected Example 1 of Section 7.5.1 (note: transition links displayed as dashed lines).

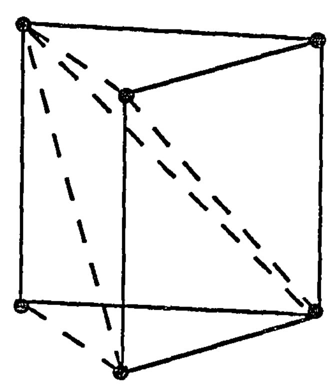


Fig. 7.8 A truss module that is foldable onto a line and which was selected Example 1 of Section 7.5.1 (note: transition links displayed as dashed lines).

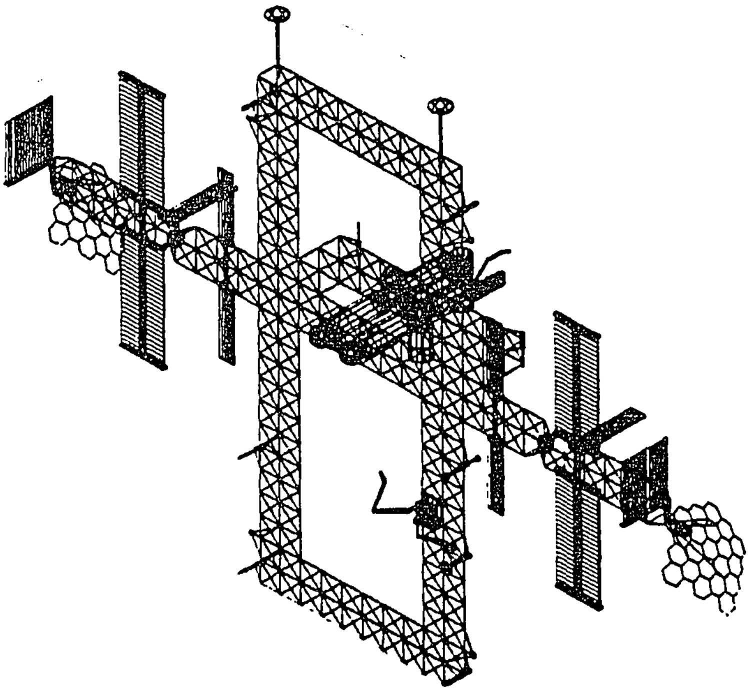


Fig. 7.9 A schematic of the proposed U.S. Space Station

Since storage space is crucial, a designer selects to fold the mast-segments onto a line. Using the same parameters as in Section 7.4.1 and employing Rules 7.1,7.2 and 7.6, leads to the selection of designs shown in Fig. 7.10. These results indicate that the truss modules of the U.S. Space Station can be folded into a very compact package, without using a large number of transition links and the associated locking mechanisms. Furthermore, the usefulness of systematic procedures for the conceptual design of deployable truss structures is hereby demonstrated, since none of the designs displayed in Fig. 7.10 have been reported in the literature.

The previous examples demonstrate the feasibility of applying expert systems as a tool in a design selection process. In particular, it has been shown to be possible to select the best available design for a particular application using the rules for design selection that were formulated in Section 7.4. Furthermore, application of these rules in conjunction with the CLIPS-shell has led to the identification of existing as well as novel designs for various applications.

PLEASE NOTE

Page(s) not included with original material and unavailable from author or university. Filmed as received.

165

University Microfilms International

CHAPTER 8

CONCLUSION

8.1 Accomplishments

This research has addressed the following three aspects of the conceptual design of deployable truss structures:

1. Geometrical synthesis (generation of the topology of deployable truss structures).
2. Dimensional synthesis (generation of the deployed and folded configurations of deployable truss modules).
3. Kinematic synthesis (generation of joint assignments for deployable truss structures).

As part of the discussion, techniques and procedures have been introduced for the automatic generation of conceptual designs of deployable truss structures. In addition, techniques and procedures have been presented for the generation of detailed conceptual designs of deployable truss modules, which form a subset of deployable truss structures. Application of these techniques has resulted in the generation of innumerable conceptual designs.

The availability of a large number of design alternatives may overwhelm a designer of deployable structures and therefore risk reducing his critical capacities

through boredom. Hence, a tool is needed that aids the designer in making a selection of structures from among all available designs. To this effect, evaluation criteria for deployable truss structures were developed and implemented in an expert system. While applying this expert system to finding the best conceptual design for various applications, many novel designs have been identified.

Summarizing, the contributions of this research in the area of deployable truss structures are the following :

1. This research has advanced the understanding of deployable truss structures by means of a thorough discussion of all aspects of the conceptual design of this type of structure.
2. Systematic design procedures for deployable truss structures have been made available to the designer of deployable truss structures.
3. A tool has been provided to the designer of deployable truss structures to aid in the selection of the best available design for any given application.
4. The availability of systematic design and evaluation procedures allows the designer to focus only on practical solutions to the design problem.
5. It has been shown that graph theory is a promising tool in the conceptual design of deployable structures.
6. Novel conceptual designs of deployable truss structures have been identified.

8.2 Directions for Further Research

The next logical step after the conceptual design is completed is the specification of joint orientations. However, due to infinite number of solutions to this design problem, these orientations must be chosen such that the deployment dynamics of a truss structure is optimal. This can only be done by selecting a particular deployment sequence for each module and subsequently carrying out a dynamic analysis as

an integral part of an optimization analysis. Hence, this research can only be conducted for a specific application and was therefore not part of the research reported here.

The present research has addressed the geometrical, dimensional and kinematic aspects of the conceptual design of truss modules. Repeated application of the expert system developed for the evaluation of this type of structures has revealed that the majority of designs selected by the expert system as the best available design for a given design problem, involves designs with special dimensions and properties of symmetry. The special dimensions of a truss module (recognized using the techniques described in Chapter 5) allow it to fold with fewer transition links than truss modules that do not have such dimensions, whereas properties of symmetry allow the truss module to fold with joints providing fewer degrees of freedom than dictated by the degree of freedom equation (see Chapter 6). Preliminary research indicates that it is possible to identify whether a truss module exhibits properties of symmetry. However, since there is no basis for the application of a general degree of freedom equation that accounts for such properties, finding the appropriate joint assignments proved to be an obstacle beyond the scope of this research.

REFERENCES

1 Crawford, R. F. ,"Strength and Efficiency of Deployable Booms for Space Applications," AAS/AIAA Variable Geometry and Expandable Structures Conference, Anaheim, California, April 1971, AIAA Paper No. 71-396.

2 Hedgepeth, John M. and Mikulas, Martin M. ,"Expandable Modules for Large Space Structures," AIAA/ NASA Conference on Advanced Technology for Future Space Systems, Hampton, Virginia, May 1979, pp 375-379.

3 Bush, Harold G. , Heard, Walter L. , Walz, Joseph E. and Rehder, John J. ,"Deployable and Erectable Concepts for Large Spacecraft," 39th Annual Conference of the Society of Allied Weight Engineers, Inc., St. Louis, Missouri, May 1980, SAWE Paper No. 1374, Index Category No. 18.

4 Bush, H. G. and Heard, W. L. ,"Recent Advances in Structural Technology for Large Deployable and Erectable Spacecraft," 31st International Astronautical Congress, Tokyo, September 1980.

5 Vaughan, D. H. ,"Modular Reflector Concept Study," NASA Contractor Report 3411, March 1981.

6 Schwartzberg, Fred R. , Coyner, John V. and Tobey, William H. ,"Development and Application of Space Deployable Box Truss Structures," 32nd Congress of the International Astronautical Federation, University of Rome, Rome, Italy, September 1981, IAF Paper No. 81-381.

7 Craighead, N. D. et al., "Design Development and Mechanization of a Precision Deployable Truss with Optimized Structural Efficiency for Spaceborne Applications," Lockheed Missiles and Space Company, Inc., 1982.

8 Adams, L. R. ,"Stacbeam: An Efficient, Low-Mass, Sequentially Deployable Structure," 17th Intersociety Energy Conversion Engineering Conference, Los Angeles, California, August 1982.

9 Adams, L. R. ,"Hinge Specification for a Square-Faceted Tetrahedral Truss," NASA Contractor Report 172272, January 1984.

10 Rhodes, M. D. and Mikulas, M. M. ,"Deployable Controllable Geometry Truss Beam," NASA Technical Memorandum 86366, June 1985.

11 Miura, K. and Furuya, H. ,"An Adaptive Structure Concept for Future Space Applications," 36th Congress of the International Astronautical Federation, October 1985, IAF Paper No. 85-211.

12 Kibe, S. et al., "Structure and Function of Deployable Truss Beam (DTB)," 15th International Symposium on Space Technology and Science, Tokyo, 1986, pp 2035-2040.

13 Takamatsu, K. , Onoda, J. and Higuchi, K. ,"New Concepts of Deployable Truss Units for Large Space Structures," 28th Structures, Structural Dynamics and Materials Conference, Monterey, California, April 1987, pp 695-704.

14 Adams, L. R. ,"Design, Development and Fabrication of a Deployable/Retractable Truss Beam Model for Large Space Structures Application," NASA Contractor Report 178287, June 1987.

15 Hedgepeth, J. M. and Miller, R. K. ,"Structural Concepts for Large Solar Concentrators," NASA Contractor Report 4075, June 1987.

16 Stoll, H. W. ,"Systematic Design of Deployable Space Structures," 2nd AIAA Conference on Large Space Platforms, San Diego, California, February 1981, AIAA Paper No. 81-0444.

17 Gvamitchava, A. S. , Myshonkoya, N. V. , Sokolov, A. G. and Spyvak, V. M. ,"The Features of the Space Structures Dynamic Design," 34th Congress of the International Astronautical Federation, Budapest, Hungary, October 1983, IAF Paper No. 83-409.

18 Freudenstein F. and Dobrjanskyj L. ,"On a Theory of Type Synthesis of Mechanisms," Proceedings of the 11th International Congress of Applied Mechanisms (Springer, Berlin), 1965, pp 420-428.

19 Crossley F. R. E. ,"The Permutations of Kinematic Chains of Eight Members or Less from the Graph-Theoretic Viewpoint," Developments in Theoretical and Applied Mechanisms, 1965, Volume 2, Ed. W. A. Shaw (Pergamon Press, Oxford) pp 467-486.

20 Dobrjanskyj L. and Freudenstein F. ,"Some Applications of Graph Theory to the Structural Analysis of Mechanisms," Transactions of the American Society of Mechanical Engineers, Journal of Engineering for Industry 89B, 1967, pp 153-158.

21 Buchsbaum F. and Freudenstein F. ,"Synthesis of Kinematic Structure of Geared Kinematic Chains and other Mechanisms," Journal of Mechanisms 5, 1970, pp 357-392.

22 Freudenstein F. ,"An Application of Boolean Algebra to the Motion of Epicyclic Drives," Transactions of the American Society of Mechanical Engineers, Journal of Engineering for Industry 93B, 1971, pp 176-182.

23 Freudenstein F. and Woo L. S. ,"Kinematic Structure of Mechanisms," in Basic Questions of Design Theory (North-Holland, Amsterdam), 1974, pp 241-264.

24 Lee, T. W. and Freudenstein, F. ,"Heuristic Combinatorial Optimization in the Kinematic Design of Mechanisms, Part 1: Theory, Part 2: Applications," Transactions of the ASME, Journal of Engineering for Industry, November 1976, pp 1277-1284

25 Freudenstein F. and Maki E. R. ,"The Creation of Mechanisms According to Kinematic Structure and Function," Journal of Environment and Planning B, 1979, pp 375-391.

26 Sohn, W. J. and Freudenstein, F. ," An Application of Dual Graphs to the Automatic Generation of the Kinematic Structures of Mechanisms," Transactions of the ASME, Journal of Mechanisms, Transmissions and Automation in Design, September 1986, Vol. 108, pp 392-398.

27 Benett, J. S. and Engelmore, R. S.,"SACON: A Knowledge-Based Consultant for Structural Analysis," Proceedings of the 6th International Joint Conference on AI IJCAI, 1979, pp 47-49.

28 Dixon, J.R.,"Expert Systems for Design: Standard V-Belt Drive Design as an Example of the Design-Evaluate-Redesign Architecture," Proceedings of the ASME Computers in Engineering Conference, Boston, MA, Aug. 1984.

29 Brown, D.C. and Chandraskeran, B., "Knowledge and Control for a Mechanical Design Expert System," IEEE Computer, July 1986.

30 Dixon, J.R.,"Artificial Intelligence and Design, A Mechanical Engineering View," Proceedings of the AIAA 5th National Conference on AI, Vol. 2, August 1986, pp 872-877.

31 Allen, R. H.,"The Nature and Evaluation of Expert System Tools for Engineering Applications," Proceedings of the 1987 ASME Computers in Engineering Conference, New York, N.Y., Aug. 1987.

32 Roberts F. S. ,"Graph Theory and its Applications to Society," Society for Industrial and Applied Mathematics, 1978.

33 Unger, S. H. ,"GIT - A Heuristic Program for Testing Pairs of Directed Line Graphs for Isomorphism," Communications of the ACM, Volume 7, Number 1, January 1964.

34 Bohm, C. and Santolini, A. ,"A Quasi-Decision Algorithm for the P-Equivalence of Two Matrices," ICC Bulletin, Vol. 3, No. 1, 1964, pp. 57-69.

35 Molian, S.,"Storage and Retrieval of Descriptions of Mechanisms and Mechanical Devices According to Kinematic Type," Journal of Mechanisms, Volume 4, pp 311-323, January 1969.

36 Corneil, D. G. and Gotlieb, C. G., "An efficient Algorithm for Graph Isomorphism," Journal of ACM, Vol. 17, No. 1, January 1970, pp. 51-64.

37 Shah, Y. J.; Davida, G. I.; McCarthy, M. K.,"Optimum Features and Graph Isomorphism," IEEE Transactions on Systems, Man, and Cybernetics, May 1974.

38 Uicker Jr., J. J.; Raicu, A.,"A Method for The Identification and Recognition of Equivalence of Kinematic Chains," Journal of Mechanisms and Machine Theory, Volume 10, pp 375-383, 1975.

39 Yan, H.; Hwang, W.,"Linkage Path Code," Journal of Mechanisms and Machine Theory, Volume 19, No 4/5, pp 425-429, 1984.

40 Gibbons, A., "Algorithmic Graph Theory," Cambridge University Press, Cambridge, 1985

41 Murthynjaya, T. S. and Raghavan, M. R., "Structural Analysis of Kinematic Chains and Mechanisms Based on Matrix Representation," Transactions of the ASME, Journal of Mechanical Design, Vol. 101, July 1979, pp. 488-494.

42 Yan, H. S. and Hall, A. S. ,"Linkage Characteristic Polynomials: Definitions, Coefficients by Inspection," Transactions of the ASME, Journal of Mechanical Design, Vol. 103, July 1981, pp. 578-584.

43 Yan, H. S. and Hall, A. S. ,"Linkage Characteristic Polynomial: Assembly Theorem, Uniqueness," Transactions of the ASME, Journal of Mechanical Design, Vol. 104, January 1982, pp. 11-20.

44 Rose, D. and Tarjan, R. E. ,"Algorithmic Aspects of Vertex Elimination," presented at SIGACT 75, Albaquerque, N. M., May 1975.

45 Rose, D. and Tarjan, R. E. ,"Algorithmic Aspects of Vertex Elimination on Directed Graphs," SIAM Journal On Applied Mathematics, Vol. 34, January 1978, pp 176-197.

46 Sandor, G. N. and Erdman, A. G.,"Advanced Mechanism Design: Analysis and Synthesis, Volume 2," Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, pp 136-137.

47 Benett, G. T.,"A New Mechanism," Engineering 76, p. 777, 1903.

48 "The CLIPS User's Guide," Johnson Space Center, Document no. JSC-IN-86-FM-25, October 1986.

49 "Riley, G.; Gulbert, C. and Savely, R. T.:"Clips: An Expert System Tool for Delivery and Training," Proceedings of the Third Conference on AI for Space Applications, November 1987.