

# A Method for the Automatic Generation of Timber Connection Patterns

**M. H. Gedig, M.A.Sc, P.Eng.**

*Structural Engineer AMEC Dynamic Structures Ltd.  
Port Coquitlam, B.C., Canada*  
and

**Dr. S. F. Stiemer, Dr.-Ing, P.Eng.**

*Professor, Department of Civil Engineering University of British Columbia  
Vancouver, B.C., Canada*

---

## ABSTRACT

*This paper describes a method for the automatic rule-based generation of design alternatives for structural connections between timber members. Using this method, a computer program for generating a list of connection options for two timber members was created. Such a computer program may be useful to engineers, architects, and others involved in the design and detailing of timber frame structures.*

## KEYWORDS

*Timber connection, automatic generation.*

---

## 1 Introduction

A method for the automatic generation of design alternatives for structural connections between timber members is described herein. The method utilizes a technique of spatial qualitative reasoning to abstract a range of design options to a single class. The approach used here is to perform a spatial discretization of the surfaces of the members involved in the connection. This technique is well suited to structural connections which utilize plates that cover parts of the surfaces of the members, because the range of possible hardware patterns may be directly related to the discretization of the surfaces of the members. Since the hardware used in the connection of timber members often includes steel plates, the technique described here is appropriate for generating connection alternatives for timber members.

The main concepts presented here are: 1) a spatial discretization scheme specifically designed to capture the key aspects of the relationship between two structural members; 2) a generation scheme which utilizes a tree data structure and a breadth-first traversal; and, 3) a rule base for classifying the types of connection between two structural members.

## 2 Assumptions

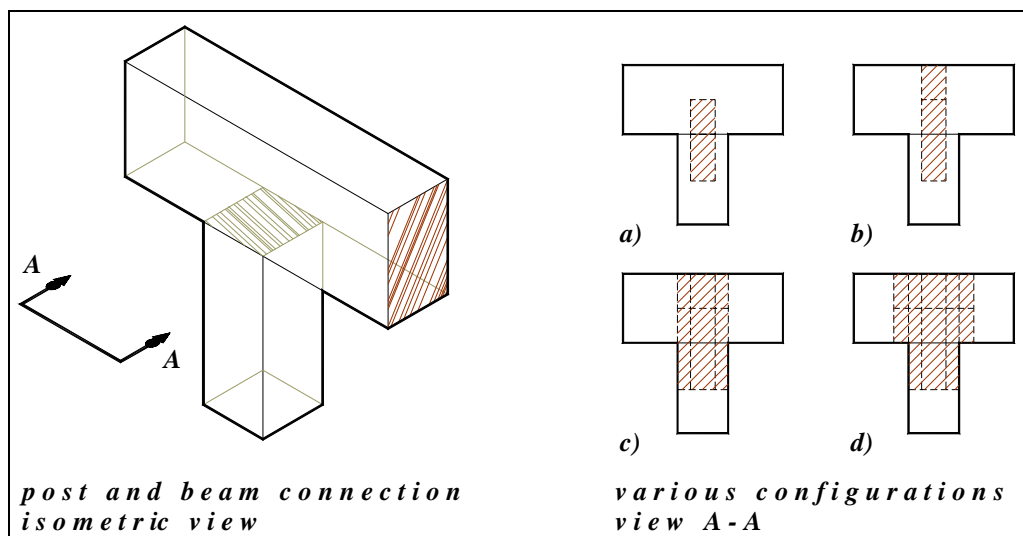
The method presented here is limited to a certain class of structural connection. The limitations of the program and the assumptions implied in the theory are listed here. The relaxation of some of some of the limitations will be covered later in this document.

The concepts introduced here are limited to the connection of two rectangular prismatic structural members. The members are assumed to be orthogonal to one another, and the members must be in physical contact with one another. The type of hardware used to connect the members is limited to plates which have the full area of at least one face in direct contact with one or both of the members.

### 3 Qualitative Spatial Reasoning

There are an infinite number of ways in which two prismatic structural members may be connected together using plates. For this reason it is impossible to enumerate all the possible configurations of plates from which the structural engineer or architect may select a design option. Qualitative reasoning is concerned with quantities which take on a finite number of values. The advantage of qualitative reasoning over traditional approaches lies in the ability to reason by exclusion, and to enumerate all solutions to a given problem.

The approach used here is to recognize the fact that although there are an infinite number of possible combinations of hardware, there are only a finite number of combinations which are *qualitatively distinct*. The concept of qualitative distinctions between geometric arrangements is discussed in the following example. Consider the connection of a beam with rectangular cross-section to a column, shown in [Figure 1](#).



**Figure 1: Qualitative spatial reasoning**

In this example, the beam is assumed to be of the same width as the column, so it is possible to connect the two members using a single flat plate as shown in [Figure 1a](#). In practice it is customary to utilize the symmetry of the members so that one such plate would be used on each side of the connection. If the column is, say, six inches square, then the width of the plate may vary from some value greater than zero to six inches. In reality of course, an extremely narrow plate provides little structural capacity and would therefore be considered an impractical design alternative. For the moment, however, we consider all possible alternatives, not discriminating on the basis of either structural strength or aesthetic value. Our objective is to find a large number of possible hardware configurations which satisfy the geometric constraints imposed by the need to physically connect the two structural members. In subsequent stages of design, those alternatives which lack sufficient structural strength or do not meet aesthetic objectives may be easily discarded.

Because we require that the face of the plate is fully in contact with the face of one or both of the members, the configuration shown in [Figure 1a](#) has the implied constraint that the width of the plate must lie between 0 and 6 inches. For this very simple situation, the number of arrangements is infinitely large, because the width of the plate may be 5.00 inches, 5.01 inches, 4.95 inches and so on. In the hardware generation scheme discussed here, one of the key concepts is the use of the qualitative distinctions *full coverage* and *partial coverage*. A variable representing the width of the plate shown in [Figure 1a](#) may take on one of two qualitative

values: *partial*, representing a width of between 0 and 6 inches; and *full*, denoting a width of exactly 6 inches. The depth of the plate may be described in a similar manner. In [Figure 1a](#), the plate extends only partially up the side of the beam, while in [Figure 1b](#), the plate extends the full depth of the beam. [Figure 1c](#) illustrates a plate that covers the full width of the column as well as the full depth of the beam. In [Figure 1d](#), a plate which partially covers the length of the beam is depicted. The reader will note that in [Figure 1](#), the options which correspond to plates covering the full length of the column or the full length of the beam have been omitted, as they rarely occur in practice except for relatively short beams or columns.

As shown in [Figure 1d](#), it is not necessary to assume that the plate is always rectangular in shape. If the plate fully covers the width of the column, this does not imply that the plate necessarily covers the full width or length of the beam. For this reason it is convenient to deal with the members involved in the connection as separate entities, and to combine the design options for each member to obtain the final solution. This process will be explained in the following paragraphs.

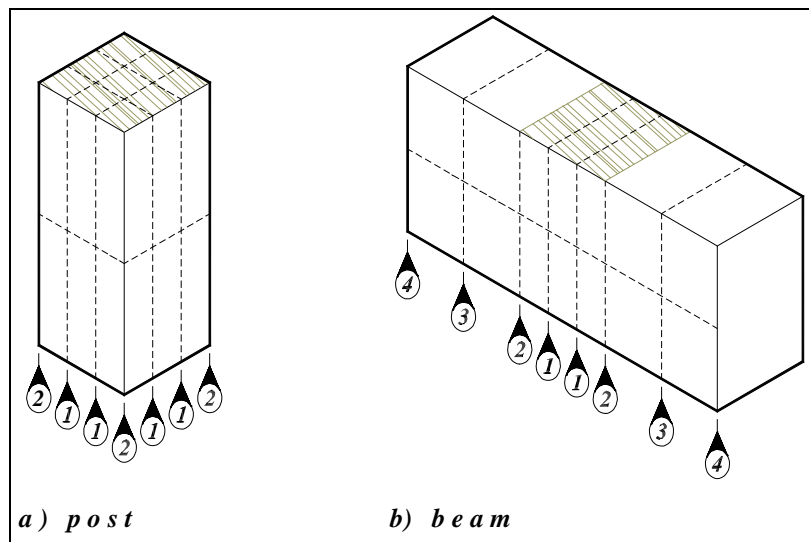


Figure 2: Surface discretization of column and beam

#### 4 Surface Discretization

For the beam to column connection shown in [Figure 1](#), the surface of the column may be discretized as shown in [Figure 2a](#). The shaded end of the column denotes the contact surface; that portion of the surface of a member which is common to the surface of the member to which it is connected. Because of the limitations imposed on the types of members which are treated in this method, each member will always have a contact surface, and that surface will always be rectangular in shape. The contact surface plays an important role in this formulation, as the qualifiers *partial* and *full* can be applied to both the overall dimensions of the members as well as to the dimensions of the contact surface. In [Figure 2a](#), each of the broken lines has a particular significance to the discretization of the surface of the column: the label '1' indicates partial coverage with respect to the contact surface, and the label '2' indicates full coverage with respect to the contact surface. The broken line at the mid height of the column indicates partial coverage of the length of the column. The discretization of the surface of the beam is shown in [Figure 2b](#). Again the contact surface is denoted by the shaded portion of the surface (the beam shown in [Figure 2b](#) is shown from a different viewing position than in [Figure 1](#)). As in [Figure 2a](#), the label '1' represents partial coverage, and the label '2' represents full coverage, both with respect to the contact surface. The label '3' represents partial coverage with respect to the length

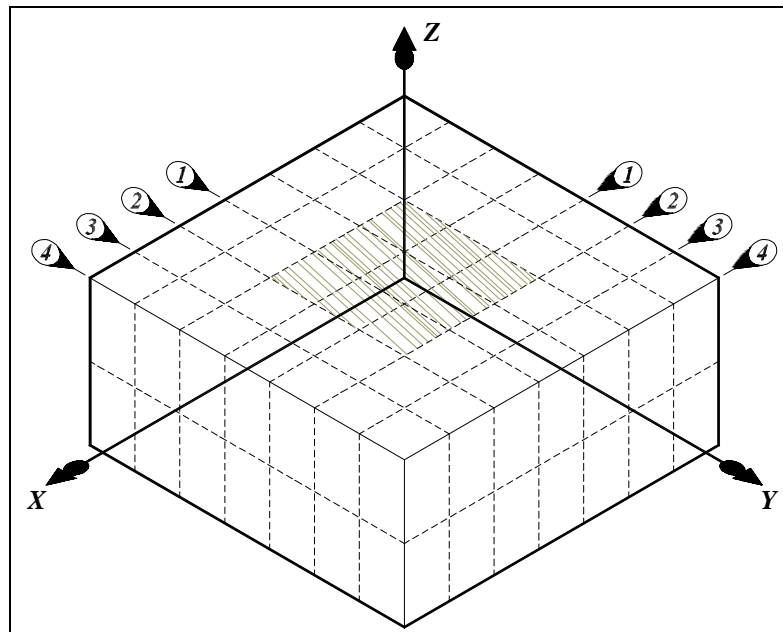
of the beam, and the label '4' represents full coverage with respect to the length of the beam. A summary of the designations used in the surface discretization is shown in [Table1](#).

**Table 2: Surface discretization summary**

Label	Description
1	Partial coverage with respect to contact surface
2	Full coverage with respect to contact surface
3	Partial coverage with respect to width/thickness of member
4	Full coverage with respect to width/thickness of member

In general, each member may be represented by a rectangular prism which is discretized as shown in [Figure 3](#). A Cartesian coordinate system is established at the center of the contact surface. In the contact plane, the x-axis and y-axis are orthogonal to one another and parallel to one of the sides of the prism. The z-axis forms a right-handed triad with the x- and y-axes. The volume of the prism is discretized so that planes corresponding to  $x = \pm 1$  and  $y = \pm 1$  represent partial coverage with respect to the contact surface. Similarly, planes corresponding to  $x = \pm 2$  and  $y = \pm 2$  denote full coverage with respect to the contact surface. The planes  $x = \pm 3$ ,  $y = \pm 3$ ,  $x = \pm 4$ , and  $y = \pm 4$  are described in a manner analogous to that shown in [Table3](#). The plane given by  $z = -1$  represents partial coverage of the member with respect to the dimension normal to the contact surface.

Using this system of discretization, the model of the column shown in [Figure 2a](#) may be obtained by cutting the prism at  $x = \pm 2$  and  $y = \pm 2$ , and by discarding the portions for which  $|x| > 2$  and  $|y| > 2$ . The surface discretization showed in [Figure 2a](#) follows directly from the volume discretization of the general model. Similarly, the surface discretization of the beam shown in [Figure 2b](#) is obtained from the general model after cutting away those parts of the prism for which  $|y| > 2$ . The discretization of the surfaces of members, and the method by which the discretized surfaces are used to generate hardware options, are covered in greater detail in the following sections.



**Figure 3: General discretization model**

## 5 Discretization of Plates

Since the plates used in the connection of two members are required to have one face in common with the surface of at least one of the members, the discretization of the surface of a member may be directly related to the discretization of the plate which shares that surface. Using the surface discretization scheme discussed here, the surface of each member is partitioned into a finite number of rectangular *surface elements*. The term *plate element* will be used to denote the portion of a plate which corresponds to a rectangular surface element. The plates which can be used to cover the surface of a member may be enumerated by finding all possible combinations of plate elements for a particular member. The plate configurations which correspond to hardware options are obtained by finding those plates on one member which intersect plates on the other member.

In order to determine the relative positions and orientations of plate elements, it is convenient to develop a labeling system. Such a labeling system is required only for a quarter of the volume used to represent numbers, as the prism has two axes of symmetry. In [Figure 4](#), each plate element is given a code of the form  $xyzf$ , in which  $x$ ,  $y$  and  $z$  are the coordinates of the element (ignoring sign), and  $f$  is a code which denotes the orientation of the element. The face code is determined as shown in the lower left hand corner of [Figure 4](#), where, for example, the code '0' denotes an element with normal parallel to the  $z$ -axis.

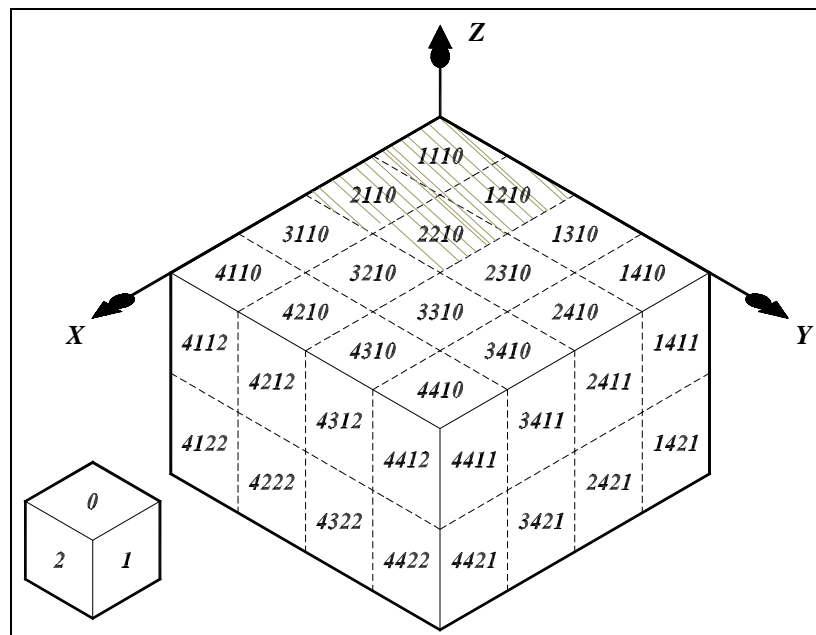
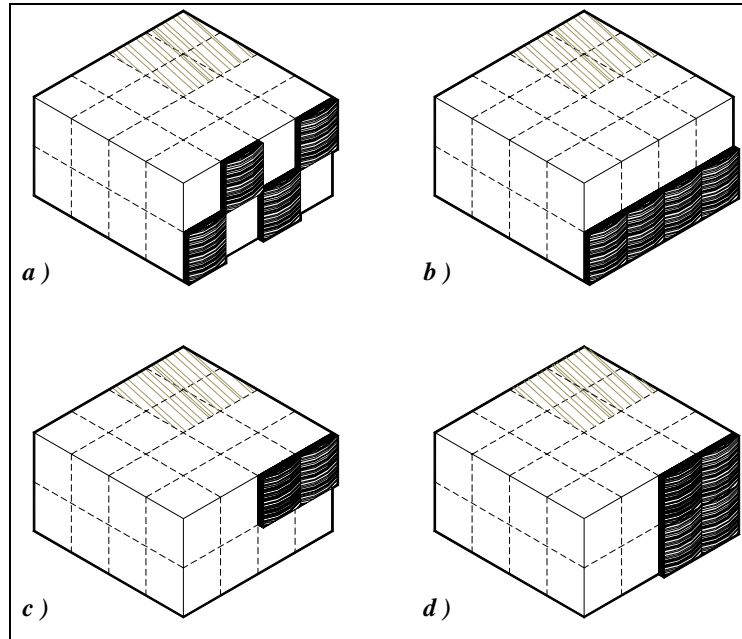


Figure 4: Surface coding system

## 6 Generation of Plate Configurations

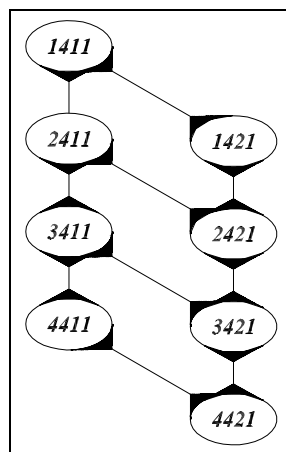
In order to generate the possible hardware configurations for two given members, the configurations of plates which suite each of the members must be determined. A plate configuration is represented by a list of plate elements. In order to conform to accepted notions of what the term “plate” means, some constraints are placed on the plate elements in the list. Examples of valid and invalid plate configurations are shown in [Figure 5](#).



**Figure 5: Plate configurations**

In [Figure 5](#), the shaded rectangles represent plate elements. The pattern shown in [Figure 5a](#) illustrates an invalid plate configuration. Each of the plate elements intersects an adjacent plate element at no more than two points. For a valid plate configuration, we require: a) that each plate element in the list has at least one complete side in common with another element in the list; and b) the plate elements in the list make up a single contiguous group. In [Figure 5b](#), although the plate elements make up a single contiguous group, the plate configuration is considered invalid because the none of the elements lies adjacent to the contact plane. The plate configuration shown in [Figure 5b](#) cannot intersect a plate on the adjoining member so it cannot lead to a viable hardware option.

It should be stressed that the classification between valid and invalid plate configurations is completely arbitrary. The main reason for establishing this classification is to limit the number of plate configurations which are considered by the system. Although the number of plate elements is finite, the generation of plate configurations is a computationally intensive process. Any move to eliminate combinations of plate elements before the generation proceeds has a beneficial impact on the performance of the system. One method of organizing the plate elements and limiting the generation of plate configurations to meaningful and simple patterns is shown in [Figure 6](#).



**Figure 6: Hierarchy of plate elements**

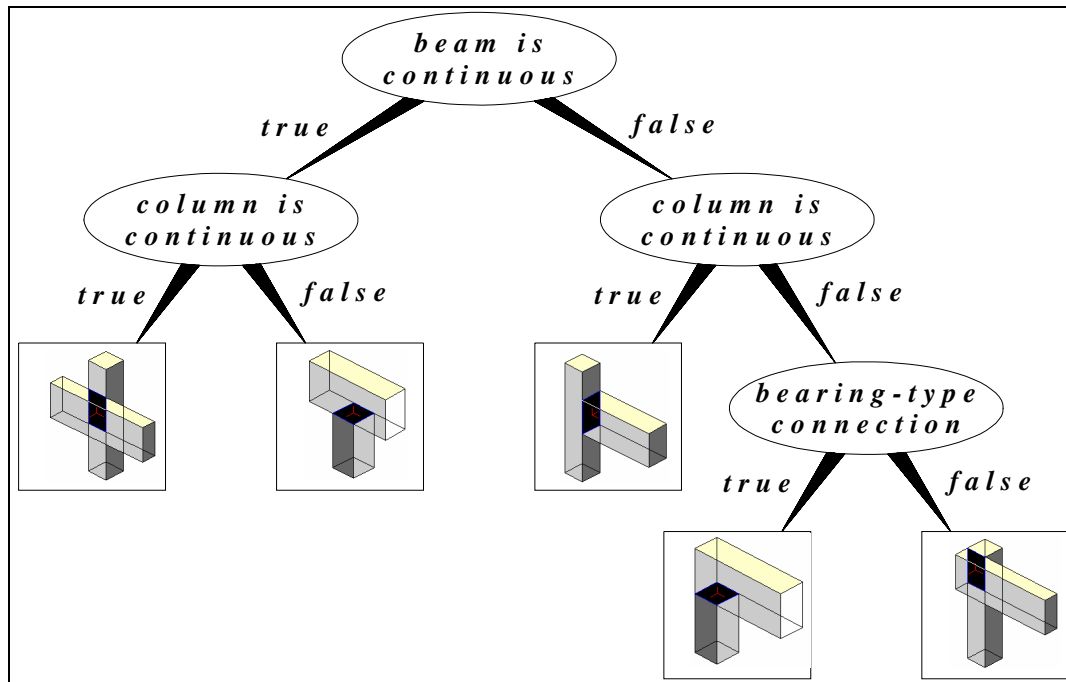
**Table 4: Generation of plate configurations**

Step	Element	Plate Configuration
1	1411	1411
2	2411	1411, 2411
3	1421	1411, 1421
4	3411	1411, 2411, 3411
5	2421	1411, 2411, 1421, 2421
6	4411	1411, 2411, 3411, 4411
7	3421	1411, 2411, 1421, 3411, 2421, 3421
8	4421	1411, 2411, 1421, 3411, 2421, 4411, 3421, 4421

In [Figure 6](#), the codes for the a number of plate elements are organized in a hierarchical structure. This structure, which is similar to the tree data structure used in computing science, expresses the requirement that in order to have one plate element in a list, other elements must be present at the same time. A line between two elements in [Figure 6](#) indicates that the uppermost element in the tree must be present in order for the lower element to exist. For example, the element 2421 cannot exist without both 2411 and 1421. These two elements, in turn, cannot exist without element 1411. All of the elements in [Figure 6](#) correspond to the right front face of the prism shown in [Figure 4](#). Comparing [Figure 4](#) and [Figure 6](#), it can be seen that a link between two elements in [Figure 6](#) indicates that the two elements are adjacent and share a common side. Using a structure like the one shown in [Figure 6](#), it is possible to generate all possible lists of contiguous plate elements for one face of a member. All valid plate configurations may be generated using a breadth-first traverse of the tree. The generation proceeds as shown in [Table2](#).

The plate configurations for steps 2 and 5 correspond to [Figure 5c](#) and [Figure 5d](#) respectively. The hierarchical data structure used here results in a controlled generation of plate configuration options, and causes many physically meaningless options to be filtered out before the generation process begins. In addition, the data structure can easily be “pruned” to represent various physical configurations of members. In the beam to column connection example studied earlier, the surface discretization models for the beam and column were each derived from the general model by “cutting away” portions of the prism. This cutting operation is performed by pruning away portions of the data structure representing the plate elements for a particular face of the member.





**Figure 7: Decision tree for selection of connection types**

A rule base is used to determine which parts of the plate element data structure must be pruned away. The following characteristics are used to classify the members involved in a connection, so that the correct portions of the data structure are eliminated:

1. Relative dimensions of connected members.
2. Orientation of members. Currently, the only distinction made is between vertical elements (columns) and horizontal elements (beams, girders, or purlins).
3. Continuity of members. A member either ends at the connection or it carries through the connection. In the beam to column example considered earlier, the beam is continuous while the column is not.
4. Type of contact between members. Prismatic members have 4 sides and 2 ends. The type of contact is defined as side-to-end (i.e., T-joint), side-to-side (i.e., lap joint), and end-to-end (i.e., butt joint).

When required, the following information is also used:

1. Transfer of forces. In some cases it is possible to distinguish between a shear-type connection and a bearing-type connection.
2. Maximization of contact surface. In some situations it is desirable to utilize only a portion of the area available for contact.

The decision tree shown in [Figure 7](#) gives an example of how the rule base is structured. In its current form, the rule base is used to select one of about 30 different types of connection situations. Once the connection type has been determined, the plate element structures are pruned and the plate configurations are generated for each member. The plate configurations of each member involved in the connection are combined to give the hardware configurations which are possible for that connection.



## 7 Generation of Hardware Configurations

The position and orientation of the plates which are associated with each of the members in a connection are defined by the position and orientation of the plate elements which make up the plate. Given the code for two plate elements, it is possible to determine whether the elements intersect. If the plate elements intersect, one can determine whether the elements intersect at a corner or along one edge. If the intersection is along an edge, the two elements can be combined to form a new plate configuration. It is possible to determine whether two plate configurations intersect simply by comparing each of the plate elements of one plate to each of the plate elements of the other plate. If the two plates have at least one pair of plate elements which intersect, then the plates intersect.

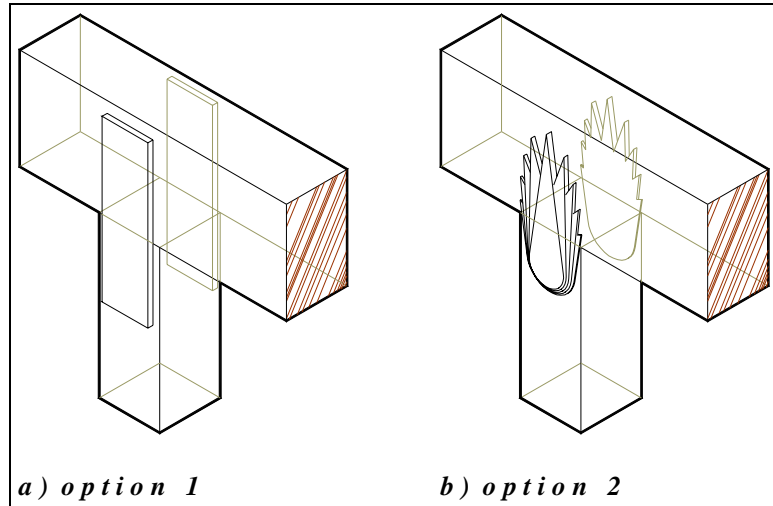
Within each member, a list of plate configurations is generated for each face of the member. Initially, each of the plate configurations is planar; that is, each of the plate elements making up the plate configuration is oriented the same way. The plate configurations for each face may be combined to give more complex plates which have elements with different orientations. An example of such a complex plate configuration is a clip angle, which has elements with two different orientations. By combining the list of plate configurations for each face, the number of plate configurations for the member is increased. It is possible to create even more complex plate configurations by combining the newly created configurations with the existing ones, but it was found that only one additional combination was suitable for most practical applications.

Once all of the plate configurations have been generated for each member, a coordinate transformation is applied to one of the members. The coordinate transformation has the effect of reflecting one of the members across the contact plane. After the transformation has been performed, it is possible to determine whether a plate configuration belonging to one of the members intersects one of the plate configurations of the other member. If the two plates intersect, they are combined into a composite plate configuration which is recorded as one of the hardware configurations which satisfies the constraints imposed by the geometry of the particular connection. In this manner, all such possible hardware configurations are determined.

## 8 Extensions

The research performed up to this point has suggested a number of extensions to the topics described here.

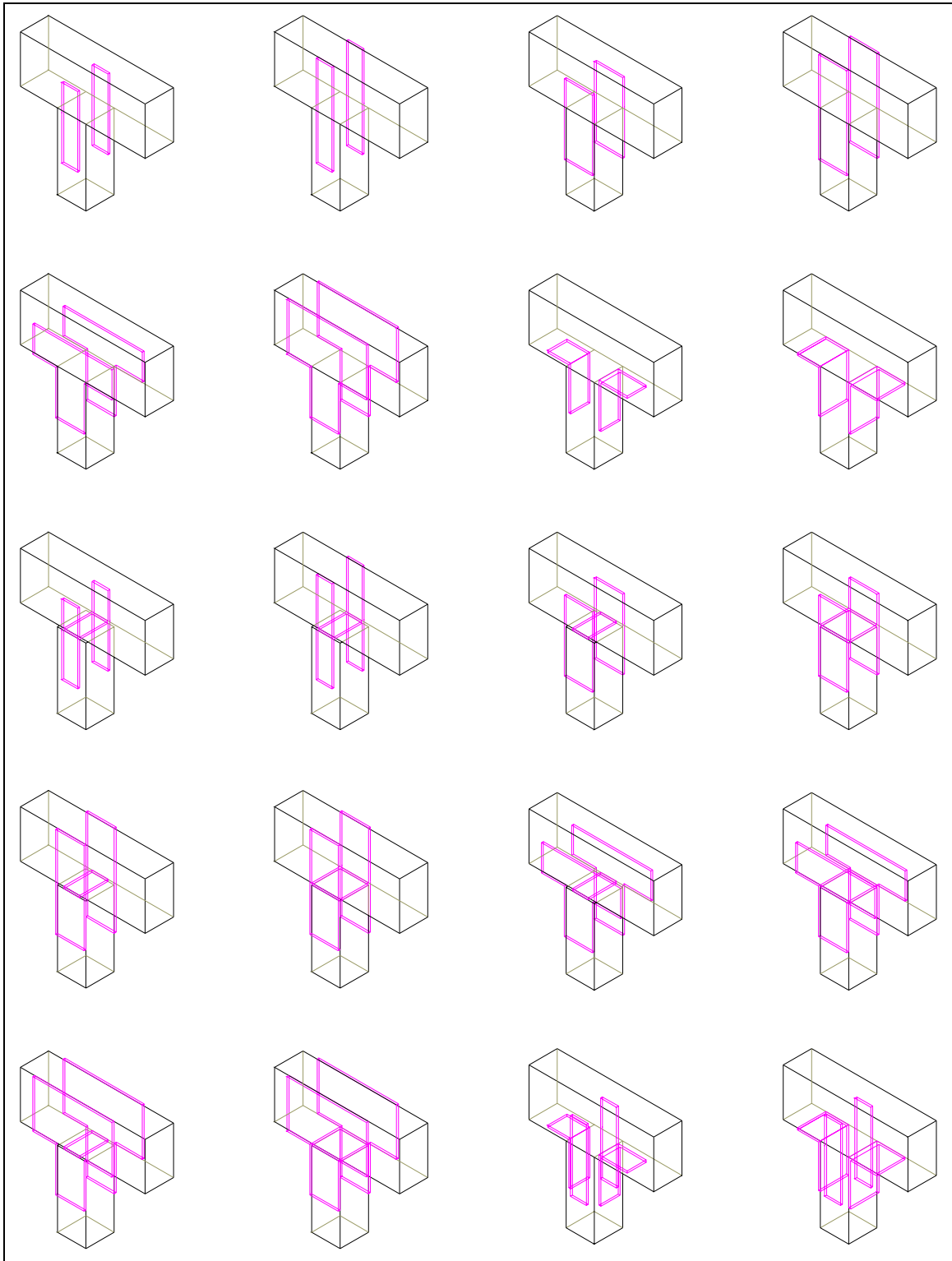
1. The method of generation discussed here gives general solutions which are easily expanded to give concrete dimensions for the connection hardware. The dimensions may be used to check the structural capacity of the hardware, which may be used as a basis for rejecting particular solutions. For a given connection type, the hardware generation does not have to be repeated when the member dimensions are scaled.
2. The representation scheme is abstract, and is therefore not strictly limited to connection hardware which resembles plates. Hardware such as bolts, dowels, split rings, pipes, and embedded plates may be inserted into the tree of plate elements and pruned as necessary.
3. The generation process for members which intersect at angles other than 90 degrees is identical to the process described here. A more sophisticated display routine would be required to facilitate the interpretation of solutions.
4. The solutions produced by the generation procedure described here define a class of connection options. The situation shown on the left in [Figure 8](#) is considered equivalent to the one on the right at the level of abstraction used in the generation.



**Figure 8: Abstraction of hardware options**

## 9 Application

The main strengths of the approach described here over previous efforts in this area of research are creativity and visualization. Work on automated selection of connection hardware has been limited to a large extent to tables or catalogues of hardware. In contrast, this research has focused on the generation of new combinations and configurations of hardware. The work presented here may prove to be a useful tool for the engineer and architect alike, as it has the ability to suggest hardware options which the designer may have not so easily conceived or visualized. In the implementation presented here, the hardware options are presented visually rather than described with words. A sample of the computer program output for the beam to post connection example used throughout this paper is shown in [Figure 9](#) and [Figure 10](#). The design options shown graphically in these figures require little explanation or translation, so that the designer may easily digest a large number of physically realizable options.



**Figure 9: Sample of output for post and beam example (Set 1 of 2)**

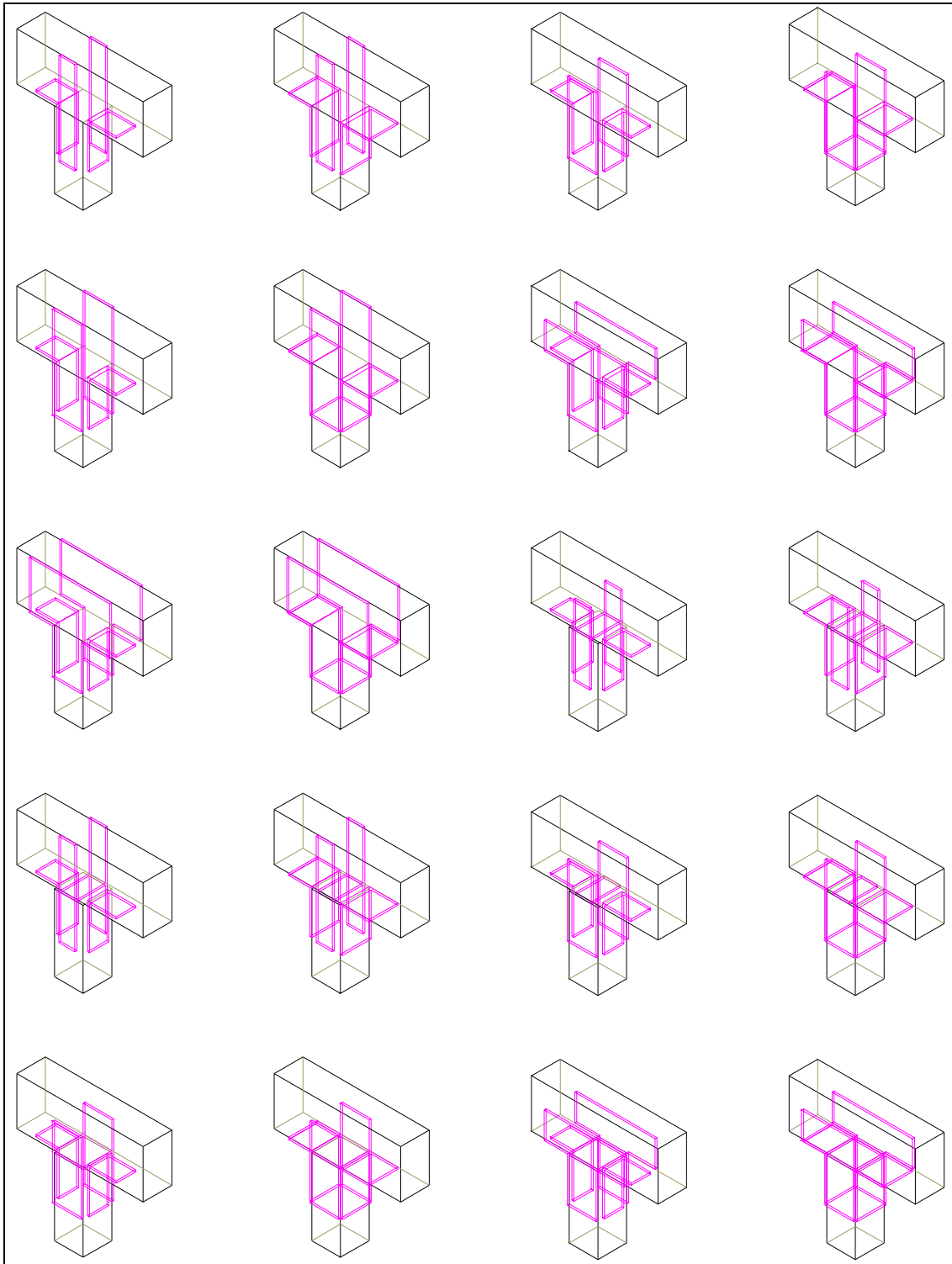


Figure 10: Sample of output for post and beam example (Set 2 of 2)

## 10 Conclusion

This paper has described a method for generating hardware options for connections between two timber structural members. This method may be used in a computer program that aids

engineers, architects and others involved in the design of timber connections. While this paper has limited the type of hardware to plates, the same technique can be readily extended to include other types of hardware including bolts, dowels and split rings. With some thought, it is possible to modify the method presented here for use with connections between other types of structural members, including rolled steel shapes.