15

Assembly Sequence Planning Using Neural Network Approach

Cem Sinanoglu and Huseyin Riza Borklu

1. Introduction

The competition between manufacturing firm's makes it necessary that those firms must supply highly quality goods with shorter time and chipper for survive in the international market. The intensive research in this field aims at augmenting methods and tools for product development and manufacturing. By the use of new and efficient methods it is possible to shorten the time from design to manufacturing and reduce the mistakes originating from humans. Therefore, full automation in the manufacturing processes can be accomplished.

An assembly can be defined as the overall function of individual parts after joining each other's, each of which has an independent function. It is possible to divide an assembly into various subassemblies depending on its complexity levels (Pahl & Beitz, 1988). Although intensive research efforts in the field of assembly sequence planning, there are still some problems to be solved. For example, there exist some shortcomings to support for full automatic assembly sequence planning or to obtain assembly plans for large systems (Singh, 1997).

By the development of efficient assembly planning systems it is possible to provide some advantages in both areas:

- CAD automation as well as
- Manufacturing performed by the use of robots

In order to develop a system that addresses computer-aided assembly sequence planning, these issues should be taken into consideration:

- The connectivity structure of parts and/or subassemblies that are used in the assembly system

Source: Manufacturing the Future, Concepts - Technologies - Visions , ISBN 3-86611-198-3, pp. 908, ARS/plV, Germany, July 2006, Edited by: Kordic, V.; Lazinica, A. & Merdan, M.

- If they have connectivity properties, then what are the theatrical number of their different connection ways
- Finally the choice of the most optimum assembly process among various alternatives

Basic approach for finding the most suitable assembly sequences is to represent assembly system in a space where it is possible to explore all different assembly sequences. Thus, some criterion may be used to obtain these sequences. Then optimum assembly sequence can be selected by the application of some other criterion. This criterion may include: the number of tool changes, part replacement and clamping on the jigs, concurrency in operations, reliable subassemblies, etc (Kandi & Makino, 1996). A new approach may be stated as a rank evaluation from new scanning factor. But this may not exactly show the differences between operation time and costs.

The initial assembly planning system was inquired to user data necessity for operation and it was formed to assembly sequences with given data. It was worked with the user interaction (De Fazio, 1987, Homem de Mello & Sanderson, 1991). The later work concentrated on assembly sequence planning systems based on geometric reasoning capability and full automatic (Homem de Mello, 1991). Then, the system included some modules that allow assembly sequence plans to be controlled and to be tested.

Since many assembly sequences share common subsequences, attempts have been made to create more compact representations that can encompass all assembly sequences. Therefore, the works in this field are graph-based approaches to represent all feasible assembly sequences (FAS).

This paper presents a neural-network-based computational scheme to generate feasible assembly sequences for an assembly product. The inputs to the networks are the collection of assembly sequence data. This data is used to train the network using the Back propagation (BP) algorithm. The neural network model outperforms the feasible assembly sequence-planning model in predicting the optimum assembly sequences.

2. System Definition

This assembly planning system is used to assembly's connection graph (ACG) for the representation product. Parts and relations among these parts are represented by this graph. Contact relations between parts are supplied by scanning module. Scanning module scans various assembly views of any product whose assembly view is produced by *CAD* packet program in the computer environment and determines to contact and interference relations between parts (Sinanoğlu & Börklü, 2004).

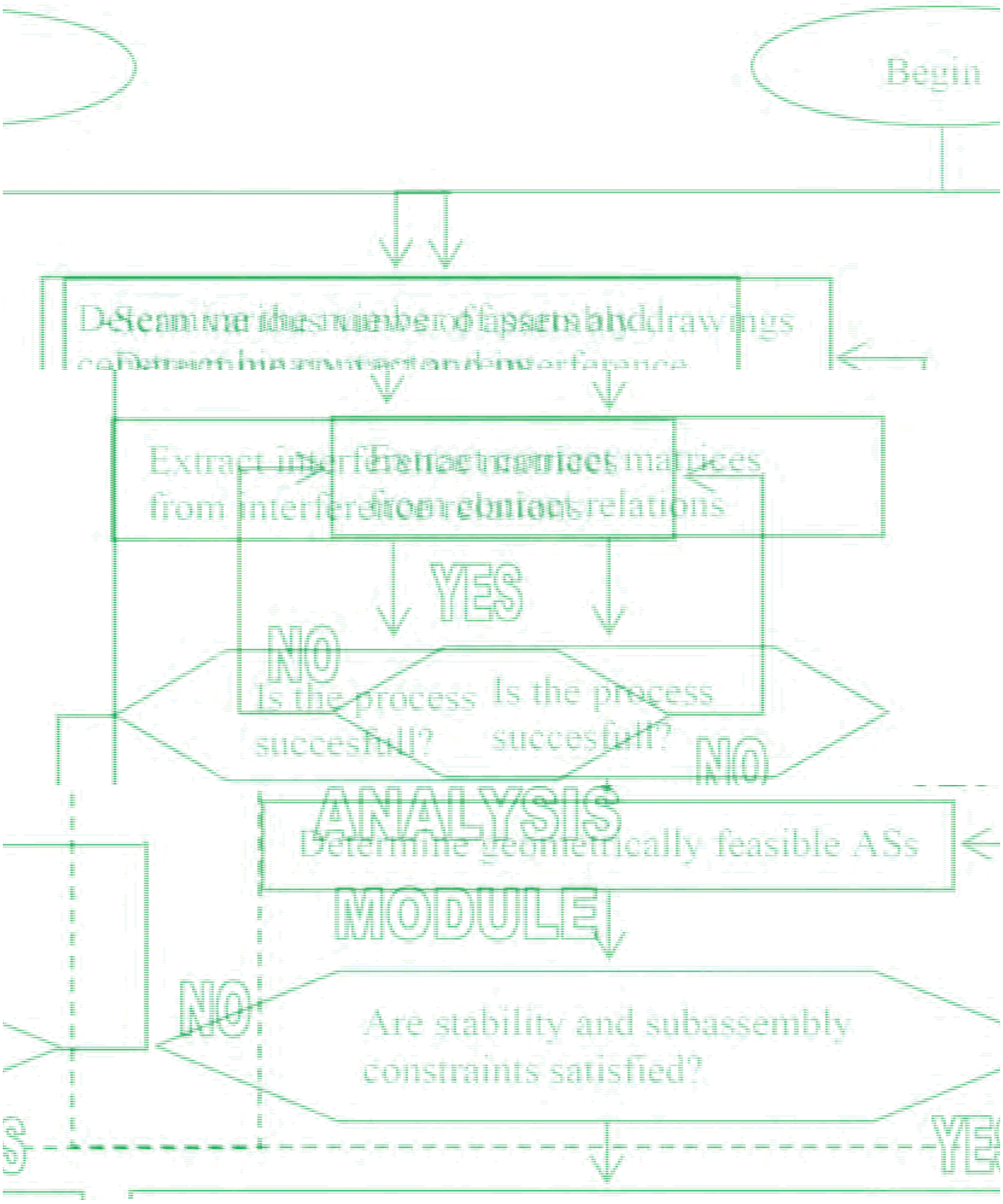


Figure 1. Block diagram of proposed assembly planning system

These relations are formed as a matrix. The system constitutes ACG of product to be assembled, by applying Boolean operators on the elements of the contact matrices according to certain rules. Moreover, interference relations are also accumulated in a form of interference matrices for determination of geometric feasibility of assembly states later (Sinanoğlu & Börklü, 2005).

Block diagram of assembly system is shown in Fig. 1. In the assembly planning system, binary vectors represent assembly states. Therefore, all binary vector representations, whether corresponding to assembly states or not, are produced by the system. By evaluating ACG and binary vector representations simultaneously with the scanning module, vector representations corresponding to assembly states are determined.

Some of the assembly states cannot take part in a feasible assembly sequence. The determination of the assembly states not corresponding to feasible assembly sequence is achieved with the analysis module. The analysis module controls all assembly states according to stability, subassembly and geometric feasibility constraints. Boolean operators apply to the elements of interference matrices determination of geometric feasibility. The feasible assembly states and assembly sequences are represented by a directed graph (Homem de Mello & Arthur, 1990). Assembly states supplying constraints are settled down in the nodes of directed graph hierarchically by the system.

Any path from the root node to terminal node in the directed graph corresponds to feasible assembly sequence. The optimum one is selected from the feasible assembly sequences with optimization module. The neural network approach has been employed for analyzing feasible assembly sequences for sample product. Due to parallel learning structure of the network, the proposed neural network has superior performance to analyze these systems.

3. Artificial Neural Networks

An artificial neural network (or simply a neural network-NN) is a biologically inspired computational structure, which consists of processing elements (neurons) and connections between them with coefficients (weights) bound to the connections.

Training and a recall algorithm are associated to every NN. NN are also called connectionist models because of the main role of the connections the weights are the result of the training phase; they are the "memory" of the structure. The connection weights change during learning (training).

A structure of a typical biological neuron is shown in Fig. 2(a). It has many inputs (in) and one output (out). The connections between neurons are realized in the synapses. An artificial neuron is defined by (Fig. 2(b)):

- Inputs $x_1, x_2, \ldots\ldots, x_n$
- Weights, bound to the inputs $w_1, w_2, \ldots, w_n$
- An input function (f), which calculates the aggregated net input
- Signal U to the neuron (this is usually a summation function)
- An activation (signal) function, which calculates the activation
- Level of the neuron: $O = g(U)$

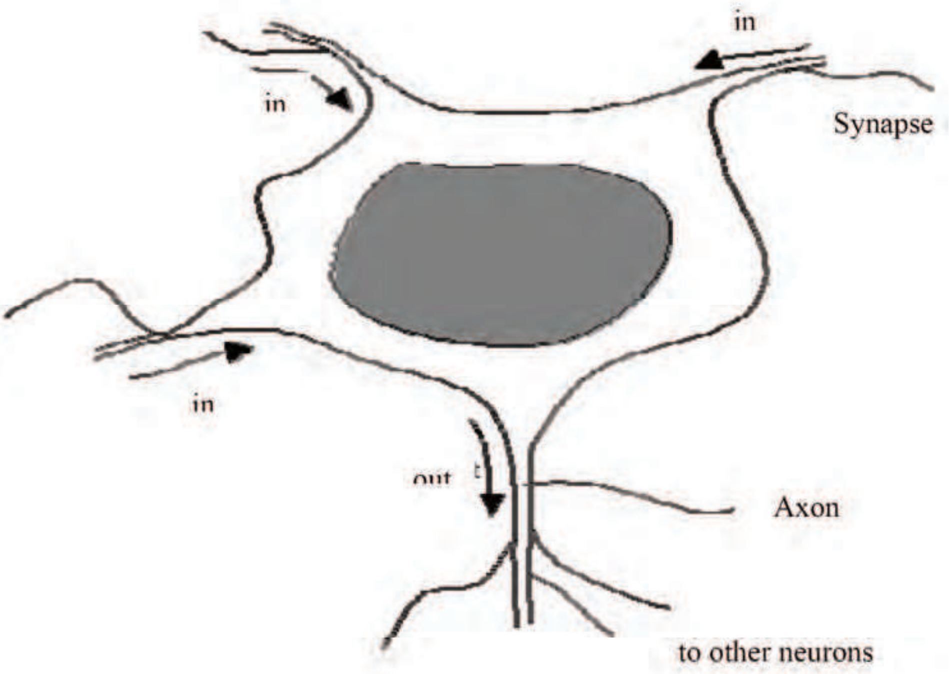


Figure 2(a). Schematic view of a real neuron

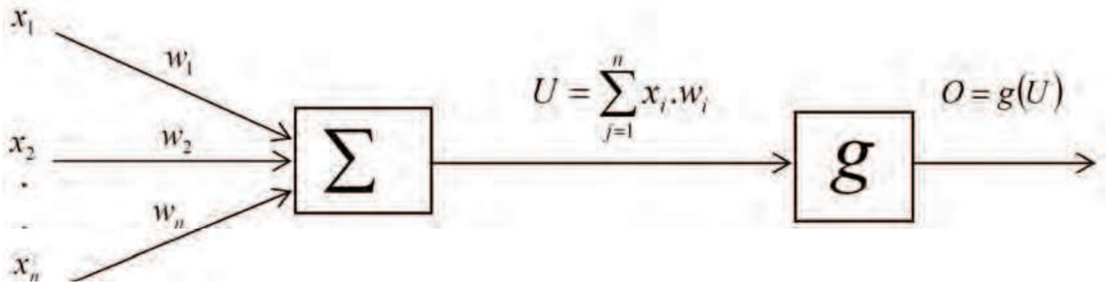


Figure 2(b) Schematic representation of the artificial neural network

Fig. 2(c) shows the currently loaded network. The connections can represent the current weight values for each weight. Squares represent input nodes; circles depict the neurons, the rightmost being the output layer. Triangles represent the bias for each neuron. The neural network consists of three layer, which are input, output and hidden layers. The input and outputs data are used as learning and testing data.

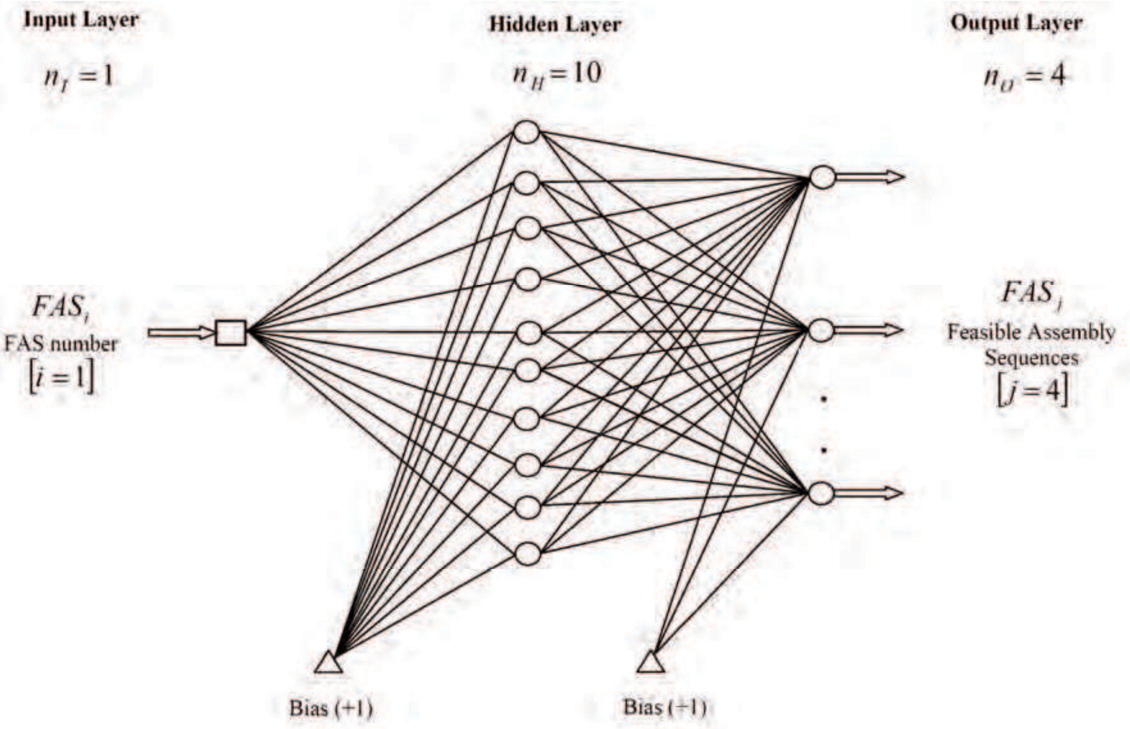


Figure 2(c) Currently loaded network

The most important and time-consuming part in neural network modeling is the training process. In some cases the choice of training method can have a substantial effect on the speed and accuracy of training. The best choice is dependent on the problem, and usually trial-and-error is needed to determine the best method. In this study, logistic function and back-propagation learning algorithm are employed to train the proposed NN.

Back propagation algorithm is used training algorithm for proposed neural networks. Back propagation is a minimization process that starts from the output and backwardly spreads the errors (Canbulut & Sinanoğlu, 2004). The weights are updated as follows;

$$\Delta w_{ij}(t) = -\eta \frac{\partial E(t)}{\partial w_{ij}(t)} + \alpha \Delta w_{ij}(t-1) \tag{1}$$

where, η is the learning rate, and α is the momentum term.

In this study, the logistic function is used to hidden layers and output layers. Linear function is taken for input layer. Logistic function is as follows;

$$y = f(x) = \frac{1}{1+e^{-x}} \tag{2}$$

Its derivative is;

$$\frac{\partial y}{\partial x} = y.(1-x) \tag{3}$$

The linear function is;

$$y = f(x) = x \tag{4}$$

Its derivative is;

$$\frac{\partial y}{\partial x} = 1 \tag{5}$$

Training and structural parameters of the network are given in Table 1.

Proposed Neural Network	η	μ	n_I	n_H	n_O	N	AF
	0.1	0	1	10	4	500000	$\log istic$

Table 1. Training and structural parameters of the proposed network

4. Modeling of Assembly System

An assembly is a composition of interconnected parts forming a stable unit. In order to modelling assembly system, it is used ACG whose nodes represent assembling parts and edges represent connections among parts. The assembly process consists of a succession of tasks, each of which consists of joining subassemblies to form a larger subassembly. The process starts with all parts separated and ends with all parts properly joined to form the whole assembly. For the current analyses, it is assumed that exactly two subassemblies are joined at each assembly task, and that after parts have been put together, the remain together until the end of the assembly process.

Due to this assumption, an assembly can be represented by a simple undirected graph $\langle P,C\rangle$, in which $P=\{p_1,p_2,...,p_N\}$ is the set of nodes, and $C=\{c_1,c_2,...,c_L\}$ is the set of edges. Each node in P corresponds to a part in the assembly, and there is one edge in C connecting every pair of nodes whose corresponding parts have at least one surface contact.

In order to explain the modeling of assembly system approach better way used for this research, we will take a sample assembly shown as exploded view in Fig. 3. The sample assembly is a pincer consisting of four components that are: bolt, left-handle, right-handle and nut. These parts are represented respectively by the symbols of $\{a\}$, $\{b\}$, $\{c\}$ and $\{d\}$. For this particular situation, the connection graph of assembly has the set of the nodes as $P=\{a,b,c,d\}$ and the set of the connections as $C=\{c_1,c_2,c_4,c_5\}$.

The connections or edges defining relationships between parts or nodes can be stated as: c_1 between parts $\{a\}$ and $\{b\}$, c_2 between parts $\{a\}$ and $\{d\}$, c_3 between parts $\{c\}$ and $\{d\}$, c_4 between parts $\{a\}$ and $\{c\}$ and finally c_5 between parts $\{b\}$ and $\{c\}$.

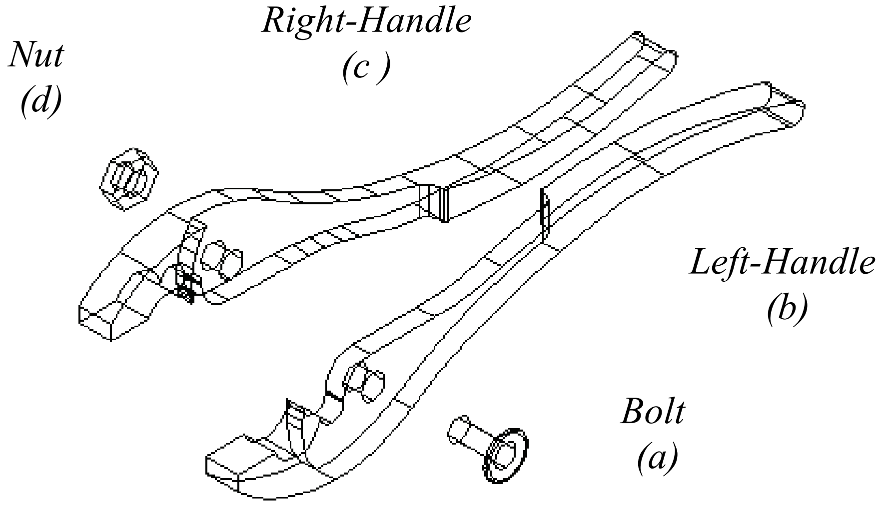


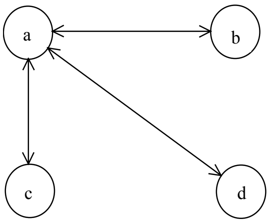
Figure 3. The pincer assembly system

4.1 Definition of Contact Matrices and ACG

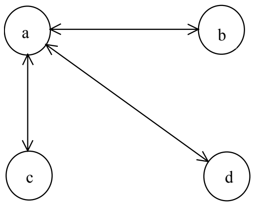
The contact matrices are used to determine whether there are contacts between parts in the assembly state. These matrices are represented by a contact condition between a pair of parts as an $\{A, B\}$. The elements of these matrices consist of *Boolean* values of *true* (1) or *false* (0). For the construction of contact matrices, the first part is taken as a reference. Then it is examined that whether this part has a contact relation in any i axis directions with other parts. If there is, that relation is defined as *true* (1), else that is defined as *false* (0).

The row and column element values of contact matrices in the definition of six main coordinate axis directions are relations between parts and that constitutes a pincer assembly. To determine these relations, the assembly's parts are located to rows and columns of the contact matrices. Contact matrices are square matrices and their dimensions are 4×4 for pincer.

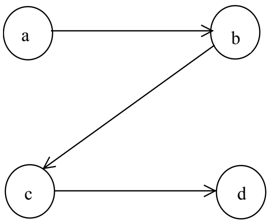
For example, $[a,b]$ element of B contact matrix in i direction is defined to whether there exists any contacts or not between parts $\{a\}$ and $\{b\}$ for the related direction and the corresponding matrix element may have the values of (1) and (0), respectively.



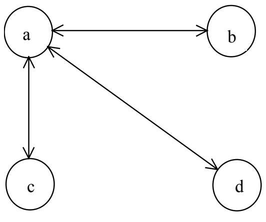
$$B_y = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$



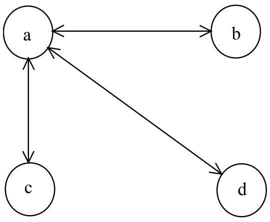
$$B_z = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$



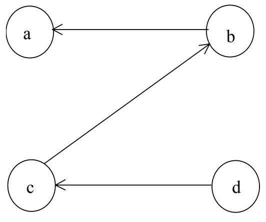
$$B_{-x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$



$$B_{-y} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$



$$B_{-z} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$



$$B_x = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Figure 4. Contact matrices and their graph representations

In this system, in order to get contact matrices in the direction of Cartesian coordinate axis, assembly view of pincer system was used. These matrices were automatically constructed (Sinanoğlu & Börklü, 2004). Contact matrices of the pincer assembly system are also shown in Fig. 4.

The connection graph can be obtained from contact matrices. To construct ACG, contact conditions are examined in both part's sequenced directions. For instance, in the manner of $\{a,b\}$ sequenced pair of parts, it is sufficient to determine contacts related sequenced direction so that its contact in any direction. Due to this reason, an $[\vee : Or]$ operator is applied to these parts. But it is also necessary contacts in any direction for inverse sequenced pairs of parts in the ACG. If these values are (1) for every sequenced pair of parts, then there should be edges between corresponding nodes of the ACG. For this purpose, every pair of parts must be determined.

- $\{a,b\}$, $\{b,a\}$ Sequenced pair of parts

To investigate whether there is an edge between $\{a\}$ and $\{b\}$ in ACG or not, it should be searched contact relations for these pairs of parts. Table 2 shows contact relations regarding $\{a,b\}$ and $\{b,a\}$ pairs of parts.

$c_1 \Rightarrow (a \div b)$	x	y	z	$-x$	$-y$	$-z$	$\vee : Or \Rightarrow$	$\wedge : And \Downarrow$
a/b	0	1	1	1	1	1	1	1
b/a	1	1	1	0	1	1	1	1
								1

Table 2. Contact relations of $\{a,b\}$ and $\{b,a\}$ pairs of parts

In this table, $\{a,b\}$ sequenced pair of parts is supplied to at least one contact condition in the related direction of $(0 \vee 1 \vee 1 \vee 1 \vee 1 \vee 1 = 1)$. $\{b,a\}$ pair of parts is also supplied to at least one contact in the related direction of $(1 \vee 1 \vee 1 \vee 0 \vee 1 \vee 1 = 1)$. An $(\wedge : And)$ operator is applied to these obtaining values. Because, these parts have at least one contact in each part sequenced direction, there is an edge between parts in the ACG. This connection states an edge in the ACG shown in Fig. 5.

If similar method is applied to other pairs of parts: $\{a,d\}$, $\{d,a\}$, $\{b,c\}$, $\{c,b\}$, $\{c,d\}$, $\{d,c\}$, $\{a,c\}$ and $\{c,a\}$, the results should be (1). Therefore, there are edges between these pairs in ACG.

The graph representation of this situation is shown in Fig. 5, where there is no edge between parts $\{b\}$ and $\{d\}$. Therefore, these parts do not have any contact relations.

Fig. 5 shows the pincer graph of connections. It has four nodes and five edges (connections). There is no contact between the left-handle and the nut. Therefore, the graph of connections does not include an edge connecting the nodes corresponding to the left-handle and the nut. By the use of the contact matrices and applying some logical operators to their elements, it is proved that it is supplied to one connection between two part in *ACG* not all contacts between them are established in every direction.

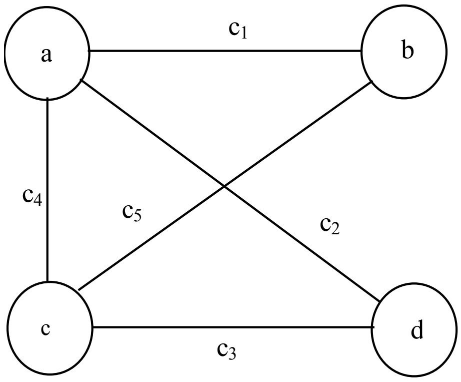


Figure 5. The graph of connections for four-part pincer assembly

5. Determination of Binary Vector Representation and Assembly States (ASs)

The state of the assembly process is the configuration of the parts at the beginning (or at the end) of an assembly task. The configuration of parts is given by the contacts that have been established. Therefore, in the developed approach an L-dimensional binary vector can represent a state of the process $(x = \{x_1, x_2, ..., x_L\})$. Elements of these vectors define the connection data between components. Based upon the establishment of the connections, the elements of these vectors may have the values of either (1) or (0) at any particular state of assembly task. For example, the i^{th} component x_i would have a value of *true* (1) if the i^{th} connection were established at that state. Otherwise, it would have a value of *false* (0). Moreover, every binary vector representations are not corresponding to an assembly state. In order to determine assembly states, the established connections in binary vectors and *ACG* are utilised together.

There are five edges in the example ACG. Because of that, the elements of vectors are five and the 5-dimensional binary vector of can represent that $[c_1, c_2, c_3, c_4, c_5]$. For instance, the initial state of the assembly process for the product shown in Fig. 3 can be represented by binary vector $[FFFFF]$ whereas the final state can be represented by $[TTTTT]$.

If the first task of the assembly process is the joining of the bolt to nut, the second state of the assembly process can be represented by $[FTFFF]$.

For example, an assembly sequence for pincer system can be represented as follows:

$$([FFFFF],[FTFFF],[TTTFF],[TTTTT]) \quad ([00000],[01000],[11100],[11111])$$

The first element of this list represents the initial state of the assembly process. The second element of the list shows the second connection c_2 between bolt and nut. The third element represents c_1 connection between right-handle and bolt and c_3 connection between right-handle and nut. The last element of the list is $[11111]$ and it means that every connection has been established.

In the developed planning system, first of all binary vector representations must be produced. The purpose of that it is classified to binary vectors according to the number of established connections. Table 3 shows vector representations for pincer assembly in Fig. 3. There are thirty-two different binary vectors. While some of them correspond to assembly state, some of them are not.

To form assembly sequences of pincer system, vector representations corresponds to assembly states must be determined. In order to determine whether the vector is a state or not, it must be taken into consideration established connections in vector representation. And then it is required that establishing connections must be determined to established connections by ACG.
For instance, if the first task of the assembly process is the joining of the bolt to the left-handle, the second state of the assembly process can be represented by $[10000]$. It is seen in Fig. 6 that it does not necessary to establish any connection so that c_1 connection between part $\{a\}$ and $\{b\}$ is establish. Therefore, $[10000]$ vector is an assembly state. Therefore, vectors only one established connection form assembly state.

LEVEL 0	LEVEL 1	*LEVEL* 2			LEVEL 3
00000	10000		11100		11111
		11000	11010	11110	
		10100	11001	11101	
		10010	10110	11011	
		10001	10101	10111	
			10011		
	01000		11100		
		11000	11010	11110	
		01100	11001	11101	
		01010	01110	11011	
		01001	01101	01111	
			01011		
	00100		11100		
		10100	10110	11110	
		01100	10101	11101	
		00110	01110	10111	
		00101	01101	01111	
			00111		
	00010		11010		
		10010	10110	11110	
		01010	10011	11011	
		00110	01110	10111	
		00011	01011	01111	
			00111		
	00001		11001		
		10001	10101	11101	
		01001	10011	11011	
		00101	01101	10111	
		00011	01011	01111	
			00111		

Table 3. Hierarchical levels of binary vector representations for pincer assembly system

Moreover, some of vectors do not correspond to an assembly state. For instance, in the $[10001]$ vector, connections of c_1 between $\{a\}$ and $\{b\}$, c_5 between $\{b\}$ and $\{c\}$ have been established (1). It has been necessary to establish c_4 connection between $\{a\}$ and $\{c\}$ so that these connections have been established (Fig. 6). But this connection has not been established in $[10001]$, $[10001]$ vector is not an assembly state.

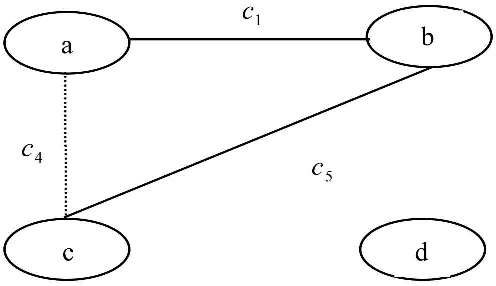


Figure 6 c_1, c_5 and c_4 connections in $[10001]$ vector

There are thirteen assembly states in pincer assembly system. These are;

$[00000], [10000], [01000], [00100], [00010], [00001], [11000], [10100], [01001], [00101], [10011], [01110], [11111]$

6. Productions and Representation of Assembly Sequences

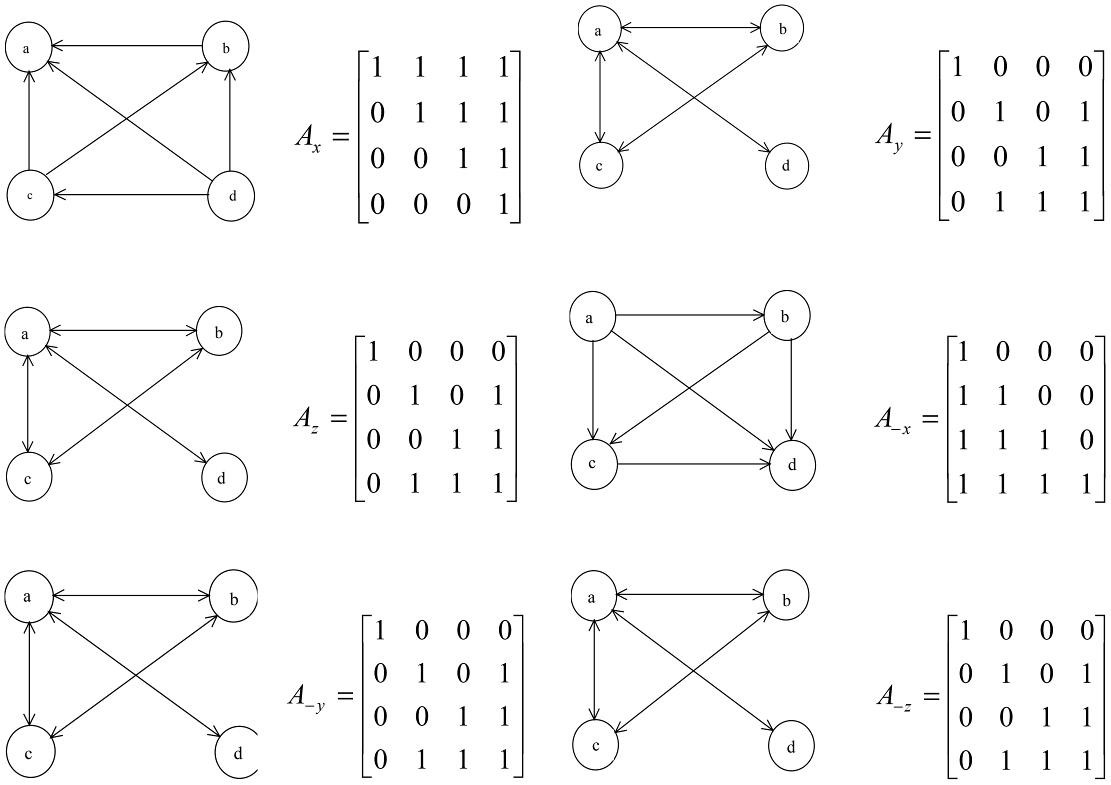
Given an assembly whose graph of connections is $\langle P, C\rangle$, a directed graph can be used to represent the set of all assembly sequences (Homem de Mello & Lee, 1991). The directed graph of feasible assembly sequences of an assembly whose set of parts is P is the directed graph $\langle x_p, T_p\rangle$ in which, x_p is the assembly's set of stable states, and T_p is the assembly's set of feasible state transitions.

In the pincer assembly, $P = \{a, b, c, d\}$ is the assembly's set of parts or set of nodes, $C = \{c_1, c_2, c_3, c_4, c_5\}$ is the assembly's set of connections or set of edges. $\langle x_p, T_p\rangle$ corresponds to directed graph of pincer system. A path in the directed graph of feasible assembly sequences $\langle x_p, T_p\rangle$ whose initial node is $\Theta_I = \{\{a\}, \{b\}, \{c\}, \{d\}\}$ and whose terminal node are $\Theta_F = \{\{a, b, c, d\}\}$. Vector representations of these sets are $[00000]$ and $[11111]$ respectively.

Assembly states not correspond to feasible assembly sequences must eliminate by some assembly constraints. In this study, three assembly constraints are applied to assembly states. These are subassembly, stability and geometric feasibility constraints.

The subassembly constraint defines feasibility of subassembly of set of partitions to established connections in assembly states. In order to form a subassembly of a set of partition, it is not a set of partition contains a pair of part has not contact relation in the ACG. Therefore, in pincer assembly bolt and left-handle has not contact relations. Because of that it is not supplied to subassembly constraint set partitions contains $\{b,d\}$ set of partition.

The second constraints is stability. A subassembly is said to be stable if its parts maintain their relative position and do not break contact spontaneously. All one-part subassemblies are stable.



$$A_x = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad A_y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

$$A_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \qquad A_{-x} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$A_{-y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \qquad A_{-z} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

Figure 7. Interference matrices and their graph representations for pincer assembly

The last constraint is geometric feasibility. An assembly task is said to be geometrically feasible if there is a collision-free path to bring the two subassemblies into contact from a situation in which they are far apart.

Geometric feasibility of binary vectors correspond to assembly states are determined by interference matrices. The elements of interference matrices were taken into consideration interference conditions during the joining parts. In the determination of geometric feasibility, it is applied to elements of interference matrices $(\wedge)$ and $(\vee)$ logical operators. At this operation, it must be utilise established connections and that is joining pairs of part. In order to, whether binary vector representations corresponds to assembly states are geometrically feasible or not, it is necessary to applying Cartesian product between sequenced pairs of parts which are representing established connections and parts which are not in this sequenced pairs.
In the determination of interference matrices elements, it is taken into consideration interference while the reference part is moving with another part along with related axis direction. If it is interference during this transformation motion, interference matrices elements are (0) if not are defined as (1).

For instance, in the A_x matrice the movement of part bolt is interfered to movement along with $\{+x\}$ axis by other parts. Therefore, the first row elements of A_x matrice defined to interference among parts along this axis are (1). But the movement of left-handle along with $\{+x\}$ axis does not interfere any parts (Fig. 3). This interference relation is illustrated to designate (0) value by element of second row and third column in A_x matrice. These matrices are also formed automatically from various assembly views.

Graph representations for the pincer assembly and construction of their interference matrices can be also determined as follows (Fig. 7).

In order to determine whether assembly states are geometrically feasible or not, it is necessary to apply Cartesian product between sequenced pairs of parts which represent established connections and parts which are not in this sequenced pairs of parts. In this situation, different interference tables are obtained and these tables are used to check geometric feasibility.

- $[01000]$ Assembly State

In this assembly state, connection of c_2 between part $\{a\}$ and $\{d\}$ has been established. To determine geometric feasibility of this assembly state, parts