



## Artificial immune systems for assembly sequence planning exploration

Chien-Cheng Chang<sup>a</sup>, Hwai-En Tseng<sup>b,\*</sup>, Ling-Peng Meng<sup>b</sup>

<sup>a</sup> Department of Industrial Design, Huaan University, No. 1, Huafan Rd., Shihdin Township, Taipei County, 222 Taiwan, Republic of China

<sup>b</sup> Department of Industrial Engineering and Management, National Chin-Yi University of Technology, 35, Lane 215, Section 1, Chung-Shan Road, Taiping City, Taichung County, 411 Taiwan, Republic of China

### ARTICLE INFO

#### Article history:

Received 24 February 2008

Received in revised form

10 March 2009

Accepted 15 April 2009

Available online 5 June 2009

#### Keywords:

Artificial immune system

Guided genetic algorithms

Memetic algorithms

Connector

Clonal selection

### ABSTRACT

Assembly sequence planning (ASP) needs to take relevant constraint factors such as the geometric characteristics and tool factors into consideration so as to work out a particular assembly sequence. At last, a product will come into being through the assembly of each part according to the assembly sequence. A problem encountered in ASP is that a larger number of components will cause more constraints to assembly a product, thus increasing the complexity of assembly problem. Therefore, it has been an objective for researchers to look for suitable methods for the solution space of feasible solutions.

Among them, traditional genetic algorithms (GAs) belong to a random searching method. When the constraints are complicated in ASP, GAs often come out with a large number of solutions not feasible. Consequently, previous research results have proposed some approaches such as Guided genetic algorithms (Guided-GAs) or memetic algorithms (MAs) to enhance the structure of GAs to cope with the complexity of constraints in ASP problems. In this study, artificial immune systems (AIS) were proposed to help solve the assembly sequence problem. In AIS algorithm, the antibody (Ab) in the immune system is simulated to encounter one or more unknown antigens (Ags). Moreover, the clonal selection concept is employed in the immune system in which a better antibody will be selected in each generation of revolution and different antibodies will be cloned to protect the infection of the original antigen. With this mechanism, the shortcoming such as the traditional GAs to converge in local optimal solution will be overcome. Practical examples have demonstrated that AIS can solve the ASP problem with complicated constraints. Compared with guided genetic algorithms and memetic algorithms, AIS can generate the same or better solutions in terms of quality and searching time.

© 2009 Elsevier Ltd. All rights reserved.

### 1. Introduction

Recently, the type of bill of order in enterprises has been transformed into a customer-oriented manufacturing mode. Instead of mass production of fewer types of products, more varieties of product configurations with smaller batches can be seen under this trend of mass customization. Therefore, the fast change of customers' needs draws forth a tremendous impact upon the current design of the manufacturing system indeed (Eimaraghy, 2006). Based upon the concept of concurrent engineering (CE), if the planning of manufacturing can be integrated in the stage of product design, both the lead time and production costs will be reduced noticeably. For most industrial products, the assembly costs take up 10–30% of total production costs (Hong and Cho, 1997). Therefore, from the viewpoint of CE, if the Research and Development (R&D) of new products can be integrated with assembly planning at the early stage, the effectiveness will be uplifted. In assembly planning

problems, many relevant constraint factors will be taken into consideration; for example, the geometric and tool factors. For that reason, how to work out a specific assembly sequence quickly and correctly and assemble parts into a product is one of the important issues in current manufacturing environment we cannot overemphasize.

Assembly planning problems can be divided into two major parts: assembly modeling and assembly sequence planning (ASP) (Fujimoto and Sebaaly, 2000). In assembly modeling, data for computer-aided design are transformed into constraints and regulations of assembly procedure. Such kinds of data consist of geometric and non-geometric information through which constraints for assembly sequence will be formed and followed. Generally, these constraints are presented by the precedence graph. In the past, the Liaison graph proposed by De Fazio and Whitney (1987) and the graphic exhaustive search were used to solve the ASP problem (Baldwin et al., 1991; Homem De Mello and Sanderson, 1991; Gottipolu and Ghosh, 1997). Though it is possible to find feasible solutions or even the optimal solution from the graphic methodology, the time and problem scale is largely limited. Moreover, because Liaison graph solves assembly planning problems with information of parts, when the number of

\* Corresponding author. Tel.: +886 4 23924505x6012; fax: +886 4 23921742.  
E-mail address: [hwai\\_en@seed.net.tw](mailto:hwai_en@seed.net.tw) (H.-E. Tseng).

parts increases dramatically, the amount of information will also increase astonishingly, adding difficulties to the searching of solutions.

To cope with the considerable expand and change of solution space in the mass customization environment, many researchers have tried AI-related methodologies. Such methodologies include A\* algorithms (Laperrière and Eimaraghy, 1996), AO\* search method (Kim et al., 1996), neural networks (Hong and Cho, 1995), simulated annealing method (Motavalli and Islam, 1997), and genetic algorithms (GAs) (Smith and Smith, 2003). Among the approaches, genetic algorithms have been attempted by most researchers in that GAs can effectively expand the solution space in the traditional graphic methods.

Reconsideration of the information of assembly parts of a product offers another approach to ASP problems. Tseng and Li (1999) once proposed a connector-based assembly planning. The so-called connector is a way of product description based upon the combination of parts. It acts as a sort of concept product building block at the design stage and may contain more engineering information to incorporate with product functions. Connectors come in various types. Sonnenberg and Sodhi (2003) have tried to generate categorizations of connectors. Chung and Peng (2006) incorporated the concept of connectors into the combination of selective disassembly planning and genetic algorithms. In order to solve complex constraints caused by large numbers of parts, Tseng (2006) proposed a Guided genetic algorithms (Guided-GAs) in which the mechanisms in GAs were changed to adapt to the ASP environment. Later, the concept of memetic algorithms (MAs) was used to upgrade the quality of solution searching but the solution time had often been found longer than that in Guided-GAs (Tseng et al., 2007).

Taking the quality and time of solution searching into consideration, the present study is focused upon uplifting the effectiveness of genetic algorithms. More importantly, it is another reasonable way to apply artificial immune systems (AIS) (De Castro and Timmis, 2003). The good quality of solution searching in its application in other domains demonstrates that AIS has been progressively accepted by researchers. For example, it has been applied in PCB flow shop scheduling (Alisantoso et al., 2003), a redundant reliability problem with multiple component choices (Chen and You, 2005) and the facility layout problem (Khilwani et al., 2008).

In addition, Cao and Xiao (2007) used the random searching method to generate the initial solution and tried an immune approach to solve the ASP problem, in which the affinity values of antibody (Ab) and antigen (Ag) were compared. In their study, a controller assembly case made up of 19 components as used to illustrate their algorithm. As mentioned earlier, in the liaison-based ASP approach, when the case is complicated, it is difficult to generate solutions because of the complex constraint factors. At the present study, the connector-based approach was used and the Adjacency List method was adopted in the generation of initial solutions so that all of them were feasible ones. Hence, it is not necessary to compare the affinity value of antibody and antigen. Instead, the variety among antibodies was investigated. More importantly, the variety among in the memory zone was maintained by deleting the affinity values between antibodies.

Briefly, it is easy to combine the concept of AIS with genetic algorithms for ASP problems. Reasonable outputs can be seen from the addition of AIS mechanism in the original GAs structure. In this paper, the relationships and application of AIS in ASP is discussed in Section 2. The concept of connectors is delineated in Section 3. In Section 4, modified AIS algorithms are proposed. Practical examples are illustrated to verify the proposed method in Section 5 and finally some concluding remarks are reached in Section 6.

## 2. Outline of artificial immune systems

The immunity system refers to the procedure of immune cells to resist infection from microorganisms or viruses, especially as a result of antibody formation. The viruses or organisms that trigger the immune system to react are defined as an antigen. From the stimulus of antigens, the immune cells that are combined with antigens are the antibody. There are many levels of defense in an organism's immune system. And the algorithm derived from the immune system boasts of specialization and memory characteristics from the adaptive immune response (De Castro and Von Zuben, 2002).

Among immune cells, B cell and T cell are the focus of people's attention. The so-called B cell refers to B lymphocyte, which is generated by the bone marrow. From the stimulus of antibody, B cell will have a reaction and secrete some antibody to combine with the antigen. T cell is the T lymphocyte that gets ripe in the thymus. T cell's major function is to recognize and destroy the outer invader. In other words, whether B cell can effectively produce antibody to resist from outer viruses depends on the help of T cell. There are many receptors on the surfaces of B cell and T cell. Through the immune stage, different epitopes will be produced from the antigen. Some bio-chemical reaction happens when these two things contact each other. The degree of their combination is called the affinity. The more closely the receptor combines with the epitope, the higher the degree of the affinity. This is the specialization characteristic of immune systems.

From the recognition in immune system, B cell will be stimulated and activated if the affinity between B cell and the antigen exceeds a threshold value. Such an affinity selection is called the affinity maturation. By doing this way, the immune system will generate a specific integrated cell to clone in a large quantity. And most of the newly generated B cells will become plasma cells serving as the antibody and the rest as the memory cells. Through blood circulation, the memory cells will reach the lymphocyte structure and be stored in T cells and B cells. The memory cells have the prevention function. If the immune system detects the infection from the same antigen, the lymphocyte cells will be reactivated. Such a procedure is called the clonal selection (De Castro and Von Zuben, 2002).

An overview of AIS can be found by some articles (De Castro and Timmis, 2003; Musilek et al., 2006). Furthermore, many studies have explored ASP and outstanding achievements can be seen from the investigation of GAs. With AIS, some problems in GAs can be overcome. First of all, the searching of solutions in original GAs mainly uses the fitness function to find the near-optimal solution. In AIS, the affinity between antibodies and the objective functions are mixed to find the near-optimal solution. In addition, the control strategy used in AIS differs from that in GAs. AIS take advantage of the memory characteristic to speed up the algorithm for local searching. The variety in memory units is sustained to keep from the falling of local solutions. Without evident memory units, GAs keep on the evolutionary procedure lacking in the stopping mechanism, making it easy to get into the local optimal solution.

If we define the antigen as the objective function for a connector-based assembly planning, the antibody can be defined as the solution to the corresponding objective function. And the affinity serves as an index for the combination of the antibody and antigen in AIS. In ASP, the variety of assembly sequence is emphasized. The fundamental concepts of the AIS algorithm specially designed for ASP are listed below:

- (1) The initial solution is built-up through greedy algorithms that will be described in Section 4.2. The initial solution should

meet the constraints described in the precedence graph so as to generate feasible solutions. Furthermore, the concept of clonal selection is used to select better antibodies in the memory cells, and split into a proper number of new cells required.

- (2) The crossover and mutation mechanisms in GAs are used to generate multiple feasible solutions to reach the variety goal of antibodies.
- (3) In terms of memory, each solution searching is recorded. The selected antibody in the memory cell area will react quickly to the antigen. Finally, an optimal antibody will be generated from the memory cell area.

### 3. Concepts and information content of connectors

#### 3.1. Connector concepts

Connectors indicate the connecting way between parts. They can serve as the concept product building block at the early

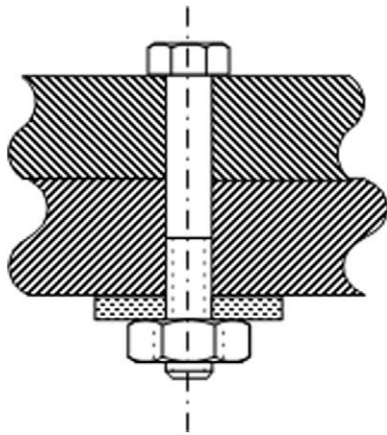


Fig. 1. Connector example: bolt–nut–washer.

Table 1  
Classification of connector types according to Akagi et al. (1980).

Type	Code	Example
<i>Fixed connector</i>		
Disassembled	FD	Screw, bolted joint, key, spline, wedge
Not disassembled	FND	Pressing fits, riveted joints, welding
<i>Movable connector</i>		
Disassembled	MD	Snap ring, bearing, spring
Not disassembled	MND	Races and ball-bearing balls

Table 2  
Classification of assembly tools.

Level	Force magnitude	Tool name	Details on assembly operation
$T_1$	None	Hand	No tools are needed, i.e., the assembly is manual.
$T_2$	Small	Work bench, handgun, screw driver, spanner, pliers	Use a simple hand tool to assemble, no strict interference occurs between components.
$T_3$	Medium	Screw driver, spanner, ratchet spanner, hand vice	Use simple hand tool to assemble; other tools are needed to support the assembly work.
$T_4$	Large	Hacksaw, heavy sledgehammer, crusher, torsional twister, chassis	Use a special tool to assemble the product; the operation may cause a destructive result.

product design stage. Fig. 1 shows the connection of a bolted joint and bolt–nut–washer. With the description of connectors, the complexity of product description can be largely reduced due to fact that the assembly properties of these three parts are integrated in the basic units of connectors (Tseng and Li, 1999).

In this study, the combination property, assembly tool, assembly direction, and the precedence relationships of connectors are considered. They are discussed as follows:

- *Combination property*: From Table 1, four types of connectors can be categorized. They include disassembleable fixed connectors, undisassembleable fixed connectors, disassembleable movable connectors, and undisassembleable movable connectors (Akagi et al., 1980). They can serve as references for initial categorization of parts.
- *Tool property*: In assembly planning, the selection of assembly tool will influence the efficiency and fluency of the assembly task. According to the difficulties of assembly tasks, four types of assembly tools are specified (Table 2).
- *Direction property*: The assembly direction will influence the process of assembly task. Six indexes,  $\pm x$ ,  $\pm y$ , and  $\pm z$ , are used in this study to describe the accessed direction.
- *Precedence relationship*: The precedence relationship of connectors is judged according to the engineering attributes or geometric information of the parts belonging to the connector. At the present study, the precedence relationships among connectors are assumed to be preliminarily set.

Take the stapler as an example. There are 18 parts in the stapler (Fig. 2(a)). Nine connectors can be assigned (Table 3) according to the combination, assembly tool and assembly direction properties. For instance, connector 7 ( $C_7$ ) is composed of three parts, steel cover, bracket spring, rivet1 compose. Its combination property belongs to tight fastening, a disassembleable fixed connector; the assembly direction is  $y$ ; the assembly tool uses a hand vice ( $T_3$ ). Fig. 2(b) shows the precedence graph of the connectors in a stapler.

#### 3.2. Describing precedence sequence by adjacency list

The adjacency listing technique in data structure is used to deal with the precedence graph of connectors. First, the Outdegree and Indegree are defined as follows:

*Outdegree*: The number of nodes connected after a specific node. Take node  $C_i$  for instance, its Outdegree means the number of nodes after  $C_i$ . If  $C_i \rightarrow C_j$  in the sequential order, then the number of  $C_j$  is the Outdegree node of  $C_i$ .

*Indegree*: The number of nodes connected before a specific node. Take node  $C_i$  as an example, its Indegree means the number of nodes before  $C_i$ . If  $C_j \rightarrow C_i$  in the sequential order, then the number  $C_j$  is the Indegree node of  $C_i$ .

In this study, the Outdegree and Indegree are used to construct the successor lists (SL) and predecessor lists (PL). SL represents the lists where successors in each node in the forward searching, from

the source to the sink, are recorded while PL represents the lists in which predecessors in each node in the backward searching, from the sink to the source, are taken down.

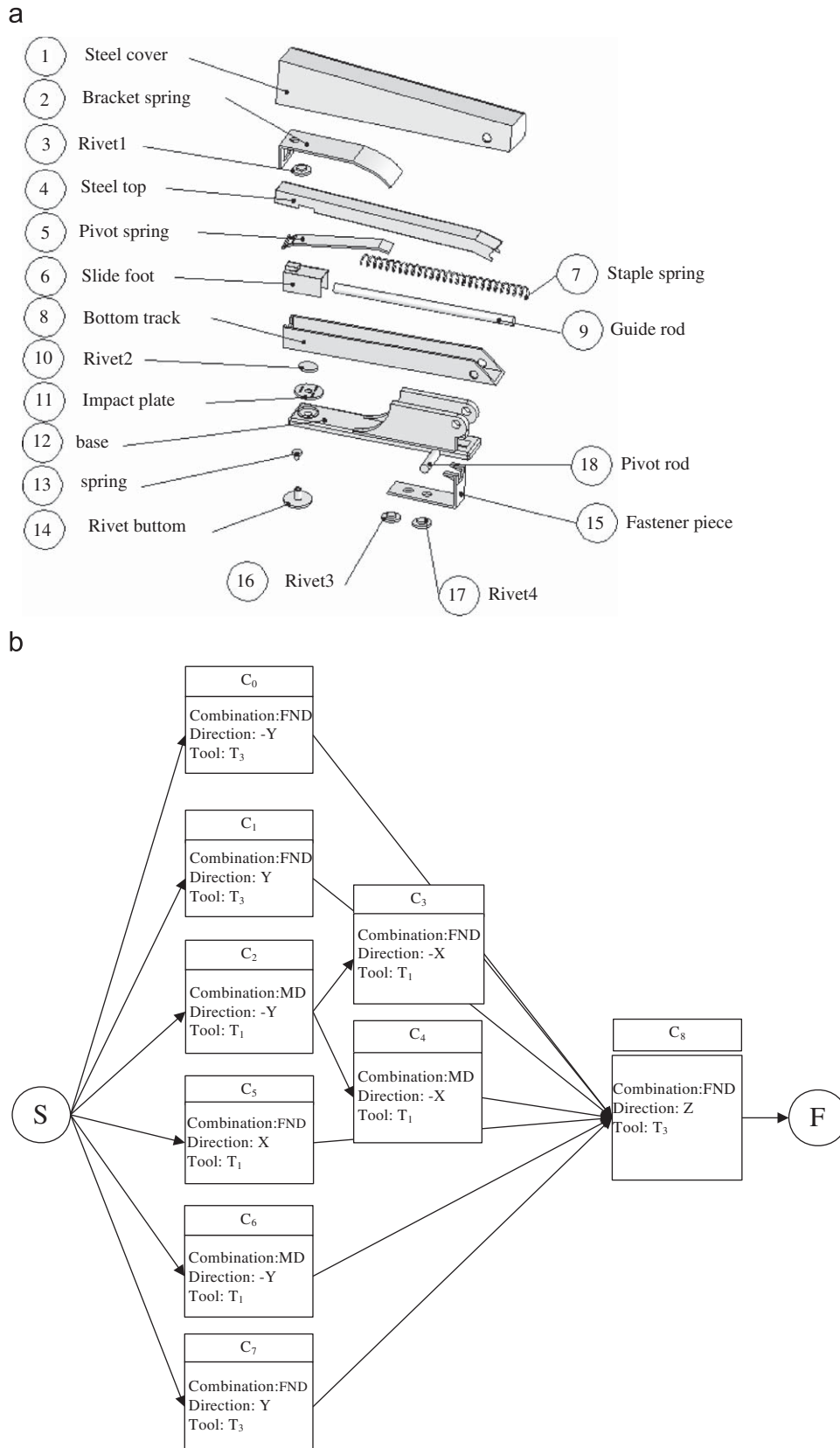


Fig. 2. (a) Stapler: diagram of parts and (b) stapler: connector-based precedence graph.

The algorithm to generate SL list based upon the Outdegree is listed below:

- Step 1: Enter information for the precedence graph of connectors.
- Step 2: From the precedence graph, find the nodes after the first node according to its Outdegree. For instance, node  $C_j$  is the Outdegree of node  $C_i$ . Since  $C_j$  is a node after  $C_i$ , the Outdegree of  $C_i$  is 1.

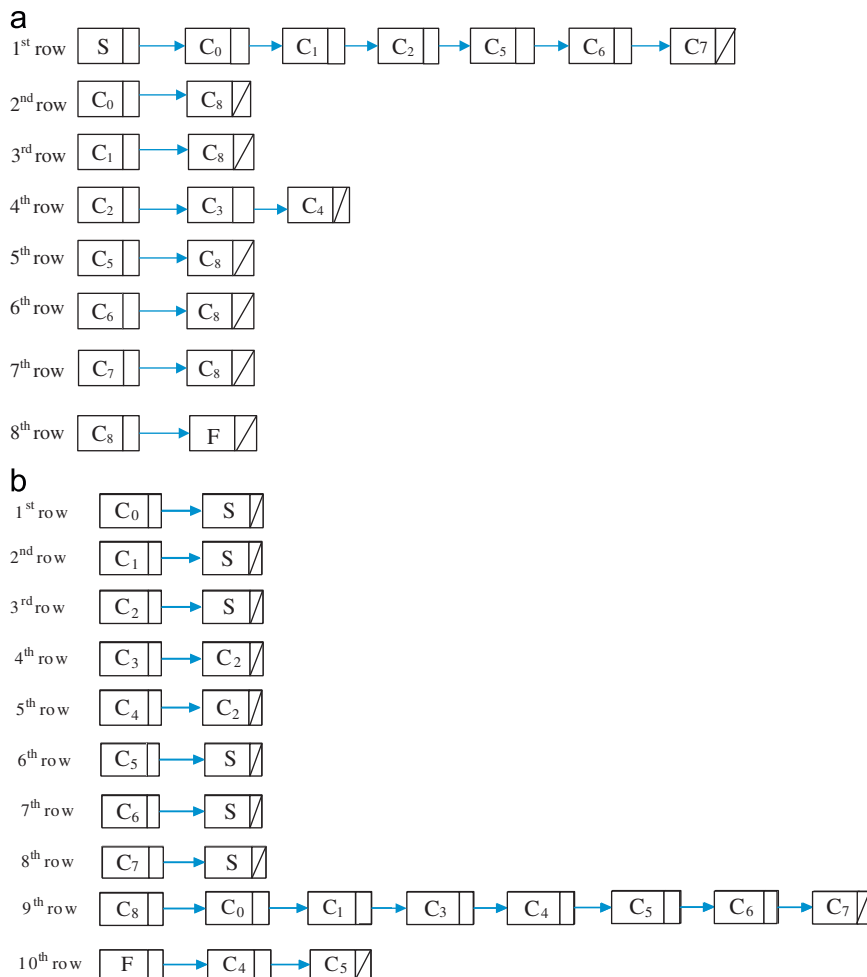
- Step 3: Connect the nodes after the first node. In the sequential order,  $C_i$  possesses a higher priority than  $C_j$ . In the case of Step 2, this can be expressed as  $C_i \rightarrow C_j$ .
- Step 4: Repeat the above steps searching for the nodes in the precedence graph. If yes, repeat Step 2. Otherwise, stop the searching.

Take the stapler parts in Fig. 2(a) as an example. Fig. 3 shows the adjacency list for stapler under the precedence situations of

**Table 3**  
Connector information for stapler.

No.	Connector name	Combination type	Direction	Tool	Component owned by connector
$C_0$	Interference fit	FND	-y	$T_3$	10, 11, 12, 13, 14
$C_1$	Interference fit	FND	y	$T_3$	12, 15, 16, 17
$C_2$	Spring	MD	-x	$T_1$	7, 9
$C_3$	Insert	FND	-x	$T_1$	6, 9
$C_4$	Spring	MD	-x	$T_1$	6, 7
$C_5$	Insert	FND	x	$T_1$	8, 9
$C_6$	Snap fit	MD	-y	$T_1$	6, 5, 4
$C_7$	Interference	FND	y	$T_3$	1, 2, 3
$C_8$	Interference	FND	z	$T_3$	1, 4, 8, 12, 18

FD: fixed fastener disassembled;  
 FND: fixed fastener not disassembled;  
 MD: movable fastener disassembled;  
 MND: movable fastener not disassembled.



**Fig. 3.** Stapler: adjacency list example: (a) successor lists (SL) and (b) predecessor lists (PL).

Fig. 2(b). The transformation procedure is described in the following two parts:

As can be seen in Fig. 3(a), the above algorithm is used to generate successor lists according to the Outdegree.

- Step 1: Enter the data of the precedence graph in Fig. 2(b).
- Step 2: In Fig. 2(b), specify the nodes connected after the first virtual node S. Judging from the Outdegree, these nodes include C<sub>0</sub>, C<sub>1</sub>, C<sub>2</sub>, C<sub>5</sub>, C<sub>6</sub>, and C<sub>7</sub>.
- Step 3: Connect the nodes C<sub>0</sub>, C<sub>1</sub>, C<sub>2</sub>, C<sub>5</sub>, C<sub>6</sub>, C<sub>7</sub> after the virtual node S. As can be seen in Fig. 3(a), the virtual node S possesses higher priority in assembly sequence than C<sub>0</sub>, C<sub>1</sub>, C<sub>2</sub>, C<sub>5</sub>, C<sub>6</sub>, and C<sub>7</sub>.
- Step 4: Step 2 is repeated because some other nodes can be found in Fig. 2(b).
- Step 2: Based upon the data of Outdegree, node C<sub>8</sub> will be connected after node C<sub>0</sub>.
- Step 3: Connect node C<sub>8</sub> to node C<sub>0</sub> as can be seen on second row in Fig. 3(a), indicating the precedence relationship of C<sub>0</sub>→C<sub>8</sub>.
- Step 4: Execute Step 2.
- Step 2: According to the Outdegree of node C<sub>1</sub>, node C<sub>8</sub> should be connected after node C<sub>1</sub>.
- Step 3: After node C<sub>1</sub>, place node C<sub>8</sub> as can be seen on third row in Fig. 3(a). The precedence relationship between node C<sub>1</sub> and node C<sub>8</sub> is C<sub>1</sub>→C<sub>8</sub>.
- Step 4: Execute Step 2.
- Step 2: For node C<sub>2</sub>, the Outdegree data shows that nodes C<sub>3</sub> and C<sub>4</sub> are connected after C<sub>2</sub>.
- Step 3: After node C<sub>2</sub>, place nodes C<sub>3</sub> and C<sub>4</sub> on fourth row in Fig. 3(a), indicating the precedence relationships of C<sub>2</sub>→C<sub>3</sub> and C<sub>2</sub>→C<sub>4</sub>.

Repeat the process until eighth row is constructed. The whole Successor list is then completed.

Besides, the predecessor lists can be generated from the Indegree data. This is described in the following steps.

- Step 1: Enter the connector data in the precedence graph.
- Step 2: Check the nodes connected after the first node in the precedence graph according to its Indegree data. For example, node C<sub>j</sub> is the Indegree of node C<sub>i</sub>. In other words, the Indegree of node C<sub>i</sub> equals to 1, indicating that C<sub>j</sub> is a node connected after C<sub>i</sub>.
- Step 3: Connect all the nodes after node C<sub>i</sub> and we can find that C<sub>j</sub> possesses higher priority than C<sub>i</sub> in precedence sequential order. From Step 2, it can be denoted as C<sub>j</sub>→C<sub>i</sub>.
- Step 4: Repeat the above steps and check whether there exist in the precedence graph other nodes for further searching. If yes, repeat Step 2; otherwise stop the transformation procedure and complete the construction of the connection of connectors.

In Fig. 3(b), Indegree is used to generate PL in a way similar to the generation of SL. This part is omitted. In this study, SL and PL lists are used to record the precedence graph. The data will be used in Section 4.3 to search for the initial solution.

### 3.3. Connector engineering data similarity matrix SS

At the present study, the engineering information similarity of connector is employed for the evaluation of the value of the design objective function. According to GÖNGÖR and GUPTA's study (1997), when the frequency of alternation of the precedence

sequence for a product is lower, the probability it will fit and survive is higher. To compare connector C<sub>i</sub> with connector C<sub>i+1</sub> from a connector set V whose precedence sequence has been set, if the contents of engineering information (combination, direction, and tools) for connectors C<sub>i</sub> and C<sub>i+1</sub> are similar, it is more likely that they will survive. To do this, we need to build up the engineering information similarity matrix SS first. SS<sub>ij</sub> represents engineering information similarity of the cell of *i*th row and *j*th column in a symmetrical matrix; i.e., the engineering similarity between *i*th connector and *j*th connector. SS<sub>ij</sub> can be calculated from Formula (1):

$$SS_{ij} = W_c \times C_{ij} + W_d \times D_{ij} + W_t \times T_{ij} \tag{1}$$

where SS<sub>ij</sub> is the engineering similarity between *i*th connector and *j*th connector. If *i* = *j*, they represent the same connectors and it is meaningless to give values to them, as shown in Fig. 4; *i* and *j* = 1,2,...,*m*, where *m* is the number of connectors. C<sub>ij</sub> is the combination condition between the connectors. When the combination property of connector C<sub>i</sub> and C<sub>j</sub> is the same, C<sub>ij</sub> = 1; otherwise, C<sub>ij</sub> = 0; D<sub>ij</sub> the direction condition between the connectors. When the direction property of connector D<sub>i</sub> and D<sub>j</sub> is the same, D<sub>ij</sub> = 1; otherwise, D<sub>ij</sub> = 0; T<sub>ij</sub> is the tool condition between the connectors. When the tool property of connector T<sub>i</sub> and T<sub>j</sub> is the same, T<sub>ij</sub> = 1; otherwise, T<sub>ij</sub> = 0. W<sub>c</sub> is weight of the combination property; W<sub>d</sub> the weight of the assembly direction and W<sub>t</sub> the weight of the assembly tool.

Take the stapler as an example. To compare with Guided-GAs (Tseng, 2006) and MAs (Tseng et al., 2007), the relative weight for combination, assembly tool and assembly direction properties are set to be equal. Since W<sub>c</sub>W<sub>d</sub>W<sub>t</sub> = 1, W<sub>c</sub> = W<sub>d</sub> = W<sub>t</sub> = 2/6. Through Formula (1), SS<sub>0,1</sub> in SS matrix is 2/6 × 1 + 2/6 × 1 + 2/6 × 0 = 0.6667. Other elements in SS matrix can be obtained in the same way. The result is shown in Fig. 4.

	C <sub>0</sub>	C <sub>1</sub>	C <sub>2</sub>	C <sub>3</sub>	C <sub>4</sub>	C <sub>5</sub>	C <sub>6</sub>	C <sub>7</sub>	C <sub>8</sub>
C <sub>0</sub>	---	0.67	0	0.33	0	0.33	0.33	0.67	0.67
C <sub>1</sub>	0.67	---	0	0.33	0	0.33	0	1	0.67
C <sub>2</sub>	0	0	---	0.67	1	0.33	0.67	0	0
C <sub>3</sub>	0.33	0.33	0.67	---	0.67	0.67	0.33	0.33	0.33
C <sub>4</sub>	0	0	1	0.67	---	0.33	0.67	0	0
C <sub>5</sub>	0.33	0.33	0.33	0.67	0.33	---	0.33	0.33	0.33
C <sub>6</sub>	0.33	0	0.67	0.33	0.67	0.33	---	0	0
C <sub>7</sub>	0.67	1	0	0.33	0	0.33	0	---	0.67
C <sub>8</sub>	0.67	0.67	0	0.33	0	0.33	0	0.67	---

Fig. 4. Stapler: similarity matrix SS for engineering information.

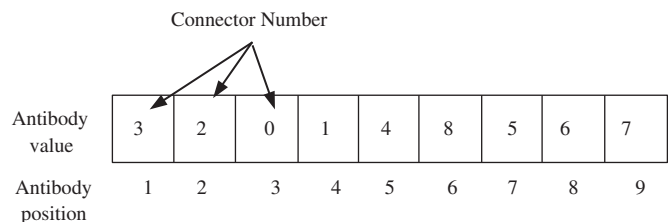


Fig. 5. Representation of an antibody for stapler example.

3.4. Representation scheme

In this study, decimal numbers are used to represent the chromosomes of a solution (an antibody). The length of the chromosomes is equal to the number of connectors in a product. Each connector has a number of which the position represents the assembly sequence of the specific connector. In the case of stapler (Fig. 5), an antibody represents the assembly sequence of stapler's connectors,  $3 \rightarrow 2 \rightarrow 0 \rightarrow 1 \rightarrow 4 \rightarrow 8 \rightarrow 5 \rightarrow 6 \rightarrow 7$ . Here, the arrow mark "→" denotes the priority order in connector's precedence sequence. For example,  $1 \rightarrow 4$  means that the priority order of  $C_1$  is higher than that of  $C_4$  in precedence sequence.

The main reason to take engineering information similarity between connectors into consideration is that if the similar connectors are connected together, the number to change the assembly tool and assembly direction will reduce, thus saving assembly time and cost (Tseng et al., 2004). Therefore, the design objective function (OF) in AIS is based upon the similarity matrix of connectors. Then the sum of engineering information similarity

in an antibody can be calculated from formula (2):

$$OF = \sum_1^m SS_{h,h+1} \tag{2}$$

where  $SS_{h,h+1}$  is the engineering information similarity between the  $h$ th and  $h+1$ th connectors in the antibody,  $h = 1,2,3,\dots,m$  and  $m$  is the the number of connectors.

In the case of the antibody in the stapler example (Fig. 5), the value of the objective function of the antibody,  $OF = SS_{3,2} + SS_{2,0} + SS_{0,1} + SS_{1,4} + SS_{4,8} + SS_{8,5} + SS_{5,6} + SS_{6,7} = 2$ , which can be calculated by Formula (3) and the similarity matrix SS in Fig. 4.

4. Artificial immune systems for assembly sequence problems

4.1. Framework of artificial immune systems for ASP

Fig. 6 shows the AIS framework for ASP. Detailed steps are listed and described as follows:

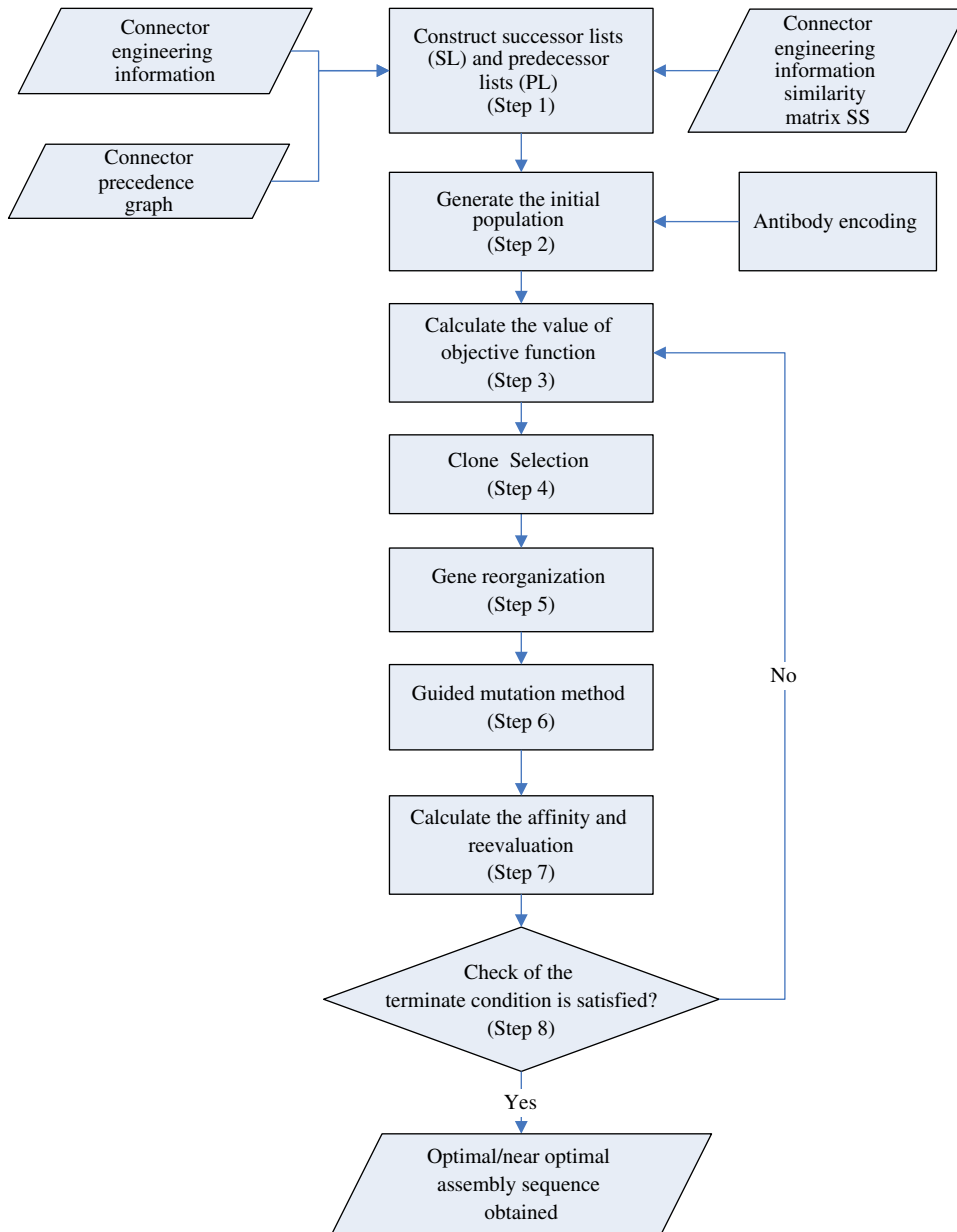


Fig. 6. The AIS framework for ASP.

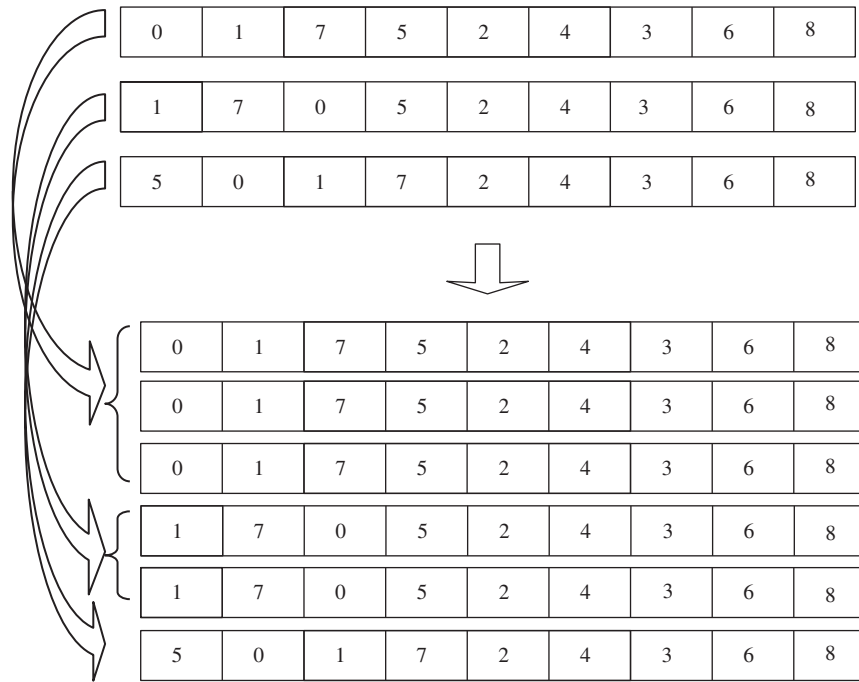


Fig. 7. Procedure of clonal selection.

- Step 1: Construct the SL and PL (refer to Section 3.2): The engineering information of connectors (Section 3.1), data in the precedence graph and engineering information similarity matrix  $SS$  (Section 3.3) are entered for the construction of SL and PL.
- Step 2: Generate the initial population from the concept of adjacency list (see Section 4.2): In the study, SL and PL are used to generate the initial solution and place it in the antibody memory zone.
- Step 3: Calculate the value of the objective function: Calculate the value of the objective function of each antibody and antigen (Formula (2)), which will serve as a standard for the selection of antibodies.
- Step 4: Clonal selection (see Section 2): From the ranking order of the objective function values of antibodies, select the best  $n$  antibodies to proliferate by cloning.
- Step 5: Gene reorganization (see Section 4.3): The guided crossover method is applied to reorganize the genes.
- Step 6: Guided mutation method (see Section 4.4).
- Step 7: Calculate the affinity and redo the selection (see Section 4.5): Evaluate the new population according to Formula (2) and add the better ones to the memory zone. Wipe out the worse antibodies in the original memory zone. Then delete the antibodies that are too similar to each other to maintain the variety of antibodies in the memory zone.
- Step 8: Check if the termination condition is satisfied: The maximum generation of evolution is used as a criterion to stop the algorithm. If the maximum generation is not reached, repeat Step 3; otherwise, find the optimal/near-optimal assembly sequence.

4.2. Apply the adjacency list to generate initial population (Step 2)

On the base of the satisfaction of the precedence relationship, the assembly sequence  $SEQ$  can be represented by an ordered list of connector, in which  $SEQ_i$  indicates the  $i$ th solution in the memory zone. If  $H(i)$  is a set of the candidate notes ( $i$  is the

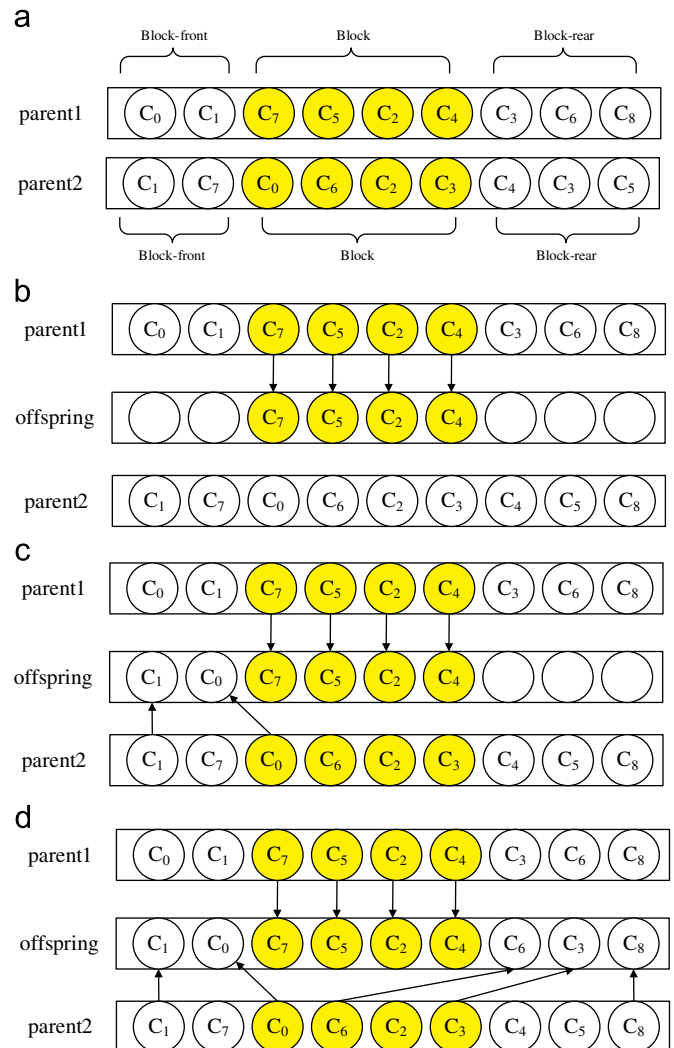


Fig. 8. Procedure of guided crossover method.



cursor), then the cursor  $i$  will point to the last connector in the set  $SEQ$ . With SL and PL lists, connectors can be placed in  $H(i)$ . Step by step, a complete  $SEQ$  will be found for the initial solution. The algorithm to generate the initial solution is listed below:

- Step 2.1: Because the set of the initial candidate connectors is  $\psi$ , and  $i$  points to the initial virtual node  $S$ , do Step 2.2.
- Step 2.2: Because  $S$  possesses Outdegree only, the SL list is used to find node  $C_i$  after  $H(S)$  and add it to  $H(i)$ .
- Step 2.3: If the current cursor points to  $S$ , then directly update  $H(S) = \{C_i\}$ . If the cursor points to  $C_i$ , find node  $C_j$  that has the biggest similarity with  $C_i$  from Formula (1) and place the cursor to pointing to  $C_j$ .
- Step 2.4: Because node  $C_i$  has been found from SL, nodes after  $C_i$  will then be found according to the Outdegree data. If Outdegree = 0, go to Step 2.6. If Outdegree  $\geq 1$ , go to Step 2.5.
- Step 2.5: Use PL to find the nodes before  $C_i$  according to the Indegree data. If Indegree = 1, then add the found node to  $H(C_i)$ . If Indegree  $> 1$ , temporarily save the found nodes in  $H(C_i)$ . Then check whether these found nodes belong to the initial candidate connector set.
- Step 2.6: End the algorithms and find an initial feasible solution.

Using the stapler in Fig. 2 as an example, the procedure to find antibodies for the initial solution is listed below:

- Step 2.1: Because  $SEQ_1 = \{\phi\}$ , and from Fig. 3(a), the cursor  $i$  is currently pointing to virtual node  $S$ ,  $H = \{S\}$ . Furthermore, because Outdegree  $> 1$ ,  $SEQ_1 = \{S\}$ . Do Step 2.2.

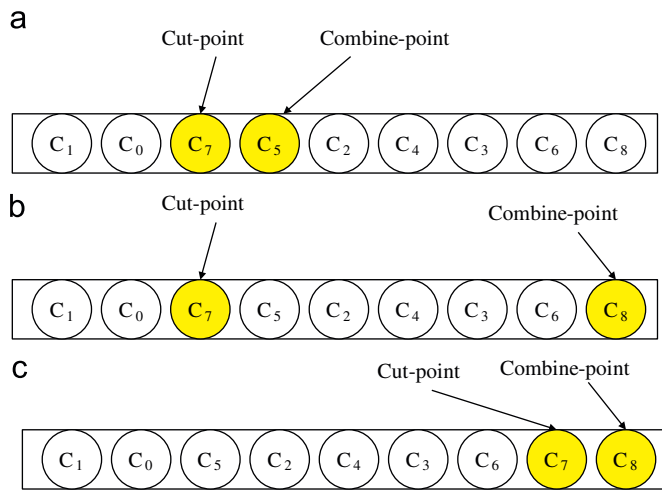


Fig. 9. Operation of guided mutation: (a) cut-point and combine-point, (b) adjusted combine-point and (c) adjusted cut-point.

- Step 2.2: From SL, find nodes before  $S$  in Fig. 3(a) and add them to  $H(S)$ . These nodes before  $S$  include  $C_0, C_1, C_2, C_5, C_6, C_7$ ; therefore,  $H(S) = \{C_0, C_1, C_2, C_5, C_6, C_7\}$ .
- Step 2.3: Because  $S$  is a virtual node, from the candidate initial connector set  $H(S)$ , randomly find node  $C_0$  and place the cursor  $i$  pointing to  $C_0$ . Currently,  $H(C_0) = \{C_1, C_2, C_5, C_6, C_7\}$ , and  $SEQ_1 = \{S \rightarrow C_0\}$ .
- Step 2.4: From SL in Fig. 3(a), we know that only  $C_8$  can be found after  $C_0$ . Now, Outdegree = 1, go to Step 2.5.
- Step 2.5: From PL in Fig. 3(b), we know that  $\{C_0, C_1, C_3, C_4, C_5, C_6, C_7\}$  is positioned before  $C_8$ . Moreover, the Indegree = 7, and neither  $C_3$  nor  $C_4$  belongs to  $H(C_0) = \{C_1, C_2, C_5, C_6, C_7\}$ . Therefore,  $C_8$  may not be added to  $H(C_0)$ . So,  $H(C_0) = \{C_1, C_2, C_5, C_6, C_7\}$ .
- Step 2.3: From connector  $S$ ,  $H(C_0) = \{C_1, C_2, C_5, C_6, C_7\}$ , and the connector engineering information similarity matrix  $SS$ , we find  $\{C_1, C_7\}$  have the biggest similarity with  $C_0$ .  $C_1$  is randomly chosen for next candidate node, cursor  $i$  pointing to node  $C_1$ . Now, we have  $H(C_1) = \{C_2, C_5, C_6, C_7\}$ , and  $SEQ_1 = \{S \rightarrow C_0 \rightarrow C_1\}$ .
- Step 2.4: From SL in Fig. 3(a), we can find only node  $C_8$  is positioned after  $C_1$ . Because current Outdegree = 1, do Step 2.5.
- Step 2.5: From PL in Fig. 3(b), we know that  $\{C_1, C_3, C_4, C_5, C_6, C_7\}$  are positioned before  $C_8$ . Now, Indegree = 6, and neither  $C_3$  nor  $C_4$  belongs to the first initial connector set. Therefore,  $C_8$  may not be added to  $H(C_1)$ .  $H(C_1) = \{C_2, C_5, C_6, C_7\}$ .
- Step 2.3: From the candidate initial connector set  $H(C_1) = \{C_2, C_5, C_6, C_7\}$  and connector engineering similarity matrix  $SS$ , we can find that  $C_7$  possesses the biggest similarity with  $C_1$ . Therefore, with cursor  $i$  pointing to node  $C_7$ ,  $C_7$  is chosen for next position. Now we have  $H(C_7) = \{C_2, C_5, C_6\}$ , and  $SEQ_1 = \{S \rightarrow C_0 \rightarrow C_1 \rightarrow C_7\}$ .

Repeat Steps 2.1–2.6, we can obtain a complete initial antibody with  $SEQ_1 = \{S \rightarrow C_0 \rightarrow C_1 \rightarrow C_7 \rightarrow C_5 \rightarrow C_2 \rightarrow C_4 \rightarrow C_3 \rightarrow C_6 \rightarrow C_8 \rightarrow F\}$ .

### 4.3. Gene reorganization (Step 5)

In this study, a guided crossover mechanism is used to reorganize genes (Tseng, 2006). The crossover procedure is listed below:

- Step 5.1: Randomly select two parent antibodies, *parent1* and *parent2*.
- Step 5.2: From these two parent antibodies, randomly generate their crossover interval and divide the antibodies into three sections: *block-start*, *block* and *block-rear*.
- Step 5.3: From one of the parent antibodies, reproduce the gene codes in the block interval to the offspring generation.

The best antibody	0	1	7	5	2	4	3	6	8	100%
Second best antibody 1	1	7	0	5	2	4	3	6	8	Number of the same individual $k = 6$ 66.7%
Second best antibody 2	5	0	1	7	2	4	3	6	8	Number of the same individual $k = 5$ 55.6%

Fig. 10. Affinity selection.

Step 5.4: According to the precedence relationships of the connectors in antibody *parent2*, duplicate the connectors in the *block-front* area of antibody *parent1* and place them

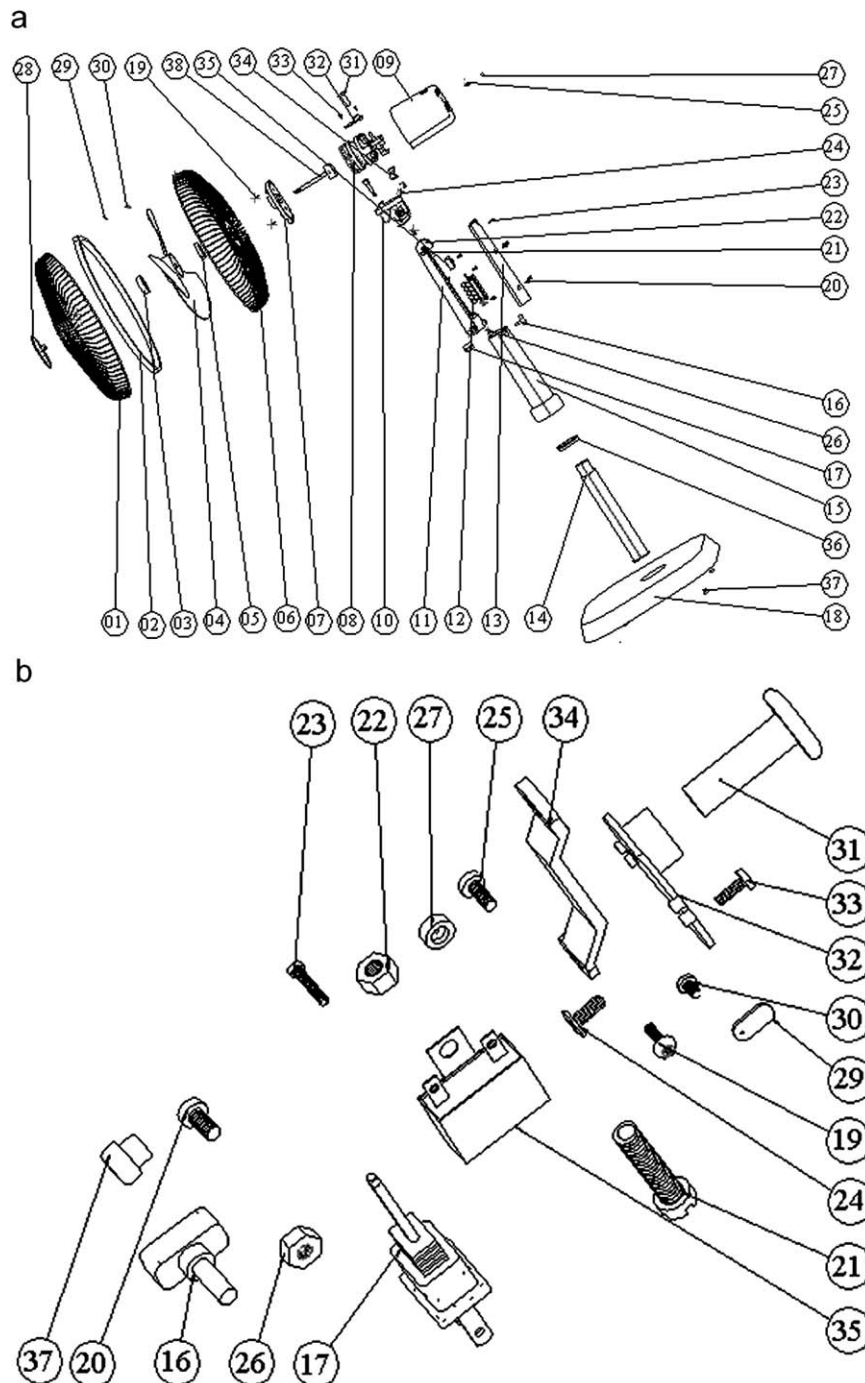
**Table 4**

Comparison between three algorithms for stapler.

Method	Average time	Average objective value	Max objective value
Guided-GAs	1.572	5.495	5.667
Memetic algorithms	2.494	5.667	5.667
AIS	2.453	5.667	5.667

on the corresponding locations of antibody *offspring1*. In the same way, duplicate the connectors in the *block-front* area of antibody *parent2* and place them on the corresponding locations of antibody *offspring2*, according to the precedence relationships of the connectors in antibody *parent1*.

Step 5.5: According to the precedence relationships of the connectors in antibody *parent2*, duplicate the connectors in the *block-rear* area of antibody *parent1* and place them on the corresponding locations of antibody *offspring1*. Similarly, duplicate the connectors in the *block-rear* area of antibody *parent2* and place them on the corresponding locations of antibody *offspring2*, according to the



**Fig. 11.** Illustration of electric fan parts: (a) exploded drawing and (b) enlarged scale drawing for small parts in (a).

precedence relationships of the connectors in antibody parent1.

of premature convergence. In this study, the guided mutation method is adopted. Such a mutation procedure is listed below:

4.4. Guided mutation method (Step 6)

The adding of mutation mechanism in AIS for ASP helps with the variety of the antibody, which can prevent the problem

- Step 6.1: Randomly select an antibody from the initial population.
- Step 6.2: Randomly generate a cut-point, and set combine-point = cut-point+1.

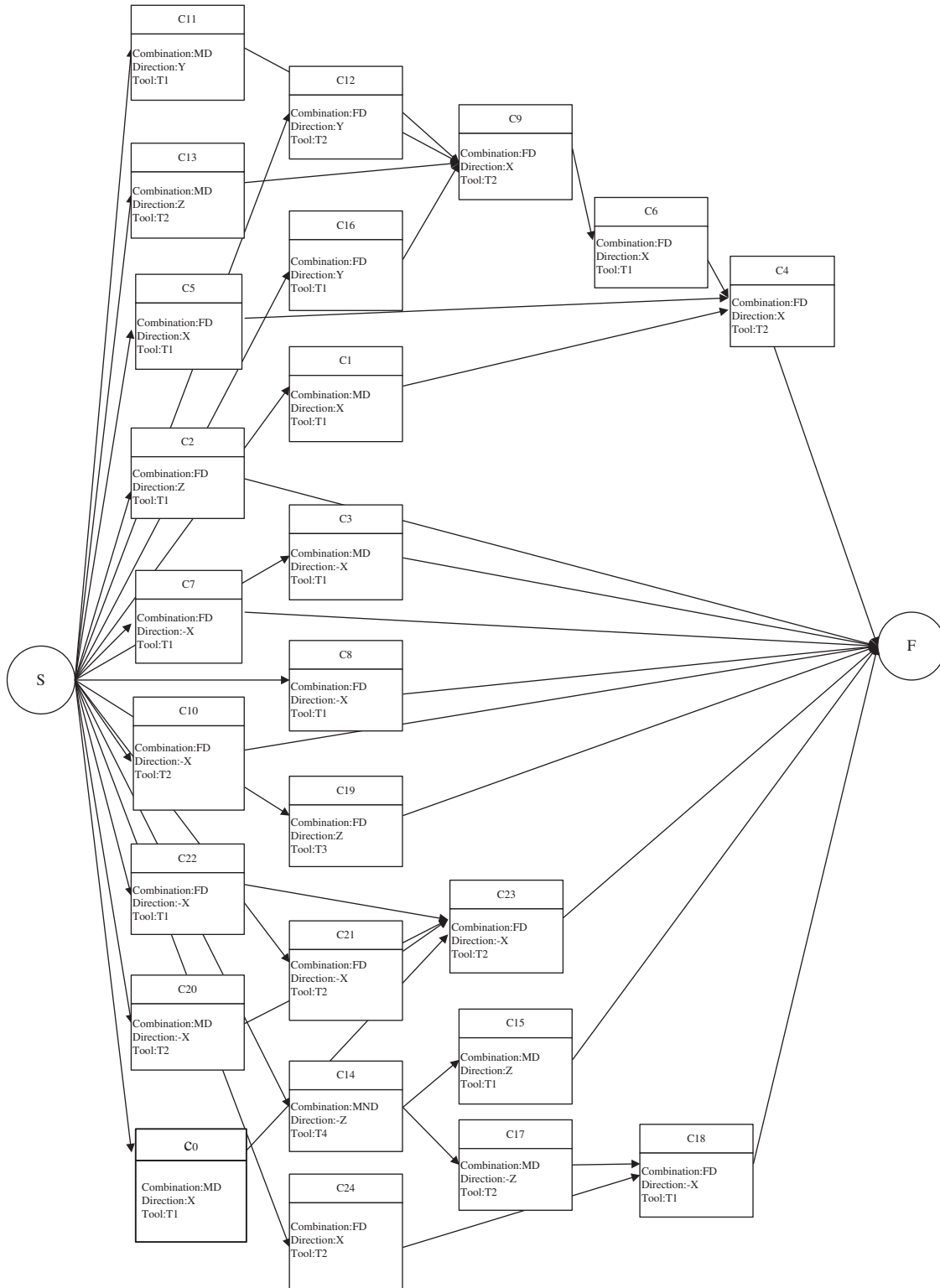


Fig. 12. Precedence graph of connectors for electric fan.

Step 6.3: Check the assembly precedence relationship between the connector on the *cut-point* and that on the *combine-point* connector.

(a) If the connector on the *cut-point* has a higher priority of assembly sequence than that on the *combine-point*, then execute Step 6.4.

(b) If the connector on the *cut-point* does not have a higher priority of assembly sequence than that on the *combine-point*, then add 1 to the *combine-point*, and execute Step 6.3 again.

Step 6.4: Insert the connector on the *cut-point* to the location of *combine-point-1*.

Step 6.5: Repeat the procedure from Step 6.1 to Step 6.5  $m/3$  times.  $m$  is the number of connectors.

4.5. Affinity maturation (Step 7)

The calculation of affinity function is mainly based on the similarity between antibodies. Greater values of affinity mean higher similarities between antibodies. To maintain the variety of antibodies in the memory zone, some antibodies that are too similar to one another will be deleted. In this study, the antibody whose objective value is best in the memory zone in each generation is considered as the base for other antibodies in terms of the gene sequential positions for the calculation of similarity between antibodies. All the other antibodies will be compared with the base antibody one by one for counting the number of different chromosome position. Then, the affinity function is used to transform the similarity data into percentage from which a threshold value is set for further selection. The calculation of affinity (Formula (3)) between antibodies is described below:

$$\text{Affinity value} = \frac{k}{m} \tag{3}$$

where  $k$  is the number of gene codes the same as those in the best antibody and  $m$  the number of connectors.

Step 7.1: From memory zone, calculate the antibody whose value of objective function is the best and select it for the best antibody.

Step 7.2: Contrast and mark the gene codes of the second best antibodies in the memory zone. Add 1 to  $k$  if the gene codes are the same and add 0 to  $k$  if the gene codes are totally different.

Step 7.3: Calculate  $k$  for every antibody whose gene codes are the same and select them according to affinity function in Formula (3).

Step 7.4: According to the threshold value, select the proper antibodies.

4.6. An illustrated example

The stapler example in Fig. 2 is used as an example to illustrate the algorithms in this section:

Step 1: Enter the connector engineering information, the precedence graph and engineering information similarity data for the construction of SL and PL (Fig. 3).

Step 2: Construct the initial solution data from SL and PL. As mentioned in Section 4.3, the current memory zone is set to 3. Then three initial solutions can be generated:

$SEQ_1 = \{S \rightarrow C_0 \rightarrow C_1 \rightarrow C_7 \rightarrow C_5 \rightarrow C_2 \rightarrow C_4 \rightarrow C_3 \rightarrow C_6 \rightarrow C_8 \rightarrow F\}$

$SEQ_2 = \{S \rightarrow C_1 \rightarrow C_7 \rightarrow C_0 \rightarrow C_5 \rightarrow C_2 \rightarrow C_4 \rightarrow C_3 \rightarrow C_6 \rightarrow C_8 \rightarrow F\}$

$SEQ_3 = \{S \rightarrow C_5 \rightarrow C_0 \rightarrow C_1 \rightarrow C_7 \rightarrow C_2 \rightarrow C_4 \rightarrow C_3 \rightarrow C_6 \rightarrow C_8 \rightarrow F\}$

The same way can be applied to other initial antibodies.

Step 3: Calculate the values of objective function: Through engineering similarity matrix  $SS$  and Formula (2), the values of objective function for all antibodies can be calculated. For example,  $SEQ_1 = \{S \rightarrow C_0 \rightarrow C_1 \rightarrow C_7 \rightarrow C_5 \rightarrow C_2 \rightarrow C_4 \rightarrow C_3 \rightarrow C_6 \rightarrow C_8 \rightarrow F\}$  and its objective value =  $SS_{0,1} + SS_{1,7} + SS_{7,5} + SS_{5,2} + SS_{2,4} + SS_{4,3} + SS_{3,6} + SS_{6,8} = 4.33$ .

Table 5 Comparison between three algorithms for electric fan.

Method	Average time	Average objective value	Max objective value
Guided-GAs	2.808	16.133	16.667
Memetic Algorithms	3.951	18.285	18.333
AIS	3.045	18.365	18.667

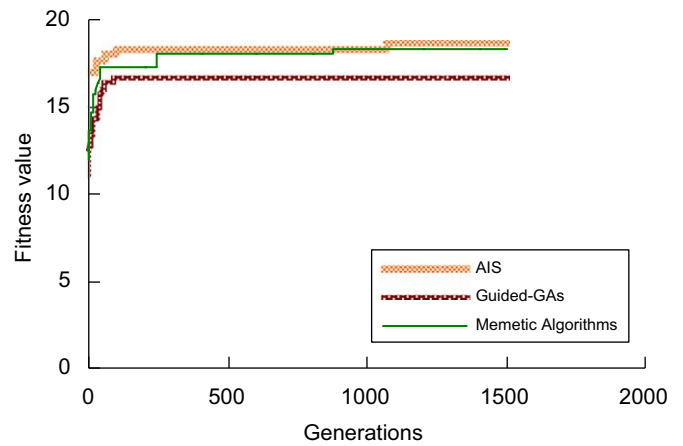


Fig. 13. Convergence plot of the electric fan.

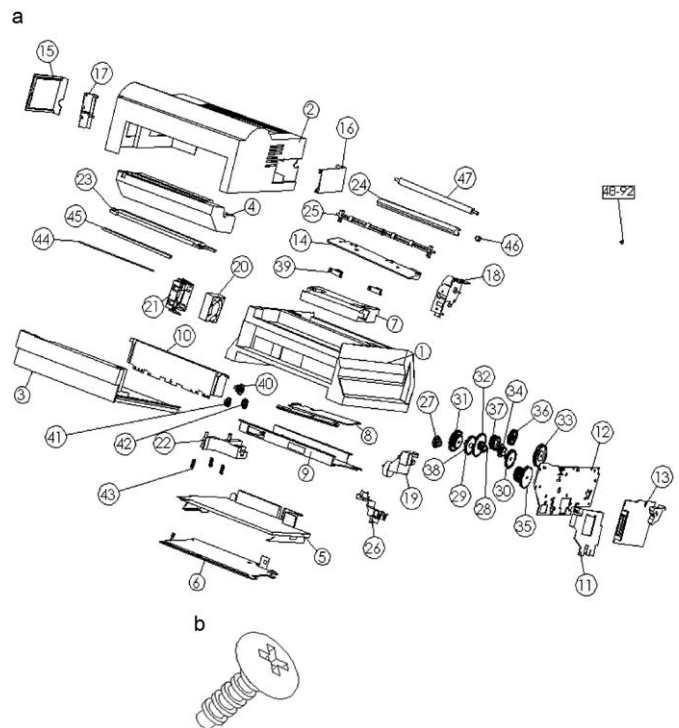


Fig. 14. Illustrations of printer parts: (a) exploded drawing and (b) enlarged scale drawing of screw parts in (a). Note: Parts numbered 48–92 are screws.

Step 4: Clonal selection: According to the objective values of antibodies, select the best  $n$  antibodies and clone them into a new population. In this case,  $n = 3$ . As can be seen in Fig. 7,

these better antibodies are  $SEQ_1$ ,  $SEQ_2$ , and  $SEQ_3$ . Then step by step, clone the first antibody into 3 antibodies; the second into 2; the third into 1 and save them in the memory zone.

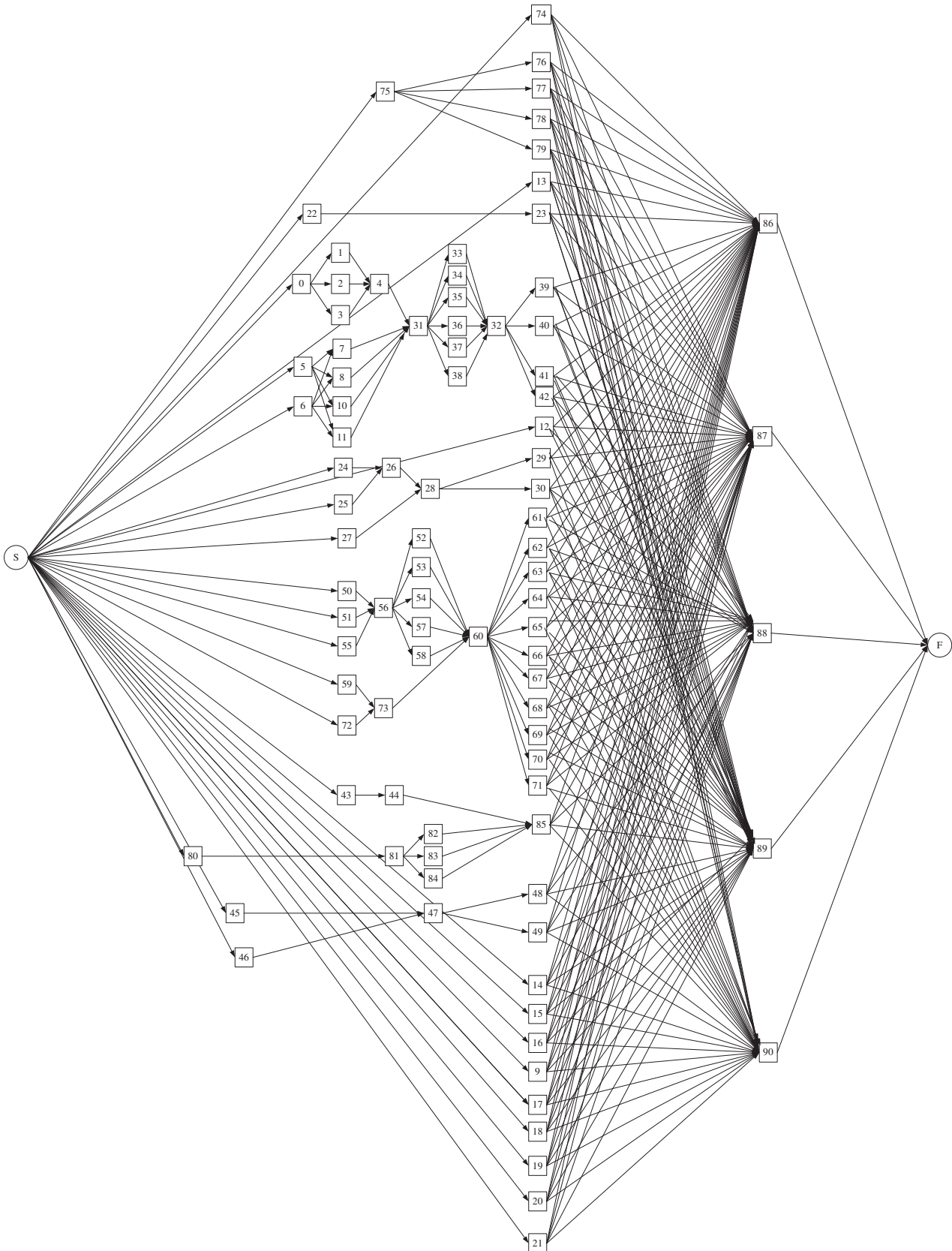


Fig. 15. Precedence graph of connectors for laser printer.

**Step 5: Guided crossover method**

**Step 5.1:** Randomly select two parent antibodies, *parent1* and *parent2* (Fig. 8(a)).

**Step 5.2:** From *parent1* and *parent2*, randomly generate their crossover interval and divide the antibodies into three sections: *block-start*, *block* and *block-rear*. (Block section is the yellow area in Fig. 8(a))

**Step 5.3:** Reproduce the gene codes in the block interval from *parent1* to the offspring generation. (Fig. 8(b)).

**Step 5.4:** According to the precedence relationships of the connectors in antibody *parent2*,  $1 \rightarrow 0$ , duplicate connector  $C_0, C_1$ , in the *block-front* area of antibody *parent1* and place them on the corresponding locations of antibody *offspring1*. This is shown in Fig. 8(c).

**Step 5.5:** Duplicate connectors  $C_3, C_6, C_8$  in the *block-rear* area of antibody *parent1* and place them on the corresponding locations of antibody *offspring1* according to the precedence relationships of the connectors in antibody *parent2*,  $C_6 \rightarrow C_3 \rightarrow C_8$ , as shown in Fig. 8(d).

**Step 6: Guided mutation method**

**Step 6.1:** Randomly select an antibody (Fig. 9).

**Step 6.2:** Randomly generate a *cut-point* = 3, and set *combine-point* =  $3+1 = 4$ .

**Step 6.3:** Check the assembly precedence relationship between the connector on the *cut-point* and that on the *combine-point* connector. Because the connector on the *cut-point* does not have a higher priority of assembly sequence than that on the *combine-point*, set *combine-point*+1 = 5, and execute Step 6.3 again. In the same way, it can be specified *combine-point* = 9, as shown in Fig. 9(b).

**Step 6.4:** Insert connector  $C_2$  on the *cut-point* to the location of *combine-point*-1, as shown in Fig. 9(c).

**Step 6.5:** Repeat the procedure from Step 4.1 to Step 4.4 three times and stop the mutation mechanism.

**Step 7: Calculate the affinity and redo the selection.**

**Step 7.1:** Fig. 10 shows the ranking order for the objective values of the antibodies in the memory zone. Therefore, the first antibody  $C_0 \rightarrow C_1 \rightarrow C_7 \rightarrow C_5 \rightarrow C_2 \rightarrow C_4 \rightarrow C_3 \rightarrow C_6 \rightarrow C_8$  is currently the best antibody.

**Step 7.2:** Contrast the gene codes of other second best antibodies with those in the best antibody. We can get  $k = 6$  for the second antibody and  $k = 5$  for the third antibody.

**Step 7.3:** Calculate the affinity values from Formula (3) and we get 66.7% for the second antibody and 55.6% for the third antibody.

**Step 7.4:** If the threshold value of affinity is set to 60%, the second antibody is deleted because it is too similar to the first antibody. A new population will be obtained.

**Step 8:** Satisfy stopping criteria and stop the algorithms.

**5. Practical examples**

Borland C# Builder 6.0 is used in this study for AIS-related programming. Under the computer specification of an Intel Celeron CPU 2.67 GHz CPU and 512 Mb memory, a stapler, an electric fan and a laser printer are used as practical examples to compare the results of AIS algorithm with those of Guided-GAs (Tseng, 2006) and MAs (Tseng et al., 2007).

**5.1. The stapler example**

In the stapler example, both Guided-GAs and MAs reach the objective value 5.667. Under the conditions of the crossover rate 70%, mutation rate 30%, parent population size 21, maximum generation 1500, cloning number 3, threshold value of affinity 80%, 10 experiments were conducted for AIS, Guided-GAs and MAs. Table 4 shows the average quality and time in solution searching, the maximum objective values of AIS, Guided-GAs and MAs.

Because the stapler is an easier example, all of three algorithms can find the optimal assembly sequence of stapler in a very short period of time. Overall, the quality for solution searching in AIS and MAs is a little bit better than that of Guided-GAs (Table 4).

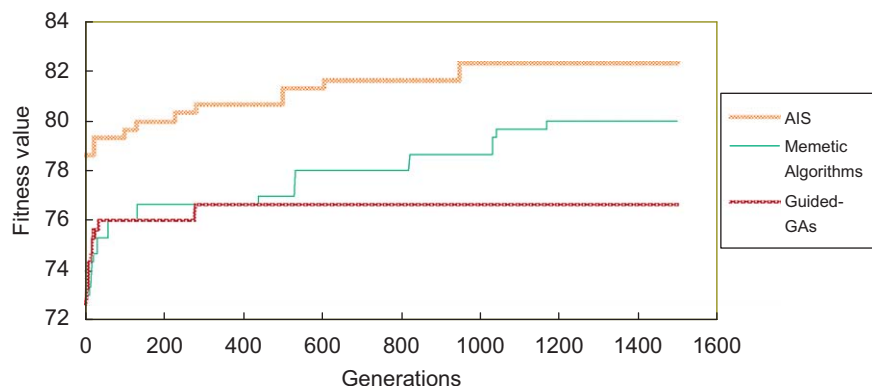
**5.2. The electric fan example**

The electric fan is composed of 40 parts (Fig. 11). Through the rules of connector, these parts can be divided into 25 connectors. Fig. 12 shows the precedence graph of the connectors in the electric fan. The execution of AIS is under the conditions of the crossover rate 70%, mutation rate 30%, parent population size 51, maximum generation 1500, cloning number 10, and threshold value of affinity 75%.

Table 5 lists the results of 10 tests of three algorithms. In terms of the average solution searching time, AIS is 20% better than MAs. For maximum objective value, AIS is 12% better than Guided-GAs and about 2% better than MAs. Generally speaking, AIS and MAs are better than Guided-GAs in the quality of solution searching. In

**Table 6**  
Comparison between three algorithms for the laser printer.

Method	Average time	Average objective value	Max objective value
Guided-GAs	18.543	75.595	76.67
Memetic algorithms	25.965	79.096	80
AIS	19.067	81.432	82.33



**Fig. 16.** Convergence plot of the laser printer.

solution searching time, AIS is better than MAs. Moreover, it can be seen from Fig. 13 that Guided-GAs converges in less than 200 generations. However, both AIS and MAs have the ability to escape from local optimization; AIS runs away from 1100 generations and MAs get free about 850 generations. For AIS, the algorithms of Affinity maturation play the role of escaping mechanism. The experiment demonstrates that both AIS and MAs are able to keep away from early convergence and jump from the local optimal solutions.

### 5.3. The laser printer example

The laser printer is made up of 92 parts (Fig. 14) and 91 connectors. The precedence graph of the connectors in the laser printer is shown in Fig. 15. The execution of AIS is under the conditions of the crossover rate 70%, mutation rate 30%, parent population size 51, maximum generation 1500, cloning number 15, and threshold value of affinity 85%.

The results of 10 trials of three genetic algorithms are listed in Table 6. In terms of the average solution searching time, AIS is 26.6% better than MAs. For maximum objective value, AIS is 7.4% better than Guided-GAs and about 2.9% better than MAs. Generally speaking, AIS and MAs are approximately the same in the quality of solution searching. But, in solution searching time, AIS is better than MAs. Moreover, Fig. 16 clearly shows that Guided-GAs converges in about 230 generations. However, both AIS and MAs have the ability to evade from local optima; AIS gets away from 1000 generations and MAs break out in 1120 generations. This demonstrates that both AIS and MAs are able to keep away from early convergence and generate better solutions.

## 6. Conclusion

In this study, an AIS algorithm was mainly proposed to solve the assembly planning problem. Moreover, its performance was compared with those of Guided-GAs and MAs (Tseng, 2006; Tseng et al., 2007). Such an AIS algorithm has two advantages. First of all, it uses the adjacency list to generate proper initial solutions. Then, the crossover and mutation mechanisms in genetic algorithms are used to upgrade the quality in solution searching. To avoid the impairment of the quality in searching solutions, common immune algorithms often eliminate the crossover mechanism. It has been demonstrated in this study that in ASP problems, a better initial solution will have more effectiveness in the quality than traditional immune algorithms. Secondly, the proposed AIS algorithm is different from traditional GAs in that it makes use of the variety characteristic of artificial immune systems, keeping from falling in the difficulty of local optimal.

Compared with past genetic algorithms, AIS is about 20% better than Guided-GAs in solution searching time and about 12% better than Guided-GAs and about 2% better than MAs in maximum objective value, taking the electric fan as an example. In the case of the laser printer, AIS is 26.6% better than MAs in average solution searching time and about 7.4% and 2.9% better than Guided-GAs and MAs, respectively, in maximum values. This demonstrates that AIS can offer effective solutions to ASP problems with more constraints, longer time for solution searching, or of the local optimal dilemma. Overall, AIS can be used to help solve the difficulty of falling in local optimization in ASP problems in Guided-GAs. Compared with MAs, AIS is about the same in quality of solution searching. But the solution searching time can be effectively shortened in AIS. However, the upper limit of the number of connectors in this study is 91 (i.e. the laser printer).

For future work suggestions, the integrations of assembly modeling and assembly sequence generator related to CAD will be helpful to the implementation of computer-aided automatic assembly planning. In addition, the parameters of AIS algorithm are obtained through constant trials and experiments. The optimal combination of parameters will differ with cases. Therefore, the dynamic adjustment of control of parameters is a direction worth of further investigation.

## References

- Akagi, F., Osaki, H., Kikuci, S., 1980. The method of analysis of assembly work based on the fastener method. *Bulletin of the JSME* 23 (184), 1670–1675.
- Alisantoso, D., Khoo, L.P., Jiang, P.Y., 2003. An immune algorithm approach to the scheduling of a flexible PCB flow shop. *The International Journal of Advanced Manufacturing Technology* 22 (11–12), 819–827.
- Baldwin, D.F., Abel, T.E., Lui, M.C.M., De Fazio, T.L., Whitney, D.E., 1991. An integrated computer aids for generating and evaluating assembly sequences for mechanical products. *IEEE Transaction on Robotics and Automation* 7 (1), 78–94.
- Cao, P.B., Xiao, R.B., 2007. Assembly planning using a novel immune approach. *The International Journal of Advanced Manufacturing Technology* 31 (7–8), 770–782.
- Chen, T.C., You, P.S., 2005. Immune algorithms-based approach for redundant reliability problems with multiple component choices. *Computers in Industry* 56, 195–205.
- Chung, C., Peng, Q., 2006. Evolutionary sequence planning for selective disassembly in de-manufacturing. *International Journal of Computer Integrated Manufacturing* 19 (3), 278–286.
- De Castro, L.N., Timmis, J.I., 2003. Artificial immune systems as a novel soft computing paradigm. *Soft Computing* 7, 526–544.
- De Castro, L.N., Von Zuben, F.J., 2002. Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation* 6 (3), 239–251.
- De Fazio, T.L., Whitney, D.E., 1987. Simplified generation of all mechanical assembly sequence. *IEEE Journal of Robotics and Automations* 3 (6), 640–658.
- Eimaraghy, H.A., 2006. Flexible and reconfigurable manufacturing systems paradigms. *International Journal of Flexible Manufacturing System* 17, 261–276.
- Fujimoto, H., Sebaaly, M.F., 2000. A new sequence evolution approach to assembly planning. *Transaction of the ASME Journal of Manufacturing Science and Engineering* 122, 198–205.
- Gottipolu, R.B., Ghosh, K., 1997. Representation and selection of assembly sequences in computer-aided assembly process planning. *International Journal of Production Research* 35 (12), 3447–3465.
- Göngör, A., Gupta, S.M., 1997. An evaluation methodology for disassembly processes. *Computer & Industrial engineering* 33 (1–2), 329–332.
- Homem De Mello, L.S., Sanderson, A.C., 1991. A correct and complete algorithm for the generation of mechanical assembly sequences. *IEEE Transaction on Robotics and Automation* 7 (2), 228–240.
- Hong, D.S., Cho, H.S., 1997. Generation of robotic assembly sequences with consideration of line balancing using simulated annealing. *Robotica* 15, 663–673.
- Hong, D.S., Cho, H.S., 1995. A neural-network-based computational scheme for generating optimized robotic assembly sequence. *Engineering Application of Artificial Intelligence* 8, 129–145.
- Khilwani, N., Shankar, R., Tiwari, M.K., 2008. Facility layout problem: an approach based on a group decision-making system and psychoclonal algorithm. *International Journal of Production Research* 46 (4), 895–927.
- Kim, G.J., Lee, S., Bekey, G.A., 1996. Interleaving assembly planning and design. *IEEE Transaction on Robotics and Automation* 12, 246–251.
- Laperrière, L., Eimaraghy, H.A., 1996. GAPP: a generative assembly process planner. *Journal of Manufacturing Systems* 15, 282–293.
- Motavalli, D., Islam, A.U., 1997. Multi-criteria assembly sequence. *Computer & Industrial Engineering* 32, 743–751.
- Musilek, P., Lau, A., Reformat, M., Wyard-Scott, L., 2006. Immune programming. *Information Sciences* 176, 972–1002.
- Smith, G.C., Smith, S.S.F., 2003. Automated initial population generation for genetic assembly planning. *International Journal of Computer Integrated Manufacturing* 16, 219–228.
- Sonnenberg, M., Sodhi, R.S., 2003. Influence of fastener on disassembly products. *International Journal of Industrial Engineering* 10 (3), 271–278.
- Tseng, H.E., Li, R.K., 1999. A novel means of generating assembly sequences using the connector concept. *Journal of Intelligent Manufacturing* 10, 423–435.
- Tseng, H.E., 2006. Guided genetic algorithms for solving the larger constraint assembly problem. *International Journal of Production Research* 44 (3), 601–625.
- Tseng, H.E., Li, J.D., Chang, Y.H., 2004. Connector-based approach to assembly planning using genetic algorithms. *International Journal Production Research* 42, 2243–2261.
- Tseng, H.E., Wang, W.P., Shih, H.Y., 2007. Using memetic algorithms with guided local search to solve assembly sequence planning. *Expert Systems with Applications* 33, 451–467.