Int J Adv Manuf Technol (2011) 55:763–782
DOI 10.1007/s00170-010-3093-5

ORIGINAL ARTICLE

A knowledge-based engineering system for assembly sequence planning

Yung-Yuan Hsu · Pei-Hao Tai · Min-Wen Wang · Wen-Chin Chen

Received: 5 December 2009 / Accepted: 1 December 2010 / Published online: 30 December 2010

Abstract In this study, we developed a knowledge-based engineering (KBE) system to assist engineers in promptly predicting a near-optimal assembly sequence. A three-stage assembly optimization approach with some heuristic working rules was employed to establish the proposed system. In the first stage, Above Graph and a transforming rule were used to create a correct explosion graph of the assembly models. In the second stage, a three-level relational model graph, with geometric constraints and assembly precedence diagrams, was generated to create a completely relational model graph, an incidence matrix, and a feasible assembly sequence. In the third stage, a robust back-propagation neural network engine was developed and embedded in the Siemens NX system. System users can easily access the volume, weight, and feature number through the Siemens NX system interface, input the related parameters such as contact relationship number and total penalty value, and predict a feasible assembly sequence via a robust engine. Three real-world examples were used to evaluate the feasibility of the KBE system. The results show that the proposed system can facilitate feasible assembly sequences and allow designers to recognize contact relationships, assembly difficulties, and assembly constraints of three-dimensional components in a virtual environment type.

Keywords Knowledge-based engineering system · Assembly optimization · Assembly sequence · Back-propagation neural networks · BPNN

Y.-Y. Hsu
Department of Mechanical Engineering, Chung Hua University,
707 Wu Fu Rd., Sec. 2,
Hsinchu 30012, Taiwan

P.-H. Tai
Delta Electronics, Inc.,
252 Shangying Road,
Taoyuan County 33341, Taiwan

P.-H. Tai · W.-C. Chen (✉)
Graduate School of Industrial Management,
Chung Hua University,
707 Wu Fu Rd., Sec. 2,
Hsinchu 30012, Taiwan
e-mail: wenchin@chu.edu.tw

M.-W. Wang
Department of Mechanical Engineering,
National Kaohsiung University of Applied Sciences,
415 Chien Kung Road,
Kaohsiung 807, Taiwan

1 Introduction

Assembly sequence planning (ASP) is a critical technology which achieves product design and facilitates realization. Product design is an important determinant of a business' competitiveness, and 80% of the costs of a product's lifecycle occur during the initial design stage [1]. Design for assembly (DFA) focuses on product-related factors such as size, weight, symmetry, orientation, and form features, as well as other assembly processes: handling, gripping, and insertion [2, 3]. The accomplishment of those numerous standard sequences can be employed without regard to the product configuration, material, or production quantity [4, 5]. In generating assembly sequences, De Fazio and Whitney [6] adopted the exhaustive concept of Bourjault

[7] to obtain a complete set of assembly sequences. They generated sequences in two stages: creating precedence relations between liaisons (i.e., physical contacts) or logical combinations of liaisons in a product and then verifying the liaison sequence in terms of graph search theory. However, some components cannot be successfully assembled due to potential geometric constraints. Homen de Mello and Sanderson [8] made a representation of directed AND/OR graphs and disassembly concerns to create feasible assembly sequences. In addition, Kroll [9] used directed graph-based procedures with conventional representations to reduce the number of sorting operations required. However, these direct approaches only apply to orthogonal assembly structures involving six orthogonal directions.

Assembly planning is also regarded as "assembly by disassembling", i.e., an assembly sequence results from systematically disassembling the final product and reversing the disassembly sequence [10]. A critical aspect of almost all objects or devices that consist of more than one part is the nature of the fasteners used to join the parts together, where fastening methods can be applied that dictate the level of difficulty during assembly/disassembly processes [11]. However, using the backward disassembly search is not necessarily to come up with the best assembly sequence. Some other researchers employ contact-based features to represent the precedence relationships of the product. However, a contact-based precedence diagram cannot effectively express the complexity of the assigned assembly relations. A complete and effective assembly plan must include other graphs, such as explosion graphs, relational model graphs (RMG), incidence matrices, assembly precedence diagrams (APDs), etc. In reality, few experts or engineers know exactly how to derive a correct explosion graph, draw a complete relational model graph or incidence matrix among the components, or determine a complete APD to generate a near-optimal assembly sequence [12–14]. However, these exact methods guaranteed to exactly find the global optimum are slow and limited due to the combinatorial nature when product parts increase [15].

Assembly process planning is commonly a crucial stage in product design and manufacturing process which contains assembly sequence planning, assembly tool and fixture planning, assembly line layout, and assembly line balancing (ALB), which are highly related to assembly sequences and precedence relations [16, 17]. However, two kinds of potentially applicability problems exist: An optimal assembly sequence of a large-scale generated assembly sequences is hard to determine; there can be great computational and highly constrained combinatorial complexity derived from the geometric reasoning and topological features [18, 19]. Assembly line modeling problems have been conventionally classified into two types: type I and type II. In type I problems, the required production rate (cycle time), assembly tasks, tasks times, and precedence relations are given. Designing a new assembly line is generally such a problem, its objective being to minimize the number of workstations. In type II problems, the number of workstations or production employees is fixed. The aim is to minimize the cycle time. Type II problems often occur when a factory wants to produce the optimum number of items using a fixed number of workstations without adding new machines [20]. Moreover, Gao et al. [21, 22] proposed an intelligent production control decision support and a bi-level genetic algorithm (GA) to deal with operative efficiency change and scheduling problems; Wong et al. [23] used GA to optimize the assignment of operators in an assembly line. The novel optimal solutions of the ASP and ALB problems have typically been obtained by following meta-heuristic methods—guided genetic algorithms and memetic algorithms (MAs), combining the global search with local search of evolutionary algorithm to improve individual solutions, simulated annealing (SA), ant colony optimization (ACO), algorithm of self-guided ants (ASGA), and particle swarm optimization (PSO); most of them can escape the local optimum and further obtain the global optimum [24–30].

The current essential technology, knowledge-based engineering (KBE) integrating geometry, configuration, and engineering knowledge, allows engineers to secure product knowledge and incorporate engineering domain skills based on design rules and powerful CAD/CAM applications that are used to design, configure, and assemble products. Examples of this include the so-called expert systems, web-based knowledge-driven databases involving engineering know-how with a knowledge-based language such as knowledge fusion (KF), which captures both geometric and non-geometric attributes of a given part or assembly, writes rules and essentially gives us the ability to capture intelligence and engineering know-how, and becomes a critical part of the business strategy [31, 32].

Other studies used the Hopfield and back-propagation neural network (BPNN) to generate optimal or near-optimal assembly sequences [33–35]. However, they merely focus on mathematical model creation such as network energy function, binary vector representation to create the assembly states, and assembly sequences by a directed and connection graph. The proposed engineering-oriented KBE system is an integrated and systematic system which can generate more feasible and reliable assembly sequences by

embedding engineers or experts' know-how and a robust neural network-based predictor [36] and combine three-stage assembly optimization technique with some heuristic working rules to facilitate near-optimal assembly sequences and allow designers to recognize contact relationships, assembly difficulties, and assembly constraints of 3D components in a virtual environment.

The remainder of this paper is as follows: Section 2 provides the related methodologies of the proposed KBE system. Section 3 describes the working concepts and procedures through the complete flow chart and KBE system model. Section 4 denotes how to conduct the three-stage assembly sequence planning of the KBE system. Section 5 demonstrates the examples for developing a KBE system. Section 6 concludes the paper with the recommendations of future work.

2 Related methodologies

The proposed methodologies herein, including BPNNs, the Taguchi method, and the response surface methodology, are briefly introduced below.

2.1 Back-propagation neural networks

An artificial intelligence (AI) system, including knowledge-based and artificial neural network (ANN) systems, has

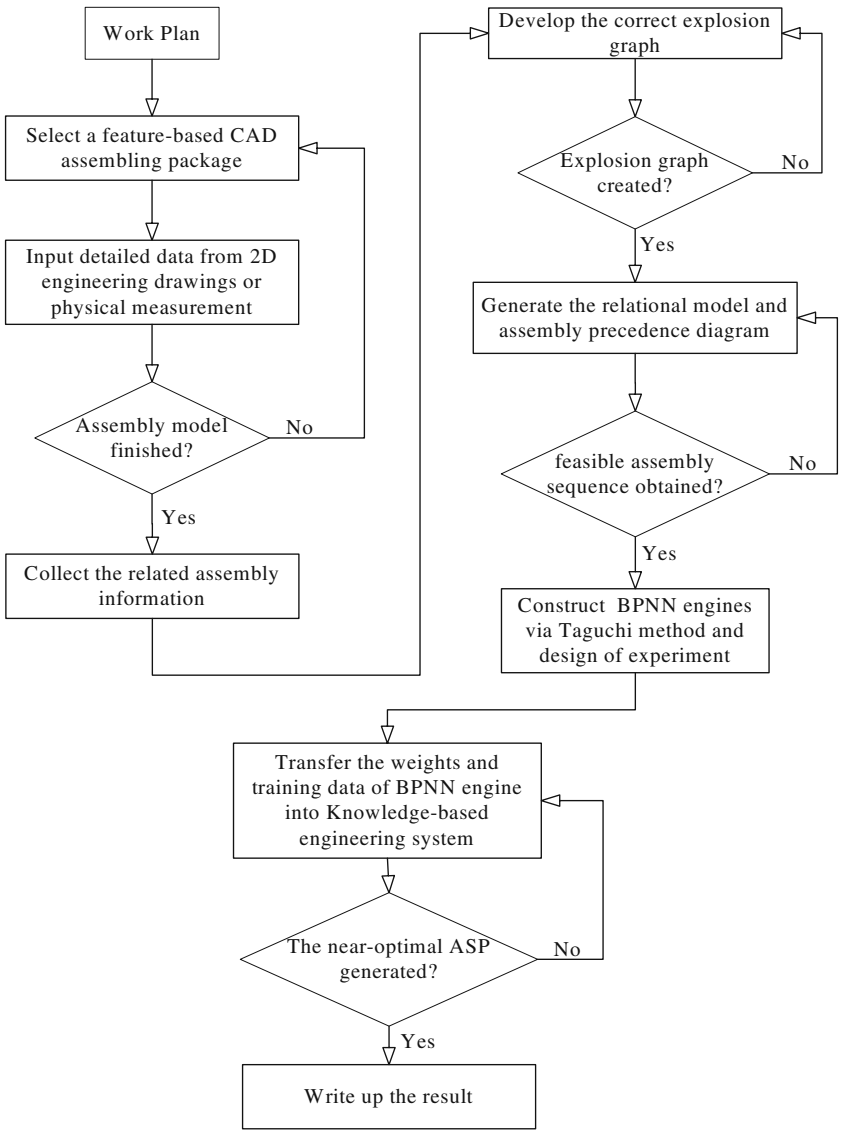


Fig. 1 Flow chart of the proposed study

three key components (representation, reasoning, and learning) and must be able to achieve three things (store knowledge, apply the knowledge stored to solve problems, and obtain new knowledge through experience) [37]. ANNs are widely applied in signal processing, pattern recognition, medical diagnosis, speech recognition, identification of geophysical features, and mortgage evaluation [38]. In much of the ANN literature, BPNNs were adopted because they have the advantages of a fast response and high learning accuracy [39–42]. The superiority of a network's functional approach depends on the network architecture and parameters, as well as a problem's complexity. If an inappropriate network architecture or parameter is selected, undesirable results may be obtained. BPNNs consist of an input layer, a hidden layer, and an output layer. Parameters for BPNNs include the number of hidden layers, number of hidden neurons, learning rate, momentum, etc. All of these parameters can significantly impact the performance of the neural network.

In this research, the steepest-descent method was used to find the weight change and minimize the error energy function. The activation function is a hyperbolic sigmoid function. According to past studies [43, 44], there are some important conditions for network learning termination: (1) when the root mean square error (RMSE) between the expected value and network output value is reduced to a preset value, (2) when the preset number of learning cycles is reached, and (3) when cross-validation takes place between training samples and test data. The first two methods are related to preset values. This research adopts the first and second approaches by gradually increasing the network training time which gradually decreases the RMSE until it is stable and acceptable. The RMSE is defined as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(d_i - y_i)^2}; \tag{1}$$

where N is the number of training samples, d_i is the actual value of training sample i, and y_i is the predicted value of the neural network of training sample i.

In network learning, input information and output results are used to adjust the weighting values of a network. The more-detailed the input training classification and the

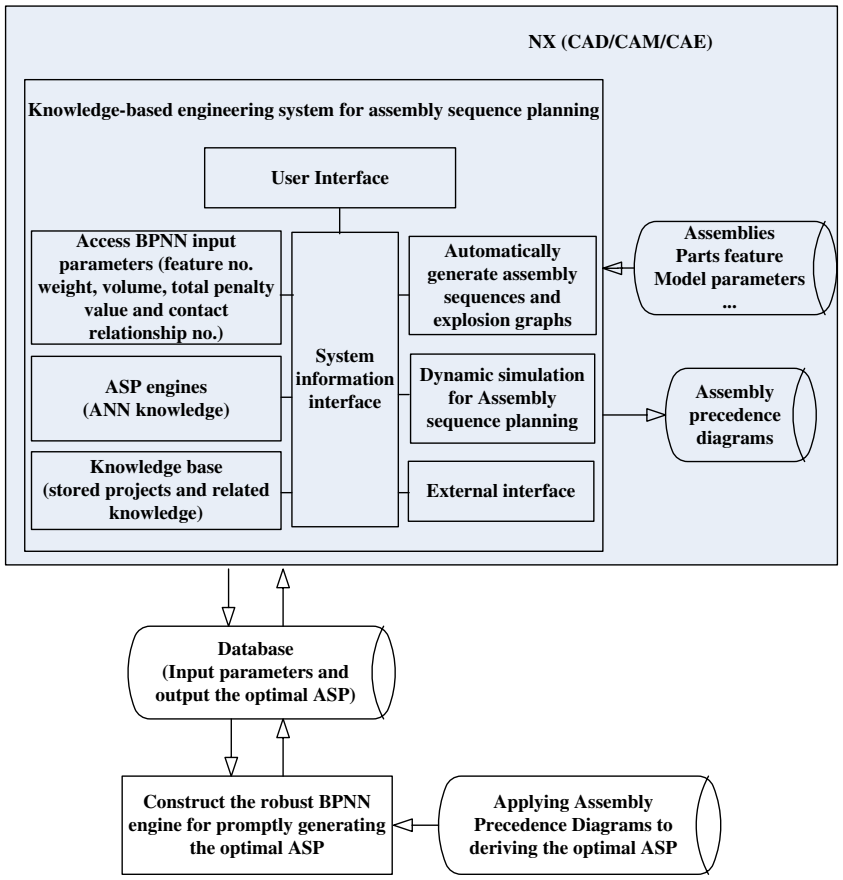


Fig. 2 The knowledge-based engineering system for assembly sequence planning

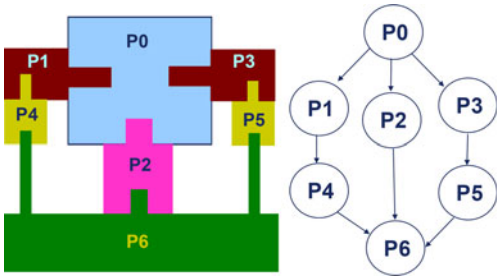


Fig. 3 An assembly precedence diagram tree used in assembly planning

greater the amount of learning information which is provided, the better the output will conform to the expected result. Since learning and verification data for BPNNs are limited by functional values, the data must be normalized by the following equation:

$$\mathrm{PN} = \frac{P - P_{\min}}{P_{\max} - P_{\min}} \times (D_{\max} - D_{\min}) + D_{\min}; \tag{2}$$

where PN is the normalized data, P is the original data, $P_{\max}$ is the maximum value of the original data, $P_{\min}$ is the minimum value of the original data, $D_{\max}$ is the expected maximum value of the normalized data, and $D_{\min}$ is the expected minimum value of the normalized data. When applying neural networking to a system, input and output values of the neural network fall in the range of [$D_{\min}$, $D_{\max}$]. When applying a neural network to a system, input and output values of the neural network fall in the range of [0.1, 0.9].

2.2 The Taguchi method

Taguchi's robust parameter design is a systematic method which normally selects an appropriate formulation of the signal/noise (S/N) ratio and calculates the S/N ratio for each treatment. There are three types of S/N ratios: nominal the best, the larger the better, and the smaller the better. Most engineers choose the highest S/N ratio treatment as the preliminary optimal initial process parameter setting. The Taguchi method has also been used to design parameters for neural networks in previous research. Santos and Ludermir [45] applied a factorial design to assist the design and implementation of a neural network. The formulae of the three types of S/N ratios are given as follows:

$$\text{nominal the best}: \qquad S/N = 10 \times \log\left(\frac{\overline{y}^2}{\overline{S}^2}\right) \tag{3}$$

the larger the better :

$$\begin{aligned} S/N &= -10 \times \log\left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{y_i^2}\right) \\ &\approx -10 \times \log\left(\frac{1}{\overline{y}^2} + \frac{3\overline{S}^2}{\overline{y}^4}\right) \end{aligned} \tag{4}$$

and the smaller the better :

$$\begin{aligned} S/N &= -10 \times \log\left(\frac{1}{n}\sum_{i=1}^{n} y_i^2\right) \\ &= -10 \times \log\left[\overline{y}^2 + \overline{S}^2\right] \end{aligned} \tag{5}$$

where y_i is the response value of a specific treatment under i replications, n is the number of replications, $\overline{y}$ is the average of all y_i values, and $\overline{S}$ is the standard deviation of all y_i values.

A crucial procedure using the Taguchi method can be conducted as follows: Using an analysis of variance of the S/N ratio, determine factors having a significant effect on the S/N ratio and then identify levels of these factors to maximize the overall S/N ratio through the main effect graphs of S/N ratio. When interactions are significant, information obtained from the plots of interactions is employed to determine the optimal settings for the corresponding factors.

Table 1 An example of a penalty index

Score rank	Rank index	Description
1	0	No contact relation
2	1~3	Simple work, forced precedent sequence, direct or absolute "above" relationship
3	4~6	A little difficult, needs careful operation, tools changed infrequently
4	7~9	Very difficult, component easily damaged, tools changed frequently

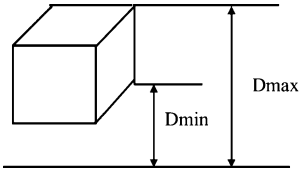


Fig. 4 Maximal and minimal dimensions of a part in the deployment direction

3 Working concepts and procedures

Generally, assembly sequence planning consists of feature-based assembly modeling (i.e., all assembly information is modeled), knowledge-based assembly sequence generation, and interactive assembly planning system demonstration. This study developed a three-stage integrated approach with some heuristic working rules to assist planners in generating the best, most-effective assembly sequence. In the first stage, the detailed data are input from 2D engineering drawings and related assembly information into a CAD assembly package. Then Above Graph (the graph illustrates that a part located absolutely or relatively above the other parts) and transforming rules were used to create a correct explosion graph of the assembly models. In the second stage, a three-level relational model graph, with geometric constraints and APDs, was generated to create a complete relational model graph, an incidence matrix, and a global feasible assembly sequence. In the third stage, the BPNN engine via parameter optimization using the Taguchi method and design of experiment was employed to predict the available assembly sequences. The aforementioned BPNN engine, created by a toy car model, as a learning (training) sample, and a toy motorbike model and a brushless DC fan as testing samples, was used to evaluate the feasibility of the proposed model in terms of differences in assembly sequences.

The working concepts and procedures fall into two parts: The first is to construct a graph-based assembly sequence planning, and the second is to develop a KBE system with an embedded robust BPNN engine. The proposed flow chart is shown in Fig. 1. A KBE system that renders an NX/KF-based operational interface to access the potential

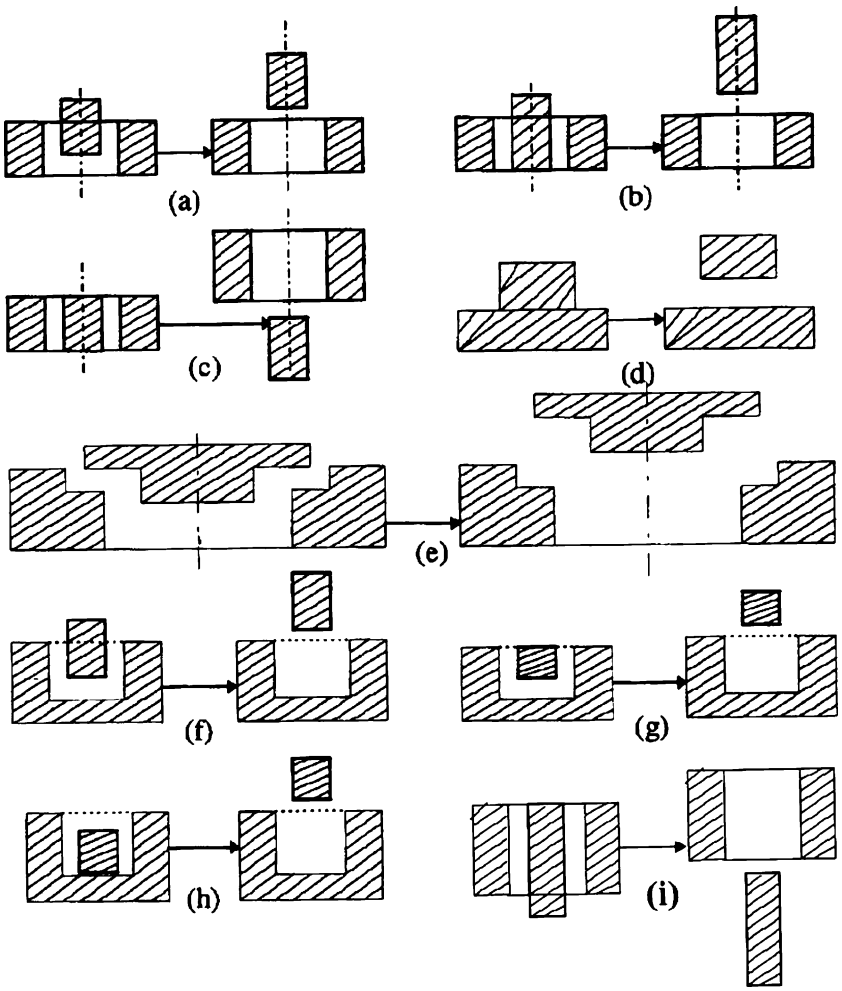


Fig. 5 Nine Above Graph sets

Fig. 6 Symbol definitions for combined parts

graphs and BPNN-related details via different types of databases is given in Fig. 2.

4 Three-stage ASP implementation of the KBE system

Knowledge-based systems as specific software programs are designed to capture and apply domain-specific knowledge and know-how to solve problems. For KBE systems, they usually (but not always) imply the use of some kind of computer system, examples of which include so-called expert systems, web-based knowledge databases, and the like. Their aims are to obtain product and process information in such a way as to allow businesses to model engineering design processes and then use the model to automate all or part of the process. KBE is also a technology that allows an engineer to create a product model based on rules that capture the methodology used to design, configure, and assemble products; it facilitates the capture of the intent behind the product design by representing the "why" and "how" in addition to "what" of a design [46], and its design model including an evaluation criterion is a decision making or an optimization model, where the "best choice" design is called the optimal design. Moreover, KBE is a process of implementing knowledge-based systems and dealing with knowledge processing, where domain-specific knowledge of parts and processes is stored with other attributes: geometry, form features, etc.

In the proposed KBE system, the object-oriented design language (i.e., KBE language which provides ways to capture geometric and non-geometric attributes) can be exploited to build a reusable and interactive smart model (a constraint- and knowledge-based model) of the product, and the exact and heuristic methods merged to implement the three-stage assembly sequence planning: (1) constructing assembly planning via an APD tree to evaluate the complexity of assembly relations, (2) deriving an explosion graph to address how to generate the global feasible assembly sequences, (3) developing the three-level relational model to describe the assembly relations of components via contact-relation symbols and their incidence values, (4) conducting the penalty matrix for assembly planning to build up the criteria levels of assembly difficulties, and (5) procedures and interfaces for creating a KBE system.

4.1 Constructing assembly planning via an APD tree

Commonly, a product's assembly precedence relationships can be represented as a weighted acyclic graph, G (N, A), shown in Fig. 3, where node (N) weights correspond to the time or cost needed to perform the operations and arcs (As) represent precedence constraints. In general, a designer can successively assign assembly relations to form an assembly plan based on a precedence diagram. However, the contact-based precedence diagram cannot effectively express the complexity of the assigned assembly relations. To avoid poor assignments, this study employed the assembly difficulty scoring method to evaluate the complexity of assembly relations. Using the difficulty rank method allows many criteria to be simultaneously considered in assembly planning. The difficulty rank represents the cost incurred (e.g., increasing operation time) by assigning an unsuitable or infeasible assembly sequence. Table 1 presents an example of a penalty index to illustrate the score weightings of some crucial factors.

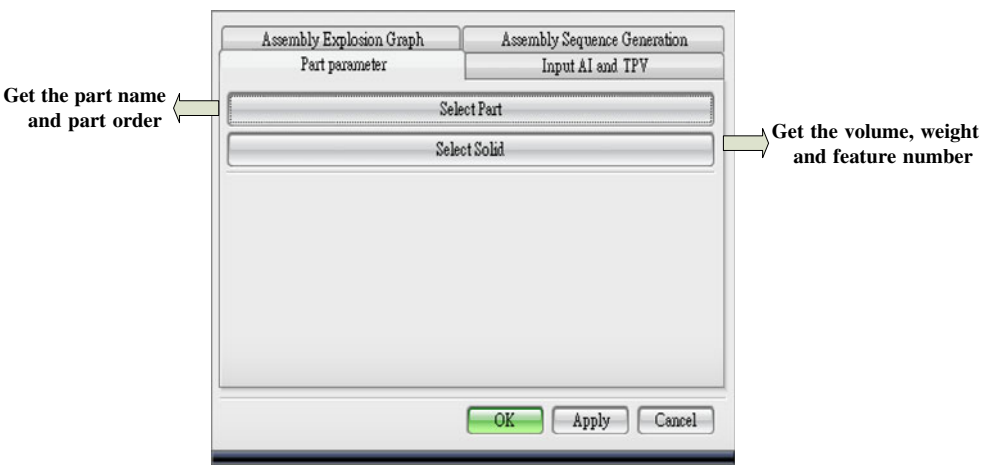


Fig. 7 Part parameters of the UI-style push button

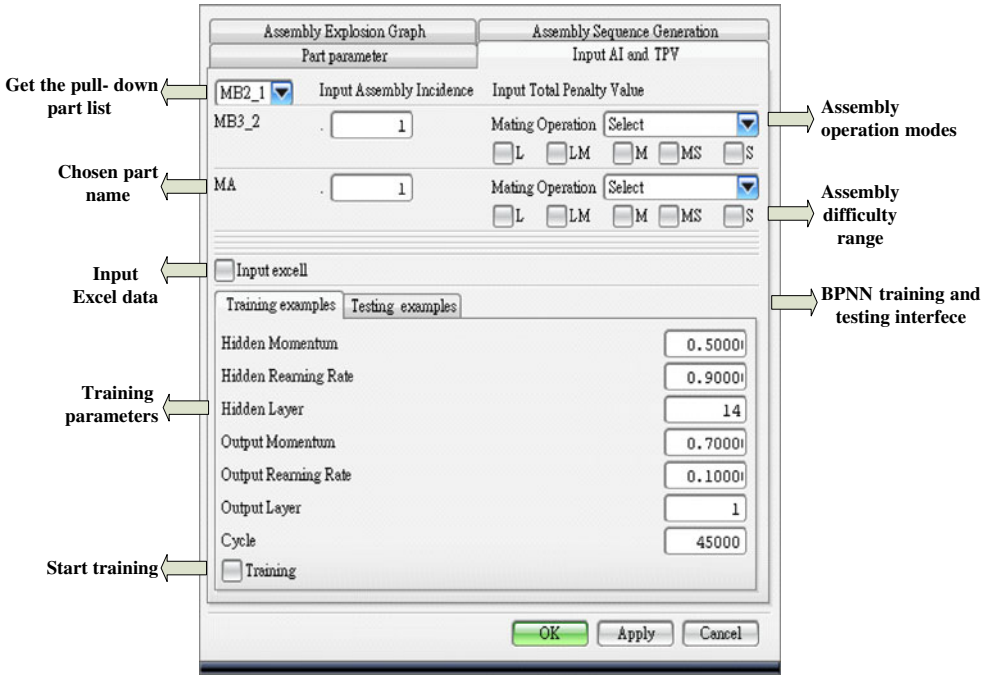


Fig. 8 BPNN parameters and UI-style assembly incidence (*AI*) and total penalty value (*TPV*) interface

4.2 Deriving an explosion graph

The explosion graph is the result of the complete disassembly a product based on relationships of its geometric contacts. This study uses the "contact above" concept to construct an Above Graph and then derives an explosion graph. "Above" implies that a part is located absolutely or relatively above another part in the *Z*-direction or outside the other part in the *X*/*Y*-directions. Two assembly parts are generally related in one of the following ways: no contact (translatable), or attached, or fastened (non-translatable). If a component can be separately pre-assembled, this study takes that component as a part and that need not be disassembled. Some transformation rules are used in this paper to define the "Above" relation of a component. The symbols ">" and "→" are used to denote the "above" and "precedent" relations in the deployment direction.

Rule 1: Set the main deployment direction along the base part composed in that direction and identify the maximum and minimum dimensions of parts as shown in Fig. 4.

Rule 2: If $D_{\min}(A)>D_{\min}(B)$, where $D_{\min}(A)$ and $D_{\min}(B)$ are the respective minimal dimensions of parts A and B along the deployment direction; then, the relation is defined as part A>part B.

Rule 3: If $D_{\min}(A)=D_{\min}(B)$; $D_{\max}(A)>D_{\max}(B)$, where $D_{\min}(A)$, $D_{\min}(B)$, $D_{\max}(A)$, and $D_{\max}(B)$ are the respective minimal and maximal dimensions of parts A and B along the deployment direction, then the relation is defined as part A>part B.

Rule 4: If $D_{\min}(A)=D_{\min}(B)$, $D_{\max}(A)=D_{\max}(B)$ and part A is larger than part B (in diameter, weight, or volume), then the relation is defined as part A>part B.

Rule 5: If part A is mounted on part B and intersects the deployment direction, then the relation is defined as part A>part B.

Rule 6: If the two components have the same priority according to rules 2~4, a component is selected according to a left-to-right and/or counterclockwise deployment.

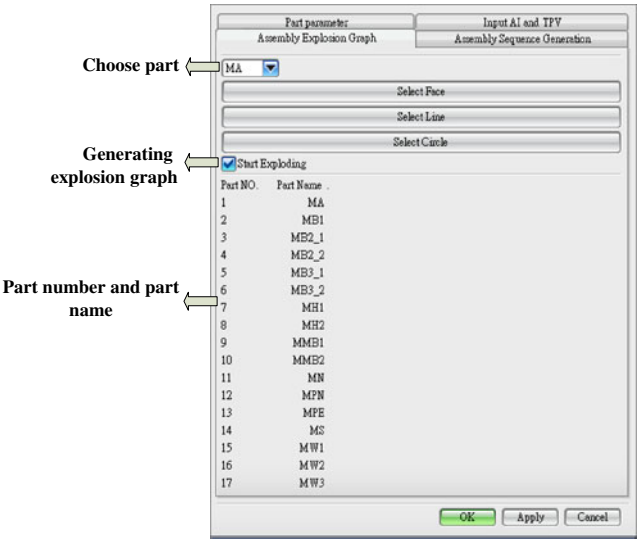


Fig. 9 Explosion graph generation with the UI-style interface

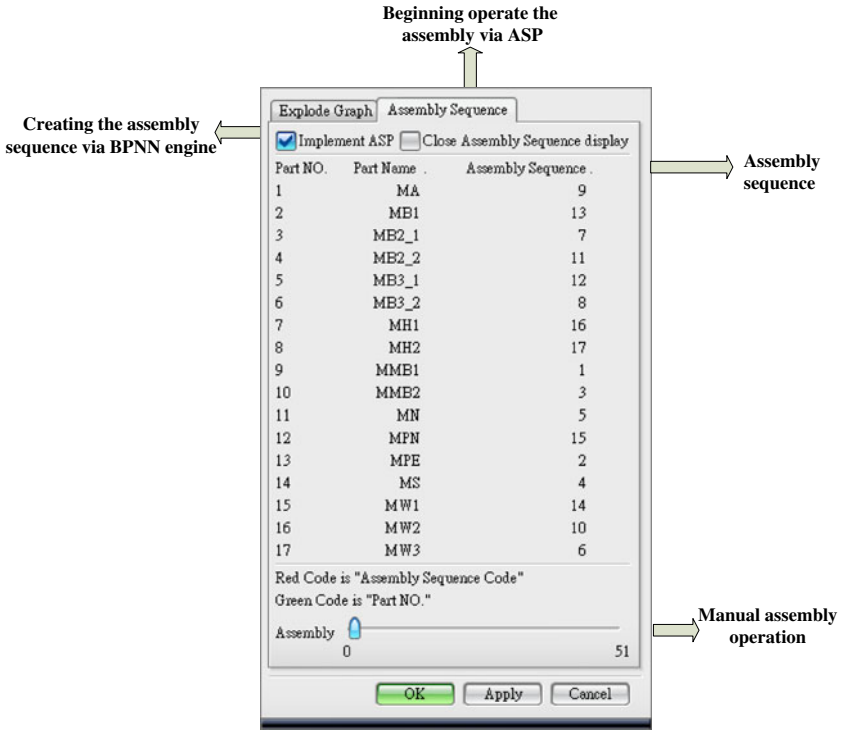


Fig. 10 Assembly sequence generation with the UI-style interface

In practice, inference becomes more difficult as the product becomes more complex. This study used the contact-based Above Graph with the heuristic transformation rules mentioned above to derive the explosion graph and simplify the inference process. Nine Above Graph sets, shown in Fig. 5, were used to guide the transformation. An APD tree, shown in Fig. 3, is normally used to present assembly plans. The APD tree is a graph used to describe

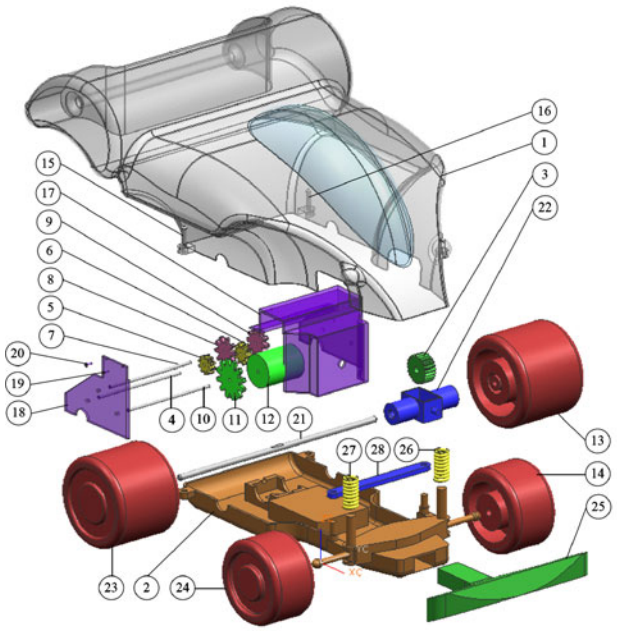


Fig. 11 Parts list and exploded view of a toy car

No	Parts' name
1	MB(MainBody)
2	CP(ChassisPan)
3	DG(DriveGear)
4	GS1_1(GearSet1_1)
5	GS1_2(GearSet1_2)
6	GS1_3(GearSet1_3)
7	GS2_1(GearSet2_1)
8	GS2_2(GearSet2_2)
9	GS2_3(GearSet2_3)
10	GS3_1(GearSet3_1)
11	GS3_2(GearSet3_2)
12	PO(Power)
13	LBW(LeftBackWheel)
14	LFW(LeftFrontWheel)
15	BS1(BaseScrew1)
16	BS2(BaseScrew2)
17	PP1(PowerPack1)
18	PP2(PowerPack2)
19	PPS1(PowerPackScrew1)
20	PPS2(PowerPackScrew2)
21	RA(RearAxis)
22	RD(RearDiff)
23	RBW(RightBackWheel)
24	RFW(RightFrontWheel)
25	SL(Spoiler)
26	SP1(Spring1)
27	SP2(Spring2)
28	SR(SteeringRack)

precedent relations of the parts to be assembled. Commonly, generating feasible sequences for an assembly plan requires an experienced planner to address the following questions:

1. How should a base part be selected?
2. What are the precedence sequences among the contacting parts?
3. What are the contact relationships (liaisons) between each pair of parts?
4. Must the case be sub-assembled in advance?

4.3 Developing the three-level relational model

The relational model, like the explosion graph, is also used to describe the assembly relations of parts. It is a hierarchical structure for organizing the relations of the parts. The following steps are followed to develop three-level relational model graphs and to create APD trees.

Step 1. Select the entire top level of the sub-assemblies.

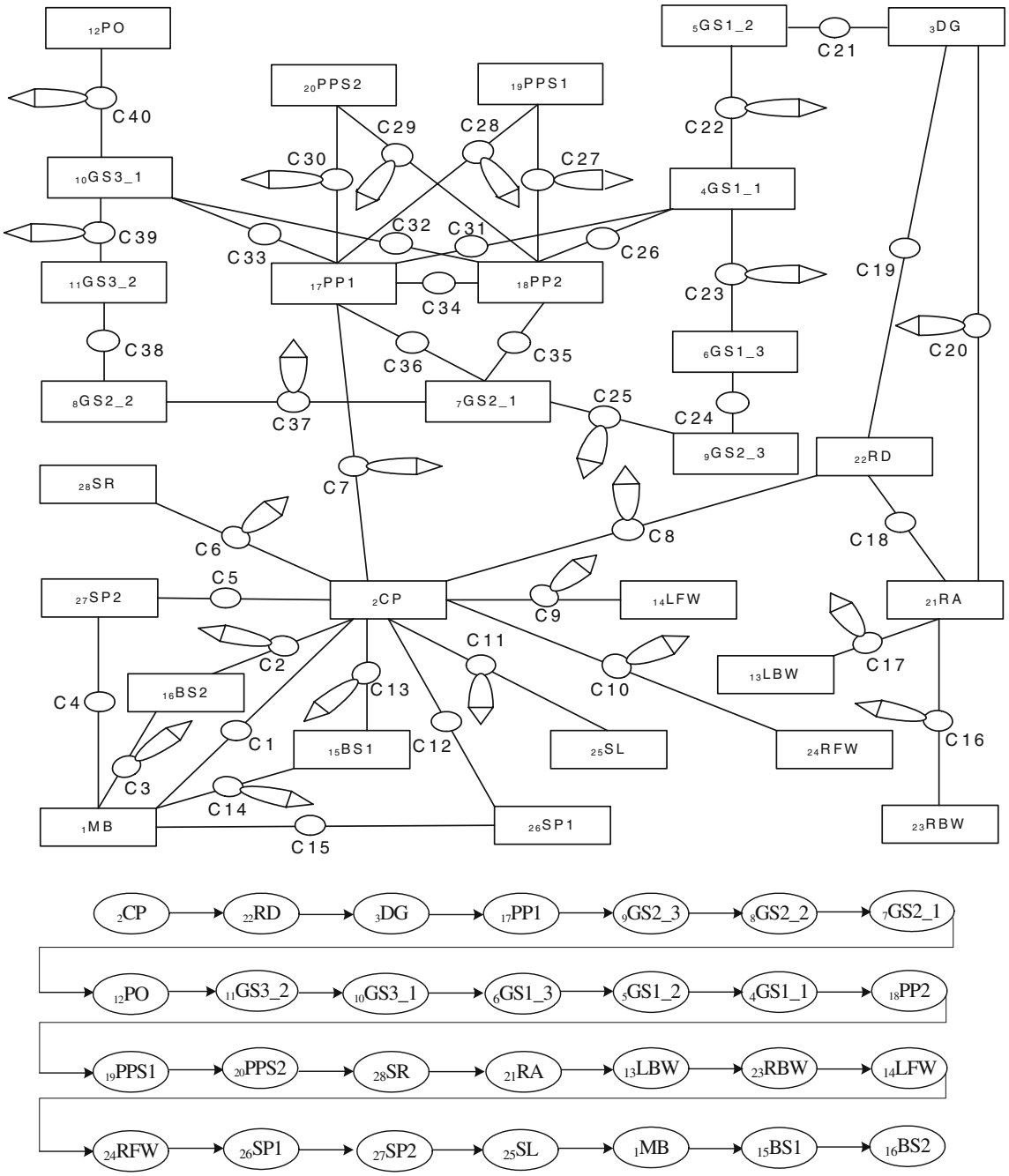
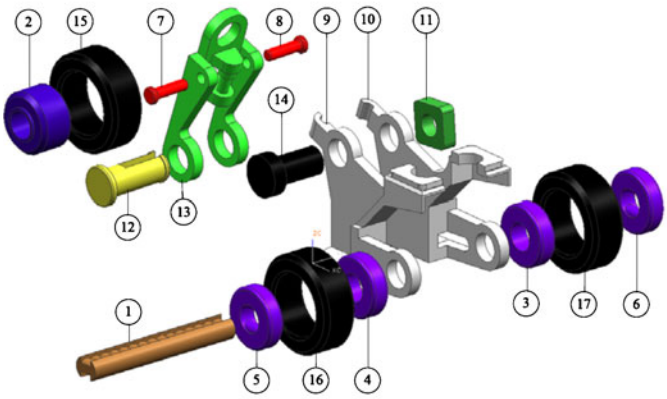


Fig. 12 The complete relational model graph and assembly precedence diagram of a toy car

Fig. 13 Parts list and exploded view of a motorbike



No	Part name
1	MA(MotorbikeAxle)
2	MB1(MotorbikeBearing1)
3	MB2_1(MotorbikeBearing2_1)
4	MB2_2(MotorbikeBearing2_2)
5	MB3_1(MotorbikeBearing3_1)
6	MB3_2(MotorbikeBearing3_2)
7	MH1(MotorbikeHandlE1)
8	MH2(MotorbikeHandlE2)
9	MMB1(MotorbikeMainBody1)
10	MMB2(MotorbikeMainBody2)
11	MN(MotorbikeNut)
12	MPN(MotorbikePin)
13	MPE(MotorbikePlate)
14	MS(MotorbikeScrew)
15	MW1(MotorbikeWheel1)
16	MW2(MotorbikeWheel2)
17	MW3(MotorbikeWheel3)

Step 2. Create the first level of the relational model. Draw out the next level of parts and sub-assemblies for each top-level of sub-assemblies according to the top level of the sub-assemblies.

Step 3. Create the second level of the relational model. Draw out the next level of parts and sub-assemblies for each upper level of sub-assemblies according to the upper level of the sub-assemblies.

Step 4. Repeat step 3 until the bottom level is reached.

Step 5. Create macro-relational model graphs. Join each lower-level relational model graph to its upper-level relational model to form macro-relational model graphs.

Step 6. Sequentially create APD trees from the macro-relational model graphs according to the rules for selecting the base part and transformation rules.

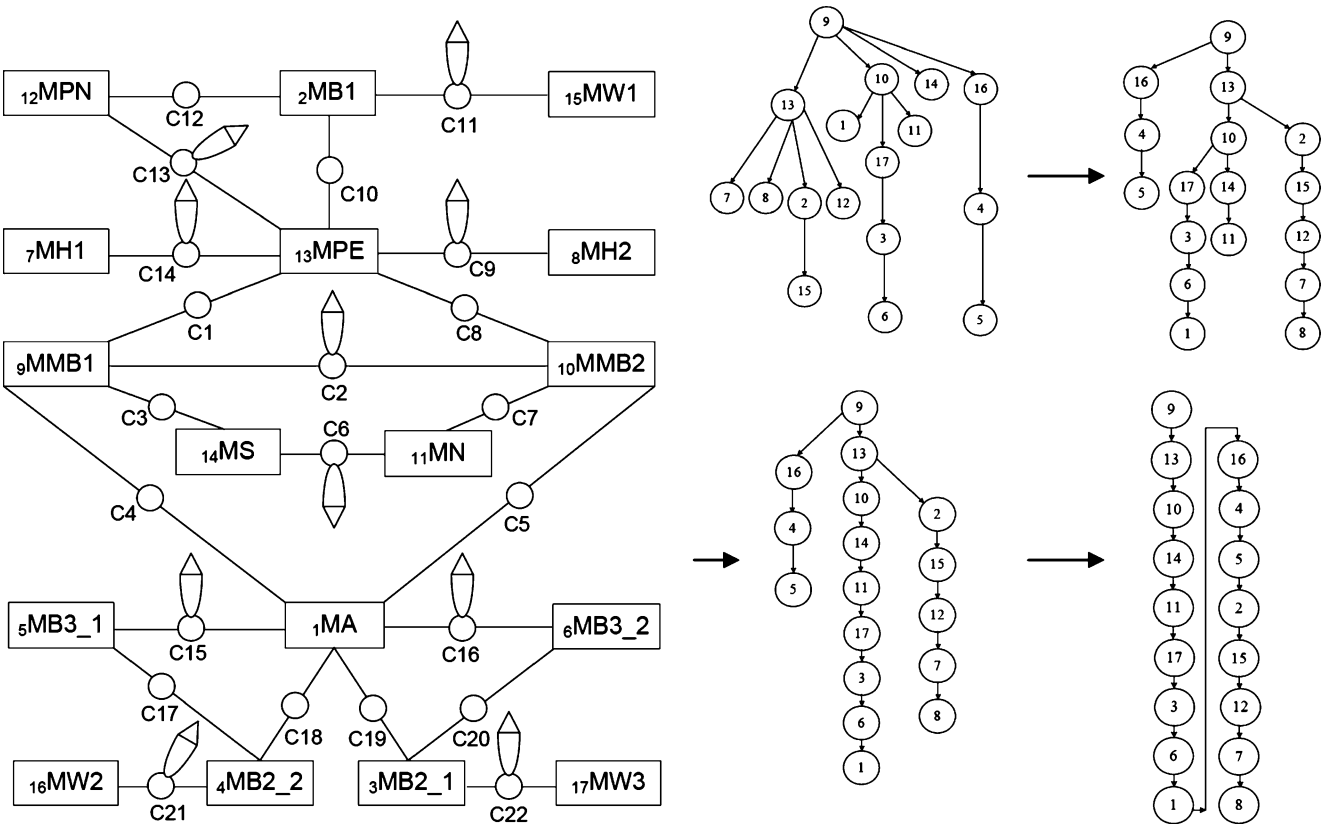


Fig. 14 The complete relational model graph and assembly precedence diagram of a motorbike

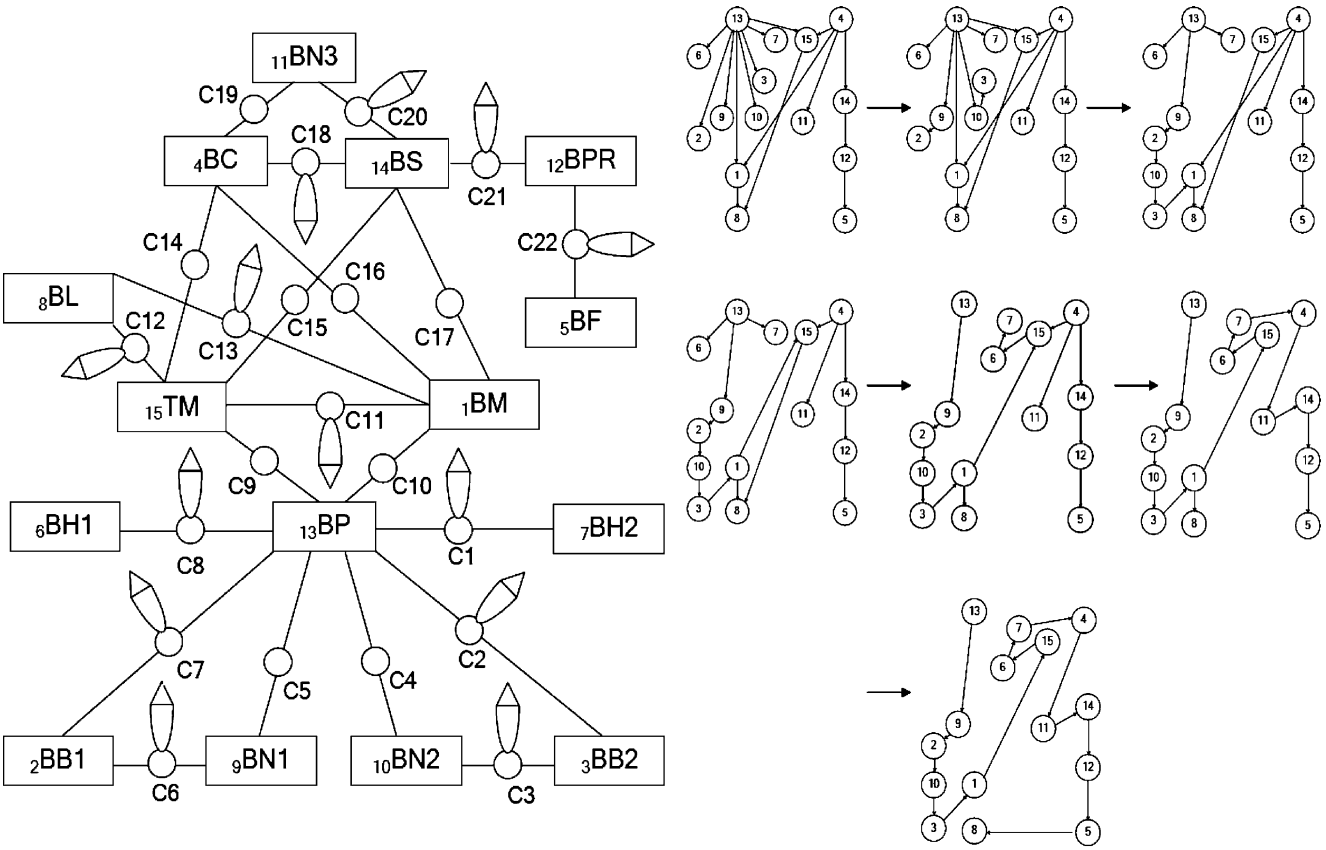


Fig. 15 The complete relational model graph and assembly precedence diagram of a toy boat

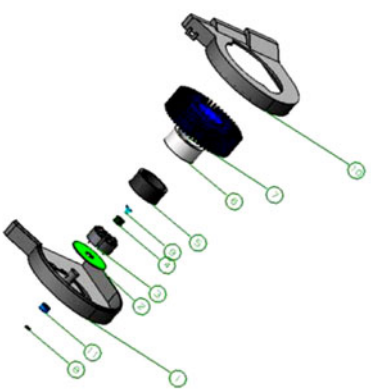
Three symbols are used herein to express the relationships of the combined parts. A square denotes the part name to be combined. A circle represents the contact relation between the two combined parts, and its incidence value was set to 1. A triangle refers to the attachment relationship, and its incidence value was set to 2. Meanwhile, if there is no contact relation between two parts, the incidence value is set to 0. The symbol definitions for combined parts are presented in Fig. 6.

4.4 Conducting the penalty matrix for assembly planning

The penalty index represents the cost incurred by assigning an unsuitable, unfeasible, or difficult assembly operation. Table 1 presents an illustrative example of penalty weightings of some critical factors. For instance, the penalty index is set to 0 if part *i* has absolutely no contact with part *j*; the penalty index is set to 7~9 if the assembly relation is a "loose above" relation or very difficult; the penalty index is set to 4~6 if the assembly relation seems to be a little difficult or requires careful operation; the penalty index is set to 1~3 if the assembly relation is simple or an absolute above relation. By employing the idea of a penalty, other critical factors concerning assembly planning, such as the frequency of changing tools, the similarity of assembly operations, the quality of assembly, the complexity of assembly, etc., can simultaneously be considered.

4.5 Procedures and interfaces for creating a KBE system

A knowledge-based ASP is created from a knowledge base and technical database in the engineering and design



No	Part Name
1	PL (Pillow)
2	BD (Board)
3	CR (Cover)
4	BR1 (Bearing 1)
5	MG (Magnet)
6	CS (Case)
7	IMP (Impeller)
8	FN (Fastener)
9	SP (Spring)
10	FR (Frame)
11	BR2 (Bearing 2)

Fig. 16 Parts list and exploded view of a DC brushless fan

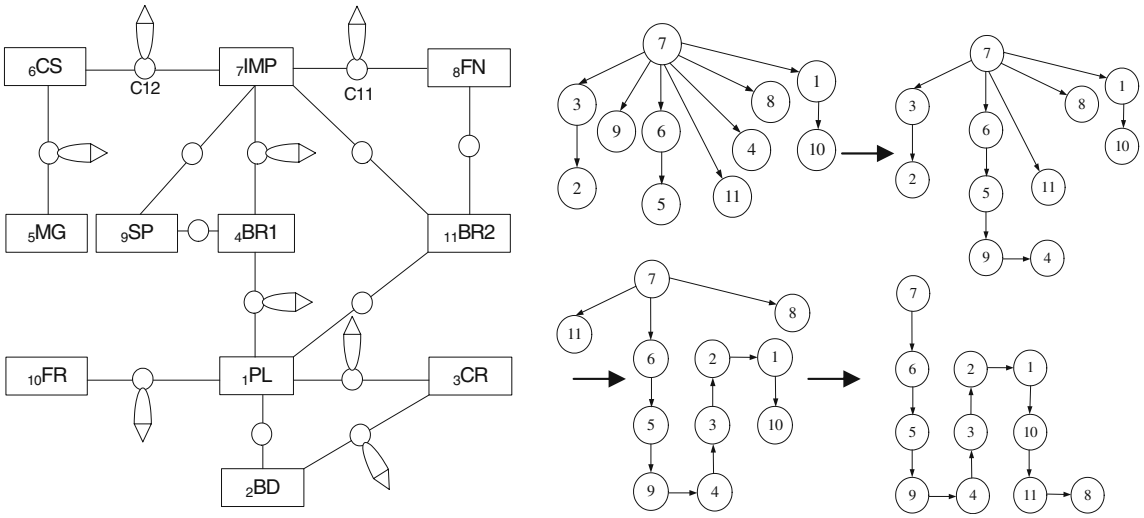


Fig. 17 The complete relational model graph and assembly precedence diagram of a DC brushless fan

Table 2 The optimal assembly sequence of a toy car

Optimal assembly sequence	Parts	AI	TPV	FN	Weight	Volume
1	$_{2}$CP	19	47	9	981.88	125,415.99
2	$_{22}$RD	4	8	10	31.42	11,246.39
3	$_{3}$DG	5	8	27	4.83	3,452.57
4	$_{17}$PP	10	29	11	83.64	29,935.98
5	$_{9}$GS2_3	3	5	22	1.96	1,397.92
6	$_{8}$GS2_2	3	5	22	1.12	802.85
7	$_{7}$GS2_1	6	16	1	3.07	392.7
8	$_{12}$PO	2	3	2	56.34	20,165.61
9	$_{11}$GS3_2	3	5	26	2.28	1,628.77
10	$_{10}$GS3_1	6	16	1	3.07	392.7
11	$_{6}$GS1_3	3	5	22	1.08	771.23
12	$_{5}$GS1_2	3	5	22	0.87	623.61
13	$_{4}$GS1_1	6	16	1	3.07	392.7
14	$_{18}$PP2	8	22	11	17.66	6,321.76
15	$_{19}$PPS1	4	6	3	0.13	14.99
16	$_{20}$PPS2	4	5	3	0.11	14.86
17	$_{28}$SR	7	4	4	27.58	3,522.26
18	$_{21}$RA	7	13	3	29.79	3,804.98
19	$_{13}$LBW	2	5	7	308.9	219,936.4
20	$_{23}$RBW	2	3	7	307.67	219,928.32
21	$_{14}$LFW	2	3	9	176.9	119,227.68
22	$_{24}$RFW	2	3	9	164.33	119,214.45
23	$_{26}$SP1	2	6	3	9.99	1,288.59
24	$_{27}$SP2	2	6	3	9.85	1,276.48
25	$_{25}$SL	2	3	2	234.01	83,756.14
26	$_{1}$MB	7	17	28	932.5	333,750.12
27	$_{15}$BS1	4	10	3	2.38	303.99
28	$_{16}$BS2	4	10	3	2.36	302.45

Table 3 The optimal assembly sequence of a toy motorbike

Optimal assembly sequence	Parts	AI	TPV	FN	Weight	Volume
1	$_{9}$MMB1	5	13	20	7.35	7,697.04
2	$_{13}$MPE	9	19	4	5.19	5,176.46
3	$_{10}$MMB2	5	17	20	6.78	7,696.73
4	$_{14}$MS	3	8	2	1.53	2,297.29
5	$_{11}$MN	3	10	3	0.78	856.50
6	$_{17}$MW3	1	9	4	8.13	7,296.14
7	$_{3}$MB2_1	4	23	3	1.2	1,931.29
8	$_{6}$MB3_2	3	16	5	1.41	1,892.18
9	$_{1}$MA	8	52	2	3.32	2,907.56
10	$_{16}$MW2	2	9	4	8	7,295.23
11	$_{4}$MB2_2	4	18	3	1.18	1,930.96
12	$_{5}$MB3_1	3	5	5	1.4	1,891.72
13	$_{2}$MB1	2	11	4	2.49	3,841.38
14	$_{15}$MW1	2	4	4	7.99	7,294.86
15	$_{12}$MPN	3	12	5	1.28	1,619.55
16	$_{7}$MH1	2	4	3	0.17	231.61
17	$_{8}$MH2	2	3	3	0.15	230.56

arena. The KBE system can be fulfilled using object-oriented language, engineering algorithms, and externally connected databases using KF programming. The first stage of creating a KBE system is to generate an explosion view using Above Graph rules. In the second stage, the global optimal assembly sequence can be created through transformation of the APD. In the final stage, the feasible assembly sequences of various products can be promptly obtained using an embedded robust BPNN engine. Some real-world products of interest are used to demonstrate the above-mentioned scheme under the NX/KF-based KBE system.

In Fig. 7, the part parameter of a UI-style push button gets the part name (*.Part) and part order in terms of the corresponding part filter (Select Part), and the solid parameters (i.e., the volume, weight, and feature number (FN)) are represented on the other filter item (Select Solid). Furthermore, the AI can be obtained in reference to relational model graphs and the total penalty value (TPV) created by engineering members in terms of the difficulty of the assembly operations. In Figs. 8, 9, and 10, the UI-style interface shows the BPNN parameters, explosion graph, and ASP generation.

5 Illustrative examples for developing a KBE system

5.1 Creating the exploded view, relational model graph, and APD

An exploded view can be directly created from the Above Graph, which possesses the contact relationships of a

Table 4 The optimal assembly sequence of a DC brushless fan

Optimal assembly sequence	Parts	AI	TPV	FN	Weight	Volume
1	$_{7}$IMP	8	45.5	140	23.95	10,888.17
2	$_{6}$CS	4	26.5	3	3.12	1,200.74
3	$_{5}$MG	2	14	2	5.25	2,627.48
4	$_{9}$SP	3	13	3	0.10	12.86
5	$_{4}$BR1	5	9.5	4	1.3	165.88
6	$_{3}$CR	4	18	17	18.87	2,028.76
7	$_{2}$BD	4	33	2	3.30	660.71
8	$_{1}$PL	8	39.5	32	28.00	15,135.53
9	$_{10}$FR	2	14.5	11	18.54	14,834.66
10	$_{11}$BR2	5	10	4	1.3	165.88
11	$_{8}$FN	3	20.5	3	0.01	7.24

Table 5 Information on the factor settings with the Taguchi method

Item	Control factor	Level 1	Level 2	Level 3
F_t	Transfer function	Hyperbolic tangent	Sigmoid	
N_h	Number of neurons in the hidden layer	8	13	18
R_l	Learning rate	0.7	0.8	0.9
M_t	Momentum	0.7	0.8	0.9
E_p	Epochs	20,000	35,000	50,000

spatial structure. Figure 11 shows the parts list, assembly codes, and exploded view. The validity of each exploded view can be confirmed by the contact relationships of the spatial structure and Above Graphs. Exact assembly plans can be derived by applying a correct exploded view. Figure 12 shows the complete RMG and APD for the proposed training sample of the case study.

5.2 Developing the BPNN engine

In this study, a toy car was used as a training sample, while a DC brushless fan and a toy motorbike were employed to test and verify the samples. Figures 13, 14. 15, 16, and 17 show the parts list, assembly codes, exploded view, and complete RMG and APD of the latter samples, respectively. The characteristics of each assembly part include the number of the AI, TPV, FN, weight, and volume. These characteristics are commonly regarded as the larger the better for the assembly sequence priority. The optimal sequences resulting in information on five characteristics of the toy car, motorbike, and DC brushless fan are given in Tables 2, 3, and 4, respectively.

The toy car, toy motorbike, and DC brushless fan were, respectively, decomposed into 28, 17, and 11 parts. Each part has five characteristic parameters: the value of the AI, TPV, FN, weight, and volume, which are used as network input parameters, and one expected network output adopts the ranking number of the optimal assembly sequence.

In parameter optimization of neural networks, the Taguchi method is used to obtain the optimal parameter settings of the BPNN. Since the number of hidden layers did not have a significant effect on convergence in this study, the number of hidden layer was set to 1. The controlling factors of the Taguchi method are the transfer function (F_t), the number of hidden neurons (N_h), the learning rate (R_l), momentum (M_t), and epochs (E_p). Information on the factor settings at different levels is given in Table 5. Under the condition of five factors, one for two levels and four for three levels, and no correlation among the five factors, the total degrees of freedom were

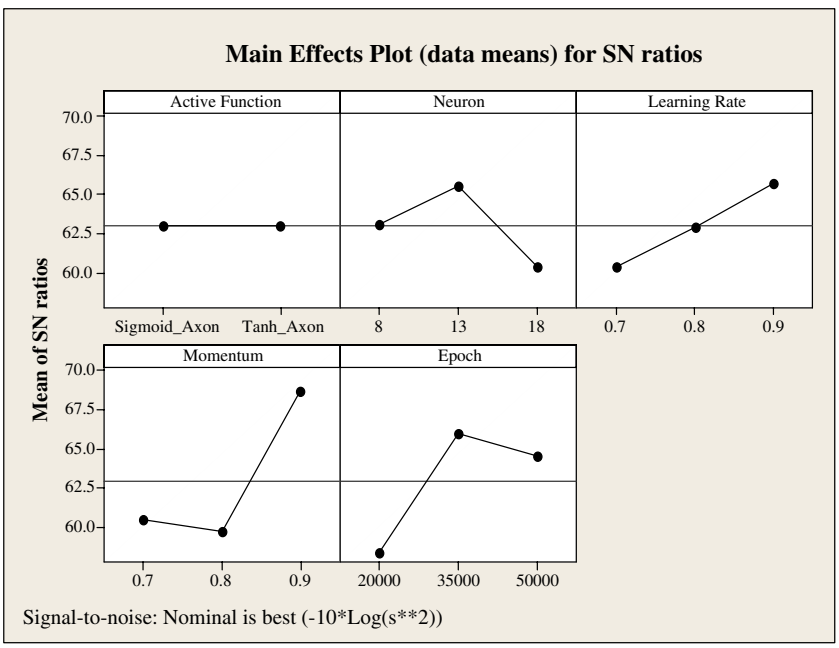


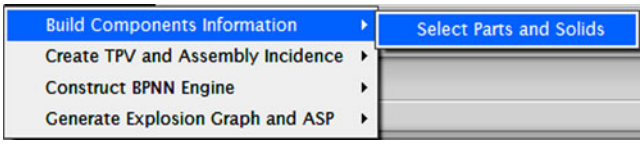
Fig. 18 Main effect plots of BPNN factors

17 (i.e., $1\times(2-1)+4\times(5-1)$). An L_{18} ($2^1\times3^4$) orthogonal array was selected to arrange the factors and carry out the experiment. In this experiment, the predicted performance (mean square error) of Y was the smaller the better for different combinations of factor levels. Υ is the average of two Y's in each replication. The optimal combination of factor levels was the experiment with the largest S/N ratio. From experimental results of the Taguchi method, the main effects plots of BPNN factors through the Taguchi method are given in Fig. 18. The optimal combination of factor levels is represented by the following: a BPNN architecture of 5-13-1, a hyperbolic tangent transfer function, 35,000 calculation generations, a learning rate of 0.9, and a momentum of 0.9. In this study, a toy car was used as a training sample, while a DC brushless fan and a toy motorbike were employed as test samples. The performance of the BPNN engine was a training RMSE of 0.055357604 and a test RMSE of 0.015026421 for the best generalization, and the prediction results of the toy motorbike assembly sequence are shown in Fig. 19.

5.3 Operational procedures of the KBE system

In this section, two real-world products, a brushless DC fan and toy motorbike, are presented to demonstrate ASP by the KBE system. The operational procedures are shown in Fig. 20 and are composed of five steps.

Step 1. Access geometric information in the KBE database on feature numbers, weight, and volume created by the NX CAD modeling system



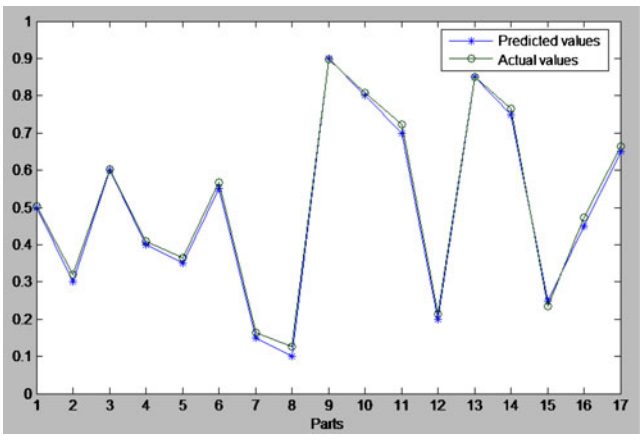


Fig. 19 An assembly sequence prediction by the BPNN engine

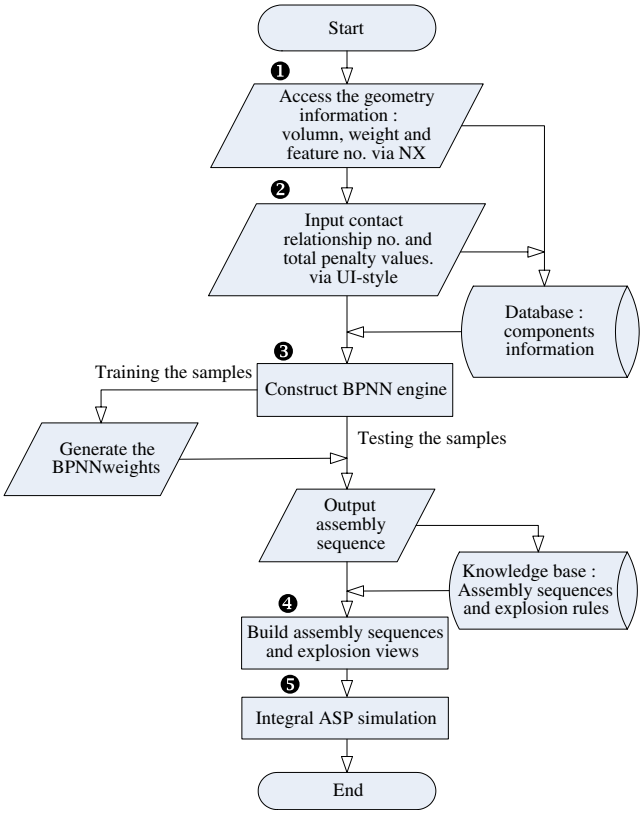
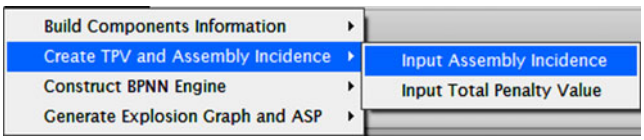
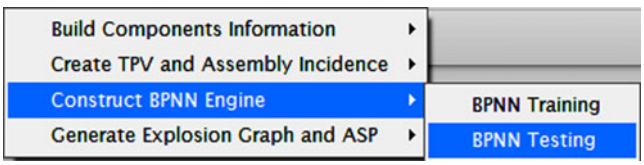


Fig. 20 Operational procedures of the knowledge-based engineering system

Step 2. Input the contact relationship number and TPVs to formulate the training and testing data using the UI-style NX tool



Step 3. Construct BPNN engines by implementing BPNN training and test processes using the NX KF language.



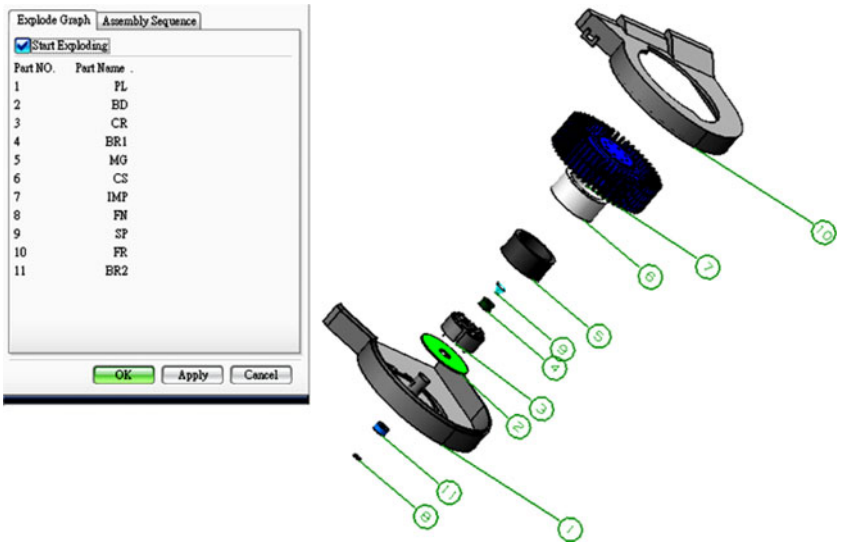


Fig. 21 Parts list and exploded view of a DC brushless fan using the NX interface

Step 4. Automatically build the explosion views and generate assembly sequences in terms of "Above" rules of the explosion graph and APD precedence relationships.

Step 5. Conduct an entire ASP simulation through steps 1~5 as instances of interest for an 11-piece DC brushless fan and a 17-piece toy motorbike respectively represented in Figs. 21, 22, 23, and 24. Figures 21 and 23 show part list and exploded view of a DC brushless fan and a toy motorbike

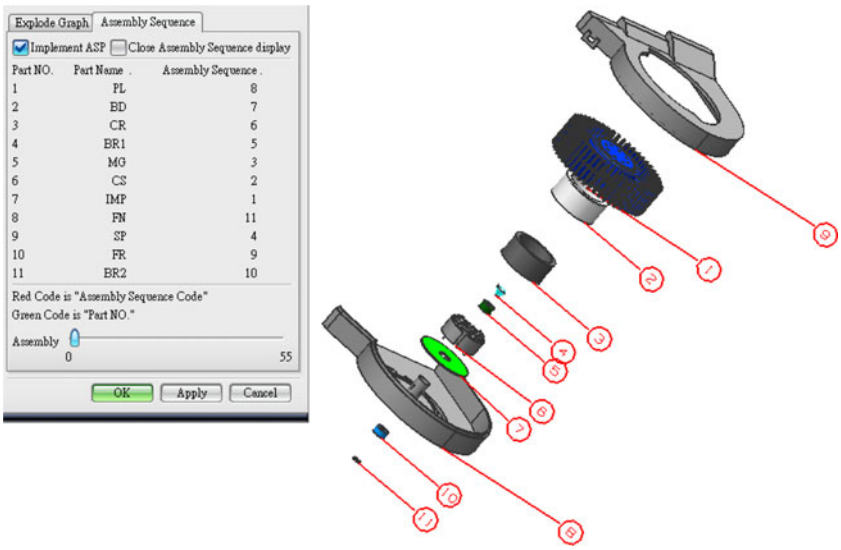


Fig. 22 Assembly sequence planning generation of a DC brushless fan using the NX interface

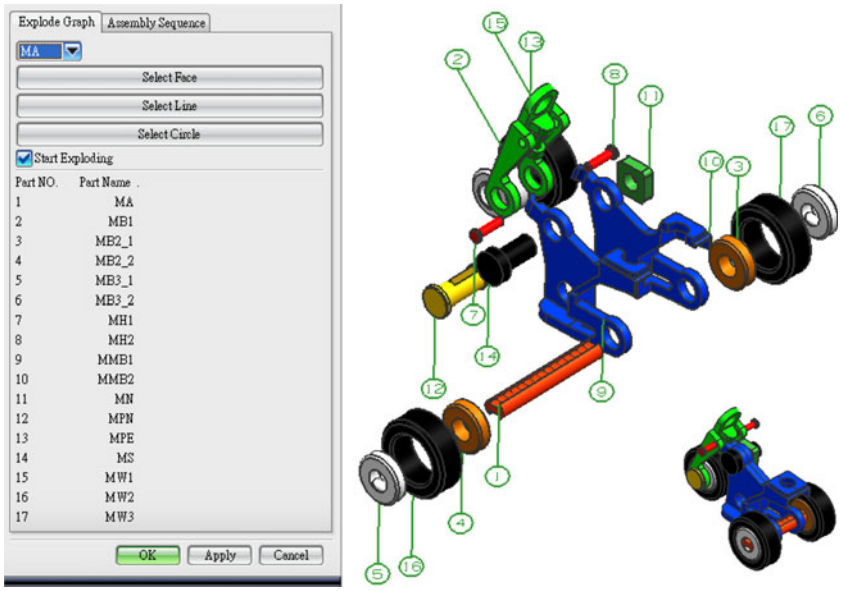


Fig. 23 Exploded view of a toy motorbike using the NX interface

using the NX interface. Figures 22 and 24 show assembly sequence planning generation of a DC fan and a toy motorbike using the NX interface.

6 Conclusions and future work

Theoretically, an assembly plan can be optimized based on factors of the shortest assembly time and assembly sequence optimization. However, there are uncertain factors prior to the determination of the optimized assembly scheme and the completion of the jig and fixture. The system proposed in this paper adopts a three-stage integrated assembly planning approach to express the complexity of the assembly relations and evaluate the feasibility of the respective assembly sequences in the design phase. The experimental results for the case study verified the feasibility of the KBE system, which facilitates the DFA in potential applications of 3D component models to facilitate the manual or automatic assembly in a virtual environment and allows the designer to recognize the relative positions, geometric constraints, and relationships of the 3D components using the following graph-oriented methods: the Above Graph, APD, and relational model graph. The planner can further generate a correct explosion graph and

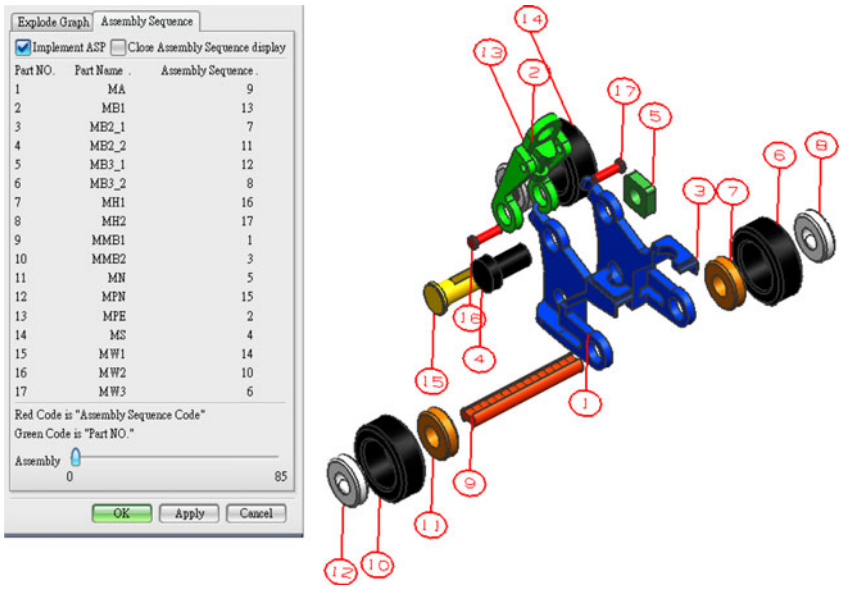


Fig. 24 Assembly sequence planning generation of a toy motorbike using the NX interface

construct an incidence matrix for validating the assembly relationships by applying the Above Graph and relation models.

The future work will be focused on enhancing the BPNN predictor accuracy through consolidating the KBE system database, comparing with various ASP results via global searching tools: guided GA, MA, SA, and PSO and the more representative products will be chosen to validate the performance of the underlying KBE system.

Acknowledgments Financial support from the National Science Council, Taiwan, ROC, under contract NSC97-2221-E-216-026 and Chung Hua University, under contract CHU-96-M-001, is gratefully acknowledged.

References

1. Mascle C, Zhao HP (2008) Integrating environmental consciousness in product/process development based on life-cycle thinking. Int J Prod Econ 12:5–17
2. Kuo TC, Huang SH, Zhang HC (2001) Design for manufacture and design for'X': concepts, applications, and perspectives. Comput Ind Eng 41:241–260
3. Kai Y, Basem EH (2003) Design for six sigma: a roadmap for product development. McGraw-Hill, New York
4. Lotter B (1989) Manufacturing assembly handbook. Butterworths, London
5. Crowson RD (2006) Assembly processes: finishing, packaging, and automation. Taylor & Francis, New York
6. De Fazio TL, Whitney DE (1987) Simplified generation of all mechanical assembly sequences. IEEE Trans Robot Autom 3 (6):640–658
7. Bourjault A (1984) Contribution á une approche méthodologique de l'assemblage automatisé: Elaboration automatique des séquences opératoires, Unpublished doctoral dissertation. Faculté des Sciences et des Techniques de l'Université de Franche-Comté, France
8. Homen de Mello LS, Sanderson AC (1991) Representations of mechanical assembly sequences. IEEE Trans Robot Autom 7 (2):211–227
9. Kroll E (1994) Intelligent assembly planning on triaxial products. Conc Eng Res Appl 1(2):311–319
10. Lee S (1989) Disassembly planning by subassembly extraction. Proceedings of the third ORSA/TIMS Conference on flexible manufacturing systems, Cambridge, pp 383–388
11. Prasad B (1997) Concurrent engineering fundamentals: integrated product development. Prentice-Hall, New Jersey
12. Chen RS, Lu KY, Tai PH (2004) Optimization of assembly plan through a three-stage integrated approach. Int J Comput Appl Technol 19(1):28–38
13. Tai PH (1997) Feature-based assembly modeling for assembly sequence planning of three-dimensional products. Unpublished master's thesis, Cranfield University, UK
14. Chen WC, Tai PH, Deng WJ, Hsieh LF (2008) A three-stage integrated approach for assembly sequence planning using neural networks. Exp Syst Appl 34:1777–1786
15. Wang L, Keshavarzmanesh S, Feng HY, Buchal RO (2003) Assembly process planning and its future in collaborative manufacturing: a review. Int J Adv Manuf Technol 41:132–144
16. Henrioud JM, Relange L, Perrard C (2003) Assembly sequences, assembly constraints, precedence graphs. Proceedings of the fifth IEEE symposium on assembly and task planning, France, July 10–11, pp 90–5
17. Su Q (2009) A hierarchical approach on assembly sequence planning and optimal sequence analyzing. Robot Comput Integr Manuf 25:224–234
18. Yin ZP, Ding H, Xiong YL (2004) A virtual prototyping approach to generation and evaluation of mechanical assembly sequences. Proc Inst Mech Eng 218:87–102
19. Dong T, Tong R, Zhang L, Dong J (2007) A knowledge-based approach to assembly sequence planning. Int J Adv Manuf Technol 32:1232–1244
20. Chen RS, Lu KY, Tai PH (2004) Optimizing assembly planning through a three-stage integrated approach. Int J Prod Econ 88:243–256
21. Gao ZX, Wong WK, Leung SYS, Fan JT (2009) Intelligent production control decision support system for flexible assembly lines. Exp Syst Appl 36(3):4268–4277
22. Gao ZX, Wong WK, Leung SYS, Fan JT, Chan SF (2008) A genetic-algorithm-based optimization model for scheduling flexible assembly lines. Int J Adv Manuf Technol 36(1–2):156–168
23. Wong WK, Leung SYS, Mok PY (2006) Developing a genetic optimization approach to balancing an apparel assembly line. Int J Adv Manuf Technol 28:387–394
24. Tseng HE, Wang WP, Shih SY (2007) Using memetic algorithms with guided local search to solve assembly sequence planning. Exp Syst Appl 33(2):451–467
25. Gao L, Qian W, Li X, Wang J (2010) Application of memetic algorithm in assembly sequence planning. Exp Syst Appl 49:1175–1184
26. Wang JF, Liu JH, Zhong YF (2005) A novel ant colony algorithm for assembly sequence planning. Int J Adv Manuf Technol 25:1137–1143
27. Marian RM, Luong LHS, Abhary K (2003) Assembly sequence planning and optimisation using genetic algorithms. Appl Soft Comput 2(3):223–253
28. Tripathi M, Agrawal S, Pandey MK, Shankar R, Tiwari MK (2009) Real world disassembly modeling and sequencing problem: optimization by Algorithm of Self-Guided Ants (ASGA). Robot Comput Integr Manuf 25(2009):483–496
29. Choi YK, Lee DM, Cho YB (2009) An approach to multi-criteria assembly sequence planning using genetic algorithms. Int J Adv Manuf Technol 42:180–188
30. Hongbo S, Shuxia L, Degang G, Peng L (2006) Genetic simulated annealing algorithm-based assembly sequence planning. International Technology and Innovation Conference, ITIC 2006 1573–1579
31. Kulon J, Broomhead P, Mynors DJ (2003) Applying knowledge-based engineering to traditional manufacturing design. Int J Adv Manuf Technol 30:945–951
32. Lovett PJ, Ingram A, Bancroft (2000) Knowledge-based engineering for SMEs—a methodology. Int J Mater Process Technol 107:384–389
33. Chen CLP (1990) Neural computation for planning and/or precedence-constraint robot assembly sequences. Proceedings of the International Conference on Neural Networks, San Diego, CA, pp 127–142
34. Hong DS, Cho HS (1995) A neural network based computational scheme for generating optimized robotic assembly sequences. Eng Appl Artif Intell 8(2):129–145
35. Sinanoğlu C (2006) A neural predictor to analyze the effects of metal matrix composite structure (6063 Al/SiCp MMC) on journal bearing. Ind Lubr Tribol 58(2):95–109
36. Chen WC, Hsu YY, Hsieh LF, Tai PH (2010) A systematic optimization approach for assembly sequence planning using Taguchi method, DOE, and BPNN. Exp Syst Appl 37(1):716–726

37. Sage AP (1990) Concise encyclopedia of information processing in systems and organizations. Pergamon, New York
38. Haupt RL (2004) Practical genetic algorithms, 2nd edn. Wiley-Interscience, USA
39. Maier HR, Dandy GC (1998) Understanding the behaviour and optimising the performance of back-propagation neural networks: an empirical study. Environ Model Softw 13:179–191
40. Liu Y, Liu W, Zhang Y (2001) Inspection of defects in optical fibers based on back-propagation neural networks. Opt Commun 198(4–6):369–378
41. Yao S, Yan B, Chen B, Zeng Y (2005) An ANN-based element extraction method for automatic mesh generation. Exp Syst Appl 29:193–206
42. Chen WC, Hsu SW (2007) A neural-network approach for an automatic LED inspection system. Exp Syst Appl 33(3):531–537
43. Hush DR, Horne BG (1993). Progress in supervised neural networks. IEEE Signal Processing Magazine, January, pp 8–39
44. Cheng CS, Tseng CA (1995) Neural network in detecting the change of process mean value and variance. J Chin Inst Ind Eng 12(3):215–223
45. Santos MS, Ludermir B (1999) Using factorial design to optimize neural networks, International Joint Conference on IEEE Neural Networks, Washington, DC, 2, pp 857–861
46. Boart P (2005) Life cycle simulation support for functional products. M.Sc. thesis, Luleå University of Technology, Sweden