An Integrated Design Evaluation and Assembly Sequence Planning Model using a Particle Swarm Optimization Approach

Feng-Yi Huang, Yuan-Jye Tseng

Open Science Index, Industrial and Manufacturing Engineering Vol:5, No:5, 2011 publications.waset.org/6707.pdf

***Abstract*—**In the traditional concept of product life cycle management, the activities of design, manufacturing, and assembly are performed in a sequential way. The drawback is that the considerations in design may contradict the considerations in manufacturing and assembly. The different designs of components can lead to different assembly sequences. Therefore, in some cases, a good design may result in a high cost in the downstream assembly activities. In this research, an integrated design evaluation and assembly sequence planning model is presented. Given a product requirement, there may be several design alternative cases to design the components for the same product. If a different design case is selected, the assembly sequence for constructing the product can be different. In this paper, first, the designed components are represented by using graph based models. The graph based models are transformed to assembly precedence constraints and assembly costs. A particle swarm optimization (PSO) approach is presented by encoding a particle using a position matrix defined by the design cases and the assembly sequences. The PSO algorithm simultaneously performs design evaluation and assembly sequence planning with an objective of minimizing the total assembly costs. As a result, the design cases and the assembly sequences can both be optimized. The main contribution lies in the new concept of integrated design evaluation and assembly sequence planning model and the new PSO solution method. The test results show that the presented method is feasible and efficient for solving the integrated design evaluation and assembly planning problem. In this paper, an example product is tested and illustrated.

Keywords—assembly sequence planning, design evaluation, design for assembly, particle swarm optimization

I. Introduction

THE purpose of assembly sequence planning is to determine a proper sequence of components and assembly operations. With the ordered sequence, the components can be located at the specified positions and fixed with the assembly operations to construct the final product. In assembly sequence planning, the components and the assembly operations are arranged in an ordered sequence under the constraints of operational constraints and precedence constraints to achieve the assembly cost objectives. In the traditional concept of product life cycle management, the activities of design, manufacturing, and assembly are performed in a sequential way. In traditional assembly sequence planning models, the assembly sequences are planned after the design of components is completed. Because the design specifications of the components are defined, the alternatives in assembly sequence planning will be confined in a small domain.

The design requirements are usually specified for the functional and aesthetic purposes. On the other hand, manufacturing and assembly activities usually consider costs. As a result, the considerations in design may contradict the considerations in manufacturing and assembly. For example, a good design with good functions can cause some difficulties in the corresponding assembly sequences. In this way, a good design may result in a high cost in the subsequent assembly sequences.

In the typical design for assembly models, the design specification of each component is analyzed to evaluate the spatial and connection relationships between the components. The spatial and connection relationships are then used to analyze the collision-free assembly directions and assembly operations. The degrees of difficulty in the assembly directions and operations are then evaluated to determine the feasibility and suitability of the design. If the product is evaluated as difficult to be assembled, then the design may need to be changed. After some components are changed, the assembly directions and operations can be improved. By changing the design of components, the difficulty in the assembly directions and operations can be reduced to achieve the goal of design for assembly.

In the typical design for assembly models, the analysis and evaluation are performed in a sequential and interactive way. Moreover, the design evaluation and the assembly sequences are not concurrently planned. Therefore, it requires an integrated design evaluation and assembly sequence planning model to improve the traditional approach.

In this research, an integrated design evaluation and assembly sequence planning model is presented. Given a product requirement, there may be several design alternative cases for designing the components for the same product. If a different design case is selected, some the components may be different. An assembly sequence is required to locate and fix the components to construct the product. Therefore, if a different

Feng-Yi Huang is with the Department of Industrial Engineering and Management, Yuan Ze University, Chung-Li, Taoyuan 320, Taiwan
(phone: 886-3-463-8800#2508; fax: 886-3-463-8907; e-mail: s948909@mail.yzu.edu.tw).

Yuan-Jye Tseng is with the Department of Industrial Engineering and Management, Yuan Ze University, Chung-Li, Taoyuan 320, Taiwan
(phone: 886-3-463-8800#2508; fax: 886-3-463-8907; e-mail: ieyjt@saturn.yzu.edu.tw).

design case is selected, the assembly sequence can be different.

In this paper, first, the design specifications of components are defined by using graph based models. An assembly precedence graph (APG) is built to represent the adjacency and precedence relationships between the components. The graph based models are transformed to an assembly precedence matrix (APM) to represent the precedence constraints. A particle swarm optimization (PSO) approach is developed by encoding a particle with a position matrix defined by the design alternative cases and the assembly sequences. The PSO algorithm simultaneously performs design evaluation and assembly sequence planning under the precedence constraints with an objective of minimizing the total assembly costs.

The presented models and algorithms were implemented and tested. This paper is organized as follows. Section 2 presents a literature review. Section 3 describes the model for integrated design evaluation and assembly sequence planning. Section 4 presents the PSO algorithm. Implementation and test results are presented in Section 5. Conclusions are discussed in Section 6.

II. Literature Review

In the related research, it can be summarized that assembly sequence planning can be performed with three stages: (1) assembly representation and modeling, (2) assembly sequence generation, and (3) assembly sequence evaluation and optimization. Lin and Chang [1] presented an assembly precedence diagram (APD) which is a directed graph representing the precedence of the components and the associated assembly operations. In Abdullah *et al.* [2], a review of assembly sequence planning methods was presented. Lai and Huang [3] presented a systematic approach for automatic assembly sequence generation. Chen and Lin [4] presented optimizing assembly planning through a three-stage integrated approach. Su [5] introduced a geometric constraint analysis method to generate assembly precedence and to evaluate feasible assembly sequences. Dong *et al.* [6] presented an assembly tree hierarchy to analyze geometric and non-geometric information for assembly sequence planning. In the recent research, Tseng *et al.* [7] presented a multi-plant assembly sequence planning model using a GA method to integrate assembly sequence planning and plant assignment. Jin *et al.* [8] presented an assembly sequence optimization method for complex mechanical product by employing a directed graph and an assembly matrix to represent the assembly relation. In Gao *et al.* [9], the memetic algorithm was used to solve the assembly sequence planning problem by combining the parallel global search nature of evolutionary algorithms with local search to improve individual solutions.

With a given set of components, sequencing the components may become a combinatorial problem. From the solution aspect, the PSO (particle swarm optimization) algorithm has been shown to be effective and efficient in solving different optimization problems. The PSO has been successfully applied to many continuous and discrete optimizations [10], [11]. Banks el al. [12] reviewed and summarized the related PSO research in the areas of hybridization, combinatorial problems, multiple objectives and constrained optimization areas.

In this research, a PSO algorithm with a new encoding scheme is developed for concurrently performing design evaluation and assembly sequence planning.

III. Representation Models

A. Assembly Precedence Graph

An assembly precedence graph (APG) is modeled for representing the components and the assembly operations.

$$APG \text{ is a directed graph } G = (C, A), \tag{1}$$

where $C = \{c_1, \ldots, c_n\}$ = the set of components,
c_i = (*component node*) = a component, $i = 1, \ldots, n$,
$A = \{a_1, \ldots, a_m\}$ = the set of operation arcs between component nodes,

As shown in Figure 1, the example product A is a mobile phone with 13 main components. The APG of the product A is shown in Figure 2.

B. Assembly Precedence Matrix

An APG is transformed into an assembly precedence matrix (APM) for use in the PSO.

$$APM = \begin{array}{c} \\ c_{i=1} \\ c_{i=2} \\ \vdots \\ c_{i=n} \end{array} \begin{array}{c} \begin{array}{cccc} c_{j=1} & c_{j=2} & \cdots & c_{j=n} \end{array} \\ \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & & \cdots & \\ \vdots & \vdots & b_{ij} & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{bmatrix} \end{array} , \tag{2}$$

where c_i and c_j are components,
$b_{ij} = 1$ represents that component c_j must be assembled before component c_i.

APM for the example product A =

$$\begin{array}{c|ccccccccccccc|} & 01 & 02 & 03 & 04 & 05 & 06 & 07 & 08 & 09 & 10 & 11 & 12 & 13 \\ 01 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 02 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 03 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 04 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 05 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 06 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 07 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 08 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 09 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 13 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

C. Design Alternative Case Table

Design alternative cases are represented in a table format. Given a product requirement, the design of the components can be represented as the original design. With the same product requirement, some of the components can be changed to satisfy the assembly constraints and to attend the assembly cost objectives. If some of the components are changed in design, the assembly constraints and the assembly costs may be reduced. By changing the design, the assembly sequences can be affected in constraints and be improved in costs. In this way, the design and assembly sequences can both be evaluated and optimized.

A design alternative case table (DCT) is developed for use in the design representation and assembly sequence planning. The general form of a DCT is shown in Table I. In the table, given an original design and the design alternative cases $d_j = 1, \ldots, m$, a value of $t_{ij} = 1$ indicates that the component c_i is changed in design d_j. A value of $t_{ij} = 0$ indicates that the component c_i is not changed in design d_j. The DCT of the product A is shown in Table II.

If a design alternative case is selected, a different set of components are used in the building of APG and APM. If different APG and APM are built, the assembly sequences will be affected. In this way, by selecting different design alternative cases, the design evaluation and assembly sequence planning can be concurrently performed

IV. Solution Using Particle Swarm Optimization Method

A PSO algorithm is presented for simultaneously performing design evaluation and assembly sequence planning.The PSO algorithm is an evolutionary computation method introduced by Kennedy and Eberhard (1995, 1997). In PSO, each particle moves around in the multi-dimensional space with a position and a velocity. The velocity and position are constantly updated by the particle's own experience and the experience of the whole swarm. Given a problem, a particle can be encoded to represent a solution. Each solution, called a particle, flies in the search space towards the optimal position.

In the original definition, a particle is defined by its position and velocity. The position of a particle i in the D-dimension search space can be represented as $X_i=[x_{i1}, x_{i2}, \ldots, x_{id}, \ldots, x_{iD}]$. The velocity of the particle i in the D-dimension search space can be represented as $V_i=[v_{i1}, v_{i2}, \ldots, v_{id}, \ldots, v_{iD}]$. Each particle has its own best position $P_i=[p_{i1}, p_{i2}, \ldots, p_{id}, \ldots, p_{iD}]$ representing the particle's personal best objective (*pbest*) at time t. The global best particle is denoted as p_g and the best position of the entire swarm (*gbest*) is denoted as $P_g=[p_{g1}, p_{g2}, \ldots, p_{gd}, \ldots, p_{gD}]$ at time t. To search for the optimal solution, each particle adjusts its velocity according to the velocity updating equation and position updating equation.

$$v_{id}^{new} = w_i \cdot v_{id}^{old} + c_1 \cdot r_1 \cdot \left(p_{id} - x_{id}\right) + c_2 \cdot r_2 \cdot \left(p_{gd} - x_{id}\right) \tag{3}$$

where d =1, …, D, i =1, …, E (number of particles),
v_{id}^{new}: the new velocity of i in the current iteration t,
v_{id}^{old}: the velocity of i in the previous iteration (t - 1),
c_1 and c_2: constants called acceleration coefficients,
w_i: the inertia weight,
r_1 and r_2: two independent random numbers with a uniform distribution [0, 1],
p_{id}: the best position of each individual particle i,
p_{gd}: the best position of the entire swarm.

$$x_{id}^{new} = x_{id}^{old} + v_{id}^{new}\ , \tag{4}$$

where x_{id}^{new} is the new position in the current iteration t, x_{id}^{old} is in the previous iteration (t - 1).

A. Encoding and Decoding Scheme

In the developed encoding scheme, a particle represents a design alternative case and the corresponding assembly sequence. A heuristic sequencing and selection rule for encoding and decoding is introduced as follows.

The position of particle i is represented by a position matrix, denoted as X_{ijk}, j = 1, …, (M+1), k = 1, …, N, where N is the number of components and M is the number of design alternative cases. In the heuristic sequencing rule, the values in the first row S of R_{s1}, R_{s2}, …, R_{sN} represent the ranked order values of the N components in an assembly sequence.

In each column, the values from row F_1 to row F_M represent the ranked assignment values for design alternative case selection. In the heuristic selection rule, the component C_k is assigned to the design alternative case with the smallest value in the column of R_{1k}, R_{2k}, …, R_{Mk}.

$$X_{ijk} = \begin{array}{c|cccccc} & C_1 & C_2 & \cdots & C_k & \cdots & C_N \\ \hline S & R_{s1} & R_{s2} & \cdots & R_{sk} & \cdots & R_{sN} \\ F_1 & R_{11} & R_{12} & \cdots & R_{1k} & \cdots & R_{1N} \\ F_2 & R_{21} & R_{22} & \cdots & R_{2k} & \cdots & R_{2N} \\ \vdots & & & & & & \\ F_j & & & \vdots & R_{jk} & & \vdots \\ \vdots & & & & & & \\ F_M & R_{M1} & R_{M2} & \cdots & R_{Mk} & \cdots & R_{MN} \end{array}\ , \tag{5}$$

where i = 1, …, E, where F_j is a plant, j =1, …, M, and C_k is a component, k =1, …, N,
R_{sk} represents the ranked order value of a component k,
R_{jk} represents the ranked selection value for component k assigned to design j.

In the heuristic rule for assembly sequencing, the values in [R_{s1}, R_{s2}, …, R_{sk}, …, R_{sN}] are sorted in an ascending order. The ranked order values represent the ordered position of component C_k in the assembly sequence. For example, if the values of row S are [4.5 1.1 3.2 7.6 5.3], then the ordered positions of (C_1, C_2, C_3, C_4, C_5) are (third, first, second, fifth, fourth). The assembly sequence is determined as (C_2, C_3, C_1, C_5, C_4).

In the heuristic rule for design alternative case selection, in each column of C_k, the component C_k is assigned to the design alternative case with the smallest ranked assignment value in R_{jk}, for j = 1, …, M. For example, if there are four design alternative cases, the values of column C_2 are $[3.1\ \ 5.8\ \ 1.5\ \ 6.9]^{\mathrm{T}}$, then the smallest value is 1.5 of design D_3. It means that the component C_2 is changed in design alternative case D_3.

B. *Fitness Function*

The cost functions include two major items. The assembly operational costs are mainly related to assembly sequencing, whereas the design related costs are primarily related to a design change and its related cost for changing the components.

1) Assembly operation cost (*AOC*): The assembly operation cost is the basic operational cost for performing an assembly operation.
2) Assembly tool change cost (*ATC*): To perform the assembly operation, proper tools are required. If two tools are different, then an assembly tool change cost is required.
3) Assembly setup change cost (*ASC*): If two consecutive setups are different, then an assembly setup change cost is required.
4) Design related cost (*DRC*): Proper design related cost for designing and changing the components in the design alternative cases.

The total cost function (*TC*) can be formulated as follows (unit: dollars).

$$TC = AOC + ATC + ASC + DRC \tag{6}$$

In the PSO evaluation, the objective is to minimize the fitness function as follows.

$$\text{Min } Fitness = TC, \tag{7}$$

Fitness: the fitness function value of a particle.

C. *Integrated Design Evaluation and Assembly Sequence Planning*

The flowchart is shown in Figure 3.

Step 1. Setup parameters.
1) Set iteration $t = 0$.
2) T_{Number}: the iteration (generation) number.
3) P_{Size}: the number of particles.

Step 2. Initialize a population of particles $i = 1, \ldots, E$,with random positions and velocities.
1) A particle i is defined by a multi-dimensional position matrix of $(N)*(M+1)$.
2) The position of particle i is defined by X_{ijk}.
3) The velocity of particle i is defined by V_{ijk}.

Step 3. Evaluate the fitness function.
1) $t = t + 1$.
2) $Fitness = TC$.

Step 4. Update the velocity of each particle i.

$$v_{id}^{new} = w_i \cdot v_{id}^{old} + c_1 \cdot r_1 \cdot (p_{id} - x_{id}) + c_2 \cdot r_2 \cdot (p_{gd} - x_{id}),$$

v_{id}^{new} is the new velocity in the current iteration t,

v_{id}^{old} is the velocity in the previous iteration $(t$-1),

Step 5. Move the position of each particle i.

$$x_{id}^{new} = x_{id}^{old} + v_{id}^{new},$$

where x_{id}^{new} is the new position in the iteration t,

x_{id}^{old} is the position in the iteration $(t$ - 1).

Step 6. Check the feasibility of the solution and the number of iteration t.
1) The precedence is checked by APM.
2) The design alternative case is checked by DCT.
3) If $(t > T_{Number})$, then go to Step 7, else go to Step 2.

Step 7. Decode the best particle position and interpret the solution.

V. Implementation and Test Results

The presented models were implemented and tested by developing software on a personal computer with a 3.0 GHz CPU and 1 GB memory. The example product A as illustrated in Figure 1 was modeled and tested. The product A is a mobile phone with 13 main components. There are 4 proposed design alternative cases. The APG of the product A is shown in Figure 2. The APM of the product A is listed in the section 3 as described earlier. TheDCT of the product A is shown in Table II. The numerical values of the PSO parameters are tested with an experiment using a Taguchi's orthogonal array to find the best combination of parameters of $T_{Number} = 80$, $P_{Size} = 20$, $w_i = 0.9$, and $(c_1, c_2) = (2, 2)$.

Figure 4 shows that the computation converges after 32 generations with a cost of 258 (unit: dollars) and a computer time of 0.0312 (unit: seconds). The position matrix of the final solution is shown in Table III.

As shown in Table IV, the position matrix of the solution particle is decoded into assembly sequence and design alternative case information. The assembly sequence can be listed as C_{13}-C_{12}-C_{11}-C_8-C_6-C_9-C_{10}-C_7-C_4-C_2-C_5-C_3-C_1.

The information of design alternative cases shows that the components C_{13}-C_{12}-C_{11} are changed in D_2. The components C_8-C_6-C_9-C_{10}-C_7 are changed in D_3. The components C_4-C_2-C_5-C_3 are changed in D_2. Finally, the component C_1 is changed in D_3 to complete the final product. As observed from the illustrative example, it shows that the developed model and algorithm present a feasible and efficient solution method.

VI. Conclusions

In this research, an integrated design evaluation and assembly sequence planning model is presented to perform two tasks, design evaluation and assembly sequence planning. A PSO algorithm is developed for simultaneously optimizing the design of components and the assembly sequence planning. First, an assembly precedence graph (APG) is built. The assembly precedence matrix (APM) is modeled for checking feasibility of the sequences. The information of design alternative cases is modeled in the design alternative case table (DCT). Next, a PSO algorithm is presented to search for the solutions. A new PSO encoding scheme is developed for assembly component sequencing and design evaluation. A particle is represented as a position matrix defined by the number of components and the design alternative cases. The fitness function is formulated by integrating assembly operation cost, assembly tool change cost, assembly setup change cost, and design related cost. The test results show that the PSO method converges fast to reach a minimized cost objective. It can be generally concluded that the developed models and the PSO algorithm are feasible and efficient for solving integrated design evaluation and assembly sequence planning. Future research should be concerned with a detailed analysis of the relationship between design parameters and assembly operations. In addition, it requires an investigation of the complexity to reduce the computational time.

REFERENCES

[1] A. C. Lin, and T. C. Chang, “An integrated approach to automated assembly planning for three-dimensional mechanical products,” *International Journal of Production Research*, vol. 31, no. 5, pp. 1201-1227, May 1993.

[2] T. A. Abdullah, K. Popplewell, and C. J. Page, “A review of the support tools for the process of assembly method selection and assembly planning,” *International Journal of Production Research*, vol. 41, no. 11, pp. 2391–2410, July 2003.

[3] H.Y. Lai, and C.T Huang, “A systematic approach for automatic assembly sequence plan generation,” *International Journal of Advanced Manufacturing Technology*, vol. 24, pp. 752-763, November 2004.

[4] Y. M. Chen, and C. T. Lin, “A particle swarm optimization approach to optimization component placement in printed circuit board assembly,” *International Journal of Advanced Manufacturing Technology*, vol. 35, no. 5-6, pp. 610-620, December 2007.

[5] Q. Su, “Computer aided geometric feasible assembly sequence planning and optimizing,” *International Journal of Advanced Manufacturing Technology*, vol. 33, pp. 48-57, May 2007.

[6] T. Dong, R. Tong, and L. Zhang, “A knowledge-based approach to assembly sequence planning,” *International Journal of Advanced Manufacturing Technology*, vol. 32, pp. 1232-1244, May 2007.

[7] Y.J. Tseng, J.Y. Chen, and F.Y. Huang, “A multi-plant assembly sequence planning model with integrated assembly sequence planning and plant assignment using GA,” *International Journal of Advanced Manufacturing Technology* , vol. 48, no. 1-4, pp. 333-345, April 2010.

[8] S. Jin, W. Cai, X. Lai, and Z. Lin, “Design automation and optimization of assembly sequences for complex mechanical systems,” *International Journal of Advanced Manufacturing Technology*, vol. 48, no. 9-12, pp. 1045-1059, June 2010.

[9] L. Gao, W. Qian, X. Li, and J. F. Wang, “Application of memetic algorithm in assembly sequence planning,” *International Journal of Advanced Manufacturing Technology*, vol. 49, no. 9-12, pp. 1175-1184, August 2010.

[10] J. Kennedy and R. C. Eberhart, “Particle swarm optimization,” in *Proc. 1995 IEEE Int. Conf. Neural Networks*, Piscataway, NJ, pp. 1942-1948.

[11] J. Kennedy and R. C. Eberhart, “A discrete binary version of the particle swarm algorithm,” in *1997 Proc. Int. Conf. Systems, Man and Cybernetics*, Piscataway, NJ, pp. 4104-4109.

[12] A. Banks, J Vincent, and C. Anyakoha, “A review of particle swarm optimization. Part II: hybridization, combinatiorial, multicriteria and constrained optimization, indicative applications,” *Natural Computing*, vol. 7, no. 1, pp. 109-124, March 2008.

Feng-Yi Huang received the MS degree in Industrial Engineering and Management at Yuan Ze University and is now a Ph.D. candidate at the Department of Industrial Engineering and Management, Yuan Ze University.

Yuan-Jye Tseng is a professor in the Department of Industrial Engineering and Management at Yuan Ze University, Taiwan. He received his M.S. and Ph.D. degrees in Industrial Engineering from the Pennsylvania State University, University Park, USA. His research interests include computer-aided design and manufacturing, assembly and disassembly planning, and supply chain system management.

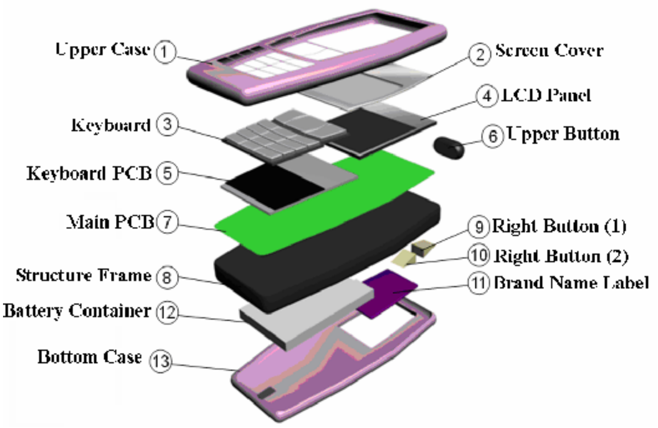


Fig. 1 The example product A is a mobile phone with 13 main components

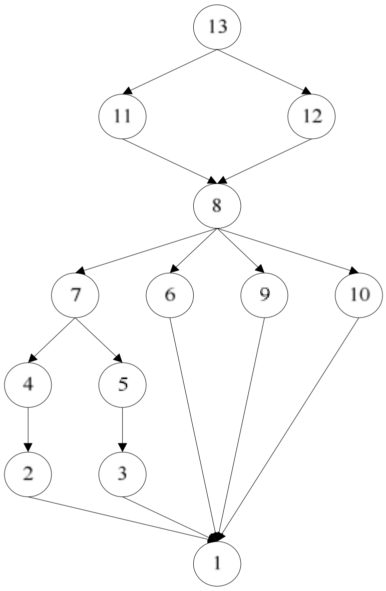


Fig. 2 The APG of the example product A

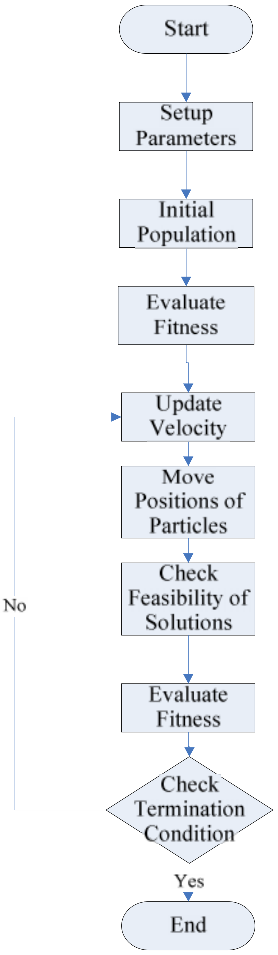


Fig. 3 The flowchart of the PSO algorithm

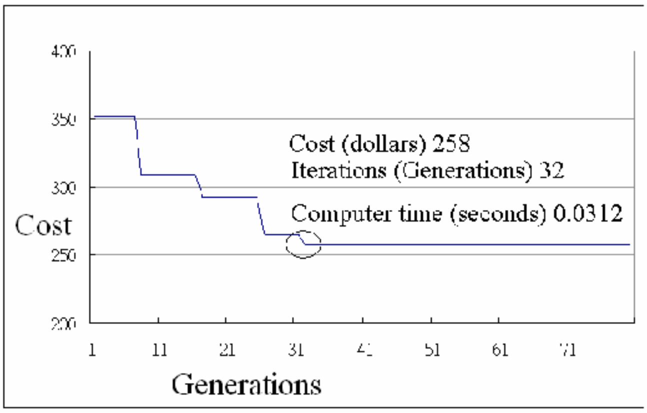


Fig. 4 The test result of the PSO for product A.

TABLE I
DESIGN ALTERNATIVE CASE TABLE (DCT)

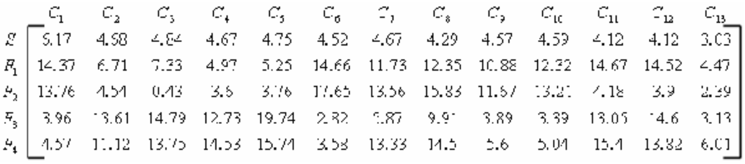
Design d_j / Component c_i	1	2	…	m
1	t_{11}	t_{12}		t_{1m}
2	t_{21}	t_{22}	t_{ij}	t_{2m}
n	t_{n1}	t_{n2}		t_{nm}

t_{ij} = 1 indicates that c_i is changed in design d_j,
t_{ij} = 0 indicates that c_i is not changed in design d_j.

TABLE II
THE DCT OF PRODUCT A.

Design d_j / Component p_i	D_1	D_2	D_3	D_4
1	0	0	1	1
2	1	1	0	0
3	1	1	0	0
4	1	1	0	0
5	1	1	0	0
6	0	0	1	1
7	0	0	1	0
8	0	0	1	0
9	0	0	1	1
10	0	0	1	1
11	0	1	0	0
12	0	1	0	0
13	1	1	1	1

TABLE III
THE SOLUTION POSITION MATRIX FOR PRODUCT A.



	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}
S	5.17	4.58	4.84	4.67	4.75	4.52	4.67	4.29	4.57	4.59	4.12	4.12	3.03
P_1	14.37	6.71	7.33	4.97	5.25	14.66	11.73	12.35	10.88	12.32	14.67	14.52	4.47
P_2	13.76	4.54	0.43	3.5	3.76	17.65	13.56	15.83	11.57	13.21	7.18	3.9	2.39
P_3	3.96	13.61	14.79	12.73	19.74	2.32	1.87	9.91	3.89	3.39	13.05	14.6	3.13
P_4	4.57	11.12	13.75	14.53	15.74	3.58	13.33	14.5	5.6	5.04	15.4	13.82	6.01

TABLE IV
THE SOLUTION OF THE INTEGRATED DESIGN EVALUATION AND ASSEMBLY SEQUENCE PLANNING FOR PRODUCT A.

Assembly sequence	Component	Design alternative case
1	13	D_2
2	12	D_2
3	11	D_2
4	8	D_3
5	6	D_3
6	9	D_3
7	10	D_3
8	7	D_3
9	4	D_2
10	2	D_2
11	5	D_2
12	3	D_2
13	1	D_2