

# GENETIC SIMULATED ANNEALING ALGORITHM-BASED ASSEMBLY SEQUENCE PLANNING

SHAN Hongbo<sup>\*</sup>, LI Shuxia<sup>†</sup>, GONG Degang<sup>‡</sup>, LOU Peng<sup>\*</sup>

College of Mechanical Engineering, Donghua University, No.2999 North Renmin Road,  
Songjiang District, Shanghai 201620, China

E-mail: [hbshan@dhu.edu.cn](mailto:hbshan@dhu.edu.cn)

<sup>†</sup>Department of Management Science and Engineering, East China University of Science & Technology,  
No.130 Meilong Road, Xuhui District, Shanghai 200237, China,

<sup>‡</sup>Dongfeng Automobile Company, No.48 East Checheng Road, Shiyan 442041, China

**Keywords:** Sequence, assembly sequence planning (ASP), genetic algorithm (GA), simulated annealing algorithm (SA), genetic simulated annealing algorithm.

## Abstract

Assembly sequence planning plays an important role in the product development process. It is an important factor that determines quality and cost of the product assembly. Cost in assembly can be reduced by the implementation of generating automatic product assembly sequences, and selecting the optimum sequence in product assembly process. Assembly sequence planning (ASP) is combinatorial problem. Graph-based algorithms are adopted for traditional ASP method. In recent years, some genetic algorithms and simulated annealing algorithms have been used to solve ASP problems, and some achievements are arrived at. However, the two kinds of algorithms have limitations for ASP. GA heavily depends on the choosing original sequence, which can result in early convergence in iterative operation, lower searching efficiency in evolutionary process, and non-optimization of final result for global variable. For simulated annealing algorithms, the principle of generating new sequence is exchanging position of the randomly selected two parts. Obviously, for complex products, a number of non-feasible solutions may appear, and the efficiency is low. In view of these limitations, the approach of combining GA and SA is proposed to build genetic simulated annealing algorithm for the optimization of ASP. In this paper, the following contents are included. Firstly, the relevant researches on assembly sequence planning and the application of GA and SA are summarized. Next, the idea of combining the two algorithms into genetic simulated annealing algorithm is put forward, which aims at improving the efficiency of problem solving. Thirdly, the genetic simulated annealing algorithm for assembly sequence planning is implemented, the method, procedure as well as key techniques of the genetic simulated annealing algorithm are addressed in detail

and the principles for selecting parameters are studied to achieve better performance of the algorithm. Fourthly, a case study is presented to validate the proposed method. In the case, GA, SA and genetic simulated annealing algorithm are applied to ASP respectively, and the results verify the advantages of the genetic simulated annealing algorithm in solving the ASP problem. At last, the work of this paper is summarized and the future researches are given.

## 1 Introduction

As the main process of realizing product function, assembly is an important part of product lifecycle. Related researches [1, 2, 3, 4, 5, 6] show that about one thirds of the human recourses are doing the work related with assembly in modern manufacturing. More than 40% of manufacturing cost is spent on assembly, and 20% to 70% of all the manufacturing work is assembly. Due to the fact that assembly accounts for 45% manufacturing work on average, involves a lot of manual operations and arouses very high cost, enhancing the assembly productivity and reducing the assembly cost surely bring great economic benefit. Assembly Sequence Planning (ASP) is the process of achieving the complete assembly plan under the circumstance that the complete description of the product and related manufacture resource are given, which can do great help in guiding the Design for Assembly (DFM), improving the quality of assembly and reducing assembly cost. Generally, the assembly planning of a product includes generation of the assembly sequence, selection of assembly path, specification of frocks and jigs, assembly time and so on. In this paper, the resolution of ASP will be mainly discussed.

ASP is a combinatorial optimization problem, and the traditional ASP method is graph searching algorithm, which is based on connection graph model. Bourjault [4] builds a two dimension topological connection graph model of product assembly, in which nodes represent parts and arcs represent assembly relation of parts.

Based on the connection assembly graph, Bourjault designs a series of Yes-or-No questions through geometry feasibility analysis of assembly operations, and educates assembly preference relation of parts and the final assembly sequence by man-machine interaction. De<sup>[5,6]</sup> and other researchers' work further reduces the questions that need to be answered by the users and gain the assembly sequence through hierarchical preference assembly model. Niu<sup>[7]</sup> advance a geometrical constraint reasoning-based kinetic space approach by introducing the concept of local freedom cone, in which the automatic reasoning of preference relation among parts and the generation of assembly sequence based on assembly preference graph are studied. Cut set method is a kind of graph theory algorithm which is often used in assembly planning based on disassembly policy. By compacting and combining the product connection graph as well as decomposing the graph repeatedly by cut set method, Homem<sup>[8,9]</sup> find all the undecomposable sub-assemblies, and puts forward a AND/OR graph-based ASP approach. However, as the process of assembly itself is very complex, which often leads to combinatorial explosion problem of complex assembly sequence and the difficulty of evaluating assembly cost, this kind of method is hard to be used for complex product assembly planning.

In order to minimize the assembly cost and further reduce the whole cost of product development, it is necessary to find the optimized assembly sequence or optimize the existed assembly sequence. In consideration of the limitation of classical ASP method, Zadeh<sup>[10]</sup> advances soft computing-based ASP method. As a new computing mode, soft computing provides an efficient way of solving multi-variable, non-linear system. Being different from the traditional hard computing, soft computing does not pursue accurate solutions. Instead, it allows for non-accuracy and uncertainty. It is a target driving mechanism built on the domain knowledge, in which the process of computing, in the long term, emphasizes on moving towards the target, not on how to find the optimization path. As a result, soft computing can reduce computing work and searching time, and is more flexible in solving complex problems. Due to the features of cost efficiency, intelligence and convenience for implementation, soft computing method such as neural network<sup>[11]</sup>, fuzzy logical control<sup>[10]</sup>, genetic algorithm<sup>[12,13]</sup>, simulated annealing algorithm<sup>[14,15]</sup> and so on have been widely used in assembly planning.

Bonneville<sup>[12]</sup> first tried to solve the combinatorial explosion problem of ASP using GA. This first used GA uses a possible assembly sequence given by an experienced assembly expert as the starting point of the evolution computing, and encodes every possible assembly plan as an assembly tree in a chromosome. Besides, as the first attempt, only cross operator and mutation

operator are used, and the selection mechanism and assembly sequence evaluation mechanism are both simple. Although good assembly sequence can be found quickly under certain circumstance, optimized assembly plan may be difficult to find in special situations. Chen<sup>[16,17]</sup> also advanced a genetic ASP method which uses five genetic operators to generate offspring and introduces a double level genetic structure to adjust the control parameters of computing dynamically. De Lit and his colleagues<sup>[18]</sup> present a Ordering Genetic Algorithm to make the assembly planning, in which every gene on the chromosome is corresponded to a assembly operation, and a special mapping mechanism is used to ensure that every offspring generated in the process of genetic operation is feasible and doesn't violate the assembly physical constraint. Dini<sup>[19, 20]</sup> advances a segment code-based genetic algorithm to plan the assembly sequence. Guan<sup>[21]</sup> brings forward a ASP solution algorithm using gene group. Saeid and his colleagues<sup>[14]</sup> plans the assembly sequence using the simulated annealing algorithm, in which the assembly preference relation was first gained according to product assembly model, and then one objective function is constructed using the multi-attribute theory according to the total assembly time and the times of re-directional of the assembly process Hong etc.<sup>[15]</sup> map the assembly constraints and assembly cost as assembly sequence energy function, and reduce the perturbation of assembly sequence energy function by simulated annealing algorithm step by step, until the energy function stayed still and the minimum cost assembly sequence was achieved.

As it shows above, some achievements are got on introducing GA and SA into the research of ASP. The two kinds of algorithms both belong to artificial intelligence optimization algorithm. For their strong ability in solving problems, they are successful and widely used in solving complex engineering optimization problems. They have less limitation to constraint of optimization design target function, meanwhile, they do well in dealing with the practical sick situation such as multi-part extreme points, part non-continuum or non-differentiable. Facing the complex problems such as big space, non-linear planning and entire optimization, the performances of GA and SA are far better than the traditional methods. Although the global optimized solution can be gained by searching randomly in GA and SA, there are still some limitations: GA heavily depended on the choosing original sequence, which can result in early convergence in iterative operation, low searching efficiency in evolutionary process, and non-optimization of final result for global variable. For SA, the principle of generating new sequence is exchanging position of the random selection two parts. Obviously, for complex products, a number of infeasible solutions may appear, and the efficiency is

low. How to combine GA and SA, and construct a better hybrid algorithm have attracted many researchers attentions.

GA makes the sequence colony evolve, SA can search the more optimized solution, so the non-optimization of final result for global variable is efficiently avoided and the global optimized solution can be achieved finally. In addition, in the whole process of optimization, the SA can renew the original sequence repeatedly, which reduce the dependence of GA and enhance the robustness. In view of this, a hybrid evolution algorithm called Genetic Simulated Annealing Algorithm is introduced by combining the advantages of GA and SA in this paper.

## 2 Expression of ASP problem

### 2.1 Encoding schema of assembly sequence

In order to carry on the assembly planning efficiently; the product assembly sequence and related operation should be expressed properly. According to the character of assembly process, in this paper, the chromosomes are constructed with genome, according to the theory that the selection of genetic operator should be corresponded to the information in the genome, the possible individual judgment mechanism is built, which ensures the individual that evolved finally is feasible. To increase the algorithm efficiency, the sorting selection mechanism based on roulette is chosen, meantime, the sampling space is enlarged so that the either the individuals generated by father generation of those generated by filial generation have the chances to be selected to the next generation, which make sure that the best individuals can always be reserved, so, the individual which makes the algorithm converge is the best or the sub-best assembly process. The figure 1 expresses a genome, represents an element and its assembly operation and related information of one element, it is a kind of combine gene coding method which include different kinds of information. In the procedure of genetic computing, the information in the genome may change so that the new individuals are generated.



Figure 1: genome.

The figure 2 expresses a chromosome constructed by genome, represents one sequence of the assembly. The  $N_i$ ,  $T_i$ ,  $D_i$ ,  $L_i$  each represents the serial number, the assembly tools, the possible assembly direction and the type of assembly operation of the element Number  $i$ .

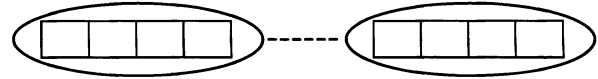


Figure 2: a chromosome (assembly sequence).

In order to simplify the expression of assembly sequence,  $SN=(N, T, D, L)$  is used to express the genome that construct the chromosome, then an assembly sequence can be described as following:  $\{SN_1, SN_2, \dots, SN_n\}$ .  $SN_i$  presents the genome corresponded to the assembly operation,  $n$  stands for the amount of elements of the product, and the corresponded assembly operand. In SA, the type of assembly is the fixed attribute, which will not change during the process of optimization; however, the selection of the assembly tools and the assembly direction is not one and only, even to the same element. They could change during the optimization. The genome code have far more information than the gene code, not only element information included, but also the related operation information, so, the assembly operation process can be expressed very well.

### 2.2 The evaluation of assembly sequence and energy function

In ASP, the assembly sequence needs to be evaluated according to some criteria. The times the assembly direction, assembly tools and type of assembly changed influenced the assistant time. Considering of the encoding schema, assembly tools, assembly direction and type of assembly are selected in this paper as standard. The equation (1) is the assembly sequence energy function;  $E$  stands for the extent of satisfactory to the sequence.

$$E = (2.5 - N_p / N)^g (2N + k_t n_t + k_d n_d + k_l n_l) \quad (1)$$

In the equation,  $N$  stands for the number of element of assembly,  $n_t$ ,  $n_d$ ,  $n_l$  each stands for the changing time of assembly tools, assembly direction and type of assembly;  $k_t$ ,  $k_d$ ,  $k_l$  stands for the weight factor,  $k_t, k_d, k_l \in [0, 1]$ ;  $N_p$ 's value is  $p$ , stands for the assembly constraint relation of the first  $p$  elements of the total  $N$  elements, the number  $p+1$  one does not fulfill it; the value of  $g$  obey the following rules: if the constraint relation is fulfilled, then  $g=1$ , if not, then  $g=-1$ . Obviously, in the process of optimization, the smaller the values of  $n_t$ ,  $n_d$ ,  $n_l$  are, while the bigger the value of  $N_p$  is, the smaller the  $E$  is. When the  $n_t$ ,  $n_d$ ,  $n_l$  are on the lowest point,  $N_p=N$ , and the constraint is fulfilled, the energy is the lowest.

## 3 Genetic simulated annealing algorithm-based ASP optimization

### 3.1 The procedure of genetic simulated annealing algorithm

The principle of algorithm shows as Figure 3.

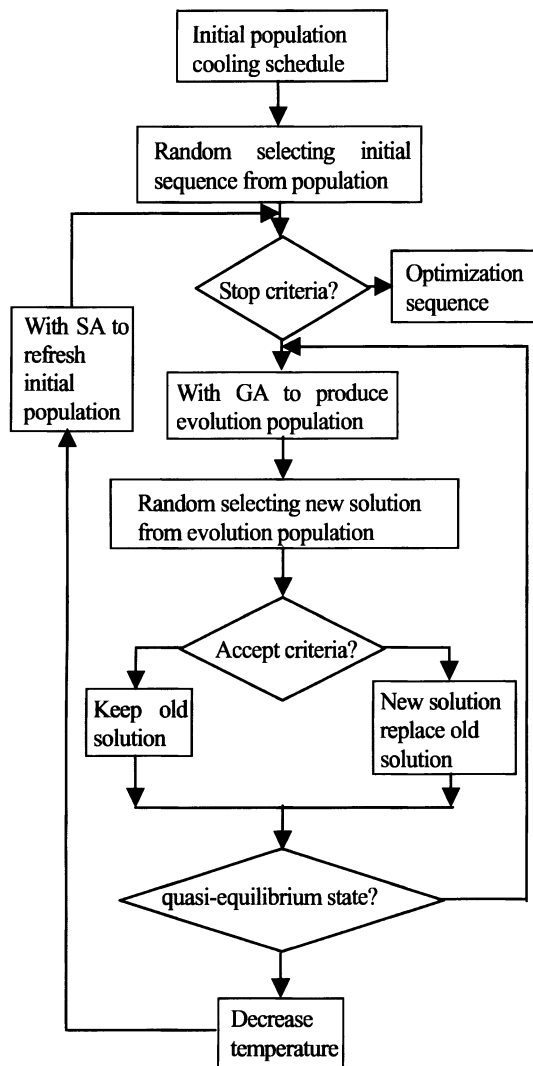


Figure 3: flow chart of genetic simulated annealing algorithm.

**Step 1** Designing the cooling schedule: the original temperature  $T$ , the max cooling schedule, cooling coefficient  $C$  and the max length of Markov

**Step 2** Designing the original sequence and scale  $pop$ , genetic evolution algebra, crossing probability  $p_c$ , and gene variation probability  $p_{ml}$ , and gene inside variation probability  $p_{ms}$

**Step 3** Judging the feasibility of each sequence, using equation (1) computing the  $E$  of each sequence.

**Step 4** Selecting a feasible sequence from the original ones as the most optimized then (old solution), the corresponded energy value is  $E_{old}$ .

**Step 5** If the cooling schedule is smaller than the max one, and the length of *Markov* is smaller than or equal with the max one, then a new sequence is selected randomly as a new solution, and the

geometry feasibility is judged, and the  $E_{new}$  is computed through Equation (1), and then  $\Delta E = E_{new} - E_{old}$ . Computing the possibility  $P = \exp(-\Delta E / KT_k)$  that corresponded to both  $\Delta E < 0$  and  $\Delta E \geq 0$ , then whether to accept the new solution is judged according it; if not, the original sequence should be transferred through method mentioned in section 2.2, and selected by simulated annealing mechanism, so that the original sequence is updated. Then, the cooling schedule should added with 1, the temperature  $T_k$  instead by  $T_k = C * T_k$ , setting the Markov to 0, turning to step 3.

**Step 6** When the new solution is accepted, and the cooling temperature reaches the threshold, the stagnation rule is used to judge it, if accepted, the new solution is outputted, and that is the most optimized assembly sequence.

### 3.2 The generation mechanism of assembly sequence

The generation mechanism of new solution to assembly sequence is important in finding the most optimized sequence. Suppose the first assembly includes  $n$  elements,  $N_i$  is used to stands for the elements number of the number  $i$  element. Next, a 9-element assembly is used to explain the mechanism.

Suppose the assembly sequence is:

$$N_1 \ N_2 \ N_3 \ (N_4 \ N_5 \ N_6 \ N_7) \ N_8 \ N_9$$

$N_4, N_5, N_6, N_7$  consist a sub-sequence.

**Mechanism 1: the shifting of sub-sequence**

The elements inside the sub-sequence do not change their sequence, but the position of sub-sequence changes, the new sequence is:

$$N_1 \ (N_4 \ N_5 \ N_6 \ N_7) \ N_2 \ N_3 \ N_8 \ N_9$$

**Mechanism 2: the antitone of sub-sequence**

The sequence of sub-sequence is reversed, without changing the position.

$$N_1 \ N_2 \ N_3 \ (N_7 \ N_6 \ N_5 \ N_4) \ N_8 \ N_9$$

**Mechanism 3: sub-sequence shift and antitone**

The sequence of the sub-sequence is reversed first, then the position is changed.

$$N_1 \ (N_7 \ N_6 \ N_5 \ N_4) \ N_2 \ N_3 \ N_8 \ N_9$$

Compared with the other two mechanisms, the mechanism 2 does more help in enhancing the efficiency.

## 4 Case study and analysis

### 4.1 Case description

The vise, as follow in figure 4, is made of 12 parts, when the small attachments are simplified.

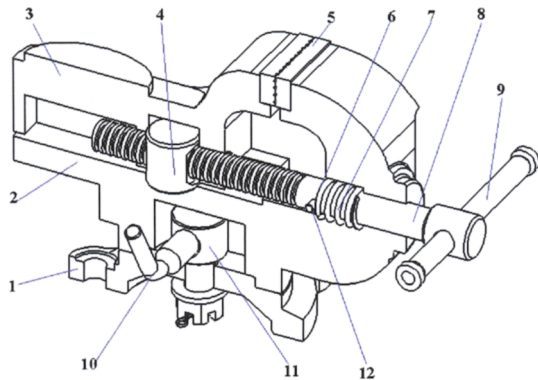


Figure 4: the assembly view graph of vise.

The meanings of numbers correspond to figure 4 as follows.

No.	Name
1	base
2	fixed body
3	flexible body
4	orientation block
5	vice grip
6	spring damper
7	spring
8	tommy screw
9	hand knob
10	eccentric rod
11	locking screw
12	locking stud

Table 1: part's name corresponding to Figure 4.

#### 4.2 Algorithm assumptions

Four-element group is used to represent the assembly sequence of figure 1 and figure 2. Every Four-element group indicates the assembly manipulation process of part. Every element of four-element group is expressed by digital which represents different aspects. The first element represents the part number. The second element represents the assembly tools which correspond to usable assembly tool set. The third element represents assembly direction which manipulates part by the assembly tools. At here, 0, 1, 2, 3, 4, 5 individually indicates the +x direction, -x direction, +y direction, -y direction, +z direction, -z direction. The four elements represent the type of assembly operation. Furthermore, weight factor of assembly tool, assembly direction, number of assembly type transformation is set 1 at function1 in order to simplification, that is  $k_t=k_d=k_f=1$ .

Based on the above, GA, SA and Genetic simulated annealing algorithm, is individually used to solute the ASP problem. In order

to compare the effect of three algorithms, the same controlling parameters are assigned the same values. For example, the original sequence and scale  $pop$  is assigned 24, crossing probability of genome  $p_c$  is assigned 0.5, variation probability of genome  $p_m$  is assigned 0.1, assembly tool variation probability of genome  $p_{mt}$  is assigned 0.1, and direction variation probability of genome  $p_{nd}$  is assigned 0.1.

#### 4.3 GA algorithm for ASP problem

With GA algorithm, the initial genetic generation is 500. Through the vast trial calculation of 12000 generation, the optimization assembly sequence is as follows:

(1 1 5 1)-(2 1 4 1)-(3 2 4 1)-(8 2 3 2)-(12 2 3 2)-(6 2 2 2)-(4 1 2 2)-(5 4 2 2)-(11 1 2 2)-(7 3 2 2)-(9 2 4 1)-(10 3 4 1)

The value of this assembly sequence energy function  $E$  is 37.823. By practical verification, this sequence is feasible, but repetition rate is higher and efficiency is lower.

#### 4.4 SA algorithm for ASP problem

With SA algorithm, the cooling schedule parameter must be set. The initial annealing temperature  $T_0$  is 1000, the length of Markov chain  $L_k$  is 500, the max cooling schedule is 150. Through the vast trial calculation, the optimization assembly sequence is follow:

(7 1 0 2)-(11 1 0 2)-(8 2 0 2)-(1 3 0 2)-(6 5 0 2)-(4 1 0 2)-(3 1 0 1)-(5 2 0 2)-(9 1 0 1)-(12 3 0 1)-(10 1 0 1)-(1 1 5 1)

The value of this assembly sequence energy function  $E$  is 56.137. By practical verification, this sequence is infeasible.

#### 4.5 Genetic simulated annealing algorithm for ASP problem

The initial population of this algorithm is identical to GA algorithm for the same ASP problem. With this algorithm, the initial genetic generation is 500, the max cooling schedule is 65. In order to compare the influence of algorithm efficiency and optimization solution from different controlling parameters, three group cooling schedule is assigned as following:

- ① The initial annealing temperature  $T_0$  is 50, temperature cooling factor  $C$  is 0.65, the max length of Markov chain  $L_k$  is 15;
- ② The initial annealing temperature  $T_0$  is 100, temperature cooling factor  $C$  is 0.75, the max length of Markov chain  $L_k$  is 40;
- ③ The initial annealing temperature  $T_0$  is 150, temperature cooling factor  $C$  is 0.85, the max length of Markov chain  $L_k$  is 80;

Through the vast trial calculation, the optimization assembly sequence is follow:

(1 1 4 2)-(10 1 4 1)-(3 1 4 1)-(9 1 3 2)-(12 2 3 2)-(6 2 2 2)-(7 2 3 2)-(8 1 2 2)-(4 1 4 2)-(11 1 2 1)-(5 1 4 1)-(2 2 4 1)

The value of this assembly sequence energy function  $E$  is 21.967.

By practical verification, this sequence is feasible and the better solution that approximately matches the actual assembly sequence.

#### 4.6 Results analysis

From the above several parts 4.1, 4.2, 4.3, 4.4, 4.5, the comparison results can be arrived from three algorithms under the same outer condition. The results of ASP optimization performance evaluation are listed in table 2.

Algorithm Item	CA	SA	Genetic simulated annealing algorithm
The times of assembly direction	1	1	0
The times of transformation Assembly tool	1	3	1
The times of transformation the type of assembly operation	1	1	2
Sequence feasible or not	Y	N	N
The value of assembly sequence energy function E	37.823	56.137	21.967
Iterative generation $n$	12000	60000	64000
Calculation time $t$	78s	113s	167s

Table 2. The results of ASP optimization performance evaluation based on GA, SA, and Genetic simulated annealing algorithm

From the Table 2, the conclusions can be arrived at. By GA algorithm, the better assembly sequence can be obtained within shorter calculation time and relatively fewer Iterative generations. Comparatively, SA algorithm not only spent longer calculation time, but also the quality of solution is evenly no better than initial sequence. SA algorithm is not convergent when the longer calculation process is over, so its solution is infeasible. Although Genetic simulated annealing algorithm also spends longer calculation time and needs much Iterative generations, the better assembly sequence can be achieved. Obviously, the effect of GA algorithm and Genetic simulated annealing algorithm for ASP problem is better than SA algorithm at this point. Through analyzing the optimization sequence solution of GA algorithm and Genetic simulated annealing algorithm, GA algorithm is limited by initial population scale and genetic generation, and its solution is often converged into local optimization sequence, not global.

The solving process of Genetic simulated annealing algorithm for ASP problem runs much iterative computation and generations. If its ergodic process is longer, the approximate error is smaller between solution sequence and ideal sequence. This is the reason why the solution of Genetic simulated annealing algorithm is the most probable global optimized sequence than GA algorithm. Furthermore, if the value of cooling schedule were raised, the cooling process were extended, the initial annealing temperature were raised, the max length of Markov chain were also enhanced, the time of achievement quasi-equilibrium state will become longer, the more feasible ergodic search process may be carried out, the quality of assembly sequence optimization solution can also be increased.

#### 5 Conclusions and the future work

The limitations can be effectively overcome by combining with merits of GA and SA. For example, in a given domain, the deterioration solution can run. This Genetic simulated annealing algorithm has better optimization performance and robustness that can be concluded from the following two aspects. The first is that the degree of dependence on the initial assembly sequence is decreased. The optimization assembly sequence still can be obtained even if the assembly sequences of initial population are infeasible. The second is that this algorithm has better convergence preference. One or several better assembly sequence can be searched and found in relatively shorter time.

However, this Genetic simulated annealing algorithm does have shortcomings. The effect of optimization is influenced by choosing initial controlling parameters, and the optimization result maybe not the global optimal solution. Furthermore, a large number of sequences, with the same or similar E value, may be generated, which can also decrease the efficiency of algorithm. It still requires subjective modifications and further optimization provided by human designers. Further efforts are required for the improvement to these problems.

#### Acknowledgments

This research was supported by grant from the National Natural Science Foundation of China (No: 59990470) and Donghua University Science fund (No: 103100044035). The authors wish to thank editors, our PhD research team and the reviewers of this paper for their valuable suggestions and comments.

#### References

- [1] G. Boothroyd, P. D. Ewhurst, W. Knight, *Product Design for Manufacture and Assembly*, New York: Marcel Dekker, 1994.
- [2] X. F. Zha, "Integrated knowledge-based assembly sequence planning", *International Journal of Advanced Manufacturing Technology*, **14**, 50-64, 1998.
- [3] W. G. Hsu, C. S. Lee, S. F. Su, "Feedback approach to design for assembly by evaluation of assembly plan", *Computer Aided Design*, **25**, 395-410, 1993.
- [4] A. Bourjault, A. Lhote, "Modelling an assembly process", *IEEE Transaction on Automation of Manufacturing Industry*, **20**, 183-198, 1986.
- [5] T. L. De Fazio, D. E. Whitney, "Simplified generation of all mechanical assembly sequences", *IEEE Transaction on Robotics and Automation*, **3**, 640-658, 1987.
- [6] D. F. Baldwin, T. E. Abell, T. L. De Fazio, D. E. Whitney, "An integrated computer aid for generating and evaluating assembly sequences for mechanical products", *IEEE Transactions on Robotics and Automation*, **7**, 78-94, 1991.
- [7] X. W. Niu, H. Ding, Y. L. Xiong, "An integrated approach to generation of precedence relations and precedence graphs for assembly sequence planning", *Chinese Journal of Mechanical Engineering*, **15**, 19-31, 2002.
- [8] L. S. Homem De Mello, A. C. Sanderson, "A correct and complete algorithm for the generation of mechanical assembly sequences", *IEEE Transactions on Robotics and Automation*, **7**, 228-240, 1991.
- [9] L. S. Homem De Mello, A. C. Sanderson, "And/or graph representation of assembly plans", *IEEE Transactions on Robotics and Automation*, **6**, 188-199, 1990.
- [10] L. A. Zadeh, "Fuzzy logic, neural networks, and soft computing", *Communications of the ACM*, **37**, 77-84, 1994.
- [11] D. S. Hong, H. S. Cho, "Neural-network based computational scheme for generating optimized robotic assembly sequences", *Engineering Applications of Artificial Intelligence*, **8**, 129-145, 1995.
- [12] F. C. Bonneville, Perrard, J. M. Henrioud, "A genetic algorithm to generate and evaluate assembly plans", *Proceedings of INRIA/IEEE Symposium on Emerging Technology and Factory Automation*, **2**, 231-239, 1995.
- [13] M. F. Sebaaly, H. Fujimoto, "A genetic planner for assembly automation", *Proceedings of the IEEE Conference on Evolutionary Computation*, 1996.
- [14] M. Saeid, "Multi-criteria assembly sequencing", *Computers Industrial Engineering*, **32**, 743-751, 1997.
- [15] D. S. Hong, H. S. Cho, "Generation of robotic assembly sequences using a simulated annealing", *Proceedings of the IEEE/RSJ International conference on Intelligent Robotic and Systems*, 1999.
- [16] S. F. Chen, "Assembly planning—a genetic approach", *ASME Proceedings of Design Engineering Technical Conferences*, DAC-5798, 1999.
- [17] S. F. Chen, Y. J. Liu, "A multi-level genetic assembly planner", *ASME Proceedings of Design Engineering Technical Conferences*, DAC-14246, 2000.
- [18] P. De lit, P. Latinne, B. Rekiek, et al, "Assembly planning with an ordering genetic algorithm", *International Journal of Production Research*, **39**, 3 623–3 640, 2001.
- [19] G. Dini, F. Failli, B. Lazzerini, et al, "Generation of optimized assembly sequences using genetic algorithms", *Annals of the CIRP*, **48**, 17-20, 1999.
- [20] B. Lazzerini, G. Dini, "Assembly planning based on genetic algorithm", *The 18th International Conference of the North American on IEEE Fuzzy Information Processing Society, NAFIPS*, 1999.
- [21] Q. Guan, J. H. Liu, Y. F. Zhong, "A concurrent hierarchical evolution approach to assembly process planning", *International Journal of Production Research*, **40**, 183-187, 2002.