Assembly Sequence Planning Based on Hybrid Artificial Bee Colony Algorithm

Wenbing Yuan, Liang Chang, Manli Zhu, Tianlong Gu

Assembly Sequence Planning Based on Hybrid Artificial Bee Colony Algorithm

Wenbing Yuan, Liang Chang, Manli Zhu, Tianlong Gu

Guangxi Key Laboratory of Trusted Software
Guilin University of Electronic Technology
Guilin 541004, China
changl@guet.edu.cn

Abstract. Intelligent algorithm provides a promising approach for solving theAssembly Sequence Planning (ASP) problem on complex products, but there is still challengein finding best solutions efficiently.In this paper, the artificial bee colony algorithmis modified to deal with this challenge. The algorithm is modified from four aspects. First, for the phase that employedbeeworks, a simulated annealing operatoris introduced to enrich the diversity of nectar sources and to enhance the local searching ability. Secondly, in order to prevent the swarm from falling into local optimal solutions quickly, atournament selectionmechanism is introduced for the onlooker beesto choose the food source.Thirdly, for the phase that scout beeworks, alearning mechanism is introducedto improve the quality of new generated food sources and to increase the convergence speed of the algorithm. Finally, afitness functionbased on the evaluation indexes of assemblies is proposedto evaluate and select nectar sources. The experimental results show that the modified algorithm is effective and efficient for the ASP problem.

Keywords: artificial bee colony algorithm, simulated annealing operators, assembly sequence planning, multi-objective optimization.

1 Introduction

Assembly sequence planning (ASP) aims to find a proper sequence of assembly operations under some operational constraints and precedence constraints.ASP is an important manufacturing process in that the quality of assembly has a direct effect on the performance of product. According to a statistical report, a good assembly sequence can reduce costs of manufacturing about 20% - 40%, and can increase the productivity about 100% - 200%.

Traditional methods of assembly sequence planning are based on cut-sets [1] or assembly knowledge representation [2]. A shortcoming of these methods is that they have to face the combinatorial state explosion problem caused by the increasing of the number of parts [3]. During the past 20 years, many researchers tried to apply intelligent algorithms to solve the ASP problem[4-9]and got many good results.

Artificial bee colony (ABC) algorithm is an artificial intelligence algorithm that was proposed for the multi-variable and multi-modal optimization of continuous functions. It simulates the emergent intelligent behaviour of foraging bees in three phases. At the initial phase, the scout bees start to explore the environment randomly in order to find a nectar source. After finding a nectar source, a bee becomes an employed bee and starts to exploit the discovered source; it returns to the hive with the nectar and share information about the source site by performing a dance on the dance area. Onlooker bees waiting in the hive watch the dances and choose a nectar source through the information shared by

employed bees by using a wheel selection. If a source is exhausted, then the employed bees that are working in that source become scouts again and start to randomly search for new sources.Compared with the genetic algorithm [4], the firely algorithm [6] and the particle swarm optimization algorithm [7], the ABC algorithm is more competitive in theability of global searching and the speed of convergence. It has been usedto solve the global optimization problem [10],global numerical optimization[11], real-parameter optimization[12], shop scheduling problem[13]and disassembly planning problem[14].

In this paper, we apply the ABC algorithm to solve the assembly sequence planning problem.In order to improve the ability of searching optimal solution; we embed the simulated annealing algorithm and a local searching algorithm into the ABC algorithm. At the same time, we use a tournament selection mechanism to replace the wheel selection mechanism for escaping from local optimum.Furthermore, we apply a cooperative learning mechanism in the scout bee phase to improve the convergence speed, and propose a suitable fitness function to evaluate and filter nectar sources. In the rest of this paper, we call the improved algorithm as hybrid artificial bee colony algorithm(HABC).

2 Optimization model of assembly sequences

In this paper, we use the following assumptions to simplify the optimization model of assembly sequences. (1) All the parts are rigid. In another word, no part is deformable in the assembly process. (2) The assembly directions are restricted to $\pm$X, $\pm$Y, $\pm$Z direction in the three orthogonal coordinate axes. (3) Along each direction only one part is assembled,and in each assembly operation process only one tool isused.

An evaluation index for the fitness function is proposed to evaluate and filter the nectar sources: geometric feasibility, assembly stability, the number of parts violating local assembly precedence, and changes of assembly directions, tools and connections. The fitness function is applied to find the optimal nectar sourceswhichare the optimal or near-optimal assembly sequences.

2.1 Geometric feasibility

An assembly sequence satisfying geometric feasibility can ensure that the movement path of part is collision-free. In this paper, we use Cartesian 6 coordinate to represent the assembly directions$d_k = \{\pm x, \pm y, \pm z\}$. The values of 0 and 1 represent the interference between the parts. The I_{ijd_k}represents the interference in thed_k direction between p_i andp_j, which is shown in formula (1);

$$I_{ijd_k} = \begin{cases} 0, \text{there is no interference in the } d_k \text{ direction between } p_i \text{and } p_j \\ 1, \text{there is an interference in the } d_k \text{ direction between } p_i \text{and } p_j \end{cases} \quad (1)$$

We assume an assembly sequence $\mathrm{AS} = \{p_1, p_2, \cdots, p_n\}$, when the subsequence $AS_1 = \{p_1, p_2, \cdots, p_m\}$was assembled and p_i will be assembled, we represent the sum total of interference between p_i and all parts of AS_1 in the d_k direction with $S_k(p_i)(\mathrm{k} = 1, 2, \ldots, 6)$ which is shown in formula (2):

$$S_k(p_i) = \sum_{j=1}^{m} I_{p_i p_j d_k} \quad (2)$$

$S_k(p_i) = 0$ indicatesthat the partp_i can be assembled in the $d_k$direction; otherwise it cannot. Then we can get a set of assembly directions of the partp_i: $\mathrm{DC}(p_i) = \{d_k \mid S_k(p_i) = 0\}$.For any part p_i, if $\mathrm{DC}(p_i) \neq \emptyset$, AS is a geometric feasible assembly sequence, infeasible otherwise. We represent the number of interference of AS with n_f, that is the number of $\mathrm{DC}(p_i) = \emptyset$.

2.2 Assembly stability

Parts or subassembly may be unstable, for its own gravity in the actual assembly process. Auxiliary devices are needed to maintain the stability of parts in the assembly process once some parts are unstable. For this reason, we consider the assembly stability as an evaluation index.

The tight fit, interference fit, screw, rivet and welding are the strong assembly connections which are the stableconnections. The clearance fit, sticking and attachment are the weak connections;the surface mating and other fits without force are the unstable connections[15].The values of s_{ij} represent the assembly connections between the $p_i$and p_j: $s_{ij} = 2$ indicates that they are joined by strong connections; $s_{ij} = 1$ indicates that they are joined by weak connections; $s_{ij} = 0$ indicates that they are joined by surface mating without force. Then the assembly stability can be represented with the stability matrix which is shown in formula (3):

$$\mathrm{SM} = \left[s_{ij}\right]_{n\times n} \tag{3}$$

It is a stable connection if $s_{ij} = 2$, otherwise not.

Given that the temporary stable subassembly is represented as $AS_1 = \{p_1, p_2, \cdots, p_m\}$ (m is the number of parts which have been assembled), the part p_i is to be assembled in the next step. The assembly stability of the part p_i with AS_1 can be concluded depending on the stability matrix SM. The correlative stability value $s_{ij}\,(1 \le \mathrm{j} \le \mathrm{m})$ can be inferred as $\{s_{i1}, s_{i2}, \ldots, s_{im}\}$. If$s_{ij} = 2\ (1 \le \mathrm{j} \le \mathrm{m})$, the part $p_i$and AS_1will be stable after they are assembled. Otherwise, the part $p_i$needs auxiliary devices in the assembly process. The assembly stability of each part will be checked and the number of parts n_s that make the subassembly self-stable in the assembly sequence can be computed.

2.3 The number of parts violating local assembly precedence

The reasonable assembly sequences not only satisfy the geometrical feasibility, but also the constraint relationships between parts. Therefore, the assembly precedence relationships of parts must be takeninto consideration. For the complex assemblies, there will be too many possible assembly sequences, but most of them are actually not ensured to be correct. Even if the assembly sequences satisfy the geometrical feasibility, some parts maybe violate the local assembly precedence still. So the violating assembly precedence is another considerable index to evaluate assembly sequences. To an assembly sequence $\mathrm{AS} = \{p_1, p_2, \cdots, p_n\}$, the number of parts violating local assembly precedence n_p can be obtained easily by assembly precedence matrix(APM) described in Section 3.2.

2.4 Changes of assembly directions, tools andconnections

The changes of the assembly directions, the assembly tools and the assembly connection types in the assembly process have a great effect on the assembly efficiency and cost.To improve the assembly efficiency and reduce the assembly cost, the parts assembled in the assembly direction, and the assembly connection types and assembly tools are suggested to be assembled in groups. The assembly directions of parts can be derived from the assembly interference matrix. Once the assembly sequence is confirmed, the number of change n_d of the assembly directions of all parts can be counted.

The assembly tools used to assemble one part may be multiple. The selected tools to assemble the current part should be the same as those used to assemble the former part so that the time for changing the tools can be shortened. All the assembly tools can be represented by a tool matrix TM = $[t_{ij}]_{\mathrm{nxm}}$, here nis the number of parts andmis the number of practicable tools to assemble the corresponding part. After the optimal or near-optimal assembly sequences have been generated, the corresponding tools are also confirmed at the same time and the number of change n_t of the assembly tools can be obtained.

It is assumed that the connection type for assembling each part is unique and the connection types of parts are determined in the design process. The assembly connections to assemble the parts can be represented as a vector of connection CM = $[c_{ij}]_{\text{nx1}}$. When the assembly sequences have been ascertained, the number of change$n_t$of the assembly connection types can also be acquired.

2.5 Fitness function

Each evaluation indexhas different effects in the assembly process, so it is reasonable to put different weights on each index. Then the fitness function can be defined as the formula(4) and (5) by using the different weights:

$$f_1 = c_s n_s + c_p n_p + c_d n_d + c_t n_t + c_c n_c \quad (4)$$

$$f_2 = c_f n_f (5)$$

c_f, c_s, c_p, c_d, c_t and c_c are the weights of the evaluation indicesrespectively.

Fitness function is the metric of evaluating assembly sequences. Similarly, it is the metric of evaluating nectar sources in the search process. The smaller the value, the better the nectar source quality, and the assembly sequence is better correspondingly. If an assembly sequence satisfies the geometrical feasibility, formula (4) will be used to calculate the assembly cost; otherwise formula (4) and (5) will be used.

3 Assembly sequences of HABC

The basic ABC algorithm was first usedto solve optimization problems of continuous functions. A hybrid ABC algorithm is proposed to solve multi-objective optimization problems in the assembly sequence planning. HABC algorithm makes full use of swarm intelligence sharing information to accelerate search efficiency.

3.1 Nectar source code

In HABC algorithm, nectar position can be represented with the decimal sequence. Each nectar source is an assembly sequence. Profitability of nectar sources is used to measure the quality of assembly sequences. Each number and its location in the sequence represent the number and the assembly order of parts respectively. Suppose that a product is made up of 9 parts, the coding scheme can be described in Fig.1:

Assembly sequence:	3	6	8	1	4	9	5	2	7
Sequence order:	1	2	3	4	5	6	7	8	9

Fig.1.*Nectar source code*

If the assembly sequence of Fig.1 is a feasible sequence, the parts are assembled orderly from 3 to 7. Its quality is evaluated by the fitness function.

3.2 Initial nectar source

If the initial nectar sources are produced random completely, it will generate large amount of infeasible sequences which can affect the efficiency of algorithm. If all initial nectar sources are feasible sequences, it will affect the diversity of nectar sources. To guarantee an initial population with certain quality and diversity, a portion of nectar sources are generated by using some priority rules whereas the rest are produced randomly.

The assembly precedence matrix (APM) is used to generate partial sequences as the initialization

nectar sources in a generator(Fig.5)assembly [16]. A value of $b_{ij} = 1$ represents that $partc_i$ must be assembled before $partc_j$; a value of $b_{ij} = 0$ represents that there is no precedence between the two parts $c_i$and c_j. If $i = j$, we set $b_{ij} = 0$. APM for a generator is shown as follows：

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
5	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1
7	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
10	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
11	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig.2.*Assembly precedence matrixfor a generator*

3.3 Employed bee phase

In the basic ABC algorithm, for eachnectar source, there is only one employed bee. Employed bees use a greedy selectionthrough its observed information to search for a better nectar source in the neighborhood of the present nectar source. To enrich the neighborhood structure and diversify of the population, simulated annealing algorithm are utilized to generate neighboring nectar sources for the employed bees[17].

The above method for generating new nectar sources follows the assembly precedence relations, thus it can improve the quality of nectar sources continually in the evolution process. ABC algorithm has strong exploration ability and weak exploitation ability. Therefore, a local search algorithm is presented in Section 3.6 in order to balance the global exploration and local exploitation of HABC.The selection of nectar sources carries out a greedy selection which is the same as in the basic ABC algorithm: if new nectar source is superior to its current nectar source in terms of profitability, the employed bee memorizes the new position and forgets the old one; otherwise, the previous position will be kept in memory.

3.4 Onlooker bee phase

Onlooker bees wait in the hive and select a nectar source to exploit with a certain probability based on the information shared by the employed bees. This selection is called the wheel selection which is easy to fall into local optimum. It maybe causes all the onlooker bees fly to the same nectar source, while the others cannot recruit onlooker bees. Instead of the wheel selection, we use the tournament selection due to its simplicity and ability to escape from local optimum.

In the tournament selection, an onlooker bee selects a nectar source in such a way that two nectar sources are picked randomly from the population and the better one is chosen. It means that half of the nectar sources will be exploited by onlooker bees and there is a greater probability to find the optimal nectar sources(the optimal or near-optimal assembly sequences). In addition, the onlooker utilizes the greedy selection as used by the employed bee. Then all the local nectar sources approach the local optimum, and this could help find better nectar sources. There are three options for the employed bee whichhas not recruited onlooker bees:

(a) It can return to the nectar source and continue to exploiting it.

(b) It can be an onlooker, waiting to be recruited by other employed bees in the hive.

(c) It can be a scout and starts searching around the hive for a new nectar source.

To guarantee the diversity of nectar sources and keep the nectar sources quantity constant at each iteration of the algorithm, we just consider the first and the third behaviors for such a bee.

3.5 Scout bee phase

In the basic ABC algorithm, if a nectar source cannot be further improved through a predetermined number of trials limit, then the nectar source is assumed to be abandoned, and the corresponding employed bee becomes a scout bee. To make full use of the swarm coordination of information sharing, we suppose that the scout bees learn from the current optimal (minimum fitness value) nectar sources when generating a new one. Since the search space around the best nectar source could be the most promising region, the new nectar source approaches the local optimum afterevery iteration of algorithm. This will increase the search efficiency and accelerate the converging speed in later convergence phase.

To avoid the algorithm trap into a local optimum, the scout bee first generates a nectar source randomly; then it learns from the current optimal nectar source with the probability P; finally, it can get a new nectar source. Next, the scout performs the function of the employed bee. The process of a scout generating a nectar source isthat the new nectar source inherit the parts of the optimal nectar source with a random probability r $< P$, the rest inherit random sequences in order.

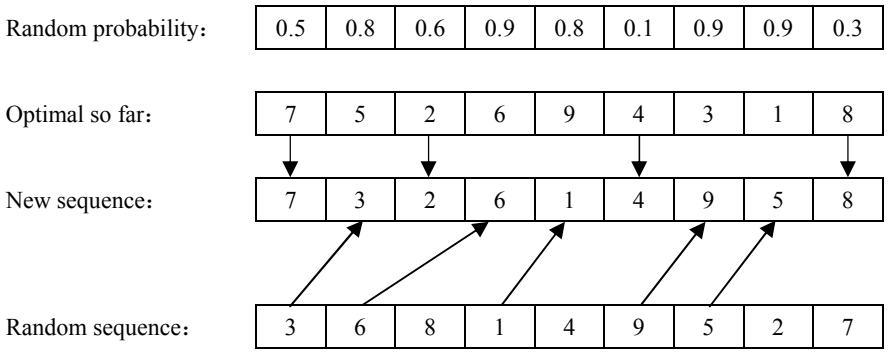

Fig.3.*The process of scout bee generate nectar source* $(P = 0.7)$

3.6 A local search algorithm

In order to enhance the exploitation capability of the algorithm, a simple local search is embedded in the proposed HABC algorithm. The local search is based on the insert and swap operator.

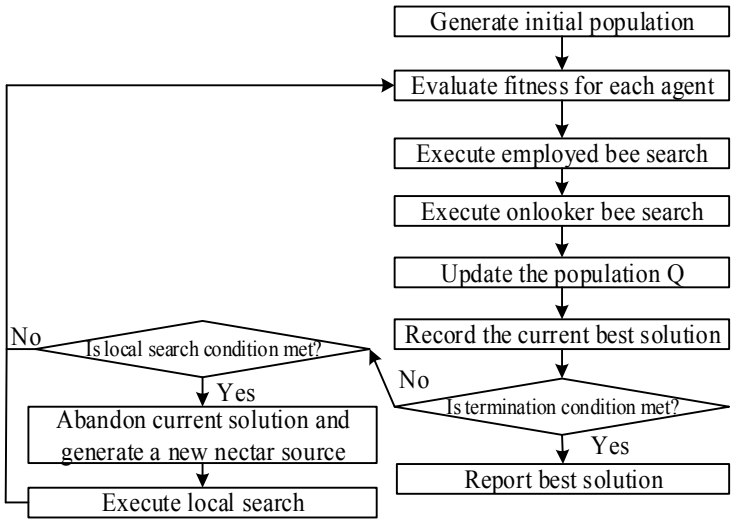

Fig.4. *The flow chart of HABC*

3.7 Computational procedure of the HABC algorithm

The complete computational procedure is described in Fig.4. Some parameters are need to set first: population size SN; learning rate P; local search probability P_L; the threshold value l_{max} of local search; round T_{max} of iteration. Then initial the population $Q = \{AS_1, AS_2, \ldots, AS_{SN}\}$, and half of Q are generated with the constraint APM and the rest are generated randomly. The employed bee search, onlooker bee search and scout bee phase which is also called local search and depend on the local search conditionl_{max} are then executed, and Q is updated by previous steps and the current best solution is recorded. The algorithm is stopped based on the termination condition, that is, the round$T_{max}$of iteration. Finally, report the best solution.

4 Experimental results and analysis

The HABC algorithm for assembly sequence planning and optimization has been programmed with Visual C++ 6.0 language on a PC with 64-bit system of Windows 7 and 8G memory. By comparing the HABC algorithm with the CPSO algorithm, firefly algorithm and improved firefly algorithm[18], the validity of this method is confirmed with actual assembly example.

4.1 Simulation example

The exploded solid model of a generator assembly is shown in Fig.5. The generator comprises 15 parts which are marked with numbers orderly. The assembly directions of all the parts are along the X and Y axes. Part 7 is viewed as the base part which has the most assembly relations with the other parts. Suppose that the assembly tool and assembly connector to assemble the base part 7 are T1 and C1respectively. The assembly interference matrix, local preference matrix, stability matrix, connection matrix and tool matrix are given based on the assembly structure and assembly process. In this case, the proper tools and the connections for assembling each part are illustrated in Table 1, the assembly interference matrix are presented in Section 3.2 and the other assembly process constraints are omitted.

Table 1.*Assembly tools and the connections for the parts*

Part number	Assembly tool	Connection type
1	T_1, T_2, T_4	C_2
2	T_1, T_4, T_5	C_1
3	T_2, T_3, T_4	C_3
4	T_1, T_2, T_5	C_1
5	T_1, T_2, T_4	C_1
6	T_1, T_2, T_3	C_1
7	T_1	C_1
8	T_3, T_4, T_5	C_3
9	T_1, T_2, T_4	C_1
10	T_1, T_2, T_5	C_1
11	T_2, T_3, T_4	C_3
12	T_1, T_4, T_5	C_1
13	T_1, T_2, T_4	C_2
14	T_3, T_4, T_6	C_3
15	T_4, T_5, T_6	C_3

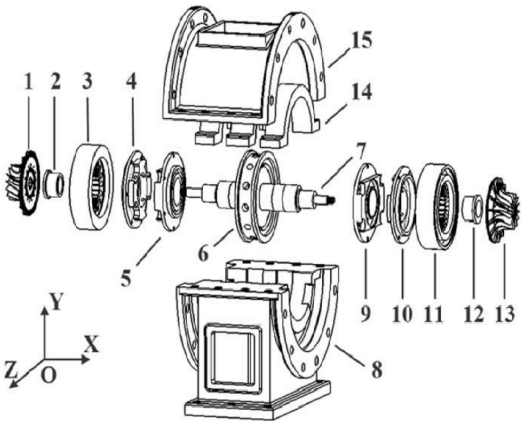

Fig.5.*The assembly of a generator*

4.2 The comparison between algorithms

By comparing the HABC algorithm with the CPSO algorithm, firefly algorithm and improved firefly algorithm, the validity of this method is confirmed. Under the same constraints, the results are shown in Table 2 and Table 3 with the same indices and weights respectively. The other parameters of CPSO algorithm are assigned as literature [7], and the other parameters of firefly algorithm and improved firefly algorithm are as literature [6]. Since CPSO algorithm can obtain sub-optimal sequences only in larger cycles, we present results with 500 cycles. All the parameters of the assembly cost function and the HPSO algorithm are assigned as: $c_f = 50$, $c_s = 0.15$, $c_p = 0.15$, $c_d = 0.40$, $c_t = 0.15$, $c_c = 0.15$, P = 0.7, $P_L = 0.8$, $l_{max} = 5$.

Table 2.*Comparison of the generated sequences of the four algorithms*

Results		Sequence	n_f	n_s	n_p	n_d	n_t	n_c	f
HABC	1	7 6 5 4 9 10 11 12 13 3 2 1 14 15 8 X X X X −X −X −X −X −X X X X X X Y T_1 T_1 T_1 T_1 T_1 T_4 T_4 T_4 T_4 T_4 T_4 T_4 T_4 T_4 C_1 C_1 C_1 C_1 C_1 C_3 C_1 C_2 C_3 C_1 C_2 C_3 C_3 C_3	0	0	0	3	1	7	2.40
	2	7 6 9 10 5 4 3 2 1 11 12 13 14 15 8 −X −X −X −X X X X X X −X −X −X −X −X Y T_1 T_1 T_1 T_1 T_1 T_4 T_4 T_4 T_4 T_4 T_4 T_4 T_4 T_4 C_1 C_1 C_1 C_1 C_1 C_3 C_1 C_2 C_3 C_1 C_2 C_3 C_3 C_3	0	0	0	3	1	7	2.40
CPSO		7 6 5 4 9 10 11 12 3 2 1 13 14 15 8 X X X X −X −X −X −X X X X −X −X −X Y T_1 T_1 T_1 T_1 T_1 T_4 T_4 T_4 T_4 T_4 T_4 T_4 T_4 T_4 C_1 C_1 C_1 C_1 C_1 C_3 C_1 C_3 C_2 C_2 C_3 C_3 C_3 C_3	0	0	0	4	1	5	2.50
Firely algorithm	1	7 6 5 4 9 10 11 12 13 3 2 1 14 15 8 X X X X −X −X −X −X X X X X X X Y T_1 T_1 T_1 T_1 T_1 T_4 T_4 T_4 T_4 T_4 T_4 T_4 T_4 T_4 C_1 C_1 C_1 C_1 C_1 C_3 C_1 C_2 C_3 C_1 C_2 C_3 C_3 C_3	0	0	0	3	1	7	2.40
	2	7 6 9 10 5 4 3 2 1 11 12 13 14 15 8 −X −X −X −X X X X X X −X −X −X −X −X Y T_1 T_1 T_1 T_1 T_1 T_4 T_4 T_4 T_4 T_4 T_4 T_4 T_4 T_4 C_1 C_1 C_1 C_1 C_1 C_3 C_1 C_2 C_3 C_1 C_2 C_3 C_3 C_3	0	0	0	3	1	7	2.40
Improved Firely algorithm	1	7 6 5 4 9 10 11 12 13 3 2 1 14 15 8 X X X X −X −X −X −X −X X X X X X Y T_1 T_1 T_1 T_1 T_1 T_1 T_4 T_4 T_4 T_4 T_4 T_4 T_4 T_4 C_1 C_1 C_1 C_1 C_1 C_1 C_3 C_1 C_2 C_3 C_1 C_2 C_3 C_3 C_3	0	0	0	3	1	7	2.40
	2	7 6 9 10 5 4 3 2 1 11 12 13 14 15 8 −X −X −X −X X X X X X −X −X −X −X −X Y T_1 T_1 T_1 T_1 T_1 T_1 T_4 T_4 T_4 T_4 T_4 T_4 T_4 T_4 C_1 C_1 C_1 C_1 C_1 C_1 C_3 C_1 C_2 C_3 C_1 C_2 C_3 C_3 C_3	0	0	0	3	1	7	2.40

Table 3.*Comparison on running time(s)*

Algorithm	Population	Cycles	c	Running time（s）
HABC	50	500	2.40	1.482
CPSO	50	500	2.50	4.652
Firefly algorithm	50	500	2.40	2.65
Improved firefly algorithm	50	500	2.40	65.29

As shown inTable 3,the HABC algorithm, firefly algorithm and improved firefly algorithm can all find two sequences with assembly cost of 2.40. The CPSO algorithm can only find the near-optimal

assembly sequences with assembly cost of 2.50. HABC algorithm has certain advantages compared with CPSO algorithm and firefly algorithm and the advantages are obvious compared with the improved firefly algorithm in the running time. Although the improved firefly algorithm has made improvements onfirefly algorithm, the results of experiment inliterature [15] showed that improved firefly algorithm is less efficient than firefly algorithm.This is alsoconfirmed in this experiment.

Literature [6] concentrates on the convergence of firefly algorithm and improved firefly algorithm. Convergence reflects the approximation of the resulting solutions and the optimal solution at each iteration. In order to study the convergence of the HABC algorithm, we used the optimal solutions (the assembly cost of the example is 2.40) that the algorithm has found at the same iterations. In the same environment, the running results of the program are shown in Table 4. From the experimental results, convergence of the HABC algorithm has been relatively close to improved firefly algorithm and HABC algorithm only spent A few one-tenth of the time of improved firefly algorithm.

Table 4.*Comparison on convergence.*(n_s*is the number of optimal solutions*)

Population / Cycles	Algorithm	30		50	
		n_s	time(s)	n_s	time(s)
200	HABC	58	0.40	97	0.65
	Improved firefly algorithm	66	9.40	118	26.14
400	HABC	123	0.75	194	1.19
	Improved firefly algorithm	147	18.87	224	52.38

To sum up, the HABC algorithm that we proposed not only has outstanding search capability (in finding the optimal sequence) and efficiency (in time), but also improves the convergence rate while solving the assembly sequence planning problems. Therefore, we have confirmed that HABC algorithm can solve the assembly sequence planning very well.

5 Conclusions and future work

The HABC algorithm presented in this paper inherits the advantages of both the ABC algorithm and the simulated annealing algorithm. We encode the assembly sequences and discrete the nectar sources to make HABC algorithm suitable for the assembly sequence planning problems. Then, based on the strong exploration ability and weak exploitation ability of ABC algorithm, we use a local searching algorithm and a simulated annealing algorithm to enhance the ability of local exploitation.Inthe employed bee phase, we use the tournament selection instead of the wheel selection to escape from local optimum. In the scout bee phase, we design the scout bees to learn from the currentoptimal (minimum fitness value) nectar sources rather than randomly generated and this can overcome the defect of the slow convergence speed. According to the experiment, it is shown that the HABC algorithm is powerful for solving the assembly sequence planning problem.

In the HABC algorithm, there are some subjective operations in setting the weights ofindexes thatare closely related to the optimal solution. If some expert knowledge can be introduced and the interaction process can be analyzed by some professional method before setting the weights, then the results of ASP may have more guiding significance in practice. Therefore, the method of setting weights is one of our future works. Another future work is to extend the HABC algorithm from the assembly in orthogonal coordinate direction to the assembly in spherical coordinate.

Acknowledgments

This work is supported by the Natural Science Foundation of China (Nos. 61262030, 61572146, 61363030, U1501252); the Natural Science Foundation of Guangxi Province (Nos. 2015GXNSFAA139285, 2014GXNSFAA118354); the Guangxi Key Laboratory of Trusted Software, and the High Level of Innovation Team of Colleges and Universities in Guangxi and Outstanding Scholars Program.

References

1. HomemM.A Correct and Complete Algorithm for the Generation of Mech-anical Assembly Sequences.IEEE Transaction Robotics and Automation, 2(7):228-240 (1991)
2. Gottipol U R B, Ghos K. A simplified and efficient representation for evaluation and selection of assembly sequences. Computers in Industry, 50(3):251-264 (2003)
3. Wang Y, Liu J H, Li L S. Assembly sequence merging based on assembly unit partitioning. International Journal of advanced manufacturing technology, 45:808-820 (2009)
4. Greg C S, Shana S F. An enhanced genetic algorithm for automated assembly planning. Robotics and Computer Integrated Manufacturing, 18 (2002):355-364 (2000)
5. CHEN R S, LU K Y, YU SC. A hybrid genetic algorithm approach on multi-objective of assembly planning proble. Engineering Applications of Artificial Intelligence, 15(4): 447-457 (2012)
6. Zeng B, LI M F, Zhang Y. Assembly sequence planning based on improved firely algorithm(in Chinese). Computer Integrated Manufacturing Systems, 4 (20):799-806 (2014)
7. Wang Y, Liu J H. Chaotic particle swarm optimization for assembly sequence planning. Robotics and Computer-Integrated Manufacturing,26(2010):212-222 (2010)
8. Ibrahim I, Ibrahim Z, Ahmad H, et al. An Assembly Sequence Planning Approach with Binary Gravitational Search Algorithm. SoMeT, 179-193 (2014)
9. Ismail Ibrahim, Zuwairie Ibrahim, Hamzah Ahmad, et al. An Assembly Sequence Planning Approach with a Multi-state Particle Swarm Optimization. Trends in Applied Knowledge-Based Systems and Data Science (2016)
10. Gao W F, Liu S Y, Huang L L. Inspired Artificial Bee Colony Algorithm for Global Optimization Problems. Acta Electronica Sinica,12(40):2396-2403 (2012)
11. Alatas B. Chaotic bee colony algorithms for global numerical optimization. Expert Systems with Applications, 37(8):5682-5687 (2010)
12. Bahriye A, Karaboga D. A modified artificial bee colony algorithm for real-parameter optimization. Information Sciences,192(1):120-142 (2012)
13. Pan Q K, Suganthan P N, Chua T J. A discrete artificial bee colony algorithm for the lot-steaming flow shop scheduling problem. Information Sciences, 181 (2011): 2455-2468 (2011)
14. Percoco G, Diella M. Preliminary Evalution of Artificial Bee Colony Algoritnm When Applied to Multi-Objective Partial Disassembly Planning. Research Journal of Applied Sciences, Engineering and Technology,6(17):3234-3243 (2013)
15. Lee S. Subassembly identification and evolution for assembly planning.IEEE Transactions on System, Man and Cybernetics,24(3):493-503 (1994)
16. Tseng Y J, Chen J Y, Huang F Y. A particle swarm optimization algorithm for multi-plant assembly sequence planning with integrated assembly sequence planning and plant assignment. International Journal of Production Research,48(10):2765-2791 (2013)
17. Yuan B A, Zhang C Y, Shao X Y. An effective hybrid honey bee mating optimization algorithm for balancing mixed-modal two-sided assembly lines. Computers & Operations Research, 53(2015):32-41(2015)
18. Zeng B, Li M F, Zhang Y. Research on assembly sequence planning based on firely algorithm (in Chinese). Journal of Mechanical Engineering, 49 (11):177-185 (2013)