A Modified Honey Bees Mating Optimization Algorithm for Assembly Line Balancing Problem

Zhicheng Zhou
Electric Power Research Institute
Jiangsu Electric Power Company
Nanjing, China
e-mail: 15105168928@163.com

Biao Yuan, Pengfei Xiao, Chaoyong Zhang*
School of Mechanical Science & Engineering
Huazhong University of Science & Technology
Wuhan, China
e-mail: zcyhust@hust.edu.cn

***Abstract*—According to the characteristics of assembly line balancing problem, a modified honey bees mating optimization (MHBMO) algorithm was proposed to solve this problem. In this algorithm, the precedence matrix of tasks is proposed to apply the coding method based on the sequence of tasks. The initial feasible population was generated by a binary tree method, and a modified crossover operator and three neighborhood structures were used to keep solutions feasible. In order to enhance the balance of between intensification and diversification, a simulated annealing algorithm was utilized as a local optimization method. Finally, experimental results validate the effectiveness of the proposed algorithm.**

Keywords—assembly line balancing; honey bees mating optimization; simulated annealing; precedence relationship matrix

I. Introduction

Assembly Line Balancing Problem (ALBP) is the important problem which needs to be taken into consideration when designing and reconstructing assembly line. It plays a vital role in increasing the productivity and competitiveness and inducing the manufacturing cost of an enterprise. In general, assembly line type used in modern enterprise mainly includes straight type, U-type and two-sided type, the first two kinds of which are often used for producing household electrical appliances, smaller volume products and the third kind of which for vehicles and other large volume products. This paper researches on solving the problem of minimizing the amount of required work stations when using these three types of assembly line under specified cycle time, which is type-I Assembly Line Balancing Problem (ALBP-I).

Currently, scholars have carried out a great deal of research on ALBP-I of straight type and U-type assembly line, relatively less on that of two-sided type. The proposed methods to solve this problem can be generally classified into three categories: (1) heuristic algorithm, such as ranked positional weight technique proposed by Helgeson [1] et al for solving the straight type ALBP and First Matching Principle proposed by Barthodi [2] for solving the two-sided type ALBP and so on; (2) exact method, for instance, goal programming method used by Gokcen [3] et al to solve the U-type ALBP and a kind of branch and bound method proposed by Hu [4] et al to solve the two-sided type ALBP and so on; (3) meta heuristic algorithm, for instance, genetic algorithm used by Hua-ming Song [5] et al to solve the U-type ALBP, modified ant colony algorithm under updating rules based on optimized pheromone components proposed by Liang Zha [6] et al to solve the straight type and U-type ALBP and sequence based modified genetic algorithm proposed by Er-fei Wu [7] et al to solve the two-sided type ALBP and so on.

Honey bees mating optimization algorithm is a kind of meta heuristic algorithm inspired by the mating process of honey bees and it was first proposed by Abbass [8] in 2001. Currently, it is mainly used for solving the combination optimization problem. For example, Marinakis [9] et al. have used it to solve the vehicle routing problem, and Marinakis [10] et al also have proposed a hybrid honey bees mating optimization algorithm to solve the probabilistic traveling salesman problem and so on. Honey bees mating optimization algorithm has shown a relatively stronger global search ability when used for this kind of problems.

From the current situation of research, some scholars have used ant colony algorithm to simultaneously solve the straight type and U-type ALBP [6,12], but nobody has yet researched on the simultaneous solving method of straight type, U-type and two-sided ALBP. Thus, aimed at the characteristics of these three types of assembly line, this paper has designed different decoding solutions and fitness functions and proposed a kind of Modified Honey Bees Mating Optimization (MHBMO) to solve the ALBP-I of these three types of assembly line.

II. Introduction of Assembly Line and Its Problems

Since Ford Company constructed the first motor vehicle flow production line (assembly line) in 1908, assembly line has gradually become an important production form in modern manufacturing, and it commonly used in large-scale products manufacturing such as household electrical appliances and vehicles.

The earliest used assembly line is straight type, each station of which is arranged in a straight line. Fig. 1 shows a straight type assembly line. As for the change of demand, straight type assembly line lacks of production flexibility, and has difficulty in readjusting to reach a new balance, leading to low productivity.

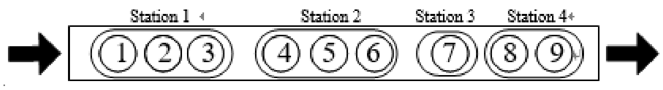

Fig. 1. Straight type assembly line

For the demand of flexibility and efficiency, the concept of U-type assembly line is proposed. Compared with the straight type which can only be assigned unidirectional tasks, U-type assembly line can carry out tasks not only in the direction from front to back or from back to front, but also in both directions simultaneously, therefore it has better production flexibility. Fig. 2 shows a U-type assembly line.

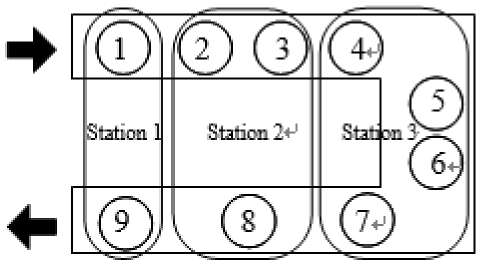

Fig. 2. U-type assembly line

Straight type and U-type assembly line are called single-sided assembly line, in which the ALBP-I can be described as: under specified cycle time (CT) and satisfying the constraint of precedence relationship among tasks, assign tasks to each station of assembly line to guarantee the amount of required stations is minimum.

Aimed at the special demand of producing large products, the concept of two-sided assembly line is proposed.

It divides station of original single-sided assembly line into two left and right stations (such as station 1 and station 2 in Fig. 3, the left and right two opposite stations are called adjoint station by each other and both in position 1), stations on each side carry on operations independently without interference.

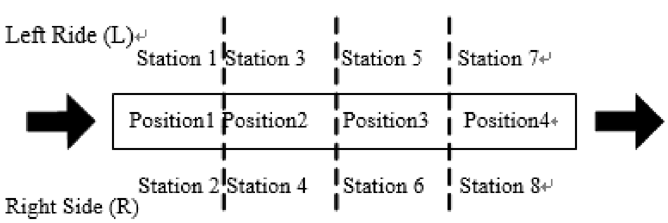

Fig. 3. Two-sided assembly line

Two-sided assembly line needs to satisfy two constraints when assigning tasks[7]: (1) satisfying the operation side demand (left side L, right side R, either side E), assign the task to the station on the according side; (2) consider not only the impact of itself operation time on balance but also the impact of its waiting time which results from the influence of its preorder task (in adjoint station) on balance.

The type-I balancing problem of two-sided assembly line can be described as: under specified cycle time (CT) and satisfying the constraint of precedence order of tasks and operation side demand, assign tasks to each station on both sides of assembly line to make sure the amount of required positions is minimum and simultaneously minimize the required amount of open stations.

III. Introduction of Honey Bees Mating Optimization Algorithm

A. *Principle of Honey Bees Mating*

Honey bee is a typical kind of social insect, which has strict division of labor. A typical bee colony consists of several queen bees, up to a thousand drones, 10000 to 60000 worker bees. The queen bee is the most important individual of the bee colony, taking charge of oviposition; drone only has one job that is mating with queen bee and supports sperms for her; worker bee takes charge of nurturing young bees. Young bees may develop from fertilized or unfertilized egg cell, the former of which will develop into queen bees or worker bees and the latter will become drones.

The mating process of honey bee colony is called mating flight. It is symbolized by the queen bee dancing in the air, after which the queen bee will choose several drones to mate. During each mating, sperms of drones will continuously accumulate in the spermathecal of the queen bee and finally form a genetic pool of the honey bee colony. The queen bee will randomly choose sperms supported by drones to fertilize her egg to reproduce young bees.

B. *Principle of Honey Bees Mating Optimization Algorithm*

In the algorithm, the queen bee represents the optimum solution of current solution space, which is one of the main individuals of producing child generation; drones support their genes for the queen bee; the worker bees nurture the child generation, as representatives of local optimization, improving the quality of solutions in the neighborhood of child generation.

This algorithm mainly includes 5 steps as below [8]:

- The mating flight begins, the queen bee choose drones to her spermathecal and then randomly choose them as individuals of reproducing child generation;
- The chromosomes of chosen drones cross over with those of the queen bee, producing new young bees;
- Worker bees carry out local optimization to young bees, enhancing the adaptability of young bees;
- Evaluate the adaptability of grown young bees;
- Compared with the queen bee, choose the better one as the new queen bee.

The mating flight of first step is relatively complex, introduced as below:

Before the mating flight begins, the queen bee has been given certain speed, energy and spermathecal of corresponding capacity. When the energy of the queen bee is under threshold or the spermathecal is full, the mating flight process is over. The speed of the queen bee determines the probability of drones entering into the spermathecal. The formula to calculate the probability is shown as below [8]:

$$Prob(Q, D) = e^{\frac{\Delta(f)}{S(t)}} \quad (1)$$

In (1), *Prob*(*Q*, *D*) represents the probability that drone *D* is chosen and enters spermathecal *Q*; Δ(*f*) represents the absolute value of the difference between the speed of the drone and the queen bee; *S*(*t*) represents the speed of the queen bee at time *t*.

The mating flight process can be considered as a series of state transformation in space region, during which the energy *E*(*t*) and the speed *S*(*t*) of the queen bee attenuate according to a coefficient α(0<α<1), specific calculation formula as below [9]:

$$S(t+1) = \alpha \times S(t) \quad (2)$$

$$E(t+1) = \alpha \times E(t) \quad (3)$$

IV. Assembly Line Balancing Based On MHBMO Algorithm

A. Task Precedence Relationship Matrix

In ALBP, digraph is commonly used to present tasks precedence relationship. Taking the Mertens problem as an example, its tasks precedence relationship is shown as Fig. 4, number in circle is the sequence number, and number outside circle is the operation time of the corresponding task, and the arrow represents the precedence relationship among tasks. In order to use the proposed algorithm expediently, the precedence relationship digraph should firstly be transformed to a precedence relationship matrix *P*. In *P*, if task *i* is completed before task *j*, then $p_{i,j}$ =1; otherwise, $p_{i,j}$ =0. Fig. 5 is the precedence relationship matrix of Mertens problem.

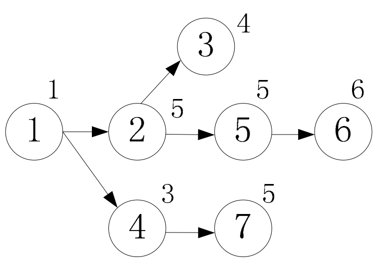

Fig. 4. Mertens task precedence relationship digraph

$$\begin{array}{c} \\ 1\\2\\3\\4\\5\\6\\7 \end{array}\begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Fig. 5. Mertens task precedence relationship matrix

B. Coding

This paper uses real number coding method. The length of chromosomes is equal to the amount of tasks, and the number on the gene locus represents the task sequence number, the former and the latter positions of which represent the precedence relationship of tasks. Taking the Mertens problem as an example, under the condition of precedence relationship, an coding form of a probable chromosome is {1, 2, 5, 4, 7, 3, 6}.

C. Decoding

1) Straight type decoding

Opening station 1, assign the task corresponding to the gene locus to station 1 according an order from left to right. When the completion time of current station is greater than the cycle time, open the next station. Repeat the above process until all tasks are assigned.

2) U-type decoding

Opening station 1, compare the task operation time of front and back ends of the chromosome, and assign the task with less operation time to the current station, after which delete the task from the chromosome and compare the task operation time of front and back ends of the chromosome again, then assign the task with less operation time to the current station in the same way. When the completion time of current station is greater than the cycle time, open the next station. Repeat this process until all tasks are assigned.

3) Two-sided type decoding

Opening position 1, judge task *i* corresponding to the first gene locus and its demanding side. If the side is L or R, then assign task *i* to the left or right side station of current position; if the side is E, then compare the earliest start time of task *i*, and assign the task to the smaller-time side, and assign it to the right side if the time is equal. When the completion time of some side station in current position is greater than the cycle time, close current position and open the next position. Repeat previous steps until all tasks are assigned. The earliest start time of task *i* is equal to the greater one between the completion time of assigned station of task *i* and the latest completion time of the former task of task *i* in the adjoint station (on the opposite side).

D. Fitness Function

1) Straight type and U-type fitness function

Aimed at the characteristics of straight type and U-type ALBP, adopt following formula:

$$f = \frac{\sum_{k=1}^{m}(\sum_{j \in S_k} t_j)^2}{m} \tag{4}$$

In (4), m is the amount of required stations, t_j is the required time for completing task j, S_k represents the k^{th} station. Numerator expression of (4) is from the document [12], which denotes the square of the sum of the task operation time of each station and impels lower load station to transfer tasks to higher load station, thereby enable to increase the probability of reducing stations amount; the denominator denotes the required stations amount. The fitness function value increases with the decrease of the stations amount. Thus, in order to get the minimum stations amount, the maximum value of fitness function should be obtained.

2) Two-sided type fitness function

Aimed at the characteristics of two-sided type ALBP, adopt following calculation formula:

$$f = w_1 n_1 + w_2 n_2 \tag{5}$$

In (5), n_1 is the amount of required positions; w_1 is the weighting coefficient of required positions amount; n_2 the amount of required stations (sum of stations on left and right side); w_2 the weighting coefficient of required stations amount; w_1 and w_2 can be chosen according to the actual goal. The goal of two-sided type ALBP in this paper is minimize the positions amount, on this premise, required stations amount is minimum. Therefore, set the weighting coefficient of positions greater than that of stations. This paper sets $w_1 = 1$, $w_2 = 1/2$ when calculating. In order to get the optimum solution, the minimum value of the fitness function should be obtained.

E. Population Initialization

when generating initial population, generate several tasks sequences first; then based on the task precedence relationship matrix, use a kind of binary tree method [13] to transform these tasks sequences into feasible solutions. Binary tree method is mainly comprised of construction and traversal two steps. When constructing a binary tree, tasks corresponding to left child node should precede over that to the root node which should precede over that to right child node.

Relevant symbols are introduced as following:

g_h: denotes the task corresponding to the h^{th} position ;

r: denotes the root node of the binary tree;

l: denotes the child node of the binary tree;

N: denotes the total amount of tasks;

Specific steps are as following:

- Step 1: Randomly generate a sequence, that is the chromosome;
- Step 2: Set h=2;
- Step 3: Set the task corresponding to g_1 as that to the root node r;
- Step 4: Set the task corresponding to g_h as that to the child node l, compare the precedence relationship between tasks corresponding to the two nodes.

If $p_{l,r}$=1, then the task corresponding to node r is completed after that to node l.

(1) If the left child node of r is non-null, then set it as the new root node r and carry out step 4;

(2) If the left child node of r is null, then insert l into it and set h=h+1, then carry out step 5;

If $p_{l,r}$=0, then there is no constraint of precedence relationship between tasks corresponding to node r and l.

(1) If the right child node of r is non-null, then set it as the new root node r and carry out step 4;

(2) If the right child node of r is null, then insert l into it and set h=h+1, then carry out step 5;

- Step 5: compare h and N and judge whether they are equal. If they are equal, then carry out step 6; otherwise, carry out step 3;
- Step 6: after binary tree in order traversal, the sequence that we get is the feasible solution satisfying the precedence relationship.

F. Crossover Operation

In order to guarantee the population after crossover operation is still satisfying the constraint of precedence relationship, we have used the improved single-point crossover operation [5], assuring the child generation feasible. Specific operation steps are as following:

- Step 1: according to the order of entering the spermathecal, choose drones as parent generation;
- Step 2: randomly generate an integer n, denoting the gene locus corresponding to the crossover point in chromosome;
- Step 3: crossover point n divides the chromosome into two sections: front section and back section, copy the genes in front section of the queen bee and the drone to child generation 1 and 2;
- Step 4: according to the precedence relationship of tasks corresponding to the chromosome of the done, rank the genes in the back section of the queen bee and copy them to the back section of child generation 1; in the same way, according to the precedence relationship of tasks corresponding to the chromosome of the queen bee, rank genes in the back section of the drone and copy them to the corresponding position of child generation 2 so as to generate child generation 2.
- Step 5: compare the fitness values of the two child generations, and choose the better one as the final child generation.

Taking the two chromosomes in the Mertens problem as an example, Fig. 6 introduces the process of crossover operation. (a) presents that choosing the 4th gene locus to divide the queen bee and the drone into front and back sections; (b) presents that copy the gene locus in front section of the queen bee to the child generation 1; (c) presents that according to the tasks precedence relationship of the drone, copy the gene locus in the back section to the child generation 1; (d) presents the produced child generation 2.

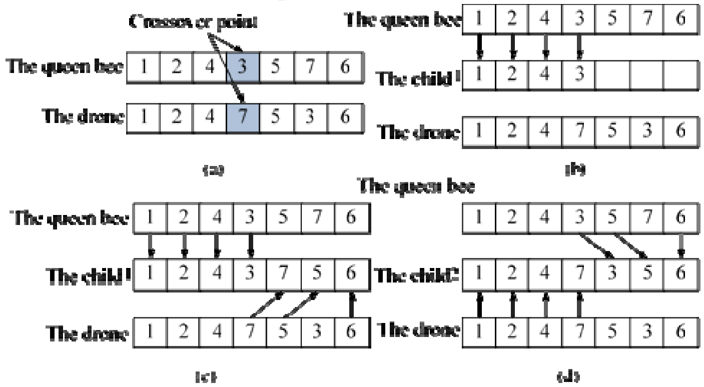

Fig. 6. Process of crossover operation

G. Local Optimization Operation

As a local optimization method, simulated annealing (SA) algorithm shows great local search ability. Thus, this paper uses simulated annealing (SA) algorithm, combined with three kinds of neighborhood structures, to design three kinds of drones so as to enhance the adaptability of child generation. When the three kinds of neighborhood structures generates new solutions, it is guaranteed that they still satisfy the constraint of precedence relationship. The three kinds of neighborhood structures are introduced as following in which a, b separately denotes the a^{th} and the b^{th} gene locus in the chromosome.

1) Insert-to-back operation

- Step 1: randomly generate a number *a*, and set *b*=*a*+1;
- Step 2: compare the precedence order of tasks corresponding to *a*, *b*. If the task corresponding to *a* precede over *b* or the value of *b* is greater than the total amount of tasks, then carry out step 3; otherwise, add 1 to *b* and then carry out step 2;
- Step 3: insert task corresponding to *a* into the position corresponding to *b*-1.

2) Insert-to-front operation

- Step 1: randomly generate a number *a*, and set *b*=*a*-1;
- Step 2: compare the precedence order of tasks corresponding to *a*, *b*. If the task corresponding to *a* precede over *b* or the value of *b* is less than 0, then carry out step 3; otherwise, subtract 1 from *b* and then carry out step 2;
- Step 3: insert task corresponding to *a* into the position corresponding to *b*+1.

3) Adjacent pairwise exchange operation

- Step 1: randomly generate a number *a*, and set *b*=*a*+1;
- Step 2: compare the precedence order of tasks corresponding to *a*, *b*. If the task corresponding to *a* precede over *b* or the value of *b* is greater than the total amount of tasks, then carry out step 1; otherwise, carry out step 3;
- Step 3: exchange the tasks corresponding to positions *a*, *b*;

V. Experiments

In order to validate the effectiveness of the proposed algorithm, choosing 6 problems from the straight type and U-type benchmark problems [14] summing up to 18 instances, and 4 problems (P24 [15]，P16、P65、P205 [16]) from the two-sided type benchmark problems summing up to 12 instances to carry out the test experiment.

As to the MHBMO algorithm, required parameters are as following: the amount of the queen bee is 1; the amounts of drones and mating flight are set according to the scale of the problem;

The capacity of spermathecal is 15; the initial energy and speed of mating flight is 10000 and the attenuation coefficient of the energy and speed is 0.95; the initial temperature of simulated annealing algorithm is 1000, and the attenuation coefficient of the temperature is 0.95. The proposed algorithm is programmed in C++ and run in the Pentium 1.86GHz CPU and 1GB RAM environment.

As for the straight type and U-type ALBP, this paper compares ant colony algorithm [11] with MHBMO, the results are presented in Table 1. As for two-sided ALBP, this paper compares Tabu Search [17] (TS) algorithm and Bee algorithm [18] (BA) with MHBMO, meanwhile, listing the optimum solution of current documents as Table 2. The number in bracket denotes the total amount of tasks of the problem.

TABLE I. Results Of Solving Straight Type And U-Type ALBP With MHBMO

Problem	Cycle Time（CT）	ACO [11]	MHBMO	MHBMO
		Straight Type	Straight Type	U-Type
Mertens	6	6	6	6
(7)	10	3	3	3
	15	2	2	2
Mitchell	14	8	8	8
(21)	21	5	5	5
	35	3	3	3
K. &Wester	57	10	10	10
(45)	92	6	6	6
	138	4	4	4
Tongue	176	21	21	21
(70)	410	9	9	9
	527	7	7	7
Arcus83	5048	16	16	16
(83)	6842	12	12	12
	8412	10	10	10
Arcus111	8847	18	18	18
(111)	10743	15	15	15
	11378	14	14	14

TABLE II. TABLE 2 RESULTS OF SOLVING TWO-SIDED TYPE ALBP WITH MHBMO

Problem	Cycle Time (CT)	Current Optimum Solution [18]	MHBMO	TS [17]	BA [18]
P16	16	6	6	6	6
(16)	19	5	5	5	5
	22	4	4	4	4
P24	18	8	8	8	8
(24)	24	6	6	6	6
	35	4	4	4	4
P65	326	17	17	17	17
(65)	490	11	11	11	11
	544	10	10	10	10
P205	**1699**	**15**	**16**	**17**	**16**
(205)	2266	12	12	13	12
	2832	10	10	10	10

Experimental results in Table 1 shows that, the same with ACO, using MHBMO has got the optimum solutions of total 18 instances of straight type and U-type ALBP, presenting its effectiveness 0f solving the two kinds of ALBPs. Experimental results in Table 2 shows that, using MHBMO has got 11 optimum solutions of 12 instances of two-sided type ALBP. In P205, when CT=1699, the solution is 16, the same with BA, and better than the solution 17 got by TS, little worse than the current optimum solution 15, presenting the effectiveness of MHBMO when solving the small-scale two-sided type ALBP and partial effectiveness when solving the large scale problem.

The above experimental results show that, MHBMO is a better algorithm to solve the 1-type problem of the straight type, U-type and two-sided type ALBP problem. It’s main advantages are: a) the proposed algorithm is in a framework of HBMO, at the same time, combined with the local search ability of SA, so that the global search and the local search can reach an advance balance, and the convergence of the algorithm is realized; b) compared with other algorithm of solving the ALBP, the proposed algorithm is not highly dependent on the characteristics of the problem, that is, it does not need to design other heuristic algorithm to help solve the problem.

VI. CONCLUSION

According to the 1-type problem of the three types of ALBPs, a modified honey bees mating optimization algorithm is proposed in this paper. The algorithm adopts the coding method based on the sequence of tasks and the decoding method satisfying the constraint of tasks assignment of the three types of assembly line, simultaneously using the crossover operator and three kinds of neighborhood structures to keep the search in solution space so that the efficiency of the algorithm is increased. Compared with other algorithms, its effectiveness is validated.

Future research can be carried out from the two following aspects: 1) consider using the proposed algorithm to solve the mixed model assembly line balancing problem and stochastic assembly line balancing problem; 2) to make the two-sided type ALBP model more close to practice, and add the limit of assigned station of partial tasks and other constraints to the model.

ACKNOWLEDGMENT

Project supported by the National Natural Science Foundation of China (no.51575211), and the Funds for International Cooperation and Exchange of the National Natural Science Foundation of China (no.51561125002).

REFERENCE

[1] HELGESON W, BIRNIE D. Assembly line balancing using the ranked positional weight Technique [J]. Journal of Industrial Engineering, 1961, 12: 394–398.

[2] BARTHODI J J. Balancing two-sided assembly lines: a case study [J]. International Journal of Production Research, 1993, 31(10): 2447-2461.

[3] GOKCEN H, AGPAK K. A goal programming approach to simple U-line balancing Problem [J]. European Journal of Operational Research, 2006, 171: 577-585.

[4] HU Xiaofeng, WU Erfei, BAO Jinsong, et al. A branch-and-bound algorithm to minimize the line length of a two-sided assembly line [J]. European Journal of Operation Research, 2010, 206: 703-707.

[5] HU Xiaofeng, WU Erfei, BAO Jinsong, et al. A branch-and-bound algorithm to minimize the line length of a two-sided assembly line [J]. European Journal of Operation Research, 2010, 206: 703-707.

[6] SONG Hua-ming, HAN Yu-qi. Genetic algorithms-based U-shaped assembly line balancing [J].Journal of Systems Engineering, 2002, 17(5): 424-429.

[7] Zha Jing, Xu Xueijun, Yu Jian-jun, et al. A Modified Ant Colony Algorithm for Simple and U-shaped Assembly Line Balancing [J]. Industry Engineering Journal, 2010, 13(6): 76-81.

[8] ABBASS H A. Marriage in honey bees optimization(MBO): a haplometrosis polygynous swarming approach [C] // Proc of Congress on Evolutionary Computation. 2001: 207-214.

[9] MARINAKIS Y， MARINAKIS M and DOUNIAS G. Honey-bees Mating Optimization Algorithm for the Vehicle Routing Problem [J]. Studies in Computational Intelligence. 2008, 129: 139‑148.

[10] MARINAKIS Y and MARINAKIS M. A Hybrid Honey Bees Mating Optimization Algorithm for the Probabilistic Traveling Salesman Problem[C]//Proc of Congress on Evolutionary Computation. 2009: 1762–1769.

[11] BAYKASOGLU A, DERELI T. Simple and U-type assembly line balancing by using an ant colony based algorithm [J]. Mathematical and Computational Applications, 2009, 14(1): 1-12.

[12] LAPIERRE S D, RUIZ A, SORIANO P. Balancing assembly lines with tabu search [J]. European Journal of Operation Research, 2006, 168, 826-837.

[13] TSENG H E. Guided genetic algorithms for solving the larger constraint assembly problem [J]. International Journal of Production Research, 2006, 44(3): 601–625.

[14] ARMIN S. Data of Assembly Line balancing Problems [J]. Schriften Zur Quantitativen Betriebswirtschaftslehre. 1993, 16: 93.

[15] KIM Y K, KIM Y, KIM Y J. Two-sided assembly line balancing: a genetic algorithm approach [J]. Production Planning and Control, 2000, 11(1): 44-53.

[16] LEE T O, KIM Y, KIM Y K. Two-sided assembly line balancing to maximize work relatedness and slackness [J]. Computers and Industrial Engineering, 2001, 40: 273–292.

[17] ÖZCAN U, TOKLU B. A tabu search algorithm for two-sided assembly line balancing [J]. International Journal of Advanced Manufacturing Technology, 2009, 43: 822-829.

[18] ÖZBAKIR L, TAPKAN P. Bee colony intelligence in zone constrained two-sided assembly line balancing problem [J]. Expert Systems with Applications, 2011, 38: 11947-11957.