A hybrid artificial bee colony algorithm for balancing two-sided assembly line with assignment constraints

Jing Xiong, Xiaokun Duan1 and Erhua Wang

School of Mechanical and Electrical Engineering, Changzhou College of Information Technology, Changzhou, China;

1E-mail: duanxk@yeah.net

Abstract. Two-sided assembly line balancing problem (TALBP) is a vital design problem for the industries. In real production process, some complex constraints should be considered in the two-sided assembly line. To solve the practical TALBP, this paper proposes a hybrid algorithm (HABC) that combines the artificial bee colony (ABC) algorithm and late acceptance hill-climbing (LAHC) algorithm. In the proposed algorithm, a well-designed decoding scheme is embedded to tackle multiple assignment constraints. Moreover, two neighborhood search strategies are implemented by employed bee and onlooker bee to explore and exploit the new solution. A set of computational experiments is performed on benchmark problems. The comparison results, best solutions, standard deviation and relative percentage index demonstrate that the HABC algorithm outperforms other algorithms published in the literature and finds 5 brand new solutions for 15 instances.

1. Introduction

Two-sided assembly lines have been widely utilized to assemble large-sized products. The assembly space of these products is largely to provide the workers on both sides to carry out assembly work simultaneously. In the two-sided assembly line balance (TALB), the assignment of the task not only satisfy all the basic constraints of the one-sided assembly lines, but also meets the operation directions constraint and the precedence relations constraint [1]. The L-type tasks must be assigned to the left side of the TALB according to the operation directions constraint, the operation of the R-type tasks same as the L-type tasks. In addition, there is a task (E-type task) that can be assigned to either side of the two-sided assembly line. A two-sided assembly line is illustrated in Figure 1.

Figure 1. Two-sided assembly line.

As shown in Figure 1, two facing stations compose a mated-station. Every task must be assigned to only one station and the finish time of all the tasks in the same mated-station must be smaller than or equal to the cycle time. Tasks are assigned to two sides of the line and the shaded rectangle is called idle time. In the TALB, the idle time is unavoidable because of each task has a set of precedence constraints [2].
An example of TALBP with 12 tasks is depicted in Figure 2. The number in the circle means the sequence number of the tasks. The label on the circle presents the operation time and the side station of the corresponding task. The arrow between the circles is the precedence relations. For instance, the operation time of task 7 is 3, and it can be assigned to the left or right side of the assembly line. And the task 7 cannot be assigned to a station before task 4 and task 5 are finished.

(2, L) (3, L) (3, E) (2, E)
(1) 4 7 10 (1, R)
(3, R) (1, E) (3, R)
(2, E) 5 8
(1, L) (2, E) (2, E)
(3) 6 9 11 (2, E)
(2, E)
(1, R) (3, R)
(1, L)

Figure 2. An example P12 of TALBP.

In the literature [3-5], the TALBP with different objective, such as maximum work-relatedness and minimize workloads, minimizing the number of stations and multi-objective function, have been solved by heuristic methods and swarm intelligence algorithms. However, all these papers ignore the assignment constraints of the TALBP in practical application.

With respect to real-world applications, more complex constraints of the TALBP should be considered. Kim et al. solved the TALBP with the positional constraints by a genetic algorithm approach [6]. Özbağır and Tapkan investigated a bees algorithm to analyze the TALBP with the zoning constraints [7]. Simaria and Vilarinho developed an ant colony optimization algorithm for the TALBP with synchronous constraints [8]. In the literature [9-13], LAHC algorithm, HTLBO algorithm, GA and Rapid Entire Body Assessment (REBA) method have been applied to solve the TALBP with one or more constraints of the above.

Considering the complexity of practical constraints, this paper proposes a hybrid algorithm that combines the ABC algorithm and the LAHC method for handling the TALBP with all above assignment constraints. Since the exact methods, such as B&B and bidirectional heuristic can find the theoretical optimal solutions of small and medium sized problems, this paper research on the large-sized problems to obtain some better solutions. According to the characteristics of the problem, a well-designed decoding scheme is embedded to tackle multiple assignment constraints. The LAHC method is utilized to accelerate the convergence rate.

2. Mathematical formulations

For given a cycle time, the objective of minimizing the number of the mated-stations is the primary objective in this paper. The notations used in this paper are listed in Table 1.

The objective of minimizing the number of the stations works as the second objective when two soluble have the same number of the mated-stations.

\[
\min (\alpha N_m + \beta N_s)
\]  

Subject to

(1) Occurrence constraint: only one station can be assigned for each task.

\[
\sum_{j \in J} \sum_{k \in K(i)} x_{ijk} = 1, \quad i \in I
\]  

(2) Precedence constraint: a task can only be assigned after its predecessors are finished.

\[
\sum_{g \in G} \sum_{k \in K(h)} g x_{hjk} \leq \sum_{j \in J} \sum_{k \in K(i)} j x_{ijk}, \quad i \in I - P_0, \quad h \in P(j)
\]  

(3) Cycle time constraint:

\[
f t_i \leq C_T, \quad i \in I
\]  

(4) Sequence-dependent finish time constraint:

\[
f t_i - f t_h + \phi (1 - \sum_{k \in K(h)} x_{hjk}) + \phi (1 - \sum_{k \in K(i)} x_{ijk}) \geq t_i, \quad i \in I - P_0, \quad h \in P(i), \quad j \in J
\]
Positional constraint: a task must be assigned to the deterministic station \((j, k)\).

Synchronous constraint: a pair of tasks \((i, h)\) must be assigned to the same mated station, and these two tasks must be operated at the same start time.

Table 1. The notations in HABLP.

| \(I\) | Set of tasks, \(I = \{1, 2, \ldots, I, \ldots, n\}\) | \(S(i)\) | Set of all successors of task \(i\) |
|---|---|---|---|
| \(J\) | Set of mated-stations, \(J = \{1, 2, \ldots, j, \ldots, m\}\) | \(K(i)\) | Set of all predecessors of task \(i\) |
| \(j\) | Indicator for the side of stations, \(k = 1\), if the station is left side; \(k = 2\), otherwise | \(K(i) = \begin{cases} 
1 & \text{if } i \in S_L \\
1 & \text{if } i \in S_R \\
2 & \text{if } i \notin \{S_L, S_R\} 
\end{cases}\) |
| \((j, k)\) | A station of mated-station \(j\) and its operation direction is \(k\) | \(t_i\) | Processing time of task \(i\) |
| \(S_L\) | Set of tasks assigned to a left station; \(S_L \subset I\) | \(f_i\) | Finish time of task \(i\) |
| \(S_R\) | Set of tasks assigned to a right station; \(S_R \subset I\) | \(x_{ijk}\) | x_{ijk} = 1, if task \(i\) is assigned to station \((j, k)\); \(x_{ijk} = 0\), otherwise |
| \(P_0\) | Set of tasks that have no predecessors | \(z_{ir}\) | \(z_{ir} = 1\), if task \(i\) is assigned earlier than task \(r\) in the same station; \(z_{ir} = 0\), otherwise |
| \(P_a(i)\) | Set of all predecessors of task \(i\) | \(\varphi\) | A very large positive number |

3. The hybrid ABC algorithm
In the ABC algorithm, different types of bees are sent to explore and exploit solutions. In the initial population, a solution of the problem is indicated as a food source. The employed bee exploits a food source and shares it with the onlooker bee. Then, the onlooker bee chooses a food source to exploit, according to its fitness. The scout bees find new food sources by random search to increase population diversity.

3.1. Encoding
In the HABC algorithm, a string of integers is an encoding of a feasible solution. Let the length of the string is equal to the number of tasks \(n\), the position of a string represents a task \(i\). A list of different priority values (integer) is generated randomly from 1 to \(n\). The priority value of task \(I\) is equal to the value of the position in the list. Suppose the solution of P12 (12-tasks of TALBP) is constructed as \{2, 5, 3, 7, 6, 1, 12, 9, 8, 11, 4, 10\}, which means that the priority value of task 6 is 1, and its assignment sequence is first in all tasks. Make a priority list of the encoding, the assignment order of these tasks is 6-1-3-11-2-5-4-9-8-12-10-7.
3.2. Decoding

Obviously, the encoding individual has no information about which station tasks are assigned to, and the assignment sequences may violate the precedence constraints. Therefore, a decoding process is designed to interpret an encoded individual into a feasible solution. In the decoding process, a set of candidate tasks contain tasks to satisfy the cycle time and precedence constraints. If the candidate task with the highest priority can satisfy the zoning constraints and synchronous constraints, then it will be assigned to the corresponding station; otherwise, it will be deleted from the candidate set and another candidate task will be chosen to check. The detailed decoding process is shown in Figure 3.

In the decoding process, the E-type tasks are assigned to the side with the earlier starting time. When a tie occurs, the task is assigned to the left side of the mated-station. After the decoding process, the tasks in the last mated-station should be checked. These tasks should be adjusted to the same station of the mated-station when all tasks satisfy all constraints and their operation directions L&E or R&E.

![Figure 3. Decoding process.](image)

The assignment process gives priority to meet the zoning constraints and synchronous constraints, while the obtained solution may violate the positional constraints. Therefore, the last step of the decoding procedure is checking the position constraints. If a solution cannot satisfy the positional constraints, a penalty coefficient should be added in the objective function.

3.3. Population initiation

In the initial population, the priority value of the task in encoding obtained by the three priority rule: random, MaxRPW and MaxT. Random rule is the most common rule in the procedure, which can
provide diverse individuals. MaxRPW and MaxT are the most effective priority rules based on the task times and the precedence constraints, which can ensure the quality of the individuals.

3.4. Local search by LAHC
The LAHC algorithm is an efficient local search with iterative process and late acceptance strategy, which is utilized to accelerate the convergence rate. The LAHC requires an initial solution and iteratively improves it by a neighborhood structure. The algorithm accepts the neighborhood solution with its cost not worse than the current several iterations before [9].

In this paper, four kinds of neighborhood structure are employed in LAHC, such as swap operator, insert operator, multi swap operator and multi insert operator. In order to explore a better food source and maintain the diversity of the population, an employed bee selects one of four neighborhood structures randomly to generate a new solution.

In order to exploit more promising solutions, VNS is combined with LAHC in onlooker bee phase. A systematic switch from one kind of neighborhood search structures (NSS) to another one is performed to find a better solution more opportunity. Two neighborhoods are utilized in VNS: multi swap operator ($N_s(S)$) and multi insert operator ($N_i(S)$). The $N_s(S)$ is utilized only if no improvement about the best solution is made by the $N_i(S)$.

3.5. Employed bee phase
In the exploration phase, each employed bee selects a current solution randomly in the initial population. In order to explore a better or potentially better solution, the LAHC with four neighborhood structures is adopted to explore a new solution using. If the objective value of the new solutions is better than the current one, the current one will be updated; otherwise, the current solution is retained.

3.6. Onlooker bee phase
Onlooker bees aim to exploit better solutions after obtaining the updated population from employed bees. To enhance the exploitation ability, onlooker bees select a current solution by binary tournament selection and exploit the current population by using the LAHC with the VNS.

In the local search of the basic ABC algorithm, the current solution will be replaced by a new solution with the greedy strategy. But the process may miss the potential solution which can produce better neighborhood solution. To prevent this condition, the new solution compared to the worst solution in the population, and it is accepted if it is better than the worst one. Otherwise, the new solution is abandoned. In this way, onlooker bees direct the population toward the most promising regions in the search space.

3.7. Scout bee phase
If a solution without improvement during a predetermined generation, the employed bee will become a scout bee. Then the scout bee will generate a new solution randomly in the search space to substitute the abandoned solution. This procedure may decrease the search efficiency because the randomly solution will most likely be a worse solution, it cannot carry better information for the population. Therefore, in order to find a new better solution and avoided to fall into local optimum, the scout bee produces a new solution by performing several neighborhood search to a randomly solution in the population.

4. Experimental results

4.1. Experimental design
An extensive computational study is performed on three large-sized benchmark problems to measure the performance of the HABC algorithm. These problems include P65, P148 [6], and P205 [9]. The
positional constraints, zoning constraints and synchronous constraints for this paper are taken from [9]. The HABC algorithm is programmed by Microsoft Visual Studio 2015.

The parameters of the HABC algorithm are set as follows. The population size is set to \( Np = 100 \), the numbers of the three bees are set to 50, 50, and 1, respectively; the limited generation which a solution cannot be improved is set to \( \text{limit} = 20 \); the length of solution list of the LAHC in the HABC algorithm is set to 20; the stop condition is a CPU time fixed to \( Ts = n \times n \times 15 \text{ms} \). Each test instance is performed 30 times independently.

4.2. Experimental results
The performance of the HABC algorithm is compared with other three algorithms, the LAHC algorithm [9] and a hybrid teacher-learning-based optimization (HTLBO1) algorithm [11] and HTLBO2 algorithm [10]. The results comparison of four algorithms is shown in Table 2.

The lower bounds (\( LB \)) is the theoretical value of the objective. The best and average solutions of \( Ns \) among 20 times are reported by \( \text{Best} \) and \( \text{Mean} \). The performance of the algorithm is measured by standard deviation (\( St.d. \)) and relative percentage index (\( RPI \)). The \( RPI \) is calculated as follows:

\[
RPI = \frac{LB - \text{Best}}{\text{Best}} \times 100
\]  

As is depicted in Table 2, among the 15 cases problems, the proposed algorithm obtains all the best solutions as so far, including 5 brand new best solutions (in bold). Especially, 4 out of 5 new best solutions are equal to \( LB \). In addition, the HABC algorithm outperforms in 11 cases over LAHC algorithm and HTLBO1 algorithm and 5 over HTLBO2 algorithm. However, the HABC algorithm have two slightly worse solutions for P205 with the cycle time 2643 and 2832. The \( St.d. \) and \( RPI \) demonstrated the proposed algorithm is stable and effective for solving the TALBP.

### Table 2. Comparison of the performance for TALBP with assignment constraints.

| Problem | Cycle time | \( LB(Ns) \) | LAHC \((Nm[Ns])\) | HTLBO1 \((Nm[Ns])\) | HTLBO2 \((Nm[Ns])\) | HABC \(\text{Best}(Nm[Ns])\) | \(\text{Mean}(Ns)\) | \(\text{St.d.}\) | \(RPI\) |
|---------|------------|--------------|-----------------|-----------------|-----------------|-----------------|----------------|--------|-------|
| P65     | 326 16     | 10[20]       | 19[17]          | 9[17]           | 17.6 0.49 0.10  | 11[21]         | 22.7 0.46 0.08 |        |       |
|         | 381 14     | 8[16]        | 17[15]          | 8[15]           | 15.1 0.30 0.08  | 11[14]         | 13.0 0.50 0.05 |        |       |
|         | 435 12     | 7[14]        | 14[13]          | 7[13]           | 13.0 0.00 0.08  | 6[11]          | 11.5 0.50 0.05 |        |       |
|         | 490 11     | 6[12]        | 13[12]          | 6[12]           | 11.5 0.50 0.05  | 6[11]          | 11.0 0.00 0.10 |        |       |
|         | 544 10     | 6[11]        | 11[11]          | 6[11]           | 11.0 0.00 0.10  | 6[11]          | 11.0 0.00 0.10 |        |       |
| P148    | 2832       |              |                 |                 |                 |                 |                 |        |       |
|         | 255 21     | 13[25]       | 24[22]          | 11[22]          | 22.7 0.46 0.08  |                 |                 |        |       |
|         | 306 17     | 11[21]       | 20[19]          | 10[19]          | 18.8 0.40 0.11  |                 |                 |        |       |
|         | 357 15     | 9[17]        | 17[16]          | 8[16]           | 15.4 0.49 0.03  |                 |                 |        |       |
|         | 459 12     | 7[14]        | 13[12]          | 6[12]           | 12.0 0.00 0.00  |                 |                 |        |       |
|         | 510 11     | 7[13]        | 12[12]          | 6[11]           | 11.0 0.00 0.00  |                 |                 |        |       |
| P205    | 1888       | 13[8]        | 15[7]           | 7[14]           | 7[14]          | 14.0 0.00 0.08  |                 |        |       |
|         | 2266       | 11[7]        | 13[7]           | 7[13]           | 6[12]          | 12.8 0.40 0.16  |                 |        |       |
|         | 2454       | 10[6]        | 12[6]           | 6[12]           | 12.0 0.00 0.20  |                 |                 |        |       |
|         | 2643       | 9[6]         | 10[6]           | 6[11]           | 6[11]          | 11.3 0.46 0.26  |                 |        |       |
|         | 2832       | 9[6]         | 10[6]           | 6[11]           | 6[11]          | 11.0 0.00 0.22  |                 |        |       |

To verify the authenticity of the experimental results, the Gantt chart of an optimal solution for P65 (\( CT=490 \)) is depicted in Figure 4. In Figure 4, the Y-axis indicate the mated-station and the side of the station, the best solution has six mated-station with five left side stations and six right side stations. The shaded rectangles boxes refer to the task with constraints. In the station “6R”, these five tasks are assigned as follows: 35-54-50-61-65, which means the precedence constraint. And the finish time of these five tasks are 132, 250, 300, 406, 471, respectively. Obviously, all the finish time of these
stations are satisfy the cycle time constraint. Therefore, the number of the mated-station of the P65 \((CT=490)\) is 6, and the number of the station is 11.

![Figure 4. Gantt chart for the best solution of P65 (CT=490).](image)

5. Conclusions

In this paper, a hybrid algorithm combines the ABC algorithm and the LAHC algorithm is suggested to solve the TALBP with three assignment constraints. In the process of optimization, the employed bees use the LAHC with four neighborhood structures to explore new solutions, by contrast, the onlooker bees implement the LAHC with the VNS to further exploit better solutions. The best solutions are close or equal to the \(LB\) value, and the value of the \(St.d.\) and \(RPI\) of the problem demonstrates that the HABC algorithm outperforms the LAHC and the HTLBO for solving the TALBP.

In the future, more realistic constraints, such as the logistics distribution capability and stochastic task times, can be considered. In addition, other objectives like minimizing the total idle time and the multi-objective optimization can be performed in TALBP, mixed-model and multi-model assembly line.

Acknowledgements

The paper is supported by the Changzhou Key Laboratory of High Technology (CM20183004).

References

[1] C Becker and A Scholl 2006 A survey on problems and methods in generalized assembly line balancing EUROPEAN JOURNAL OF OPERATIONAL RESEARCH 168 694-715,

[2] A Baykasoglu 2006 Multi-rule multi-objective simulated annealing algorithm for straight and U type assembly line balancing problems JOURNAL OF INTELLIGENT MANUFACTURING 17 217-232

[3] Z Li, Q Tang and L Zhang 2017 Two-sided assembly line balancing problem of type I: Improvements, a simple algorithm and a comprehensive study Computers & Operations Research 79 78-93

[4] H Alper 2018 Balancing of mixed-model two-sided assembly lines using teaching-learning based optimization algorithm Pamukkale University Journal of Engineering Sciences 24
682-691

[5] Y Lv and J Zhang 2019 A genetic regulatory network based method for multi-objective sequencing problem in mixed-model assembly lines 16 1228-1243

[6] Y K Kim, Y H Kim and Y J Kim 2000 Two-sided assembly line balancing: a genetic algorithm approach PRODUCTION PLANNING & CONTROL 11 44-53

[7] L Ozbakir and P Tapkan 2011 Bee colony intelligence in zone constrained two-sided assembly line balancing problem EXPERT SYSTEMS WITH APPLICATIONS 38 11947-11957

[8] A S Simaria and P M Vilarinho 2009 2-ANTBAL: An ant colony optimisation algorithm for balancing two-sided assembly lines COMPUTERS & INDUSTRIAL ENGINEERING 56 489-506

[9] B Yuan, C Zhang and X Shao 2015 A late acceptance hill-climbing algorithm for balancing two-sided assembly lines with multiple constraints Journal of Intelligent Manufacturing 26 159-168

[10] Q Tang, Z Li, L Zhang, C A Floudas and X Cao 2015 Effective hybrid teaching-learning-based optimization algorithm for balancing two-sided assembly lines with multiple constraints Chinese Journal of Mechanical Engineering 28 1067-1079

[11] Q Tang, Z Li, L Zhang and C Zhang 2017 Balancing stochastic two-sided assembly line with multiple constraints using hybrid teaching-learning-based optimization algorithm Computers & Operations Research 82 102-113

[12] O Polat, Ö Mutlu and E Özgormus 2018 A Mathematical Model For Assembly Line Balancing Problem Type 2 Under Ergonomic Workload Constraint The Ergonomics Open Journal 11 1-10

[13] A Yadav, P Verma and S Agrawal 2019 A Mathematical Formulation for Mixed Model Two Sided Assembly Line Balancing Problem to Consider Boundary Conditions Global Journal of Researches in Engineering 19 17-25