

An Ant Colony Algorithm for Two-Sided Assembly Line Balancing Problem Type-II

Zeqiang Zhang, Junyi Hu and Wenming Cheng

Abstract Two-sided assembly lines are widely used in the assembly of large-sized products, such as automobiles, buses, or trucks. Two-sided assembly line problem is more difficult than single-sided assembly line problem as the operation directions constrain of tasks and the requirement of parallel work in the task distribution procedure. In this paper, the mathematical model of two-sided assembly line balancing problem type-II (TALBP-II) is given first. And then, an improved ant colony algorithm was constructed to solve TALBP-II. In the algorithm, a hybrid ant-based search rule and a heuristic task distribution rule were used in order to establish a feasible solution, global pheromone trail update, and the optimum solution search strategy are also considered. The feasibility of this algorithm was indicated by a case of a loader final assembly line.

Keywords Two-sided assembly lines · Balancing · Ant colony algorithm

1 Introduction

Considering the product's size and the complexity of the assembly line, two-sided assembly lines are widely used in assembly of large-sized products such as automobiles, buses, or trucks. It is very important for manufacturers to design an efficient two-sided assembly line. Two-sided assembly line balancing problem (TALBP) can be divided into two versions: TALBP-I consists of assigning tasks to

Z. Zhang (✉) · W. Cheng
School of Mechanical Engineering, Southwest Jiaotong University,
610031 Chengdu, China
e-mail: zhagnzeqiang@gmail.com

J. Hu
CSR Qishuyan Institute Co., Ltd, 213011 Changzhou, China

work stations, such that the number of open position of the two-sided assembly line is minimized for a given production rate, and TALBP-II is to minimum cycle time for a given number of open position in the assembly line. In actual manufacturing enterprises, the optimization goals often refer to the second category in order to reduce the production cycle time.

TALBP are received widely attention in the past few years for the importance of two-sided assembly lines. Bartholdi [1] first proposed two-sided assembly line balancing problem, and a first-fit rule-based heuristic algorithm is given for TALBP-I. Kim et al. [2] and Lee et al. [3] used genetic algorithm for solving TALBP-I and join the group distribution strategy to optimize the correlation indicators between tasks. Simaria and Vilarinho [4] used ant colony optimization algorithm, named 2-ANTBAL, for solving TALBP-I. Baykasoglu and Dereli [5] first introduced the concept of regional constraints in TALBP-I and then proposed an ant colony algorithm to solve the problem. Wu and Hu proposed a branch-and-bound algorithm for TALBP-I [6].

Compare to TALBP-I, TALBP-II is less studied [7]. Ozcan [8] presented a Petri net-based heuristic to solve simple assembly line balancing problem type-II. To solve TALBP-II, Kim et al. [9] presented a mathematical model and a genetic algorithm, which adopted the strategy of localized evolution and steady-state reproduction to promote population diversity and search efficiency. In this paper, a new ant colony algorithm is proposed to solve the two-sided assembly line balance problem as the common problems of manufacturing enterprises, respectively, using the ant colony comprehensive search rules and task assignment rules to select and assign tasks. Considering the special features of TALBP-II, an ant colony the overall search process is introduced in this paper.

This paper is organized as follows: The TALBP-II is discussed in Sect. 2, and mathematical model is presented. In Sect. 3, the modified ant colony algorithm for TALBP-II is explained in detail. An example problem of a loader final assembly line is solved using the proposed method in Sect. 4. Finally, some conclusions are presented in Sect. 5.

2 Problems Statement and Formulation

As shown in Fig. 1, in two-sided assembly line, both sides of the station can operate different tasks in parallel. Stations 1 and 2 can be described as a pair of

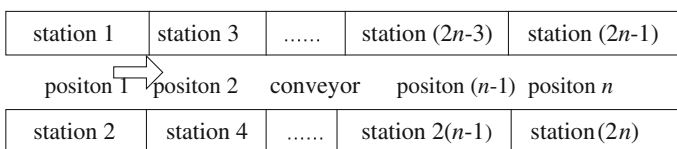


Fig. 1 Two-sided assembly line

stations (mated-station), the station 1 can be called an accompanied station of the station 2 (a companion), and the remaining work stations have the similar relationship. Tasks can be operated simultaneously when they do not have priority constraints.

The mathematical model of two-sided assembly line balancing problem can be described as below. First, list the relevant variables: I —task set, $I = \{1, 2, \dots, N\}$; I_D —to split the task set by the task operating position, $D = L(\text{left}), R(\text{right}),$ or $E(\text{both sides}), I_E$ refers to the tasks can be assigned to both sides of the position, and so on; I'_j, I''_j —sets of tasks assigned to station j and j' ; T_j, R_j —the total operating time and total delay time of station j ; T'_j, R'_j —total operating time and total delay time of station j' ; $P(i)$ —set of all predecessors of task i ; S —set of all the task i whose $P(i)$ is empty, all this tasks can be used for distribution into the station because their pre-order tasks are already located; S' —set of tasks meet the cycle time constraints from S , the tasks can be assigned to the station directly; W —set of unassigned tasks; n —the number of opened position; N_S —opened work stations; x_{ipk} —the value is 1, if task i is assigned to position p , and the operating position is $k, k \in \{L, R\}$; 0, otherwise.

Constraints:

$$\text{If } x_{ipk} = 1, \text{ then } x_{hgk} = 0, h \in P(i), g, p = 1, 2, 3, \dots, n, \text{ and } g > p \quad (1)$$

$$\sum_{i \in I} \sum_{p=1}^n x_{ipk} = 1, k \in \{L, R\} \quad (2)$$

$$T_j = \sum_{i \in I} x_{ipk} \cdot t_i, \text{ and } k = L, j = 2p - 1 \quad (3)$$

$$T'_j = \sum_{i \in I} x_{ipk} \cdot t_i, \text{ and } k = R, j' = 2p \quad (4)$$

$$\text{In any position } p, T_j + R_j \leq C, \text{ and } T'_j + R'_j \leq C \quad (5)$$

Objective:

$$\text{Minimize } C \quad (6)$$

$$\max \text{LE} = \left(\sum_{i \in I} t_i \right) / (N_S C) \quad (7)$$

$$\min \text{SI} = \sqrt{\frac{\sum_{i=1}^{N_S} (T_{\max} - T_i)^2}{N_S}} \quad (8)$$

Lower limit formulation of cycle time:

$$C_{\min} = \max\{|T_{\text{SUM}}/2n|, |T_{\text{SUML}}/n|, |T_{\text{SUMR}}/n|, t_{\max}\} \quad (9)$$

Constraint (1) is the assignment constraint which ensures that each task is assigned to exactly one station and all precedence relations between tasks are satisfied. Constraint (2) ensures the task must be assigned to a position p , and the allocated operating position only can be left or right. Constraints (3) and (4), respectively, refers to total operating time of station j and j' . Constraint (5) refers to the operating time of any mated-stations in any position p must satisfy the cycle time constraints. Objective function (6) as the main objective function this paper focused on TALBP-II, that is to minimize the cycle time when the number of positions opened in the assembly lines is determined. Functions (7) and (8) are secondary evaluation objective, to maximum line efficiency LE and minimum smoothness index SI can balance the stations' load. When $LE = 1$ and $SI = 0$, every position has the same load. Formulation (9) is lower limit formulation of cycle time for TALBP-II; T_{SUM} is the sum of task time in the set I ; T_{SUML} , T_{SUMR} , and T_{SUME} were the sum operating time of all tasks in I_L , I_R , and I_E , respectively; t_{max} is the task which has the longest operating time.

3 The Ant Colony Algorithm for TALBP-II

3.1 Ant Colony Comprehensive Search Rules

Selected tasks from the set S which have the earliest start time. These tasks constitute FT (the tasks in FT have the same earliest start time); if FT has only one task, select this task by task assignment rules. If FT has more than one tasks, then use the hybrid search mechanism to select the task, this mechanism borrow ideas called improved pheromone rule from the literature [10]:

$$i = \begin{cases} I_1 : p_{ij} = \frac{\left(\sum_{k=1}^j \tau_{ih}\right)^\alpha (\eta_i)^\beta}{\sum_{s \in FT} \left(\sum_{h=1}^j \tau_{sh}\right)^\alpha (\eta_s)^\beta} & 0 \leq r \leq r_1 \\ I_2 : \text{random selection of } i \in FT & r_1 < r \leq 1 \end{cases} \quad (10)$$

r —random number between (0, 1); r_1 —the user-defined parameter to meet $0 \leq r_1 \leq 1$; FT —in the current sequence position j , the tasks that ant can be selected with the earliest start time; α , β —parameters determine the relative importance of pheromone intensity and heuristic information; η_i —the heuristic information of task i , using pw_i that is the position weight of task i as heuristic information:

$$\eta_i = pw_i = t_i + \sum_{j \in F_i} t_j, \quad i = 1, \dots, N \quad (11)$$

F_i —set of successor tasks of task i . Position weight heuristic rules account the sum of successor task operating time and the largest number of successor tasks

simultaneity, tasks with largest successor task numbers and longer sum operating time of successor tasks has the greater probability of being select. In formulation (10), the random variable r determines using the rules below to select a task form FT : If the random number r satisfies $0 \leq r \leq r_1$, then randomly select a task from the FT with the probability p_{ij} ; if the random number r satisfies $r_1 < r \leq 1$, then randomly choose a task from the candidate FT .

3.2 Heuristic Task Distribution Rules

Rule 1: If the task’s operating position constraint is L (or R), it is placed on the left (or right) station; Rule 2: If this task’s operating position constraint is E , then put it into the side which it can began earlier; if it placed on both sides with the same start time, then selected one side randomly.

3.3 Construct the Feasible Solution in Ant Colony Algorithm

As shown in Fig. 2, the open position is the determined value n , and the optimization goal is to minimize the cycle time. In order to ensure that all tasks can be completely distributed, in the first $n - 1$ positions, the constraints of the optimal cycle time are used; in the final position of n , eliminating the cycle time constraint to assign all tasks into this location to meet the total number of the open position is exactly n , and feasible solutions are generated. Find out the latest completion time from all positions in this feasible solution as the cycle time searched by this ant.

3.4 Pheromone Updating Rule

Local pheromone update: The role is to reduce the attractive effect from earlier ants to the later one, thereby this method strengthens the random search capabilities of ants. When the ant assigned task i to the arranged location j , the pheromone in edge ij updated as formulation (12):

$$\tau_{ij} \leftarrow (1 - \rho_1)\tau_{ij} + \rho_1\tau_0 \tag{12}$$

ρ_1 local pheromone evaporation coefficient, $0 < \rho_1 < 1$;

τ_0 initialized pheromone value, $\tau_0 = 1/(N \cdot K^*)$, $K^* = \left[\left(\sum_{i \in I} t_i \right) / C_{\min} \right]$.

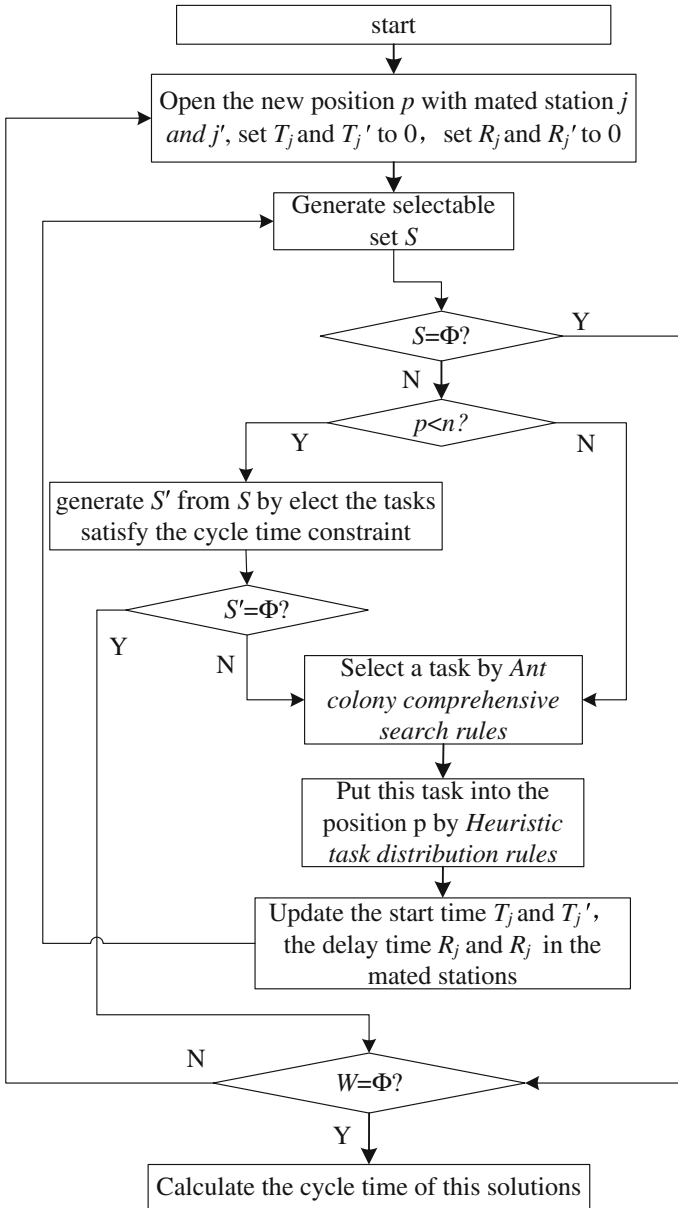


Fig. 2 Assignment of tasks to construct an ant solution

Global pheromone update rule: As the establishment of an optimal solution set for each generation of ants' search results in this paper, every optimal solution in

the collection releases pheromone to the path. The global pheromone update formula is as follows:

$$\tau_{ij} \leftarrow (1 - \rho_2)\tau_{ij} + \rho_2\Delta\tau_{ij} \tag{13}$$

where

$$\Delta\tau_{ij}^{gb} = \begin{cases} 1/N_S, & \text{if } (i, j) \in \text{global-best-tour} \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

ρ_2 is global pheromone evaporation coefficient, $0 \leq \rho_2 \leq 1$

The whole search plan of ant colony algorithm: The algorithm starts search the optimal solution at the theoretical minimum cycle time, C_{min} . The first generation of ants search from the smallest cycle time C_t ($C_t = C_{min}$) at the start search. If it do not find the solution of C_t , the standard value C_t increased by 1 in the next generation of ant colony search, $C_t = C_t + 1$; if a generation of ant colony obtained the current search criteria C_t , then the next generation of search will narrow the search criteria, that is, $C_t = C_t - 1$; the entire ant colony uses this plan to search for the theoretical optimal solution until find out the optimal solution or the maximum search iteration is reached.

4 Computational Study

Precedence diagram of a loader assembly line [11] is given in Fig. 3. The proposed ant colony algorithm was coded in MATLAB 7.8 and run on a Notebook PC equipped with Intel Core i5-2410 M 2.30 GHz and 4 GB memory. In the proposed ACO, several parameters, such as number of ants n_{ant} , maximize iteration number NC_{max} , α , β , ρ and e , affect the performance of the algorithm. Excessive experimental tests were conducted and found out the following value of parameters are more appropriate: $n_{ant} = N$, $NC_{max} = 400$, $\alpha = 1$, $\beta = 2$, $\rho_1 = 0.1$, $\rho_2 = 0.1$, $r_1 = 0.8$. By changing the number n (number of opened positions in the assembly line), the computational study shows the four kinds of optimization solutions as shown in Table 1. Figure 4 is the Gantt chart output by MATLAB as the second optimization solution in Table 1.

In Table 1, four kinds of optimization results are all with a good balance rate. Assignment schedules of scheme 2 for the test problem generated by ACO refer to Table 2. To increase the practicality of the algorithm and the intuitive of the results, the Gantt chart display module was added to this program, refer to Fig. 3. The progress bar's length is in accordance with the tasks' operation time, and the number on top of the progress bar is the start time and end time of tasks; the horizontal axis is for the timeline for the task start and end time, the vertical axis is for the station, which on the left station of every position is odd, the right station in

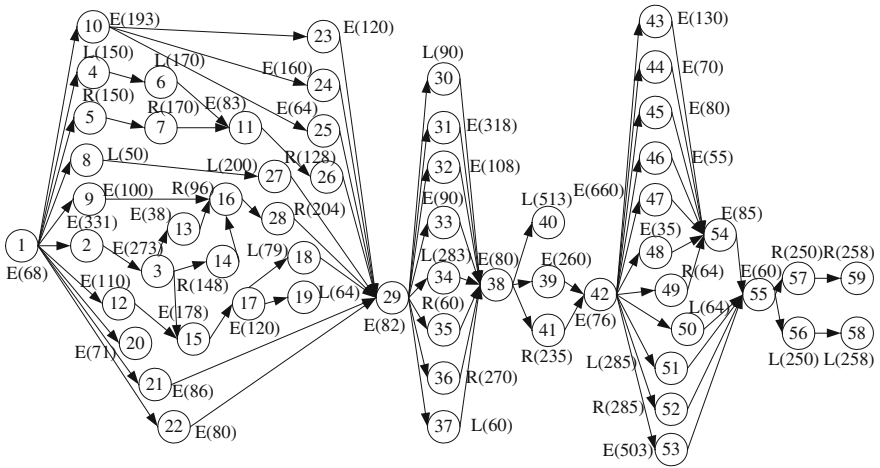


Fig. 3 Precedence diagram of a loader assembly line

Table 1 Four types of assignments for the test problem generated by ACO

Scheme	C(s)	LE (%)	SI (s)
Scheme 1(5)	1,021	94.0	79.73
Scheme 2(6)	860	93.0	70.22
Scheme 3(7)	771	89.0	152.47
Scheme 4(8)	660	91.0	81.01

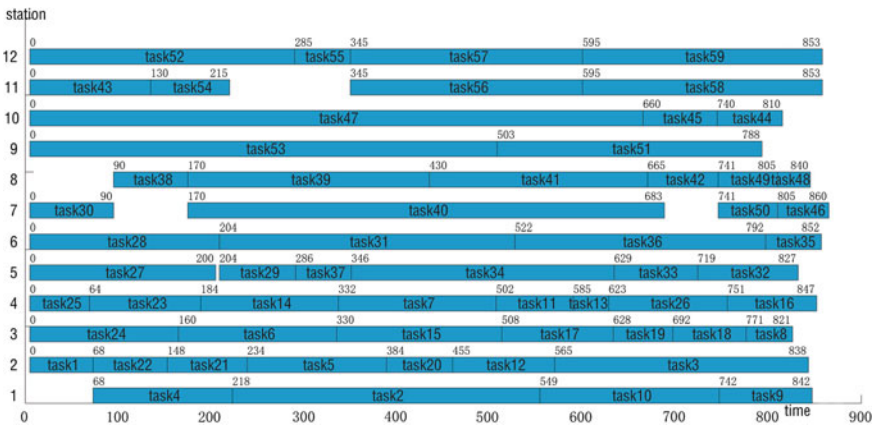


Fig. 4 Gantt chart of assignment of tasks for the example problem

Table 2 Assignment schedules of scheme 2

Work station	Assigned tasks	Station (waiting) time (sec)
1 (1L)	4, 2, 10, 9	774 (68)
2 (1R)	1, 22, 21, 5, 20, 12, 3	838
3 (2L)	24, 6, 15, 17, 19, 18, 8	821
4 (2R)	25, 23, 14, 7, 11, 13, 26, 16	847
5 (3L)	27, 29, 37, 34, 33, 32	823 (4)
6 (3R)	28, 31, 36, 35	852
7 (4L)	30, 40, 50, 46	722 (138)
8 (4R)	38, 39, 41, 42, 49, 48	750 (90)
9 (5L)	53, 51	788
10 (5R)	47, 45, 44	810
11 (6L)	43, 54, 56, 58	723 (130)
12 (6R)	52, 55, 57, 59	853

the same position is even. The time delayed by the pre-order tasks in the mated-station is clear in Fig. 1, so the Gantt chart display module has obvious practical significance.

5 Conclusions

In this paper, an ant colony algorithm was introduced to solve two-sided assembly line balance problem type-II. An ant colony search rules, heuristic task distribution rules were used for generate a feasible solution. The whole search plan of ant colony algorithm controls the optimal solution search process. Computational experiment demonstrated the validity of the proposed algorithm. In the future studies, we hope to apply the method to extensions of the TALBP-II, such as multiple-object TALBP-II.

Acknowledgments This research was partially supported by the National Natural Science Foundation of China (No. 51205328), the Youth Foundation for Humanities and Social Sciences of Ministry of Education of China (No. 12YJJCZH296), the PhD Programs Foundation of Ministry of Education of China (No. 200806131014), the Foundation of Sichuan Province Cyclic Economy Research Center (No. XHJJ-1205), and the Fundamental Research Funds for the Central Universities (No. SWJTU09CX022; 2010ZT03).

References

1. Bartholdi JJ (1993) Balancing two-sided assembly lines: a case study. *Int J Prod Res* 31:2447–2461
2. Kim YK, Kim YH, Kim YJ (2000) Two-sided assembly line balancing: a genetic algorithm approach. *Prod Plan Control* 11:44–53

3. Lee TO, Kim Y, Kim YK (2001) Two-sided assembly line balancing to maximize work relatedness and slackness. *Comput Ind Eng* 40:273–292
4. Simaria AS, Vilarinho PM (2009) 2-ANTBAL: an ant colony optimisation algorithm for balancing two-sided assembly lines. *Comput Ind Eng* 56:489–506
5. Baykasoglu A, Dereli T (2008) Two-sided assembly line balancing using an ant-colony-based heuristic. *Int J Adv Manuf Technol* 36:582–588
6. Wu EF, Jin Y, Bao JS, Hu XF (2008) A branch-and-bound algorithm for two-sided assembly line balancing. *Int J Adv Manuf Technol* 39:1009–1015
7. Scholl A, Becker C (2006) State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *Eur J Oper Res* 168:666–693
8. Ozcan K (2010) A Petri net-based heuristic for simple assembly line balancing problem of type 2. *Int J Adv Manuf Technol* 46:329–338
9. Kim YK, Song WS, Kim JH (2009) A mathematical model and a genetic algorithm for two-sided assembly line balancing. *Comput Oper Res* 36:853–865
10. Zhang Z-Q, Cheng W-M, Zhong B, Wang J-N (2007) Improved ant colony optimization for assembly line balancing problem. *Comput Integr Manuf Syst CIMS* 13:1632–1638
11. Wu E (2009) Research on balancing two-sided assembly line. School of mechanical engineering, vol 120. Doctor of Science. Shanghai Jiao Tong University, Shanghai (2009)