

Research Article

A Novel Assembly Line Balancing Method Based on PSO Algorithm

Xiaomei Hu, Yangyang Zhang, Ning Zeng, and Dong Wang

Shanghai Key Laboratory of Intelligent Manufacturing and Robotics, School of Mechatronic Engineering and Automation, Shanghai University, Mailbox 232, No. 149 Yanchang Road, Shanghai 200072, China

Correspondence should be addressed to Xiaomei Hu; sufeimasohxm@163.com

Received 19 January 2014; Accepted 31 March 2014; Published 30 April 2014

Academic Editor: Jonathan N. Blakely

Copyright © 2014 Xiaomei Hu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Assembly line is widely used in manufacturing system. Assembly line balancing problem is a crucial question during design and management of assembly lines since it directly affects the productivity of the whole manufacturing system. The model of assembly line balancing problem is put forward and a general optimization method is proposed. The key data on assembly line balancing problem is confirmed, and the precedence relations diagram is described. A double objective optimization model based on takt time and smoothness index is built, and balance optimization scheme based on PSO algorithm is proposed. Through the simulation experiments of examples, the feasibility and validity of the assembly line balancing method based on PSO algorithm is proved.

1. Introduction

With the progress and development of science and technology, manufacturing has transformed from the simple process documented by individual behavior and single machine to the complicated one conducted by manufacture system composed of many manufacture components [1]. Assembly line is an effective combination of man and machine, and it is the most widely used production mode in the manufacturing industry. The advanced manufacturing models, such as Flexible Automation Manufacturing, Agile Manufacturing, JIT Manufacturing, and Networked-Manufacturing, are constantly emerging. Manufacturing technology develops towards systematical, flexible, integrated, reconfigurable, networked, intelligent, green renewable, and other directions [2]. The transformation of manufacturing mode and manufacturing technology has influenced the manufacturing system planning, design, operation, and management deeply.

Manufacturing enterprises mainly produce in the continuous production line with multiple and fine-sorted process. Fine-sorted work forms are used in the assembly production. On the one hand, it improves the proficiency of operators and the efficiency of production; on the other hand, it also causes each assembly process not to be able to consume the same

time in theory and reality. Load unbalance in assembly line can lead to workpiece accumulation and even the termination of production line [3–6]. Assembly line balancing is to equalize all the assembly processes by adjusting the work load of each process to make the job time of each process as close as possible. Balancing assembly line is beneficial to enhance the work efficiency of operator and equipment and reduce the consumption of single product working hours [7–9]. Therefore, the study of the assembly line balancing problem has important theoretical value and practical significance.

At present, some studies aim to solve the line balancing problem. Charlton uses mathematical analysis method and proposes the branch and bound method to solve the assembly line balancing [10]. Gao and Sun put forward a measure of action analysis improvement to improve the production rate of balance by reducing the operation time [11]. With the deepening of the research, the intelligent algorithm is considered to solve the production line balance problem recently. Chen and Zhang propose improved ant colony algorithm, and the ant scheme generation measuring method is given to compute homework allocation plan [12], and the research on line balancing problem with certain station numbers is also carried on [13]. However, there have been few researches on multiobjective optimization of production

line balance problem nowadays. Takt time and smoothness index are two important indicators to measure the assembly line balancing. Takt time is referring to the interval time of completing the same two products continuously and smoothness index is a measurement index of the discrete state of location homework time distribution.

Multiobjective optimization of production line balancing problem is studied in this paper. The integrated optimal function of takt time and smoothness index is defined, and a novel optimization method of assembly line balancing based on PSO algorithm is proposed since the PSO algorithm has the advantage of high solution speed, high solution quality, and good robustness [14].

2. The Mathematical Model of Assembly Line Balancing Problem

2.1. The Description of the Assembly Line Balancing Problem

2.1.1. The Main Parameters of Production Line Balancing Problem. The main parameters of production line balancing problem are represented as follows [15].

(1) *Job Element and Standard Work Time.* Job element divides assemble into unit operations, and these unit operations cannot or need not be divided in general. Standard work time is the time to complete an operation of a job element. Job element and standard work time are one to one correspondence relation. $1, 2, 3, \dots, n$ express the job elements and t_j is the standard work time of job element j .

(2) *Production Line Balancing Rate P .* Production line balancing rate P is the expression of the balance of the whole or part of production lines and continuous condition. It is an important index to measure the production line balancing. It can be defined as

$$P = \frac{\sum_{j=1}^n t_j}{m * \max(T_i)}. \quad (1)$$

In (1), t_j is the expression of standard work time of the j job elements, n represents the number of the work elements, m represents the number of total stations in assembly lines, T_i represents the work time in the i station, and $\max(T_i)$ represents the biggest station operating time, namely, the bottleneck station time.

(3) *Takt Time TT .* Takt time TT is the work time between two consecutive products or batches. It can be defined as

$$TT = \frac{Tw}{Q}. \quad (2)$$

In (2), Tw represents the total effective work time in the plan period and Q represents the production quantity in the plan period.

(4) *Smoothness Index SI .* Smoothness index is the evaluation of time distribution of discrete conditions of the whole or part of the production line. It represents the deviation degree of operation time between each station in the whole or part of the production line. The greater the value of SI , the greater

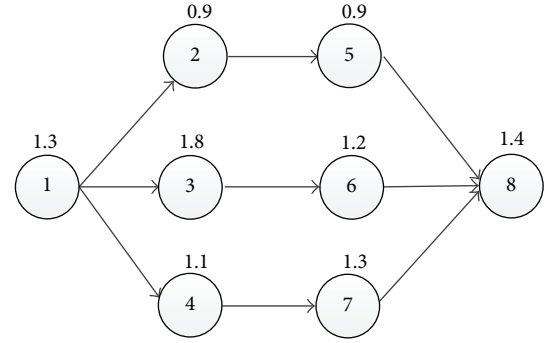


FIGURE 1: Precedence relations diagram.

the homework time distribution of production line location deviation. It can be defined as

$$SI = \frac{\sqrt{\sum_{i=1}^m (TT - T_i)^2}}{m}. \quad (3)$$

(5) *The Precedence Relations Diagram and Relationship Matrix.* The precedence relations diagram is a directed graph G without a loop. It expresses the process sequence of job elements in graphic method according to the priority constraints:

$$G = (E, P). \quad (4)$$

In (4), node E is a set of job elements:

$$E = \{1, 2, 3, \dots, n\}. \quad (5)$$

Arc P represents the successive relationship sets between assembly work in the graph.

Relationship matrix M is the matrix transformed by the priority constraints based on precedence relations diagram. The conversion relationship between the precedence relations diagram and relationship matrix can be expressed as follows.

When $M_{ij} = 1$, the job element i is the first operation before the job element j . At this time, the station of the job element i must be arranged before the job element j [16]. The precedence relations diagram of a simple assembly line with eight operating elements is shown in Figure 1. And its relationship matrix is shown in

$$M = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (6)$$

2.1.2. *The Classification of the Line Balancing Problem.* There are three kinds of line balancing problem.

- (1) The objective function of the first kind assembly line balancing problem is to maximize the production line balancing rate P with certain takt time TT . It is equivalent to minimizing the number of stations m with certain takt time. It can be expressed as

$$\max P = \frac{\sum_{j=1}^n t_j}{m * \max(T_i)} \times 100 \iff \min m. \quad (7)$$

- (2) The objective function of the second kind assembly line balancing problem is to maximize the production line balancing rate P with certain station number m . It is equivalent to minimizing the takt time TT with certain location numbers m . It can be expressed as

$$\max P = \frac{\sum_{j=1}^n t_j}{m * \max(T_i)} \times 100 \iff \min TT. \quad (8)$$

- (3) The objective function of the third kind assembly line balancing problem is to minimize the smoothness index SI with certain takt time TT and station numbers m . It can be expressed as

$$\min SI = \frac{\sqrt{\sum_{i=1}^m (TT - T_i)^2}}{m}. \quad (9)$$

The general description of the assembly line balancing problem with certain station numbers is shown as follows [17].

Assume that the assembly line is expressed with a given directed graph without a loop $G = (E, P)$; assembly line balancing problem with certain station numbers m can be described to confirm the division $\{S_k \mid k = 1, 2, 3, \dots, m\}$ in $E = \{1, 2, 3, \dots, n\}$ on the premise of successive relationship.

2.2. *A Multiobjective Optimization Model.* According to the classification description of the assembly line balancing problem, three kinds of assembly line balancing problems belong to optimization problem in fact [18]. Optimizing a certain goal in the process of actual assembly individually may cause the neglect of relevance of balance goals. As a result, the optimization results are often not accord with the actual situation of assembly process [19]. In this paper, the fitness function f_1 and the objective function $\min f_1$ are constructed to realize the comprehensive optimal goals of the takt time TT and smooth index SI in assembly lines in consideration that the layout of assembly line and the station number certain m is fixed.

Fitness function is defined as

$$f_1 = b_1 * TT + b_2 * SI. \quad (10)$$

Constraint conditions are as follows:

$$TT = \max(T_i), \quad (i = 1, 2, \dots, m), \quad (11)$$

$$S_i \cap S_j = \emptyset, \quad (i \neq j, i, j = 1, 2, \dots, m), \quad (12)$$

$$E = \bigcup_{k=1}^m S_k, \quad (k = 1, 2, \dots, m), \quad (13)$$

$$T(S_k) \leq TT, \quad (k = 1, 2, \dots, m), \quad (14)$$

$$\text{if } M_{ij} = 1, \quad i \in A_x, \quad j \in A_y; \quad (15)$$

$$\text{then } x \leq y, \quad M = (M_{ij})_{n \times n}.$$

The fitness function f_1 takes advantage of the linear weighted average model of takt time TT and smoothness index SI in assembly line. The first part of f_1 is used to evaluate the balance of assembly line and the second part of f_1 is used to evaluate the speed of assembly line production. b_1 and b_2 are weight value, and $b_1 + b_2 = 1$. Specific values can also be used according to the specific situation.

In the constraint conditions, n and m are constant; n is the number of job elements; m is the number of location. In (12), there is no overlap between the different stations, because the same job elements could only be distributed in one location. In (13), all the job elements should be assigned [20]. In (14), the maximum operation time of station is the takt time. Formula (15) is used to establish the distribution relations of job elements on the basis of precedence relation matrix.

2.3. *The Solution of the Assembly Line Balancing Problem.* Two distribution solutions are applied to solve the objective function $\min f_1$.

- (1) To solve a problem, a production line balancing problem with certain station numbers, a series of classification schemes of job elements $\{S_k \mid k = 1, 2, 3, \dots, m\}$ with smaller takt time from job elements set $E = \{1, 2, 3, \dots, n\}$ are concluded. A kind of coding based on two-dimensional particle is proposed to express and update the arrangement of operating elements.
- (2) According to classification schemes of job elements S_k in step one, each value of smooth index in job element arrangement scheme is calculated and the classification scheme of job elements $\{S_k \mid k = 1, 2, 3, \dots, m\}$ with minimum value of objective function $\min f_1$ is taken as the optimal arrangement scheme.

3. The Assembly Line Balance Optimization Algorithm Based on PSO

3.1. Particle Swarm Optimization Algorithm

- (1) The basic idea of particle swarm optimization: the particle swarm optimization was first put forward in 1995 by the American social psychologist J. Kennedy

and electrical engineer R. Eberhart on the basis of the study of the behavior of birds group in the early time [21]. PSO treats each individual as a particle which has no weight and volume in n dimensional search space and the particle flies at a certain speed in the search space. The speed is dynamically adjusted based on the flying experience of individual and group.

- (2) The basic mathematical model of PSO algorithm [22–24].

In the n dimensional space, $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$ express a current position of particle i .

$V_i = (v_{i1}, v_{i2}, \dots, v_{in})$ express that the speed v_i is the current speed of particle i .

$P_i = (p_{i1}, p_{i2}, \dots, p_{in})$ express the best experiencing position of particle i ; namely, P_i has the smallest fitness.

Assume that $f(x)$ is the minimized objective function, the best position of particle i is calculated as follows:

$$P_i(t+1) = \begin{cases} P_i(t) & f(P_i(t)) \leq f(X_i(t+1)) \\ X_i(t+1) & f(P_i(t)) > f(X_i(t+1)). \end{cases} \quad (16)$$

Assume that the number of particles is s in population; all particles have experienced the best position $P_g(t)$; then

$$P_g(t) \in \{P_0(t), \dots, P_s(t)\} \mid f(P_g(t)) = \min \{f(P_0(t)), \dots, f(P_s(t))\}. \quad (17)$$

PSO algorithm basic solving equations are as follows [25, 26]. Speed evolution equation is shown as

$$V_i(t+1) = wV_i(t) + c_1r_1(p_i(t) - x_i(t)) + c_2r_2(p_g(t) - X_i(t)). \quad (18)$$

Position evolution equation is shown as

$$X_i(t+1) = X_i(t) + V_i(t+1). \quad (19)$$

In (18), c_1 and c_2 are the acceleration constant, which take the value in $[0, 2]$. r_1 and r_2 both are a random number between 1 and 2; w is inertia coefficient between 0 and 1 and it has the ability of keeping inertia expansion of particle movement to explore new areas.

3.2. Parameter Setting of PSO. The parameters of particle swarm optimization algorithm include population quantity N , the number of iterations G , inertia weight w , accelerated constant c , and the initial position and velocity of each particle.

3.3. Coding of Assembly Line Balancing Problem. The most important job is to code in order to solve the assembly line balancing problem using PSO algorithm. There are many coding methods for the scheduling problem. Among the numerous coding methods, coding based on workpiece,

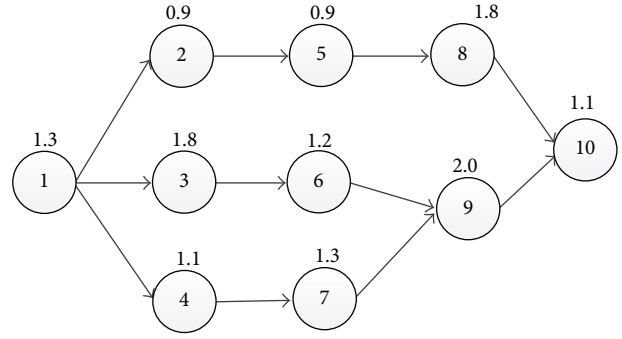


FIGURE 2: Precedence relations diagram with ten job elements.

coding based on machine, and coding based on operation are used in the optimization problem frequently [27–29].

In view of the particularity of assembly line balancing problem, as well as the limitations of general coding method, a coding method based on job elements is adopted in this paper. AOV-net coding basic ideas are used to encode the precedence relations diagram of job element [30].

The basic steps of AOV-net basic coding are as follows.

- (1) Output the node which has no precursors in AOV-net table.
- (2) Remove the selected node and the arc which takes the selected node as starting point from the map; the rest of the vertices still constitute the AOV-net.
- (3) Repeat (1) and (2) until outputting all the nodes, the vertex sequencing is a topological sort.

Since the precedence relations diagram of job element contains two parts, namely, job element and standard work time, a two-dimensional particle expressive method is proposed [31]. Assume the total number of job elements are N ; a two-dimensional particle with length N is created: the first dimension corresponds the order of job elements; the second dimension corresponds standard work time t_i ($i = 1, 2, 3, \dots, N$) of each job element, and each job element has a corresponding standard work time. If the order of job element changes, the arrangement of standard work time also changes.

The precedence relations diagram of an assembly line with ten job elements is shown in Figure 2.

Two particles are coded and shown in Tables 1 and 2.

3.4. The Particle Updating Formula of PSO Algorithm. The particle updating formula is defined as

$$X_i(t+1) = c_2 \otimes g \{c_1 \otimes g [w \otimes h_{x,y}(X_i(t)), pB_i(t)], gB_i(t)\}. \quad (20)$$

In (20), $X_i(t+1)$ express the positions of particle i in the t iterations.

TABLE 1: The code of particle one.

	Job element									
	1	2	3	4	5	6	7	9	8	10
Standard work time (t_i)	1.3	0.9	1.8	1.1	0.9	1.2	1.3	2.0	1.8	1.1

TABLE 2: The code of particle two.

	Job element									
	1	3	4	2	6	5	7	9	8	10
Standard work time (t_i)	1.3	1.8	1.1	0.9	1.2	0.9	1.3	2.0	1.8	1.1

c_1 and c_2 are the acceleration constant, and w is the inertia coefficient. Moreover, their values are between 0 and 1. $h_{x,y}(X_i(t))$ express the exchange of the x component and the y component. x and y are two different random integers between 1 and n (the number of job elements). $pB_i(t)$ is the best position of particle i in t iterations. $gB_i(t)$ is the best position of group in t iterations.

The formula (20) includes three parts.

The first part is the updating of particle position:

$$V_i(t) = w \otimes h_{x,y}(X_i(t))$$

$$= \begin{cases} h_{x,y}(X_i(t)) & \text{rand}() < w \\ X_i(t) & \text{rand}() \geq w. \end{cases} \quad (21)$$

The second part is the adjustment of the particle $pB_i(t)$ based on its best position:

$$F_i(t) = c_1 \otimes g(V_i(t), pB_i(t))$$

$$= \begin{cases} g(V_i(t), pB_i(t)) & \text{rand}() < c_1 \\ V_i(t) & \text{rand}() \geq c_1. \end{cases} \quad (22)$$

The third part is the adjustment of the particle based on the best position $gB_i(t)$ in the group:

$$X_i(t+1) = c_2 \otimes g(F_i(t), gB_i(t))$$

$$= \begin{cases} g(F_i(t), gB_i(t)) & \text{rand}() < c_2 \\ F_i(t) & \text{rand}() \geq c_2. \end{cases} \quad (23)$$

$\text{rand}()$ is a random uniform distribution number. The meaning of function $g(V_i(t), pB_i(t))$ is the adjustment of particles according to its best position in the i iteration. The specific implementation processes are as follows.

- (1) When $\text{rand}() \geq c_1$, $F_i(t) = V_i(t)$.
- (2) When $\text{rand}() < c_1$, execute $g(V_i(t), pB_i(t))$. Take $pB_i^t(t)$ as the best position in the i iteration; n is the number of job elements. An integer from 1 to n is

randomly selected to make $pB_i^t(t)$ be divided into two sets $\{S(1)_{i,j}\}, \{S(2)_{i,j}\}$.

- (3) An expressive method of two-dimensional particle is adopted to update the arrangement of job elements. $x_{1,j}(t)$ express the job elements arrangement of $V_i(t)$ and $z_{1,l}(t)$ expresses the new arrangement $F_i(t)$. Take $l = 1, j = 1$.
- (4) If $x_{1,j}(t) \in S(1)_{1,j}$, $z_{1,l}(t) = x_{1,j}(t)$; if $pB_i(t) \in S(2)_{1,j}$, $z_{1,l}(t) = pB_i(t)$, $l = l + 1, j = j + 1$.
- (5) If $l = n$, output $z_{i,l}(t)$ and $F_i(t) = z_{i,l}(t)$.
- (6) The formula $g(F_i(t), gB_i(t))$ expresses the adjustment of particle i according to the optimal particle $gB_i(t)$ in group, and specific operation is the same as the function $g(V_i(t), pB_i(t))$.
- (7) If the new particle $X_i(t+1)$ can satisfy the relationship matrix, this iteration is successful.

3.5. *The Specific Steps of PSO Algorithm.* The specific steps of PSO algorithm are as follows.

- (1) Algorithm initialization: set the initialization parameters w, c_1, c_2, b_1, b_2 , confirm the number of iterations G , and generate N initialized population.
- (2) The initialization of $pB_i(t)$ and $gB_i(t)$: $pB_i(t)$ is the local optimal particle and $gB_i(t)$ is the best particle in N population. Take the N population which is generated initially as $pB_i(t)$, and calculate each particle's fitness according to the objective function $\min f_1$ to confirm $gB_i(t)$.
- (3) Use a part of the best particle $pB_i(t)$ instead of a part of the carrier particle in the current iteration population in order to ensure the diversity of population during iteration.
- (4) In G iterations, update each particle's position according to the particle update formula, calculate the takt time of each particle, select these particles with small takt time, and calculate their smoothness index. Determine the value of the new $pB_i(t)$ and $gB_i(t)$ according to the fitness function $\min f_1$ and complete the update of $pB_i(t)$ and $gB_i(t)$.
- (5) End the iteration and output the result after G iterations.

The operation process of assembly line balancing problem based on PSO algorithm is shown in Figure 3.

4. The Results of Simulation and Analysis

4.1. *Operating Environment.* The configuration of computer is CPU Intel Core i7, CPU Clock Speed 2.60 GHz, internal storage 12.00 GB, and windows 7 operating system. Matlab is used to realize the PSO algorithm.

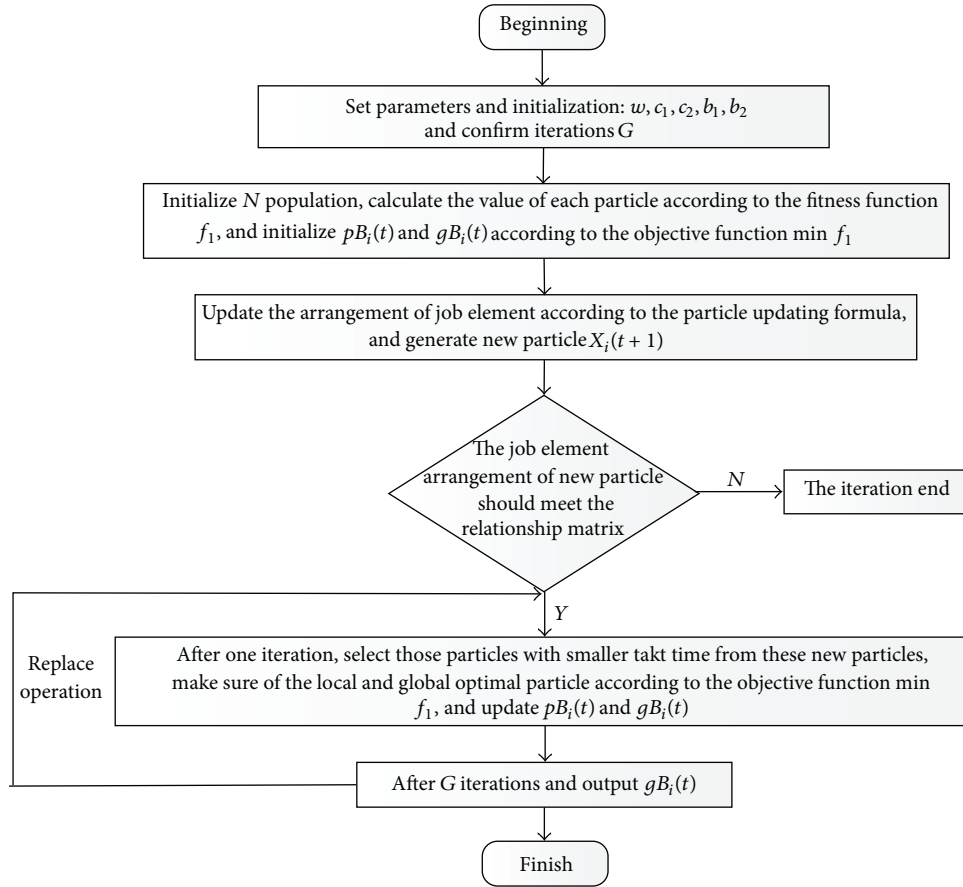


FIGURE 3: The operation process of assembly line balancing problem based on PSO algorithm.

4.2. Example 1. Take a simple assembly line with 12 job elements and 5 stations as an example. The precedence relations diagram is shown in Figure 4, and the relationship matrix is shown in (24). The PSO algorithm, Manual Balance Search algorithm, and Tabu Search algorithm are used to optimize the assembly line and the optimization results are compared in Table 3. The parameters used in PSO algorithm are as follows: population $N = 5$; iteration number $G = 100$; $w = 0.5$; $c_1 = 0.8$; $c_2 = 0.8$; $w_1 = 0.5$; $w_2 = 0.5$. One has the following:

$$M = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (24)$$

From Table 3, PSO algorithm has the smallest fitness f_1 , takt time TT, smoothness index SI, and the largest

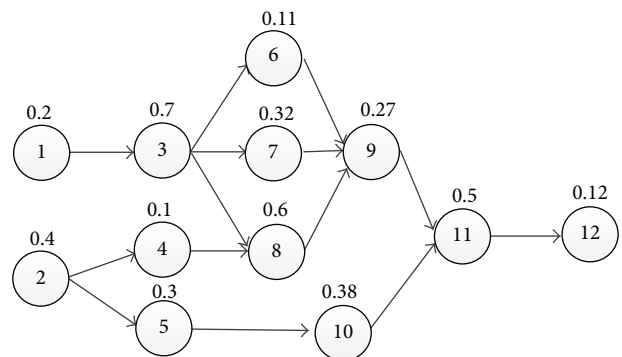


FIGURE 4: Precedence relations diagram with 12 job elements.

balance rate P , which shows the effectiveness of PSO algorithm.

4.3. Example 2. In the APXV9R20B antenna assembly shop, there are 8 work stations in each assembly line, and there are 28 job elements during the assembly of antenna. The job element of antenna assembly and standard work time is shown in Table 4. The precedence relations diagram of antenna relationship matrix with 28 job elements is shown in Figure 5. The relationship matrix is in the following:

TABLE 4: The job element of antenna assembly.

Operation sequence	Number job element	Standard work time (s)
1	Put the chassis on the assembly line and clean	3.4
2	Install the high frequency oscillator	6.96
3	Install the wall components	5.29
4	Install the input cable components	2.58
5	Install the high frequency isolation	12.13
6	Install the low frequency isolation	3.9
7	Install high frequency isolation block	4.2
8	Install low frequency oscillator	10.46
9	Low frequency cable wiring	6.12
10	High frequency cable wiring	10
11	Install low frequency ground plate	12.59
12	The high frequency oscillator cable welding to the oscillator	21.8
13	Install high frequency ground plate	12.98
14	Install the low frequency oscillator cable in place	9.29
15	Install low frequency switching	4.38
16	Low frequency cable welding and cleaning	28.89
17	Install the high frequency oscillator cable in place	8.43
18	The high frequency cable welding and cleaning	22.66
19	Place the low frequency phase shifter components	1.76
20	Place the high frequency phase shifter components	1.63
21	Install low frequency flat	6.14
22	Install high frequency flat	4.97
23	Low frequency plate welding	26.90
24	High frequency plate welding	33.43
25	Install phase shift	11.63
26	Fixed the guide box and the front panel	5.23
27	The support installation	3.07
28	Install support	7.37

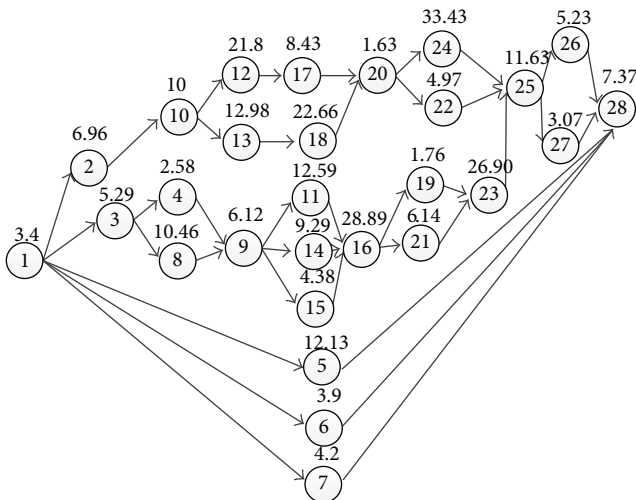


FIGURE 5: Precedence relations diagram with 28 job elements.

Balance Search and Tabu Search algorithm, because it has the smallest takt time TT, smoothness index SI, and the largest balance rate P.

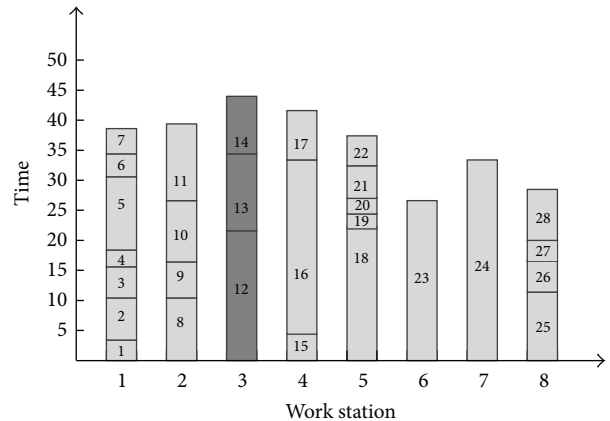


FIGURE 6: The optimization result of manual balance search.

5. Conclusion

The assembly line balancing problem is a key question in the field of assembly line design and management. The PSO algorithm is applied to solve the assembly line balancing problem with optimization goals smoothness index SI and

TABLE 5: The 28 job elements assembly balancing comparison result.

Station	PSO		Manual Balance Search		Tabu Search algorithm	
	Job Element j	Time T_j	Job Element j	Time T_j	Job Element j	Time T_j
1	1, 5, 3, 2, 8	38.24	1, 2, 3, 4, 5, 6, 7	38.46	1, 2, 3, 4, 8, 10	38.69
2	4, 10, 12	34.38	8, 9, 10, 11	39.17	6, 7, 9, 12	36.02
3	9, 14, 15, 11	32.38	12, 13, 14	44.07	11, 13, 14	34.86
4	13, 18	35.64	15, 16, 17	41.7	15, 16, 19	35.03
5	16, 21, 19	36.79	18, 19, 20, 21, 22	37.16	17, 18, 20, 21	38.86
6	17, 20, 23	36.96	23	26.9	22, 24	38.4
7	24, 22	38.40	24	33.43	23, 25	38.53
8	7, 25, 6, 27, 26, 28	35.40	25, 26, 27, 28	27.3	5, 26, 27, 28	27.8
TT		38.40		44.07		38.86
P		93.81%		81.74%		92.70%
SI		3.035		9.988		4.491
f_1		20.72		27.03		21.68

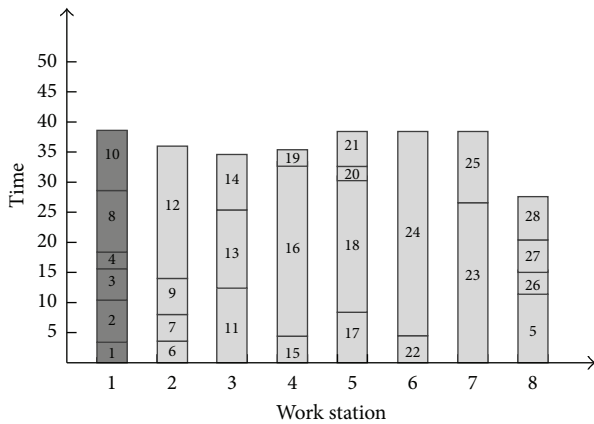


FIGURE 7: The optimization result of Tabu Search algorithm.

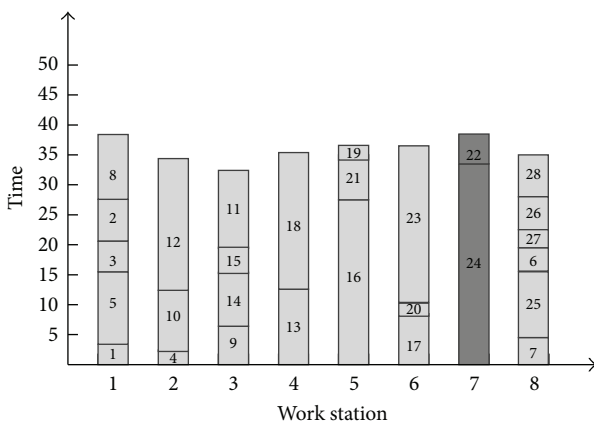


FIGURE 8: The optimization result of PSO algorithm.

takt time TT. The simulation results show that the PSO algorithm can optimize assembly line balancing problem with the highest assembly line balance rate and the smallest discrete conditions than the other two methods, which shows

the effectiveness of the algorithm. However, in the PSO algorithm, particles are easy to lose diversity and lead to premature after several iterations. Therefore, the study of combining PSO algorithm with other algorithms to overcome the defects in the PSO algorithm is the focus of study in the future.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by NSFC (Project no. 41101454), the Grand Science & Technology Program, Shanghai, China (no. 13111101300), and Industrial Innovation Grand Projects (no. 07CH-008).

References

- [1] C.-H. Chen, *Research on Line Balancing Problem of Manufacturing Enterprises Based on Simulation Optimization*, Hefei University of Technology, 2007.
- [2] R. Lujic, G. Simunovic, T. Saric, and N. Majdandzic, "Applying artificial intelligence to the scheduling problem in the ERP system," in *Proceedings of the 27th International Conference on Information Technology Interfaces (ITI '05)*, pp. 149-153, June 2005.
- [3] Z. Qiong, C. Xue-Fang, T. Shi-Yong, and Z. Jie, "The application and research of a simulation-based streamline balance method," *Industrial Engineering and Management*, vol. 13, no. 4, pp. 110-113, 2008.
- [4] D.-M. Lei and Z.-M. Wu, "Pareto archive multi-objective particle swarm optimization," *Pattern Recognition and Artificial Intelligence*, vol. 19, no. 4, pp. 475-480, 2006.
- [5] M. Reimann, "Guiding ACO by problem relaxation: a case study on the symmetric TSP," in *Hybrid Metaheuristics*, vol. 4771 of *Lecture Notes in Computer Science*, pp. 45-56, Springer, 2007.

- [6] M. Guntsch and M. Martin, "Applying population based ACO to dynamic optimization problems," in *Ant Algorithms*, vol. 2463 of *Lecture Notes in Computer Science*, pp. 111–122, Springer, 2002.
- [7] W. Lin, *Production Planning and Scheduling Method Research Under Multi-Objectives Multi-Constrains Environment*, Donghua University, 2008.
- [8] W.-J. Xia, Z.-M. Wu, W. Zhang, and G.-K. Yang, "Application of particle swarm optimization in the job-shop scheduling problem," *Journal of Shanghai Jiaotong University*, vol. 39, no. 3, pp. 381–385, 2005.
- [9] X. Xu, *A Thesis Submitted in Fulfillment of the Requirements for the Degree of Master of Management Science*, Jiangsu University of Science And Technology, 2010.
- [10] O. Guschinskaya, A. Dolgui, N. Guschinsky, and G. Levin, "A combined heuristic approach for optimization of a class of machining lines," in *Proceedings of the IEEE Conference on Automation Science and Engineering (IEEE-CASE '05)*, pp. 154–159, August 2005.
- [11] G.-Z. Gao and J.-H. Sun, "The application of program analysis method in streamline balance," *Machinery Design and Manufacture*, vol. 7, pp. 204–206, 2007.
- [12] J.-X. Chen and Q.-S. Zhang, "The application of Ant colony algorithm in assembly line balancing problem," *The Computer Age*, vol. 12, pp. 20–22, 2008.
- [13] T. Xu, *Study on Assembly Line Balance*, Shang Hai Jiao Tong University, 2011.
- [14] J.-C. Zeng and Z.-H. Cui, "A guaranteed global convergence particle swarm optimizer," *Journal of Computer Research and Development*, vol. 41, no. 8, pp. 1333–1338, 2004.
- [15] K. N. McKay, "Unifying the theory and practice of production scheduling," *Journal of Manufacturing Systems*, vol. 18, no. 4, pp. 241–255, 1999.
- [16] X.-F. Gu, *The Research of Job Shop Scheduling Based on Intelligent Algorithm*, Shenyang Ligong University, 2009.
- [17] Q.-J. Fang, Y. Chen, W. Zhuang, and J.-X. Lu, "Multi-line optimal scheduling research based on PSOGA hybrid optimization algorithm," *Modern Manufacturing Engineering*, vol. 5, article 008, 2012.
- [18] F.-B. Shao, *Study and Application of Dynamic Scheduling for Multi-Objective and Flexible Manufacturing in MES*, Shang Hai Jiao Tong University, 2008.
- [19] J. Li and X.-W. Yi, "The job-shop scheduling based on a kind of new GA in ERP," *Manufacture Information Engineering of China*, vol. 11, article 004, 2007.
- [20] A. Ghanbari, H. Esmail, and A.-N. Salman, "An intelligent ACO-SA approach for short term electricity load prediction," in *Advanced Intelligent Computing Theories and Applications, With Aspects of Artificial Intelligence*, vol. 6216 of *Lecture Notes in Computer Science*, pp. 623–633, Springer, 2010.
- [21] X.-L. Lin, A.-P. Li, and B.-S. Chen, "Scheduling optimization of mixed model assembly lines with hybrid particle swarm optimization algorithm," *Industrial Engineering and Management*, vol. 11, no. 1, pp. 53–57, 2006.
- [22] J. Na, "Application of particle swarm optimization with adaptive mutation to job shop scheduling problem and its software implementation," *Information and Control*, vol. 34, no. 3, pp. 365–368, 2005.
- [23] F. Y. H. Ahmed, S. M. Shamsuddin, and S. Z. M. Hashim, "Improved SpikeProp for using particle swarm optimization," *Mathematical Problems in Engineering*, vol. 2013, Article ID 257085, 13 pages, 2013.
- [24] W. Wei, J.-R. Tan, Y.-X. Feng, and R. Zhang, "Multi-objective optimization method research on flexible job shop scheduling problem," *Computer Integrated Manufacturing Systems*, vol. 15, no. 8, pp. 1592–1598, 2009.
- [25] D. Sendrescu, "Parameter identification of anaerobic wastewater treatment bioprocesses using particle swarm optimization," *Mathematical Problems in Engineering*, vol. 2013, Article ID 103748, 8 pages, 2013.
- [26] Y. Ma, M. Zamirian, Y. Yang, Y. Xu, and J. Zhang, "Path planning for mobile objects in four-dimension based on particle swarm optimization method with penalty function," *Mathematical Problems in Engineering*, vol. 2013, Article ID 613964, 9 pages, 2013.
- [27] X.-H. Li, *Research on Job-Shop Scheduling Problem Based on Particle Swarm Optimization Algorithm*, Wuhan University of Science and Technology, 2009.
- [28] S. Talatahari, E. Khalili, and S. M. Alavizadeh, "Accelerated particle swarm for optimum design of frame structures," *Mathematical Problems in Engineering*, vol. 2013, Article ID 649857, 6 pages, 2013.
- [29] M. Khalilzadeh, F. Kianfar, A. Shirzadeh Chaleshtari, S. Shadrokh, and M. Ranjbar, "A modified PSO algorithm for minimizing the total costs of resources in MRCPSP," *Mathematical Problems in Engineering*, vol. 2012, Article ID 365697, 18 pages, 2012.
- [30] Y.-P. Du and Y.-X. Yang, "The research into the problems of assembly line balance," *Machinery Design and Manufacture*, vol. 2, pp. 104–106, 2003.
- [31] P.-L. Qiao, L.-L. Ma, and L. Zheng, "Research on job-shop scheduling problem based on improved particle swarm optimization," *Journal of Harbin University of Science and Technology*, vol. 16, no. 2, pp. 35–39, 2011.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

