

A Novel Assembly Line Balancing Optimization Algorithm based on PSOSA

Xiaomei Hu^{1,a}, Yubin Wang^{1,b}, Yangyang Zhang^{1,c}, Xinran Shi^{2,d}

¹The Key Laboratory of Intelligent Manufacturing and Robotics, School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, 200072, China

²Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI, 48105, U.S.

^asufeimasohxm@163.com, ^bwangjiayiyubin@qq.com, ^c904542902@qq.com, ^dxinrshi@umich.edu

Keywords: assembly line balancing; optimization; PSOSA; hybrid optimization algorithm

Abstract. Assembly line is widely used in manufacturing system and assembly line balancing problem directly affects the productivity of the whole manufacturing system. In the actual situation of assembly process, optimizing a certain goal in the process of actual assembly individually may cause the neglect of relevance of balance goals, so a dual-objective optimization model is proposed. Although there are many ways to solve the assembly line balancing problem, most of them have a slow convergence speed and poor optimize result. A PSOSA hybrid optimization algorithm is put forward to improve the solution quality. The Metropolis criterion of SA algorithm is integrated into PSO algorithm to overcome the precocious phenomenon of PSO and long optimization time of SA. So the PSOSA hybrid optimization algorithm has the advantages of PSO and SA algorithm. The simulation results show the feasibility and effectiveness of the assembly line balancing optimization algorithm based on PSOSA algorithm.

Introduction

With the progress and development of science and technology, the Manufacturing enterprise competitive advantage embodies in quality, cost, delivery, service and safety etc. How to enhance the overall efficiency of the assembly line, reduce lost work hour and logistics jam, balance the work load are paid more and more attention to in the assembly line production mode[1].

The extent of assembly line balancing determines the equipment utilization in manufacturing system. 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 to enhance the work efficiency of operator and equipment, reduce work hour consumed on a single piece of product. Therefore, the study of the assembly line balancing problem has important theoretical value and practical significance.

At present, some studies aim to solve the production line balancing problem and most of the studies concentrate in using a single method or algorithm to solve it. However, there have few researches on multi-objective optimization of assembly line balancing problem based on hybrid algorithm [2]. Takt time and smoothness index are two important indicators to measure the assembly line balancing. In this paper, assembly line balancing dual-objective optimization model about takt time and smoothness index is studies and a novel hybrid algorithm based on PSO algorithm and SA algorithm is proposed to solve the assembly line balancing dual-objective optimization problem [3-4].

Assembly line balancing optimization bases on PSOSA

A Dual-objective optimization model.

According to the classification description of the assembly line balancing problem in the paper of Li and Xin [5], three kinds of assembly line balancing problems belong to optimization problem in fact. 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 [6]. 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:

$$f_1 = b_1 * TT + b_2 * SI \quad (1)$$

Constraint conditions are as follows:

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

$$S_i \cap S_j = \emptyset, (i \neq j, i, j = 1, 2, \dots, m) \quad (3)$$

$$E = \bigcup_{k=1}^m S_k (k = 1, 2, \dots, m) \quad (4)$$

$$T(S_k) \leq TT; (k = 1, 2, \dots, m) \quad (5)$$

$$\text{if } M_{ij} = 1, i \in A_x, j \in A_y; \text{ then } x \leq y, M = (M_{ij})_{n \times n} \quad (6)$$

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 (3), there is no overlap between the different locations, the same job elements could only distribute on one location. In (4), it is ensuring that all the job elements are assigned [7]. The meaning of the formula (5) is the maximum operation time of location is the takt time. In (6), it is used to establish the distribution relations of job elements on the basis of precedence relation matrix.

Parameter Setting and coding.

The parameters of PSOSA algorithm include population quantity N , the number of iterations G , inertia weight w , accelerated constant c , weight coefficient b_1 and b_2 , temperature function $T_{X+1} = 0.8 * T_X$.

In order to use PSOSA algorithm to solve the assembly line balancing problem, the key job is the coding design. 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.

The basic steps of AOV-net 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 take the selected node as starting point from the map, the rest of the vertices are still constitute the AOV-net.
- ③ Repeat ①, ② until all the nodes are outputted, the vertex sequencing is a topological sort.

Since the precedence relation diagram of job element contains two parts, namely, job element and standard work time, a two-dimensional particle expressive method is proposed. Assume the total number of job elements is 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 relation diagram of an assembly line with twelve job elements is shown in Figure 1:

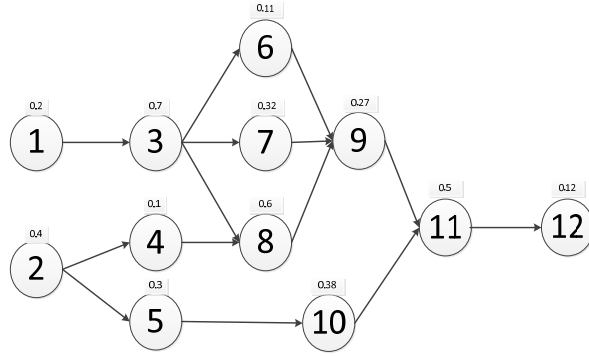


Figure 1. Precedence relation diagram with ten job elements

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

Table 1. The code of particle one

Job element	1	3	2	4	5	6	7	8	9	10	11	12
Standard work time	0.2	0.7	0.4	0.1	0.3	0.11	0.32	0.6	0.27	0.38	0.5	0.12

Table 2. The code of particle two

Job element	2	4	5	1	3	6	7	8	10	9	11	12
Standard work time	0.4	0.1	0.3	0.2	0.7	0.11	0.32	0.6	0.38	0.27	0.5	0.12

The particle updating and evolving formula of PSOSA algorithm.

1) The particle updating formula of PSOSA algorithm

The particle updating formula is defined as:

$$X_i(t+1) = c_2 \otimes g\{c_1 \otimes g[w \otimes h_{a,b}(X_i(t)), pB_i(t)], gB_i(t)\} \quad (7)$$

In (1), $X_i(t)$ expresses the positions of particle i in the t iterations, $X_i(t+1)$ expresses the positions of particle i in the $t+1$ iteration. c_1 and c_2 are the acceleration constant, and w is the inertia coefficient. Moreover, their values are between 0 and 1. $h_{a,b}(X_i(t))$ expresses the exchange of the a component and the b component in particle i . a and b 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 (1) includes three parts:

The first part is the updating of particle i position:

$$V_i(t) = w \otimes h_{a,b}(X_i(t)) = \begin{cases} h_{a,b}(X_i(t)) & rand() < w \\ X_i(t) & rand() \geq w \end{cases} \quad (8)$$

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

$$F_i(t) = c_1 \otimes g(V_i(t), pB_i(t)) = \begin{cases} g(V_i(t), pB_i(t)) & rand() < c_1 \\ V_i(t) & rand() \geq c_1 \end{cases} \quad (9)$$

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

$$X_i(t+1) = c_2 \otimes g(F_i(t), gB_i(t)) = \begin{cases} g(F_i(t), gB_i(t)) & rand() < c_2 \\ F_i(t) & rand() \geq c_2 \end{cases} \quad (10)$$

In (3) and (4), $rand()$ is a random uniform distribution number. The meaning of function $g(V_i(t), pB_i(t))$ is the adjustment of particle i according to its best position $pB_i(t)$ in the t iteration. The specific implementation processes are as follows:

(1) When $rand() \geq c_1$, $F_i(t) = V_i(t)$;

(2) When $rand() < c_1$, execute $g(V_i(t), pB_i(t))$. Take $pB_i(t)$ as the best position of particle i in the t iteration, n is the numbers of job element. $\{z(t)_{1,n}\}$ expresses the arrangement of job element

in $F_i(t)$, $\{y(t)_{1,n}\}$ expresses the arrangement of job element in $V_i(t)$, and $\{x(t)_{1,n}\}$ expresses the arrangement of job element in $pB_i(t)$;

(3) Because the numbers of job element are n , the length of code is n and there have $n-1$ intersection locations in the n codes. An integer c from 1 to $n-1$ is selected randomly to divide $\{x(t)_{1,n}\}$ into $\{x(t)_{1,c}\}$ and $\{x(t)_{c+1,n}\}$, and to divide $\{y(t)_{1,n}\}$ into $\{y(t)_{1,c}\}$ and $\{y(t)_{t+1,n}\}$;

(4) Delete the job elements in $\{y(t)_{1,c}\}$ if they are also in $\{x(t)_{1,c}\}$. And delete the job element $x(t)_{c+1}$ in $\{x(t)_{c+1,n}\}$. Then arrange the remaining elements in $\{x(t)_{1,c}\}$ and $\{x(t)_{c+1,n}\}$ in the original order to generate the new set of job elements $\{x'(t)_{c+1,n}\}$. Take $\{z(t)_{1,c}\} = \{y(t)_{1,c}\}$, $\{z(t)_{c+1,n}\} = \{x'(t)_{c+1,n}\}$. Merge $\{z(t)_{1,c}\}$ and $\{z(t)_{c+1,n}\}$ into $\{z(t)_{1,n}\}$;

(5) Set job element and its standard work time in $z(t)_{1,n}$ to generate new arrangement $F_i(t)$;

(6) The function $g(F_i(t), gB_i(t))$ 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 M , this iteration is successful.

2) The local and global optimal particle evolution formulas of PSOSA algorithm

After population particles are updated, the local optimal particle of various groups is updated according to formula (25), and the global optimal particle is updated according to formula (12).

The evolution formula of local optimal particle $pB_i(t+1)$ is shown as:

$$pB_i(t+1) = \begin{cases} i_{t+1} & \Delta f_1 > 0 \text{ } p_r > rand(0,1) \\ pB_i(t) & p_r < rand(0,1) \end{cases} \quad (11)$$

In (11), $pB_i(t+1)$ expresses the local optimal particle of particle i after $t+1$ iterations, i_{t+1} expresses the new particle in the $t+1$ iterations. Take boltzman constant $k=1$, $p_r = \exp^{-(\Delta f_1)/T}$, $\Delta f_1 = f_1(pB_i(t)) - f_1(i_{t+1})$. $rand(0,1)$ expresses a random number between 0 and 1.

The evolution formula of global optimal particle $gB_i(t+1)$ is shown as:

$$gB_i(t+1) = \begin{cases} i_{best} & \Delta f_2 > 0 \cup p_r > rand(0,1) \\ gB_i(t) & p_r < rand(0,1) \end{cases} \quad (12)$$

In (6), $gB_i(t+1)$ expresses the global optimal particle of particle i after $t+1$ iterations. i_{best} expresses the global optimal particle of particle i after $t+1$ iterations. $gB_i(t)$ expresses the global optimal particle of particle i after t iterations, and $\Delta f_2 = f_1(gB_i(t)) - f_1(i_{best})$.

Experiment and analysis

Take muffler assembly line with 23 job elements and 5 stations in commercial vehicles H1 as example. The job element of muffler assembly and standard work time is shown in Table 3. The precedence relations diagram is shown in Figure 2. The parameters used in PSO algorithm are as follows: population $N=20$; iteration number $G=800$; $w=0.5$; $c_1=0.8$; $c_2=0.8$; $b_1=0.5$; $b_2=0.5$. The parameters used in PSOSA algorithm are as follows: population $N=20$; iteration number $G=800$; $w=0.5$; $c_1=0.8$; $c_2=0.8$; $b_1=0.5$; $b_2=0.5$. $T_0 = -f_1(gB_i(1)) / \ln(0.2)$; $T_{x+1} = 0.8 * T_x$.

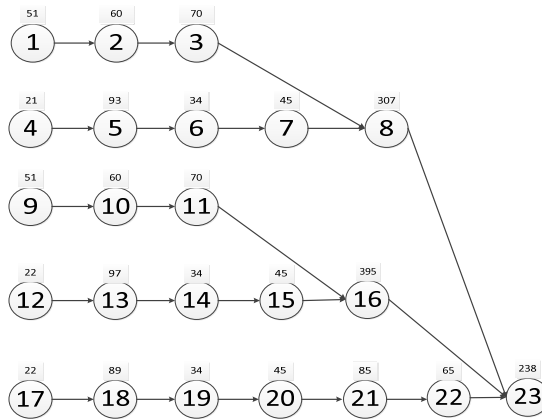


Figure 2. Precedence relations diagram with 23 job elements

By the optimization of PSOSA algorithm, the Gantt chart with time and job elements distribution is shown in Figure 3 .Their assembly line balancing comparison result is shown in Table 4.

Table 3. The Job Element of Antenna Assembly

operation sequence number	job element	standard work time (s)
1	The secondary assembly welding clapboard	51
2	The secondary assembly welding plate	60
3	The secondary assembly with glass fiber	70
4	The secondary assembly blanking	21
5	The secondary assembly pipe bending	93
6	The secondary assembly scrap material	34
7	The secondary assembly welding flange	45
8	The secondary assembly welding	307
9	The first assembly welding clapboard	51
10	The first assembly welding plate	60
11	The first assembly with glass fiber	70
12	The first assembly blanking	22
13	The first assembly pipe bending	97
14	The first assembly scrap material	34
15	The first assembly welding flange	45
16	The first assembly welding	395
17	Blanking	22
18	Elbowing	89
19	Removing the excess stock	34
20	The front flange electric welding	45
21	The front flange welding	85
22	The front heat shield welding	65
23	Assembly welding	238

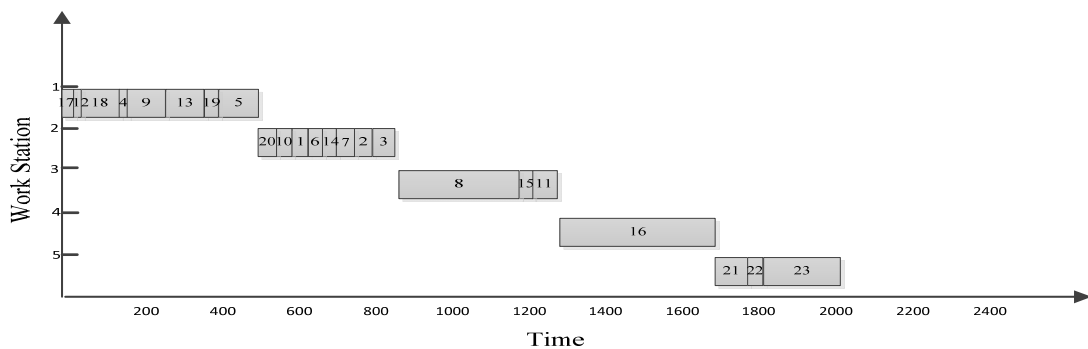


Figure 3. The Gantt chart of PSOSA algorithm

Table 4. The 23 job elements assembly balancing comparison result

Station	Base on PSO algorithm		Base on PSOSA algorithm	
	Job Element i	Time T_i	Job Element i	Time T_i
1	17,12,18,4,9,13,19,5	429	17,1,4,5,2,18,19,20	415
2	20,10,1,6,14,7,2,3	399	12,13,14,9,10,15,11,6	413
3	8,15,11	422	3,7,8	422
4	16	395	16	395
5	21,22,23	388	21,22,23	388
Takt Time TT	429		422	
Balance Rate P	94.77%		96.35%	
Smoothness Index SI	27.52		20.07	
Fitness f_i	228.26		221.04	

From Table 4, PSOSA algorithm has a faster convergence speed, smaller fitness f_i , Takt Time TT and Smoothness Index SI , and larger Balance Rate p , which shows that PSOSA algorithm has better performance than PSO algorithm in assembly line balancing dual-objective optimization.

Conclusion

In this paper, according to the number of a certain location line balancing problem, the PSOSA algorithm is proposed, which has the advantages of the PSO algorithm and SA algorithm, and use the algorithm to produce the beat TT and smoothness index SI as the optimization goal line balancing problem. The simulation experimental results show that the PSOSA hybrid algorithm to solve the optimization results compared to the PSO algorithm has a higher rate of balance, the balance index of smaller SI and production rhythm TT . Therefore PSOSA hybrid algorithm can effectively solve the problem of the balance of the production line, is a kind of feasible solutions.

References

- [1] Wu, X.Y. 2007. "Research on the assembly production line balance." Shanghai jiaotong university.
- [2] Das, G., P. K. Pattnaik, and S. K. Padhy. 2014. "Artificial Neural Network trained by Particle Swarm Optimization for non-linear channel equalization." *Expert Systems with Applications* 41(7): 3491-3496.
- [3] van den Bergh, F., and A. P. Engelbrecht. 2006. "A study of particle swarm optimization particle trajectories." *Information sciences* 176(8):937-971.
- [4] Pang, F. 2006. "The principle of SA algorithm and algorithm's application on optimization problem." Jilin university.
- [5] Li. J., X. W.Y. 2007. "The Job-Shop Scheduling Based on a kind of new GA in ERP." *Manufacture Information Engineering of China* 11:004.
- [6] Ghanbari, A., E. Hadavandi, and S. Abbasian-Naghneh. 2010. "An intelligent ACO-SA approach for short term electricity load prediction." *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence Springer Berlin Heidelberg*: 623-633.
- [7] Du. Y.P., and Y. X. Yang. 2003. "The research into the problems of assembly line balance." *Machinery Design and Manufacture* 2:104-106.