COMPLEX SYSTEM MODELING AND SIMULATION
ISSN 2096-9929 01/06 pp 1−11
Volume 3, Number 1, March 2023
DOI: 10.23919/CSMS.2022.0026

Load Optimization Scheduling of Chip Mounter Based on Hybrid Adaptive Optimization Algorithm

Xuesong Yan, Hao Zuo, Chengyu Hu*, Wenyin Gong, and Victor S. Sheng

Abstract: A chip mounter is the core equipment in the production line of the surface-mount technology, which is responsible for finishing the mount operation. It is the most complex and time-consuming stage in the production process. Therefore, it is of great significance to optimize the load balance and mounting efficiency of the chip mounter and improve the mounting efficiency of the production line. In this study, according to the specific type of chip mounter in the actual production line of a company, a maximum and minimum model is established to minimize the maximum cycle time of the chip mounter in the production line. The production efficiency of the production line can be improved by optimizing the workload scheduling of each chip mounter. On this basis, a hybrid adaptive optimization algorithm is proposed to solve the load scheduling problem of the mounter. The hybrid algorithm is a hybrid of an adaptive genetic algorithm and the improved ant colony algorithm. It combines the advantages of the two algorithms and improves their global search ability and convergence speed. The experimental results show that the proposed hybrid optimization algorithm has a good optimization effect and convergence in the load scheduling problem of chip mounters.

Key words: Surface Mount Technology (SMT); chip mounter; load optimization scheduling; adaptive genetic algorithm; ant colony algorithm

1 Introduction

With the increasing demand for electronic products, their design is becoming increasingly miniaturized and refined, and their functions are continuously improved. This development makes the once-common perforated plug-in components no longer suitable for large-scale, highly integrated circuits.

Manufacturers aim to reduce costs, expand the profit space, and enhance the competitiveness of their products by developing highly integrated electronic products, which promote Surface Mount Technology (SMT) and its process of rapid development. An SMT production line mainly includes a plate feeding machine, dispensing machine, chip mounter, reflow welding furnace, plate machine, and other equipment. A chip mounter is the core equipment in the SMT production line, responsible for finishing the mount operation, which is the most complex and time-consuming stage in the production process. Therefore, optimizing the load scheduling of the mounter to improve the mounting efficiency is of great significance for improving the production line efficiency[1–3].

The SMT line is used to mount a single Print Circuit Board (PCB) using multiple chip mounters. Each chip mounter is responsible for the mounting task of a part of the components. Hence, how to reasonably allocate components for each chip mounter is the first stage of

• Xuesong Yan, Hao Zuo, Chengyu Hu, and Wenyin Gong are with the School of Computer Science, China University of Geosciences, Wuhan 430074, China. E-mail: yanxs@cug.edu.cn; zuohao@cug.edu.cn; huchengyu@cug.edu.cn; wygong@cug.edu.cn.
• Victor S. Sheng is with the Department of Computer Science, Texas Tech University, Lubbock, TX 79409−3104, USA. E-mail: victor.sheng@ttu.edu.
* To whom correspondence should be addressed.
※ This article was recommended by Associate Editor Lining Xing.
Manuscript received: 2022-11-12; revised: 2022-11-29; accepted: 2022-12-01

the SMT production line optimization and is also the basis of the optimization of feeding through discharge and mount sequences. Moreover, it is very important to rationally allocate the type and quantity of mounted components for the chip mounter on the production line to improve the actual production capacity of the production line[4, 5]. In the study on the load optimization scheduling of mounters, the heuristic optimization algorithm is common[6]. Heuristic algorithms, such as genetic algorithms[7−11], swarm intelligence algorithms[12−15], and differential evolution algorithm[16, 17], are common optimization algorithms. Besides the above algorithms, other intelligent optimization algorithms have also been applied to the optimization problems of SMT production lines, such as multi-modal and multi-objective optmization algorithms[18−21], bio-inspired optimization algorithms[22−26], dymanic multi-objective optimization algorithms[27], and data and knowledge-driven memetic algorithms[28−32]. In the early stage of the development of chip mounters, the load-balancing optimization problem of components is solved by a linear programming model. However, with the continuous expansion of the scale in the actual production process, such as the increasing types and quantities of components and numerous factors limiting the component mount, the linear programming model is no longer suitable for the current actual production optimization. In addition, to achieve the global optimum, load balancing optimization and mount path optimization must be considered at the same time[33]. A material allocation method will correspond to an optimal sorting method, and an optimal solution will be obtained. It may be a local optimal solution because the corresponding optimal solution may be superior under another allocation method. Based on the above reasons, intelligent optimization algorithms have been applied to the optimization research of the SMT production line. Hsu[34] designed a particle swarm optimization algorithm to optimize the feeder position and mount sequence. Guo et al.[35] designed a hybrid genetic algorithm to optimize feeder allocation. Castellani et al.[36] designed an improved bee colony algorithm to optimize the combination of the component placement sequence and feeder allocation. In reference, Gao et al.[37] proposed a two-stage method for the combinatorial optimization of the component placement sequence and feeder allocation. With many types of mounting components, the frequent replacement of suction nozzles will also affect the mounting efficiency[38]. Luo et al.[39] constructed a two-stage mixed-integer linear programming model by combining nozzle replacement, feeder allocation, and component placement sequence for optimization. Li and Yoon[40] designed a tabu search algorithm based on the nearest-neighbor domain, which combines the optimization of the picking sequence, nozzle replacement, and feeder position.

2 Problem Description and Model

2.1 Problem description

In the SMT production line, after a chip mounter completes the mounting task of a PCB, it sends the PCB to the next chip mounter for the next part of the mount and receives the PCB mounted by the previous chip mounter for the next stage of the mounting task. When this mounter finishes its task, the next mounter does not finish the mount work mounter. Hence, this mounter must wait for the next mounter to finish the mount work before sending the PCB to the next mounter and receiving the PCB transmitted by the previous mounter. That is, the time required by an SMT production line composed of multiple chip mounters to complete a stage of the mounting task is the maximum time for these chip mounters to complete a stage of the mount task time. Theoretically, the SMT production line efficiency can be maximized when chip mounters no longer wait for one another, that is, when all chip mounter stages work simultaneously. However, in the actual production process, it is not realistic to make the working time of each stage of the patch machine exactly the same, so we can only continuously reduce this time. The phase working time of the production line is determined by the chip mounter with the longest working time. Thus, in the load optimization scheduling process, the material distribution of the chip mounter is optimized with the goal of minimizing the maximum phase working time.

2.2 Problem model

The main structure of the SMT production line of a company is shown in Fig. 1. The core equipment of the SMT production line is composed of a high-speed chip mounter (high-speed machine) and a multifunctional chip mounter (medium-speed machine). The high-speed machine has two independent working tables with high-speed and high precision, and its main task is to mount small components. The medium-speed machine’s mount speed is slow, but the machine can

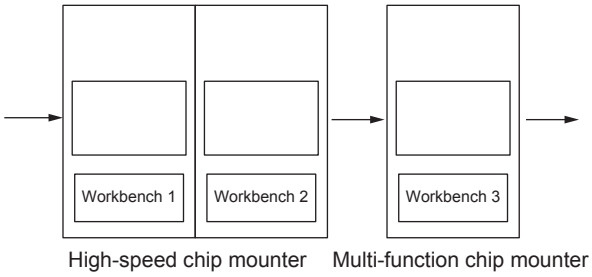


Fig. 1 Structure of the SMT production line.

mount all types of components, including some large irregularly shaped components. In general, components that cannot be mounted on high-speed machines are placed on medium-speed machines for mounting tasks.

The PCB is sent to the high-speed mounter by the working equipment, which is assembled by the No. 1 and No. 2 tables of the high-speed mounter in turn, and then sent out by the high-speed mounter. The multifunctional mounter receives the PCB to complete the remaining component mount task. Finally, it is sent to the reflow welding and curing furnace. The mounter's mount task is completed. Most of the components on the PCB are fed by the belt feeder, while some components need the disc feeder.

The disc feeder needs to be installed on the multifunctional mounter. The multifunctional mounter with the disc feeder can no longer install the belt feeder. Therefore, a single multifunctional mounter is usually used to mount the components that need the disc feeder for the feeding task. Only the optimal load scheduling problem of the belt feeder in each patch machine is considered. The high-speed mounter has two workbenches, which are independent of each other for the mounting task. Thus, the load optimization scheduling task of the mounter is not based on the mounter as the unit but on the independent workbenches as the unit for the optimization task. In the series structure of the chip mounter shown in Fig. 1, the stage working time of the production line is determined by the maximum working time of No. 1, No. 2, and No. 3 workbenches. The optimization goal is to minimize the maximum working time of the stage specific to the workbench to achieve the maximum working efficiency of the SMT production line.

During component allocation, the components with special requirements are assigned first. For instance, some large irregularly shaped components can only be mounted on the multifunctional chip mounter as high-speed chip mounters cannot mount such components. In addition, the speed mode of the chip mounter has certain requirements. Components under MODEL 1608, that is, components smaller than 1.6 mm in length and 0.8 mm in width, can only be mounted on high-speed chip mounters. After completing the special component allocation task, the remaining component types of both mounters can mount the task, and these components can be reasonably allocated to complete the load optimization scheduling. In the mounting process of components, the nozzle types for picking up different components are also different. Each arm of the chip mounter involved in this study has eight mounting heads to assemble the suction nozzle, and up to eight different types of suction nozzles can be assembled. When there are too many types of components on the PCB, only eight types of suction nozzles are not enough to absorb all components. Then, the boom should be transferred to the suction nozzle exchange station to replace the suction nozzle. A multifunctional chip mounter is used to mount large irregularly shaped components. These components often have many kinds, but the required number is very small. Many such components only need one, so in the multifunctional chip mounter mounting process, it is easy to replace the suction nozzle operation. In the load optimization scheduling process, the time consumed by replacing the suction nozzle should be considered. For medium and large components, due to their large size, the suction nozzle will lead to the phenomenon that the adjacent suction nozzle cannot absorb the component, and the absorption of the component mount cycle cannot be fully loaded. This phenomenon should also be considered in the load optimization scheduling process, so that the material separation results are highly scientific and reasonable. Through the above analysis, the problem model established is shown in the following:

The objective functions are as follows:

$$\min f = \min(\max_{i \in I} X_i) \tag{1}$$

$$X_i = w_i + N_i \tag{2}$$

$$w_i = \sum_{j \in J, f \in F} t_{ifj} p_j z_{ifj} \tag{3}$$

$$p_j = \sum_{k \in K} a_{kj},\ j \in J \tag{4}$$

The constraints are as follows:

$$\sum_{j \in J_i} \sum_{f \in F} s_j z_{ifj} \leqslant m_i,\ i \in I \tag{5}$$

$$\sum_{i\in I}\sum_{f\in F} z_{ifj} = 1,\ j \in J \tag{6}$$

$$w_i > 0 \tag{7}$$

$$\sum_{l\in L} b_{l1j} = 1,\ j \in J \tag{8}$$

$$\sum_{l\in L} b_{l2j} = 1,\ j \in J \tag{9}$$

$$\sum_{l\in L} b_{l3j} = 1,\ j \in J \tag{10}$$

In the above equations, Eq. (1) represents the objective function, which means minimizing the maximum working time of the three workbenches. Equation (2) represents that the working time of the workbench is divided into two parts, which are composed of the time for the workbench to be attached and the time for the mounting head to change the nozzle at the nozzle exchange station. Equation (3) represents the calculation method of the workbench patch time. Equation (4) represents how many points component j has on the PCB. Formula (5) represents the total number of station locations required to place components on workbench stage, i, which cannot exceed the limit of the total number of workbench station locations. Equation (6) represents that the components of the same type can only be placed on one workbench, and multiple workbenches cannot mount one component at the same time. Formula (7) represents that each workbench should be assigned a mounting task. Equations (8)−(10) represent that when the same component is mounted on the same workbench, only the same nozzle can be used for the material retrieval operation.

The definitions of each variable in the equations are as follows:

• i represents the workbench number. i is an integer, and its values are 1, 2, and 3. The workbench number is shown in Fig. 1. I represents the set of all workbenches i.

• j represents the component type number, and J is the set of all components j.

• k represents the number of each point component on the PCB and K is the set of all components k.

• l represents the type of suction nozzle, and L is the set of all types of suction nozzles.

• X_i represents the total working time of the workbench i, including the time to replace the nozzle from the nozzle exchange station.

• w_i represents the mount component time of the workbench i.

• N_i represents the time for the workbench i to switch the suction nozzle.

• m_i represents the number of feeding slots of the workbench i. The sum of component stations of the feeding results cannot exceed this maximum limit.

• f represents the speed mode of the workbench. The value can be 1, 2, and 3, representing one speed mode. The speed ranges from fast to slow, F is the set of speed modes.

• t_{ifj} represents the expected time of attaching the j-th component to the i-th table in the speed mode f. For example, the time of attaching a small material to the high-speed chip mounter in speed mode 1 is 0.05 s/piece.

• p_j represents how many points the j-th component has on the PCB.

• a_{kj} represents that the value of component k is 1 if it is a component of type j, and 0 otherwise.

• s_j represents how many feeding slots are occupied by j-type components. When the value is greater than 1, this component is a medium or large component, and the suction nozzle will lead to the phenomenon that the adjacent suction nozzle cannot absorb components.

• z_{ifj} represents that a component of type j is set to mount on the workbench i in the speed mode f if the value is 1; otherwise, it is 0.

3 Algorithm Description

3.1 Hybrid adaptive optimization algorithm

The crossover probability and mutation probability of the standard genetic algorithm are fixed. This condition has a great impact on the convergence of the algorithm and cannot meet the needs. In addition, to search the solution space at the beginning of the genetic algorithm, the algorithm needs a large crossover and mutation probability. In the later stage of the genetic algorithm, the population constantly approaches the optimal solution, and a large crossover and mutation probability are not conducive to the final fast convergence. Moreover, the fixed probability will affect the efficiency of the algorithm. To solve the above problems, this paper proposes an adaptive genetic algorithm to dynamically adjust the crossover and mutation probability. The algorithm increases the crossover and mutation probability when the population diversity is small and reduces the crossover and mutation probability when the population diversity

is large.

The ant colony algorithm has the characteristics of fast convergence and strong stability, but it also has the defects of long search time and easy premature phenomenon. To improve the performance of the ant colony algorithm and the search efficiency, the pheromone calculation and update methods are improved in this study. The way that ants search for the next city according to the pheromone concentration is represented by Eqs. (11) and (12) in the following:

As one time, when $q < q_0$,

$$s = \arg\max [\tau(i,j)]^{\alpha}[\eta(i,j)]^{\beta},\ j \in J_k(i) \tag{11}$$

Otherwise,

$$p_k(i,j) = \begin{cases} \dfrac{[\tau(i,j)]^{\alpha}[\eta(i,j)]^{\beta}}{\sum [\tau(i,j)]^{\alpha}[\eta(i,j)]^{\beta}},\ j \in J_k(i); \\ 0,\ \text{otherwise} \end{cases} \tag{12}$$

where α and β represent the importance degree of the pheromone and heuristic factor, respectively. $\tau(i,j)$ represents the pheromone concentration on the edge (i,j). $\eta(i,j) = 1/d(i,j)$ is the heuristic factor, and $d(i,j)$ is the distance between cities i and j. $J_k(i)$ represents the next city that can be searched at city i. q is a random number with a uniform distribution of 0 and 1, and q_0 is the threshold set to determine the probability of the probabilistic search and path selection with prior knowledge. When $q < q_0$, according to the pheromone left by the last iteration, the ant chooses the path with the largest pheromone concentration as the next search path. Otherwise, it will conduct a probabilistic search according to the pheromone. The pheromone is updated by Eqs. (13) and (14) in the following:

$$\tau(i,j) = (1-\rho_2)\cdot\tau(i,j) + \rho_2 \cdot \Delta\tau(i,j) \tag{13}$$

$$\tau(i,j) = \begin{cases} (L_{best})^{-1},\ (i,j) \in best; \\ 0,\ \text{otherwise} \end{cases} \tag{14}$$

where ρ_2 is the global pheromone volatilization factor, L_{best} is the fitness value of the optimal path, and *best* represents the optimal path.

In this study, the adaptive genetic algorithm and improved ant colony algorithm are mixed, and the advantages of the two algorithms are combined to improve the global search ability and convergence speed of the algorithm. Then, the hybrid algorithm is applied to the load optimization scheduling of the chip mounter.

3.2 Problem coding

The type of components and the number of workbenches are represented by numbers, and the components correspond to the workbenches. The workbenches of the mounter in Fig. 1 are coded as 1, 2, and 3 from left to right, in which No. 1 and No. 2 are high-speed chip mounters, and No. 3 is a multifunctional chip mounter. Due to the different types of components, the types of chip mounters that can mount different types of components are also different, so component coding can be divided into three categories. The first category consists of components that can only be mounted on the high-speed chip mounter. This kind of component corresponds to workbenches 1 and 2. The second category consists of components that can only be mounted on a multifunctional chip mounter, corresponding to workbench 3. The third category refers to components that can be mounted on both types of chip mounters, corresponding to all workbenches. Assuming that there are eight kinds of components to be mounted, coded from 1 to 8, No. 4 component is the component that can only be mounted on the high-speed chip mounter, and No. 3 and No. 5 components are the components that can only be mounted on the multifunctional chip mounter. Their coding methods are shown in Fig. 2. No. 4 component can only be mounted on the high-speed chip mounter, so the value of the workbench code is 1 or 2. No. 3 and No. 5 components can only be mounted on the multifunctional chip mounter, so the value can only be 3. The other components can be mounted on the two chip mounters without restriction, so the value can be any integer from 1 to 3.

3.3 Genetic operation

The selection operator selects the roulette selection operator, calculates the sum of fitness values, and takes the ratio of fitness to the sum as the selection probability. The crossover operator selects two points to cross, randomly sets two points in the parent

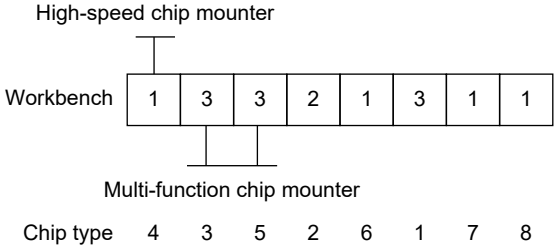


Fig. 2 Problem coding method.

individual, and exchanges the part between the two set points in the parent individual to form two offspring individuals. The relative position of the exchanged chromosome parts in the parent individual is the same, ensuring that the type of element can be mounted on the corresponding workbench.

The mutation operator selects a single-point mutation operator, randomly selects a component, and randomly mutates to another workbench number. If the component can only be mounted on the high-speed chip mounter, it will be changed from 1 to 2 or 2 to 1. If it can only be mounted on the multifunctional chip mounter, it will not be changed. If there is no limit, a positive number will be randomly selected in the interval [1, 3]. If the number of chip mounters is more, only the variation interval can be expanded.

The crossover and mutation probabilities are calculated,

$$P_{\mathrm{c}}=\begin{cases} k_1\times\dfrac{F_{\max}\times f'}{F_{\max}\times F_{\mathrm{avg}}}, & f'\geqslant F_{\mathrm{avg}};\\ k_2, & f'<F_{\mathrm{avg}}\end{cases} \tag{15}$$

$$P_{\mathrm{m}}=\begin{cases} k_3\times\dfrac{F_{\max}\times f}{F_{\max}\times F_{\mathrm{avg}}}, & f\geqslant F_{\mathrm{avg}};\\ k_4, & f<F_{\mathrm{avg}}\end{cases} \tag{16}$$

where f is the fitness of the current mutant individual, f' is the greater fitness of two crossed individuals; $F_{\max}$ and F_{avg} are the maximum and average fitness values, respectively; the values of k_1 and k_3 are the real number [0, 1], and the value of k_2 and k_4 is set to 0.8 to ensure that the inferior individuals, whose fitness value is lower than the average fitness value of the population, will carry out the cross-mutation operation with a great probability.

3.4 Pheromone calculation and update

Different from the TSP problem, the SMT production line load optimization scheduling does not have a conventional path, so there are some special aspects in the pheromone calculation and update. To minimize the working time difference of each workbench, the pheromone calculation is set as follows: The pheromone is updated with the difference between the maximum and minimum working time of the three chip mounter workbenches of the optimal individual. Q is the pheromone intensity left by the ant after the path and is constant, and the working time X_i is the working time of each workbench of the optimal individual. ρ is the pheromone evaporation rate, and n represents the component type. The calculation process is shown in the follwing:

$$\begin{aligned}&\tau(n,n+1)=\\&(1-\rho)\cdot\tau(n,n+1)+\rho\cdot\Delta\tau(n,n+1)\end{aligned} \tag{17}$$

$$\tau(n,n+1)=\frac{Q}{\max(X_i)-\min(X_i)},\ \ i\in I \tag{18}$$

3.5 Repair operation

For the randomly generated chromosomes, the inevitably generated components occupy more stations than the limit of the number of feeding slots, resulting in infeasible solutions. Such infeasible solutions should be repaired during the operation of the algorithm. In all the components that exceed the limit number of the feeding slots on the workbench, select one at random and assign the component to a workbench that will not exceed the limit number of the feeding slots. Then, continue to cycle this operation until the phenomenon that the number of component stations is equal to the limit number of the feeding slots.

For randomly generated chromosomes, it is possible that $w_i=0$. That is, there are no components to be assigned. For this infeasible solution, randomly select a workbench whose w_i is not 0, randomly select a component that meets the mounting conditions on this workbench, and assign the component to the undivided workbench.

3.6 Algorithm process

The process of the hybrid adaptive optimization algorithm proposed in this paper is as follows:

Step 1: Parameter and pheromone initialization.

Step 2: Determine the population size N of the adaptive genetic algorithm, initialize the population randomly, and maximize the number of iterations T.

Step 3: Calculate the individual fitness of the population, determine the crossover and mutation probability, and reserve the individuals with the mutation and crossover probability of 0. The probabilities of crossover and mutation are determined by Eqs. (15) and (16).

Step 4: Check whether the termination condition is met. If yes, output the result and end the algorithm; otherwise, go to Step 5.

Step 5: Select the population by the roulette, sequence crossover, and mutation operations to obtain the offspring population of size N, and calculate the fitness value. The fitness value of the optimal fitness individual in the offspring is compared with *best*, and *best* is updated. Then, use *best* to update the global

pheromone with Eqs. (17) and (18).

Step 6: Use pheromones to guide ants in their search. Set the threshold q_0, take random number q, and determine the search mode of the ant search according to the size of q and q_0. When $q < q_0$, the path with the maximum pheromone concentration is selected using the prior knowledge; otherwise, a probabilistic search is performed.

Step 7: Each ant performs global pheromone updates during the search process.

Step 8: Among the m paths searched by ants, select some optimal paths to replace the worst individuals in the population, keep the population number N unchanged, update the population, and return to Step 4.

4 Results and Discussion

4.1 Experimental data and parameters

In this study, three kinds of PCBs of different sizes in actual production are used for the simulation experiment. The chip mounters are of two types: one is CPM-2 (multifunctional chip mounter), and the other is CPM-3 (high-speed chip mounter). CPM-3 is equipped with two workbenches, and the parameters of the three PCBs are shown in Table 1. The data source is from an intelligent equipment company.

Table 1 shows the basic parameters of the three PCBs. Board 351-1405 is single-sided, so only one side needs to be mounted. Boards 351-1540 and 5 352 360 are double-sided, and there are component points on both sides to be mounted. 351-1540-TOP and 5 352 360-TOP are the front component data of the PCB. 351-1540-BOT and 5 352 360-BOT are the back component data of the PCB. The experiment does not involve the turning-over operation, and the two sides of the same PCB are optimized independently.

In the hybrid algorithm, the population size is 50, the maximum iteration is 300 generations, and the crossover probability and mutation probability change dynamically with the algorithm. The number of ants is 30, the pheromone increase coefficient is 100, and the pheromone evaporation coefficient is 0.2. The hybrid algorithms are used to optimize five load-balancing tasks of the three PCBs.

Table 1 Related data of the PCB.

PCB board	Number of component points	Number of component types
351-1405	74	38
351-1540-BOT	38	29
351-1540-TOP	221	60
5 352 360-BOT	254	104
5 352 360-TOP	113	46

The hardware running environment of the experiment is as follows: 64-bit OS, Windows 8, 8 GB memory, and Intel® Core(TM) I7-3537U@CPU 2.00 GHz processor.

4.2 Experimental results and discussion

The load optimization scheduling of the chip mounter is carried out on five surfaces of the three kinds of PCB. It is assumed that the working time of the multifunctional mounter (medium-speed machine) workbench is T_1, and the working time of the two high-speed mounters are T_2 and T_3, respectively. The working time is expressed in cycles. The experimental results are compared with the load scheduling mode of the chip mounter: connecting software. Then, the result of the load optimization scheduling is evaluated by the deviation rate (dev). The calculation method of the deviation rate is shown in the following:

$$dev = \frac{\max(T_i) - \min(T_i)}{\text{avg}(T_i)} \tag{19}$$

where T_i represents the working time of the workbench, and the deviation rate is the ratio between the difference of the maximum and minimum working time of the three workbenches to the average working time ave (T_i). The smaller the deviation rate, the closer the working time of each workbench, and the better the load optimization scheduling effect. The experimental results are shown in Table 2.

The data of PCB 351-1540-BOT have 29 kinds of components and 38 component points, so it is the smallest in scale and the lowest in complexity among the five groups of test data. The load optimization scheduling effect of the two methods is the same. The data of PCB 5 352 360-BOT have 104 kinds of components and 254 component points, so it is the most complex and largest set of experimental data. The connecting software has the worst effect, whereas the hybrid optimization algorithm obtains good results. When the numbers of component types and component points on the PCB are larger, the scale of the optimization problem is larger, the effect of the connecting software method is worse, and the hybrid algorithm can obtain better results. When the sizes of the component points and component types on the PCB are small, the two methods obtain the same results. The

Table 2 Comparison of experimental results.

PCB board	Number of component points	Number of component types	Algorithm	T_1 (s)	T_2 (s)	T_3 (s)	*dev* (s)
351-1405	74	38	Connecting	21.03	11.97	7.38	1.01
			Hybrid	19.31	13.01	8.86	0.70
351-1540-BOT	38	29	Connecting	7.90	3.28	6.47	0.78
			Hybrid	7.90	3.28	6.47	0.78
351-1540-TOP	221	60	Connecting	42.09	39.22	16.31	0.79
			Hybrid	37.11	29.23	29.37	0.25
5 352 360-BOT	254	104	Connecting	49.31	42.89	21.61	0.73
			Hybrid	34.26	33.23	34.01	0.03
5 352 360-TOP	113	46	Connecting	22.02	16.16	14.03	0.54
			Hybrid	20.55	15.77	15.01	0.32

larger the size of the PCB is, the better the hybrid algorithm is compared with the connecting software.

Table 3 shows the information on the components that can only be mounted on the multifunctional mounter and high-speed mounter, and the information on the component points that can be mounted on both kinds of chip mounters. Based on the data in Table 3, the number of limited components will affect the effect of the load optimization scheduling algorithm. There are many limited components in the data of PCB 351-1540-BOT, and the types and number of components that can be freely distributed in the two chip mounters are small. Thus, the optimization space is small, so the results obtained by the two methods are the same. When there are many freely allocated components in the data, such as PCB 5 352 360-BOT, the hybrid algorithm has a large optimization space, and the working time of the three workbenches will be similar. Based on the result of the experiment data, the multifunctional chip mounter tends to need more work time than the high-speed chip mounter. The occurrence of this kind of phenomenon is attributed to the generally large irregularly shaped components fixed on the multifunctional mounter. Its large size leads to a small number of components for each mount cycle, so more cycles are required. In addition, due to the large volume and mass of the large irregularly shaped components, to avoid the components from falling, the low-speed mode is often used for mounting work. Therefore, the multifunctional chip mounter often needs more time for the mounting task. When the number of large irregularly shaped components is large, the working time of the multifunctional chip mounter workbench and high-speed mounter workbench are often large.

Figures 3 and 4 show the average fitness value of the hybrid algorithm in each generation and the best fitness value of the hybrid algorithm in each generation of PCB 5 352 360-BOT. The results show that the convergence speed of the hybrid algorithm is relatively fast and has a good convergence effect. When the hybrid algorithm performs load optimization scheduling for the PCB data of different sizes, the efficiency of the algorithm will significantly decrease with the expansion of the scale, but the effect of obtaining load optimization scheduling will be significantly improved.

5 Conclusion

The load optimization scheduling of chip mounters is studied for the SMT production line. The mounting task of a PCB will be carried out with the cooperation

Table 3 Component mount limit information.

PCB board	Multi-function mounter		High-speed mounter		Free point	
	Number of types	Number of component points	Number of types	Number of component points	Number of types	Number of component points
351-1405	5	11	2	6	31	57
351-1540-BOT	2	4	23	27	4	7
351-1540-TOP	16	22	2	18	42	181
5 352 360-BOT	25	35	31	73	38	146
5 352 360-TOP	10	34	18	28	18	51

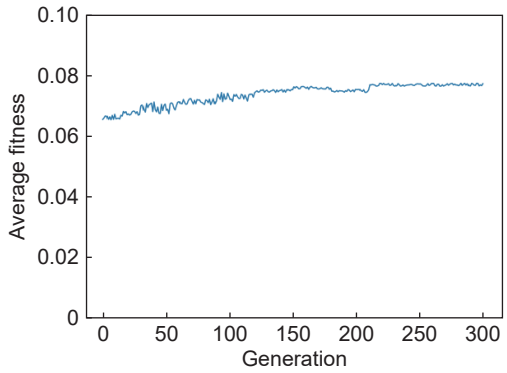


Fig. 3 Average fitness value of the hybrid algorithm in each generation.

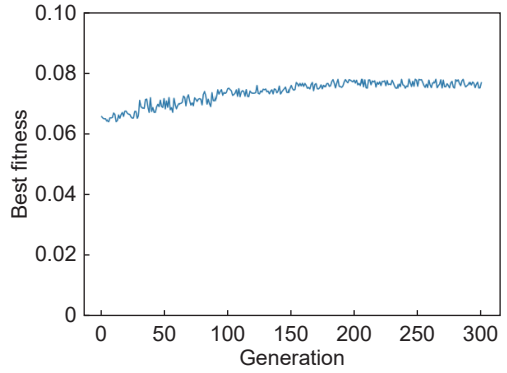


Fig. 4 Best fitness value of the hybrid algorithm in each generation.

of multiple mounters. The components required by a specific printed circuit board will be allocated to the tasks of each mounter rationally through optimization, so the time required by the mounter to complete each part of the task is as close as possible. Moreover, the idle waiting time of the mounter can be shortened. In this study, combined with the working characteristics of the chip mounter in the production line, the common constraints in production, such as the size of components, speed mode, and operation of changing the nozzle, are introduced to make the model highly suitable for the actual production. In the process of the experiment, three kinds of PCBs in the actual production are selected, and the data of five aspects are tested. The complexity and scale of the five groups of data are also different. The proposed hybrid adaptive optimization algorithm is used to optimize the load scheduling of the chip mounter in the production line, and the results are compared with the optimization method of the machine. The results show that the proposed hybrid optimization algorithm has a good optimization effect and convergence in the load scheduling problem. With the increase of the problem size, the optimization effect of the proposed hybrid algorithm becomes more obvious, which proves the effectiveness of the proposed hybrid algorithm on the load optimization scheduling problem of chip mounters.

Acknowledgment

This paper was supported by the National Natural Science Foundation of China (Nos. U1911205, 62073300, and 62076225) and the National Key Research and Development Program of China (No. 2021YFB3301602).

References

[1] F. J. Cazorla, P. M. W. Knijnenburg, R. Sakellariou, E. Fernandez, A. Ramirez, and M. Valero, Predictable performance in SMT processors: Synergy between the OS and SMTs, *IEEE Trans. Comput.*, vol. 55, no. 7, pp. 785–799, 2006.

[2] X. J. Lan, Y. Chen, and H. T. Tang, Balancing and continuous improvement of SMT production line, (in Chinese), *Industrial Engineering and Management*, vol. 11, no. 2, pp. 109–111, 2006.

[3] M. Ayob and G. Kendall, A survey of surface mount device placement machine optimisation: Machine classification, *Eur. J. Oper. Res.*, vol. 186, no. 3, pp. 893–914, 2008.

[4] H. Y. Lin, C. J. Lin, and M. L. Huang, Optimization of printed circuit board component placement using an efficient hybrid genetic algorithm, *Appl. Intell.*, vol. 45, no. 3, pp. 622–637, 2016.

[5] W. Wang, P. C. Nelson, and T. M Tirpak, Optimization of high-speed multistation SMT placement machines using evolutionary algorithms, *IEEE Trans. Electron. Packag. Manufact.*, vol. 22, no. 2, pp. 137–146, 1999.

[6] W. Ho and P. Ji, An integrated scheduling problem of PCB components on sequential pick-and-place machines: Mathematical models and heuristic solutions, *Expert Syst. Appl.*, vol. 36, no. 3, pp. 7002–7010, 2009.

[7] O. Kulak, I. O. Yilmaz, and H. O. Günther, PCB assembly scheduling for collect-and-place machines using genetic algorithms, *Int. J. Prod. Res.*, vol. 45, no. 17, pp. 3949–3969, 2007.

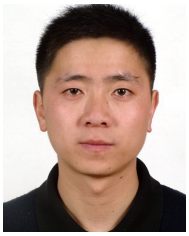
[8] S. Y. Li, C. F. Hu, and F. H. Tian, Enhancing optimal feeder assignment of the multi-head surface mounting machine using genetic algorithms, *Appl. Soft Comput.*, vol. 8, no. 1, pp. 522–529, 2008.

[9] A. García-Nájera, C. A. Brizuela, and I. M. Martínez-Pérez, An efficient genetic algorithm for setup time minimization in PCB assembly, *Int. J. Adv. Manuf. Technol.*, vol. 77, no. 5, pp. 973–989, 2015.

[10] X. S. Yan, J. Sun, and C. Y. Hu, Research on contaminant sources identification of uncertainty water demand using genetic algorithm, *Cluster Comput.*, vol. 20, no. 2, pp. 1007–1016, 2017.

[11] X. S. Yan, H. M. Liu, Z. X. Zhu, and Q. H. Wu, Hybrid genetic algorithm for engineering design problems,

Cluster Comput., vol. 20, no. 1, pp. 263–275, 2017.
[12] C. H. Wu, D. Z. Wang, A. Ip, D. W. Wang, C. Y. Chan, and H. F. Wang, A particle swarm optimization approach for components placement inspection on printed circuit boards, *J. Intell. Manuf.*, vol. 20, no. 5, pp. 535–549, 2009.
[13] B. Wu, C. H. Qian, W. H. Ni, and S. H. Fan, The improvement of glowworm swarm optimization for continuous optimization problems, *Expert Syst. Appl.*, vol. 39, no. 7, pp. 6335–6342, 2012.
[14] Q. H. Wu, Z. X. Zhu, X. S. Yan, and W. Y. Gong, An improved particle swarm optimization algorithm for AVO elastic parameter inversion problem, *Concurr. Comput.*: *Pract. Exper.*, vol. 31, no. 9, p. e4987, 2019.
[15] W. Q. Zhang, W. L. Hou, C. Li, W. D. Yang, and M. Gen, Multidirection update-based multiobjective particle swarm optimization for mixed no-idle flow-shop scheduling problem, *Complex System Modeling and Simulation* , vol. 1, no. 3, pp. 176–197, 2021.
[16] W. Y. Gong and Z. H. Cai, Parameter extraction of solar cell models using repaired adaptive differential evolution, *Solar Energy*, vol. 94, pp. 209–220, 2013.
[17] S. J. Li, W. Y. Gong, X. S. Yan, C. Y. Hu, D. Y. Bai, and L. Wang. Parameter estimation of photovoltaic models with memetic adaptive differential evolution, *Solar Energy*, vol. 190, pp. 465–474, 2019.
[18] X. S. Yan, J. Zhao, C. Y. Hu, and D. Z. Zeng, Multimodal optimization problem in contamination source determination of water supply networks, *Swarm Evol. Comput.*, vol. 47, pp. 66–71, 2019.
[19] Y. J. Song, L. N. Xing, M. Y. Wang, Y. J. Yi, W. Xiang, and Z. S. Zhang, A knowledge-based evolutionary algorithm for relay satellite system mission scheduling problem, *Comput. Ind. Eng.*, vol. 150, p. 106830, 2020.
[20] J. W. Ou, J. H. Zheng, G. Ruan, Y. R. Hu, J. Zou, M. Q. Li, S. X. Yang, and X. Tan, A pareto-based evolutionary algorithm using decomposition and truncation for dynamic multi-objective optimization, *Appl. Soft Comput.*, vol. 85, p. 105673, 2019.
[21] S. J. Li, W. Y. Gong, X. S. Yan, C. Y. Hu, D. Y. Bai, L. Wang, and L. Gao, Parameter extraction of photovoltaic models using an improved teaching-learning-based optimization, *Energy Convers. Manage.*, vol. 186, pp. 293–305, 2019.
[22] X. S. Yan, P. P. Li, K. Tang, L. Gao, and L. Wang, Clonal selection based intelligent parameter inversion algorithm for prestack seismic data, *Inf. Sci.*, vol. 517, pp. 86–99, 2020.
[23] Y. J. Song, X. Ma, X. J. Li, L. N. Xing, and P. Wang, Learning-guided nondominated sorting genetic algorithm II for multi-objective satellite range scheduling problem, *Swarm Evol. Comput.*, vol. 49, pp. 194–205, 2019.
[24] P. F. Yu and X. S. Yan, Stock price prediction based on deep neural networks, *Neural Comput. Appl.*, vol. 32, no. 6, pp. 1609–1628, 2020.
[25] X. S. Yan, M. Z. Zhang, and Q. H. Wu, Big-data-driven pre-stack seismic intelligent inversion, *Inf. Sci.*, vol. 549, pp. 34–52, 2021.
[26] W. Y. Gong, Z. W. Liao, X. Y. Mi, L. Wang, and Y. Y. Guo, Nonlinear equations solving with intelligent optimization algorithms: A survey, *Complex System Modeling and Simulation*, vol. 1, no. 1, pp. 15–32, 2021.
[27] Y. R. Hu, J. H. Zheng, S. Y. Jiang, S. X. Yang, and J. Zou, Handling dynamic multiobjective optimization environments via layered prediction and subspace-based diversity maintenance, *IEEE Trans*. *Cybern*., doi: 10.1109/TCYB.2021.3128584.
[28] Y. H. Du, L. Wang, L. N. Xing, J. G. Yan, and M. S. Cai, Data-driven heuristic assisted memetic algorithm for efficient inter-satellite link scheduling in the BeiDou navigation satellite system, *IEEE/CAA J. Autom. Sin.*, vol. 8, no. 11, pp. 1800–1816, 2021.
[29] S. Xiang, L. Wang, L. N. Xing, Y. H. Du, and Z. Q. Y. Zhang, Knowledge-based memetic algorithm for joint task planning of multi-platform earth observation system, *Comput. Ind. Eng.*, vol. 160, p. 107559, 2021.
[30] J. Y. Gong, X. S. Yan, and C. Y. Hu, An ensemble-surrogate assisted cooperative particle swarm optimisation algorithm for water contamination source identification, *Int. J. Bio-Inspired Comput.*, vol. 19, no. 3, pp. 169–177, 2022.
[31] Q. H. Wu, B. Wu, and X. S. Yan, An intelligent traceability method of water pollution based on dynamic multi-mode optimization, *Neural Comput*. *Appl*., doi: 10.1007/S00521-022-07002-0.
[32] C. Y. Hu, Q. M. Wang, W. Y. Gong, and X. S. Yan, Multi-objective deep reinforcement learning for emergency scheduling in a water distribution network, *Memetic Comput.*, vol. 14, no. 2, pp. 211–223, 2022.
[33] J. Han and Y. Seo, Mechanism to minimise the assembly time with feeder assignment for a multi-headed gantry and high-speed SMT machine, *Int. J. Prod. Res.*, vol. 55, no. 10, pp. 2930–2949, 2017.
[34] H. P. Hsu, Solving feeder assignment and component sequencing problems for printed circuit board assembly using particle swarm optimization, *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 2, pp. 881–893, 2017.
[35] S. J. Guo, F. Geng, K. Takahashi, X. H. Wang, and Z. H. Jin, A MCVRP-based model for PCB assembly optimisation on the beam-type placement machine, *Int. J. Prod. Res.*, vol. 57, no. 18, pp. 5874–5891, 2019.
[36] M. Castellani, S. Otri, and D. T. Pham, Printed circuit board assembly time minimisation using a novel Bees algorithm, *Comput. Ind. Eng.*, vol. 133, pp. 186–194, 2019.
[37] J. S. Gao, X. M. Zhu, A. B. Liu, Q. Y. Meng, and R. T. Zhang, An iterated hybrid local search algorithm for pick-and-place sequence optimization, *Symmetry*, vol. 10, no. 11, p. 633, 2018.
[38] D. B. Li, T. He, and S. W. Yoon, Clustering-based heuristic to optimize nozzle and feeder assignments for collect-and-place assembly, *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 2, pp. 755–766, 2019.
[39] J. X. Luo, J. Y. Liu, and Y. M. Hu, An MILP model and a hybrid evolutionary algorithm for integrated operation optimisation of multi-head surface mounting machines in PCB assembly, *Int. J. Prod. Res.*, vol. 55, no. 1, pp. 145–160, 2017.
[40] D. B. Li and S. W. Yoon, PCB assembly optimization in a single gantry high-speed rotary-head collect-and-place machine, *Int. J. Adv. Manuf. Technol.*, vol. 88, no. 9, pp. 2819–2834, 2017.



Xuesong Yan received the BEng degree in computer science and technology and the MEng degree in computer application from China University of Geosciences, China in 2000 and 2003, respectively, and the PhD degree in computer software and theory from Wuhan University, China in 2006. He is currently an associate professor at the School of Computer Science, China University of Geosciences, Wuhan, China, and was as a visiting scholar at the Department of Computer Science, University of Central Arkansas, Conway, AR, USA. His research interests include evolutionary computation, data mining, and computer application.

Hao Zuo received the BEng degree from Nanchang Hangkong University, China in 2022. He is currently a master student at the School of Computer Science, China University of Geosciences, Wuhan, China. His research interests include intelligent optimization and scheduling optimization.

Chengyu Hu received the BEng and MEng degrees in automation and control from Wuhan University of Technology, China in 2000 and 2003, respectively, and the PhD degree in automation control from Huazhong University of Science and Technology, China in 2010. He is currently a professor and vice dean at the School of Computer Science, China University of Geosciences, Wuhan, China. His current research interests include evolutionary algorithm, swarm intelligence, and cloud computing.

Wenyin Gong received the BEng, MEng, and PhD degrees in computer science from China University of Geosciences, Wuhan, China in 2004, 2007, and 2010, respectively. He is currently a professor at School of Computer Science, China University of Geosciences, Wuhan, China. His research interests include evolutionary algorithms, evolutionary optimization, and their applications. He has published over 70 research papers in journals and international conferences.

Victor S. Sheng received the PhD degree in computer science from the University of Western Ontario, Canada in 2007. He is currently an assistant professor at the Department of Computer Science, Texas Tech University, USA. His current research focuses on data mining, machine learning, and artificial intelligence.