S & M 2707

# Component Placement Process Optimization for Multi-head Surface Mounting Machine Using a Hybrid Algorithm

Cheng-Jian Lin[1,2*] and Chun-Hui Lin[3]

[1]Department of Computer Science and Information Engineering, National Chin-Yi University of Technology,
No. 57, Sec. 2, Zhongshan Rd., Taiping Dist., Taichung 41170, Taiwan
[2]College of Intelligence, National Taichung University of Science and Technology,
No. 129, Sec. 3, Sanmin Rd., North Dist., Taichung City 404, Taiwan
[3]Department of Computer Science and Information Engineering, National Cheng Kung University,
No. 1, University Road, Tainan City 701, Taiwan

(Received April 25, 2021; accepted August 24, 2021)

***Keywords:*** multi-head placement machine, nearest neighbor search, scheduling optimization, printed circuit board, pick-and-place sequence problems

This paper addresses the optimization problem of printed circuit board (PCB) assembly scheduling for surface mount placement (SMP) machines with multiple heads. Most of the related studies have oversimplified the practical problems and limited their focus to three sub-problems: the feeder assignment problem (FAP), the nozzle setup problem, and the component pick-and-place sequence problem (PAPSP). To extend the range of problems that can be addressed, in this study, we propose a hybrid algorithm to consider the component height, the pick-and-place restrictions, and simultaneous pickup restrictions. A roulette wheel (RW) is designed to generate an automatic nozzle changer (ANC) sequence, with the nearest neighbor search (NNS) used to find an initial placement path. Furthermore, we apply the 2-optimization (2-opt) method to improve the placing sequence. Practical PCB datasets provided by an EVEST EM-780 multi-head SMP machine are adopted as experimental examples. Experimental results show that the proposed algorithm effectively enhances the productivity of the multi-head SMP machine and reduces the cycle time in PCB assembly.

## 1. Introduction

Owing to technology advancement in recent years, the information technology, communication, and consumer electronics industries have flourished. The electronics industry is well developed and has become the most important industry around the world. The printed circuit board (PCB) is one of the essential and indispensable components in the manufacturing process of electronic products. The increasing demand for PCBs has led to the prosperity of the PCB industry.

As the complexity of electronic devices increases, circuits and components are placed increasingly close to each other on PCBs. PCB production companies are seeking to reduce

[*]Corresponding author: e-mail: cjlin@ncut.edu.tw

https://doi.org/10.18494/SAM.2021.3579

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

production costs and assemble more components on PCBs of the same area. Followed by the surface mount technology (SMT) developed for assembling components on PCB, the scheduling of the assembly is obviously the key constraint of the whole system.

Many scholars have studied the assembly scheduling problem. Each proposed method has considered the differences existing in mechanical structures. The assembly scheduling problem is normally divided into a few sub-problems including (1) the feeder assignment problem (FAP), the nozzle setup problem, and (3) the component pick-and-place sequence problem (PAPSP). The assembly scheduling for a placement machine is optimized to maximize the throughput of a PCB, minimize the production cycle time, and minimize the total travel distance of the head.

Numerous approaches have been proposed to optimize multi-head placement machines based on the sub-problems mentioned above. Lee *et al.*(1) conducted an in-depth study of the multi-head placement machine optimization problem by considering the installation of multiple nozzles. The scheduling problem includes the line balancing problem, FAP, and PAPSP. FAP is classified as a quadratic assignment problem. As a traveling salesman problem (TSP), PAPSP is to minimize the travel distance of the head, simultaneously pick up the largest number of components, and reduce the number of nozzles exchanged to minimize the production cycle time of the assembly. Huang *et al.*(2) developed a hierarchical multi-objective optimization model in which the first master hierarchy was for nozzle optimization. The component mounting sequence, feeder optimization, and picking order were optimized in parallel in the second hierarchy based on the results of the first hierarchy optimization. The experimental results showed that the proposed method can significantly improve the overall performance. Ayob and Kendall(3) used a greedy algorithm to optimize the feeder slot assignment by considering the feeder type and the number of feeders. Oh and Park(4) discussed a heuristic path-planning algorithm for multi-head dispensers to increase productivity in PCB assembly lines. The component placement path was treated as a TSP, and they used a nearest neighbor search (NNS) to find an initial path and applied the 2-optimization (2-opt) method to improve it. Chen and Shen(5) proposed an integrated mathematical model to compare the total assembly time of the component placement process between single-nozzle and multiple-nozzle placement machines. The results showed that the latter is more efficient. Gao *et al.*(6) combined a local search with integer programming in an iterated hybrid local search (IHLS) algorithm to solve the pick-and-place optimization problem. A greedy algorithm with a distance weight strategy and the convex-hull strategy was used in the local search section, and the integer programming model was built to solve FAP. The experimental results showed that IHLS requires less time than the genetic algorithm (GA) and a memetic algorithm (MA) and is suitable for solving large-scale problems. Chen and coworkers(7,8) used the Tabu search to assign the feeder and used an improved shuffled frog-leaping algorithm to improve the pick-and-place sequence to simultaneously solve FAP and PAPSP. They combined the diversification disturbance and variation factor with the Tabu search to greatly improve the production efficiency. Liu *et al.*(9) built an optimization algorithm based on a heuristic strategy and a scatter search method to minimize the PCB assembly time and pointed out strong constraints in existing optimization algorithms that lead to unsatisfactory performance in practice. Jeevan *et al.*(10) solved PAPSP by the use of GA. They used the travel distance of the head as a fitness value and suggested using GA to replace a variable neighborhood search because GA can provide an approximately optimal solution with better quality. Garcia-Naijera and Brizuela(11) proposed an efficient GA to solve PAPSP and FAP by setting the fitness function as the assembly time of the PCB. Neammanee *et al.*(12) developed an MA to solve the placement sequence problem and FAP. MA gives a better solution; however, it requires a huge amount of calculation. Zhang *et al.*(13) attempted to optimize the PCB assembly process by a hybrid GA. Lim *et al.*(14) applied a hybridized cuckoo search–GA to path optimization for the PCB hole drilling process. Jiang and coworkers(15–17) focused on placement sequence optimization and reduction of the nozzle changing frequency for a placement machine with four nozzles. The topic was regarded as a TSP, and an improved ant colony algorithm was applied to optimize the placement sequence. He *et al.*(18) proposed an adaptive clustering-based GA to optimize the pick-and-place operation of a dual-gantry component placement machine. The nozzle and component allocation decisions were made before the evolutionary search of GA to improve the efficiency of the algorithm. Experimental results showed that the proposed algorithm reduced the total gantry moving distance by 5.71% and 4.07% on average compared with the large clusters of operations algorithm and multi-phase planning heuristic algorithm, respectively. Mumtaz *et al.*(19) focused on optimizing the multi-level planning and scheduling problem by minimizing the cycle time of PCB assembly lines. They also used a mixed integer linear programming model to solve the component allocation problem and component placement sequence problem simultaneously. A novel hybrid spider monkey optimization (HSMO) algorithm was also proposed in their study, and results indicated that the HSMO algorithm performed better than GA and particle swarm optimization in terms of efficiency and effectiveness. Li *et al.*(20) altered the original PCB assembly optimization problem to a newly defined component allocation problem, which is to decide the component type handled by each head in each PAP cycle. Cell division GA was proposed to solve the component allocation problem. The experimental results demonstrated that the proposed method dramatically reduces the PCB assembly time when applying industrial PCB datasets.

The above evolutionary computation optimization methods, although able to improve scheduling problems, usually require a long computing time to generate a feasible solution and are obviously not applicable in practical production lines due to research constraints. To reduce the complexity of the problem, many researchers ignore some of the restrictions on the placement machine, resulting in schedules that are not applicable in real situations. Most of the related studies oversimplified the practical problems and limited their focus to three sub-problems: (1) FAP, (2) the nozzle setup problem, and (3) component PAPSP. In this study, we propose a hybrid optimization method to optimize PCB assembly scheduling for a surface mount placement (SMP) machine with multiple heads within a short time to solve (1) PAPSP, (2) the nozzle assignment problem, and (3) the automatic nozzle changer (ANC) assignment problem under the restrictions of (1) component height, (2) pick, (3) place, and (4) simultaneous pickup.

This paper is organized as follows. Section 2 gives the problem statement including the assumptions and mathematical model. The solution methodology is described in Sect. 3. Computational experiments and results are given in Sect. 4, and conclusions and future work are discussed in Sect. 5.

## 2. Problem Statement

### 2.1 Assumptions

A surface mount device (SMD), also known as a placement machine, is an automated, high-speed, high-precision component placement device. The SMD is the most critical and complex equipment in the whole SMT production process. The main task of the placement machine is to pick and place the components in the correct positions to assemble all the components on the PCB. The placement machine (EM-780, EVEST Corporation, Taoyuan County, Taiwan) considered in this paper is shown in Fig. 1. It is equipped with many sensors such as a dual servo motor drive, a high-pixel camera, a motion simulation system, and a unique lighting design, and its architecture is shown in Fig. 2.

- Heads: Each head can have a nozzle installed to pick up a component; EM-780 has eight heads.
- Nozzles: A nozzle is installed in a head; usually there are different types of nozzles for different component shapes.
- ANC: This is an automatic nozzle-switching device that detaches or attaches as the nozzles are switched to accommodate different types of components during the process.
- Feeders: Feeders are used to contain and supply components for the placement machine. Each feeder only contains one type of component.
  (a) Tape feeders: A tape feeder is the most commonly used type of feeder. Typical tape widths are 8, 12, 16, and 24 mm.
  (b) Stick feeders: Stick feeders are designed for components packed in linear sticks (small ICs issued in low volumes). Components are moved to the pickup location by gravity or vibration.
  (c) Matrix tray feeders: Matrix tray feeders are used for large, delicate, or expensive components.

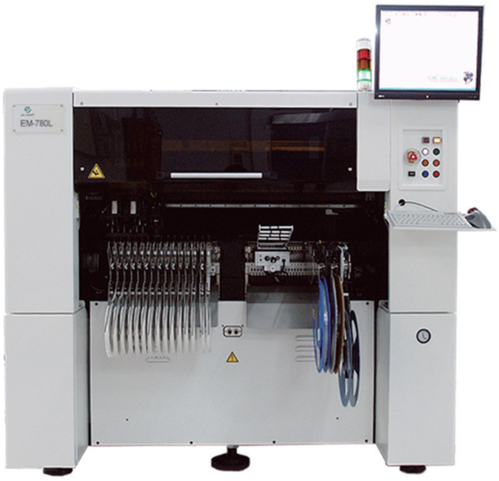


Fig. 1. (Color online) SMD used in this study.

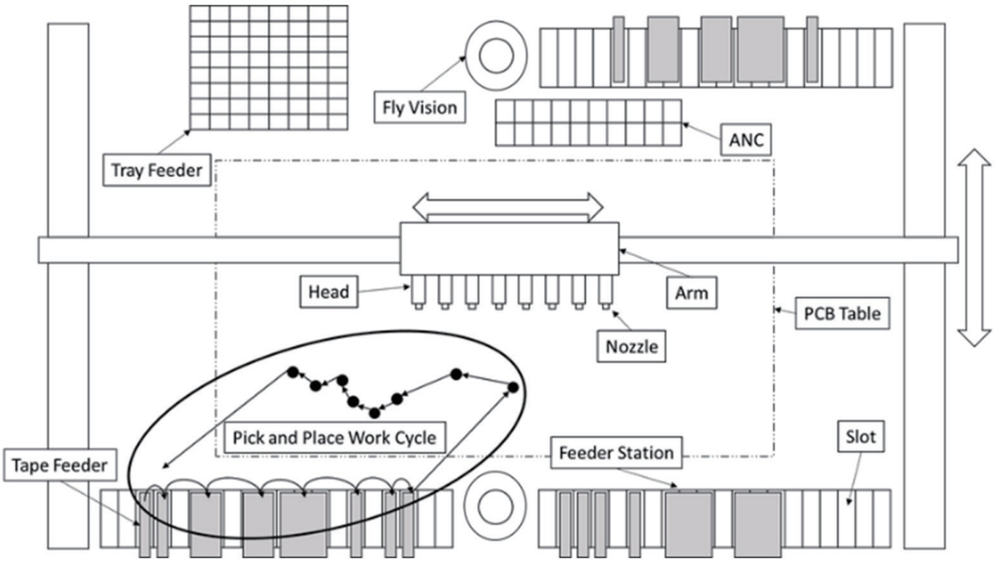


Fig. 2. Architecture of EM-780 machine.

- Feeder slot: A slot used to install a feeder.
- Feeder station: The area where the feeder slots are set.
- PCB table: A base that holds the PCB.
- Arm: Multiple heads are located on the arm and are controlled as a mobile *X*–*Y* coordinate system.

The arm with multiple heads moves to the feeder station and picks up as many components as possible until each nozzle is loaded with a component, then moves to the PCB and places the components one by one in the correct positions. This is called a pick-and-place work cycle (cycle) as shown in Fig. 2. The placement machine iterates this process until all the components are mounted on the PCB. A flowchart of the movement processes of the placement machine is shown in Fig. 3. However, the restrictions of the placement machine need to be considered to generate an applicable schedule as follows.

- Component height: The height of each component may be different, and the components are picked and placed in the ascending order of height to avoid collision.
- Time required for nozzle changes: The shapes of the components are different, and each component matches a unique corresponding nozzle. When the nozzle is not applicable, the head must exchange its nozzle, which takes time. The time required for nozzle changes has been ignored in some studies.
- Picking restrictions: In the placement machine, not every head can pick up components from all feeders because there are some positions that some heads cannot reach.
- Placing restrictions: As above, there are some destinations that some heads cannot reach.
- Simultaneous pickup restrictions: Simultaneous pickup is important in the placement machine process. Practically, we want to simultaneously pick up as many components as possible to reduce the time required for picking components.

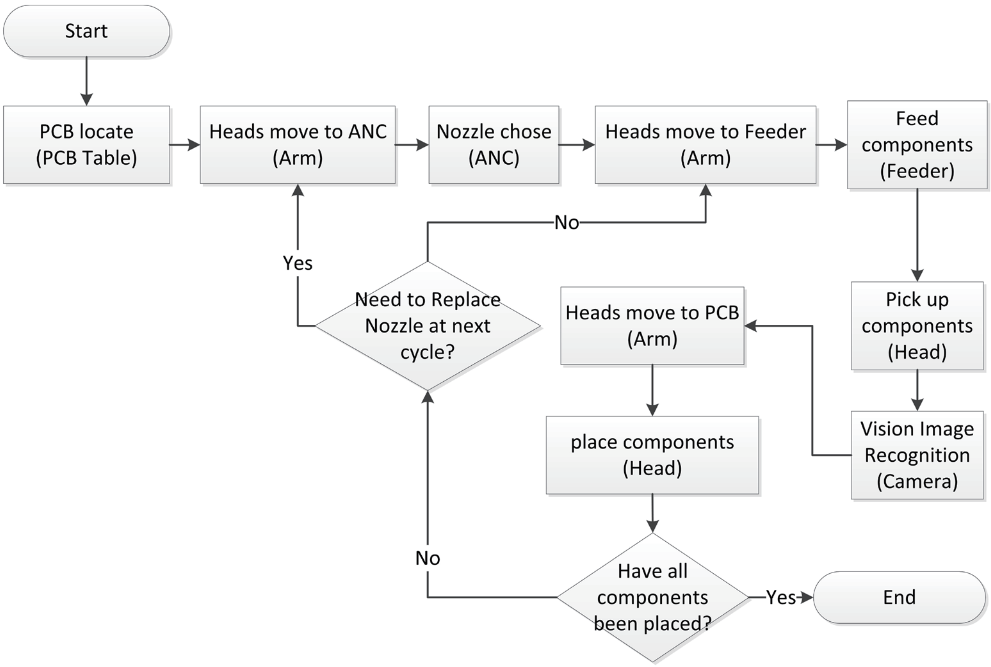

Start
PCB locate (PCB Table)
Heads move to ANC (Arm)
Nozzle chose (ANC)
Heads move to Feeder (Arm)
Feed components (Feeder)
No
Yes
Need to Replace Nozzle at next cycle?
Heads move to PCB (Arm)
Pick up components (Head)
Vision Image Recognition (Camera)
place components (Head)
No
Have all components been placed?
Yes
End

Fig. 3. Block diagram of movement processes of placement machine.

On the basis of the machine descriptions in Sect. 2.1, the aim of this study is to generate a schedule with the shortest assembly time. After visiting industry representatives and discussing the problems, we realized that engineers will normally assign the positions of feeders, but the feeder allocation is not changed often. It is common that the same components are placed on different PCBs. Sometimes, the feeder allocation remains unchanged even when different PCB products are produced. This is the reason that the feeder allocation is treated as a given condition in this study. Thus, three more issues concerning PCB assembly to be discussed are listed as follows.

- ANC assembly problem: First, the number of different nozzles should be determined for nozzle changes in ANC before the PCB assembly is scheduled. The ANC has 16 small cells and four large cells corresponding to different nozzle sizes. The number of nozzles is a factor in both the nozzle setup problem and the pick-and-place problem to generate configurations with the best performance.
- Nozzle setup problem: The shapes of assembly components differ, and each component has a fitting nozzle. Here, we consider (1) pick-and-place restrictions and (2) simultaneous pickup restrictions when assigning a suitable sequence of nozzles to heads. The purpose is to maximize the number of components for each picking and minimize the total number of pickings. Such a sequence is called an ANC sequence in this paper.
- Component PAPSP: The first priority is to generate the shortest head movement path when assigning the pick-and-place sequence of components. The component height and nozzle setup are considered in this issue to generate a feasible sequence.

### 2.2 Mathematical model

To assess the performance of our proposed method based on the actual operation of the placement machine, a mathematical model is developed to calculate the total cycle time of the assembly. The notations are defined in Table 1.

The process is divided into three parts: (1) pick time, (2) place time, and (3) ANC time. The mathematical formulation of the total cycle time of the assembly is

$$T_{total} = \sum_{c=1}^{NC} \left( T_{pick}(c) + T_{place}(c) + T_{change}(c) \right), \tag{1}$$

where $T_{pick}(c)$, $T_{place}(c)$, and $T_{change}(c)$ are the pick time, place time, and ANC time at the $c$th cycle of the PCB assembly, respectively. The mathematical formulation of the pick time is

$$T_{pick} = \begin{cases} \sum_{i=1}^{NP-1} \left( T(d(S_i, S_{i+1})) + T_z(i+1) + TD_{i+1} + TP_{i+1} \right), \; c = 1 \text{or last cycle have replaced nozzle,} \\ T(d(p_N, S_1)) + T_z(1) + TD_1 + TP_1 + \sum_{i=1}^{NP-1} \left( T(d(S_i, S_{i+1}) + T_z(i+1) + TD_{i+1} + TP_{i+1} \right), \text{ otherwise,} \end{cases} \tag{2}$$

where $T(d(p_N, S_1))$ is the travel time of the head moving from the $N$th component position to the first slot position, $T_z(i + 1)$ is the travel time of the $z$-axis at the $(i + 1)$th pickup of the cycle,

Table 1
Notations used in this study.

| Notation | Meaning |
|---|---|
| $Min(T_{total})$ | Objective function of the optimization problem |
| $T_{pick}$ | Pick time |
| $T_{place}$ | Place time |
| $T_{change}$ | ANC time |
| $NC$ | Total number of cycles |
| $N$ | Number of heads |
| $NP$ | Total number of pickups |
| $p$ | Pickup of the cycle |
| $TD$ | Vacuum-on delay time |
| $TP$ | Pick up wait time |
| $TB$ | Blow time |
| $TL$ | Load wait time |
| $TN$ | Time cost of replacing a nozzle |
| $CN$ | Total number of nozzles needing to be replaced |
| $PC$ | Position of the component |
| $S$ | Position of the slot |
| $A$ | Position of the ANC |
| $d()$ | Chebyshev distance between two positions |
| $T()$ | Travel time of head movement |
| $T_z(c)$ | Travel time of $z$-axis |

$T(d(S_i, S_{i+1}))$ is the travel time of the head moving from the $i$th slot position to the $(i + 1)$th slot position, and $TD_{i+1}$ and $TP_{i+1}$ are the vacuum-on delay time and the pickup wait time at the $(i + 1)$th pickup of the cycle, respectively. The mathematical formulation of the place time is

$$T_{place} = T(d(S_N, PC_1)) + T_z(1) + TB_1 + TL_1 + \sum_{j=2}^{N}\left(T(d(PC_{j-1}, PC_j)) + T_z(j) + TB_j + TL_j\right), \quad (3)$$

where $T(d(S_N, PC_1))$ is the travel time of the head moving from the $N$th slot position to the first component position, $T_z(j)$ is the travel time of the $z$-axis at the $j$th component of the cycle, $T(d(PC_{j-1}, PC_j))$ is the travel time of the head moving from the $(j + 1)$th component to the $j$th component of the cycle, and $TB_j$ and $TL_j$ are the blow time and the load wait time of the $j$th component of the cycle, respectively. The mathematical formulation of the ANC time is

$$T_{change} = T(d(PC_N, A_1)) + T(d(A_{CN}, S_1)) + \sum_{k=1}^{CN}\left[T(d(A_k, A_{k+1})) + TN_k\right], \quad (4)$$

where $T(d(A_k, A_{k+1}))$ is the travel time of the head moving from the $k$th nozzle to the $(k + 1)$th nozzle that needs to be replaced, $TN_k$ is the time cost of replacing the $k$th nozzle, $T(d(PC_N, A_1))$ is the travel time of the head moving from the $N$th component to the first position of the ANC, and $T(d(A_{CN}, S_1))$ is the travel time of the head moving from the $CN$th position of the ANC to the first component.

## 3. Solution Methodology

We propose an efficient hybrid algorithm to optimize the placement machine scheduling in a short time to meet on-site requirements. The flowchart of the efficient hybrid algorithm is shown in Fig. 4. First, the number of nozzles is assigned proportionally based on the pick loading of the nozzles (Sect. 3.1). Second, a roulette wheel (RW) is applied to generate an ANC sequence (Sect. 3.2). Third, an NNS is performed to generate an initial pick-and-place sequence (Sect. 3.3). Fourth, the sequence is optimized through the 2-opt method (Sect. 3.4). These procedures are repeated until the entire assembly is completed. The time models are compared, and the best assembly scheduling is recorded. Examples are used in the following subsections to easily explain the procedures.

### 3.1 Distribute the numbers of nozzles proportionally

The numbers of nozzles are distributed proportionally to maximize the benefits for each nozzle. As an example, the steps involved when using small nozzles are as follows:

Step 1. Generate the component dataset shown in Table 2 and assign one nozzle for each nozzle type to the ANC based on the same nozzle size.

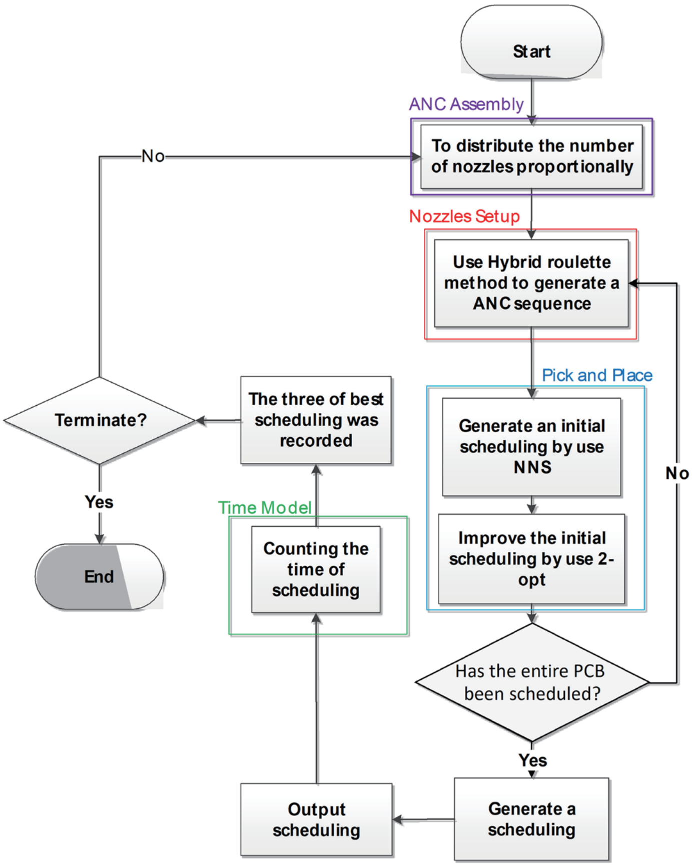

Start
ANC Assembly
To distribute the number of nozzles proportionally
No
Nozzles Setup
Use Hybrid roulette method to generate a ANC sequence
Pick and Place
Generate an initial scheduling by use NNS
No
Improve the initial scheduling by use 2-opt
Has the entire PCB been scheduled?
Yes
Generate a scheduling
Output scheduling
Time Model
Counting the time of scheduling
The three of best scheduling was recorded
Terminate?
Yes
End

Fig. 4. (Color online) Block diagram of the proposed efficient hybrid algorithm.

Table 2
Information of components and nozzles.

| Component type | Nozzle type | Number of nozzles | Total number of nozzles | Nozzle size | Proportion (%) |
|---|---|---|---|---|---|
| A | AN2 | 150 | 200 | Small | 56 |
| B | AN2 | 50 | | | |
| C | AN3 | 30 | 30 | | 9 |
| D | AN4 | 80 | 125 | | 35 |
| E | AN4 | 45 | | | |
| F | AN6 | 10 | 10 | Large | 66 |
| G | AN7 | 5 | 5 | | 34 |

Step 2. Split different nozzle sizes. Then assign the numbers of nozzles proportionally based on the pick loading of nozzles for each nozzle size. For example, the pick loadings of small nozzles AN2, AN3, and AN4 are 200, 30, and 125 in Table 2, respectively. The proportions are 56, 9, and 35%, respectively. Because the total number of nozzle cells of ANC is 16 (Fig. 5) and one of each type of nozzle is already assigned in step 1, the remaining spaces allocated to AN2, AN3, and AN4 are 7, 1, and 5 nozzle cells, respectively.

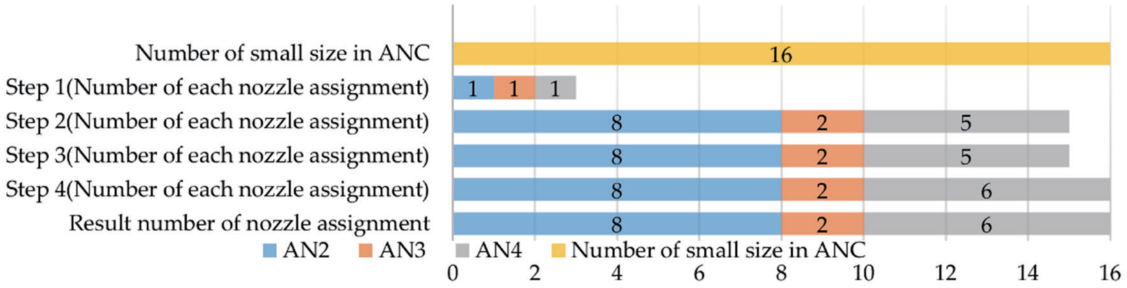


Fig. 5. (Color online) Assignment of the small nozzles to ANC.

Step 3. Check the number of nozzles for each nozzle size. An infeasible distribution is when the number of nozzles exceeds the number of heads detected by sensing. If this occurs, allocate the excessive number of nozzles to the nozzle type with the second highest proportion. Otherwise, go to step 4.

Step 4. Check whether there are empty nozzle cells left. If yes, then assign these nozzle cells to the nozzle type with the highest proportion. Similar to that in step 3, if the number of nozzles exceeds the number of heads detected by sensing, allocate the excessive cells to the nozzle types with the second highest proportion, the third highest proportion, and so on, until all the cells are allocated. Figure 5 shows the final solution for this example.

### 3.2 Use RW to generate the pickup sequence and ANC sequence

During the processing, different nozzles are utilized to pick up components, and the nozzles are assigned by an ANC sequence to the head. There are 10 kinds of nozzles, AN1–AN10, for EVEST EM-780. As shown in Fig. 6, AN4-AN2-AN2-AN2-AN3-AN2-AN5-AN8 is an ANC sequence, and such a sequence can be used in many pick-and-place work cycles.

An RW to generate an ANC sequence was designed in this study, and the RW considers the simultaneous pickup restrictions to generate an ANC sequence with the most simultaneous pickups and the fewest pickups. The simultaneous pickup restrictions are as follows:

- The component is a tape feeder.
- The component is allowed to be picked up simultaneously.
- The components are inspected by the same camera.
- The supply angles of the components are the same.

Step 1. To avoid component collisions, the components must be placed in ascending order of height in the PCB assembly. An RW is used to determine each component picked by each head. For example, if the allowable height range is 2 mm, we set component heights of 0.6 to 2.5 mm as a group (see Table 3).

Step 2. Generate a random number and match each number to the RW to determine the selected component for head 1. Match the component to the feeder station and generate a virtual ANC array. Apply the same procedure to select a component for head 2 and check the first virtual array. If the component already exists in the first virtual array,

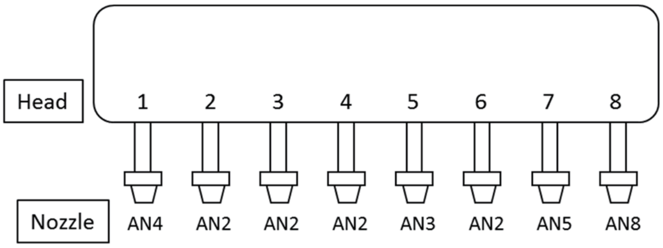


Fig. 6. ANC sequence.

Table 3
List of component heights.

| Component height (mm) | Component | Number of components | Nozzle |
|---|---|---|---|
| 0.6 | R1 | 35 | AN2 |
| 0.7 | R3 | 44 | AN2 |
| 1.3 | H2 | 20 | AN3 |
| 1.6 | H1 | 30 | AN2 |
| 2.5 | H5 | 10 | AN1 |
| 2.7 | R5 | 15 | AN4 |

assign the component to its head number according to the first virtual array. Then, select another component for head 2. Otherwise, assign this component to head 2 and generate a second virtual array. Repeat this procedure to generate the component sequence for all heads.

Step 3. Generate the corresponding ANC sequences based on each component in accordance with the applicable nozzles. As shown in Table 4, for instance, head 1 uses nozzle AN2 to pick component R1, head 2 uses nozzle AN2 to pick component H1, and so forth, until all the assignments are completed. In addition, heads 1, 3, 4, 5, and 8 pick components R1, H5, H1, R5, and H3 in the first sequence, and heads 2, 6, and 7 pick components H1, H3, and H2 in the second sequence, respectively.

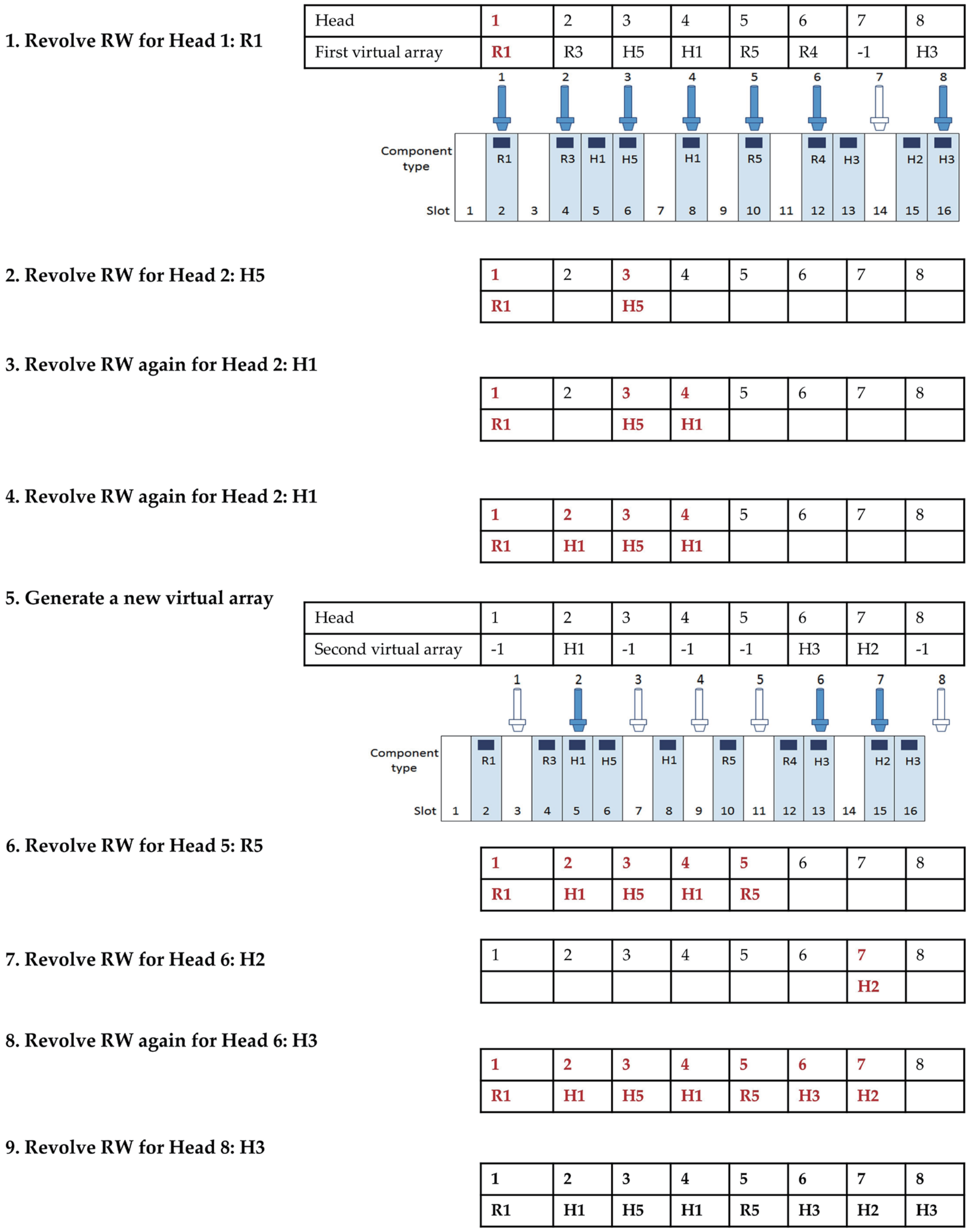
Figure 7 shows an example to illustrate the procedures in detail. First, based on the result of the RW, the first virtual array is {R1, R3, H5, H1, R5, R4, −1, H3} (“−1” indicates that no component is picked up for this head). Assign head 1 to pick up component R1. Second, allocate a component for head 2 using the RW. In this example, component H5 is selected, but H5 has already appeared in the first virtual array. Therefore, revolve the RW again for head 2; H1 is the new selected component for head 2. Since H1 is already listed in the first virtual array, we assign component H1 to head 4. Revolve the RW for head 2 again to select component H1. Assign H1 to head 2 and use the RW for heads 5 to 8. The final arrangement for heads 1 to 8 is {R1, H1, H5, H1, R5, H3, H2, H3}.

### 3.3 NNS

The NNS is deployed to determine the initial placement path based on the ANC sequence. Start from the feeder slot to find the nearest point on the PCB for the first component in the ANC sequence and repeat this process until all places in the sequence have been generated.

Table 4
Pick sequence for components.

| Head | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Pick sequence | 1 | 2 | 1 | 1 | 1 | 2 | 2 | 1 |
| Component | R1 | H1 | H5 | H1 | R5 | H3 | H2 | H3 |
| ANC | AN2 | AN2 | AN1 | AN2 | AN4 | AN5 | AN3 | AN5 |



**1. Revolve RW for Head 1: R1**

| Head | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| First virtual array | R1 | R3 | H5 | H1 | R5 | R4 | -1 | H3 |

1 2 3 4 5 6 7 8
Component type
R1 R3 H1 H5 H1 R5 R4 H3 H2 H3
Slot 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

**2. Revolve RW for Head 2: H5**

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| R1 |  | H5 |  |  |  |  |  |

**3. Revolve RW again for Head 2: H1**

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| R1 |  | H5 | H1 |  |  |  |  |

**4. Revolve RW again for Head 2: H1**

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| R1 | H1 | H5 | H1 |  |  |  |  |

**5. Generate a new virtual array**

| Head | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Second virtual array | -1 | H1 | -1 | -1 | -1 | H3 | H2 | -1 |

1 2 3 4 5 6 7 8
Component type
R1 R3 H1 H5 H1 R5 R4 H3 H2 H3
Slot 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

**6. Revolve RW for Head 5: R5**

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| R1 | H1 | H5 | H1 | R5 |  |  |  |

**7. Revolve RW for Head 6: H2**

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
|  |  |  |  |  |  | H2 |  |

**8. Revolve RW again for Head 6: H3**

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| R1 | H1 | H5 | H1 | R5 | H3 | H2 |  |

**9. Revolve RW for Head 8: H3**

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| R1 | H1 | H5 | H1 | R5 | H3 | H2 | H3 |

Fig. 7. (Color online) Example of ANC sequence selection.

Three assembly components, R1, H1, and C1, are illustrated as an example in Fig. 8. The NNS begins the search from the feeder station to find the nearest point on the PCB, which is component R1. Then the selected point is set as the starting point to find the next nearest point on the PCB (component H1). The searching process is repeated until all the components are placed on the PCB.

### 3.4 2-opt

Consider the initial placement path generated by the NNS as a TSP, then the 2-opt method is applied to enhance the performance for the initial placement path. This method exchanges two points of the path and generates some different paths that may be shorter than the initial path. As illustrated in Fig. 9, 2-opt generates the first new path by swapping the first two points in the initial placement, then creates the second new path by exchanging the second point with the third point in the first new path. The third path is generated by switching the second and third

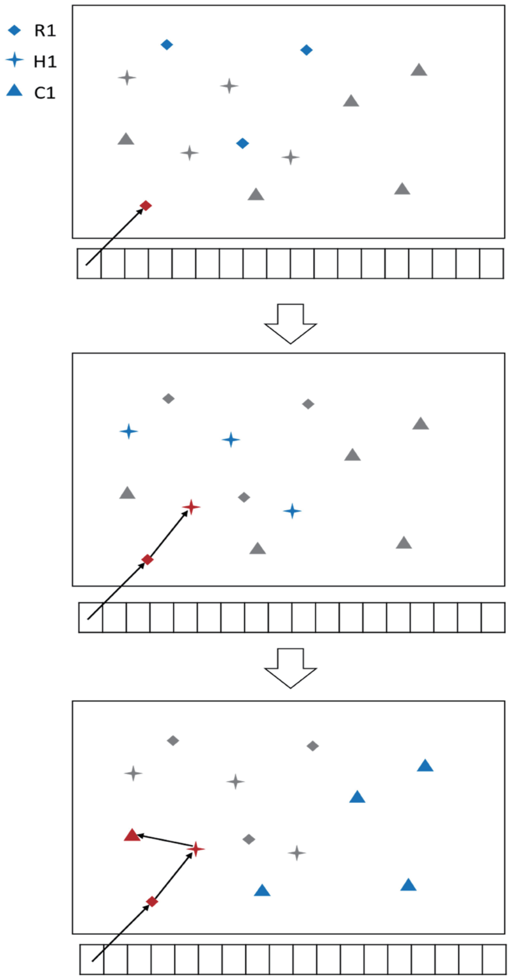


Fig. 8. (Color online) Example of the NNS.

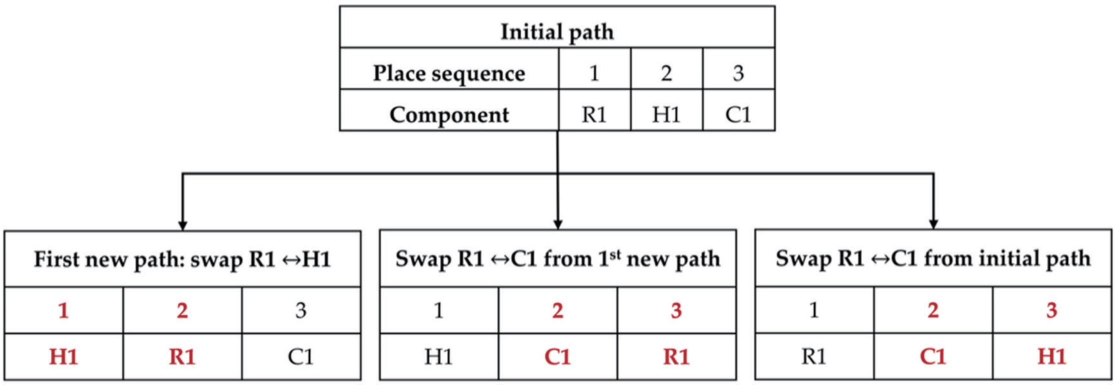


Fig. 9. (Color online) Example of 2-opt.

points from the initial path. Subsequently, select the shortest distance between the second and third paths as the final path. Then, the RW is used to generate the next pickup sequence and ANC sequence, and this procedure is repeated until all the scheduling for the assembly has been completed. Finally, the time consumed for each model is evaluated.

## 4. Results

Ten different examples of PCB assembly provided by a company are used to evaluate the performance of our proposed algorithm. The PCB information, including the number of components, component types, nozzle types, and heads, is shown in Table 5. The component height is 2 mm and the optimization time is set to 10 min. The algorithm is executed for each PCB 20 times.

Normally, it takes 8–10 h for a factory engineer to generate a feasible schedule, whereas our proposed algorithm can complete the work within 10 min, saving an enormous amount of time. Our best schedule showed an improvement of 15% as compared with the engineer’s schedule (shown in Table 6), indicating the superior performance of the proposed algorithm. Since the RW is based on a random strategy, there is a higher likelihood of generating a better schedule with a longer optimization time.

Table 5
Ten examples of PCB assembly.

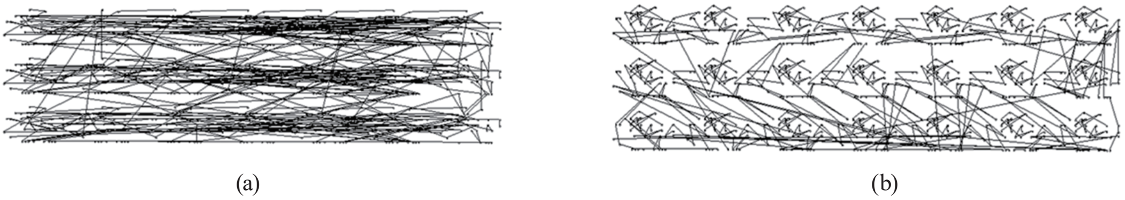
| PCB | Number of components | Number of component types | Number of nozzle types | Number of heads |
|---|---|---|---|---|
| 1 | 48 | 3 | 1 | 8 |
| 2 | 55 | 8 | 3 | 8 |
| 3 | 62 | 8 | 3 | 8 |
| 4 | 78 | 2 | 1 | 8 |
| 5 | 177 | 7 | 2 | 8 |
| 6 | 204 | 3 | 1 | 8 |
| 7 | 316 | 6 | 1 | 8 |
| 8 | 396 | 14 | 4 | 8 |
| 9 | 735 | 7 | 2 | 8 |
| 10 | 796 | 6 | 1 | 8 |

Table 6
Pick sequence for components.

| PCB | Engineer's schedule | Proposed algorithm | Improvement (%) |
|---|---|---|---|
| | Assembly time (s) | | |
| 1 | 34 | 32 | 5 |
| 2 | 101 | 86.69 | 14 |
| 3 | 112 | 108 | 3.5 |
| 4 | 140 | 119 | 15 |
| 5 | 311 | 294.38 | 5 |
| 6 | 146 | 141 | 3 |
| 7 | 221.835 | 221.746 | 0.04 |
| 8 | 738 | 691 | 6 |
| 9 | 1291 | 1180 | 8.5 |
| 10 | 552 | 521 | 5 |

Table 7
Comparison of head moving distance.

| PCB | Engineer's schedule | Proposed algorithm |
|---|---|---|
| | Distance (mm) | |
| 1 | 12255 | 10859 |
| 2 | 11035 | 10309 |
| 3 | 12596 | 12225 |
| 4 | 9716 | 5945 |
| 5 | 24799 | 18025 |
| 6 | 46511 | 43351 |
| 7 | 72998 | 71267 |
| 8 | 83275 | 81887 |
| 9 | 98201 | 76695 |
| 10 | 190415 | 184333 |


(a)

(b)

Fig. 10. Component placement paths of PCB 1 for (a) engineer's schedule and (b) proposed algorithm.

Table 7 presents the distances of the head movement in this study. All the distances obtained with the proposed algorithm are clearly shorter than those in the schedule generated by the engineer. Figures 10(a) and 10(b) show the placement paths from the engineer's schedule and the proposed algorithm for PCB 1, respectively. Our algorithm can reduce the number of repeated paths and significantly decrease both the distance and the time required for head movement.

## 5. Conclusions

Considering the restrictions on placement machine hardware, a hybrid algorithm for placement machine scheduling optimization was proposed in this study. The NNS was first used to find an initial placement path. Then the 2-opt method was used to improve the placement path. The algorithm does not require complex calculations or a long execution time to devise a feasible schedule. The proposed hybrid algorithm optimizes PCB assembly scheduling for an SMP machine with multiple heads within 10 min to solve (1) the component pick-and-place sequence, (2) the nozzle assignment problem, and (3) the ANC assignment problem under the restrictions of (1) component height, (2) pick, (3) place, and (4) simultaneous pickup. Ten examples were used to compare the performance of our proposed algorithm and a conventional heuristic algorithm performed by a factory engineer. The results show that the proposed algorithm has better performance, i.e., a placement path with a shorter movement distance and a shorter cycle time. This study provides a reference for optimizing the assembly scheduling in the PCB industry. There are many kinds of components in a PCB, and sometimes the components have different sizes, which might influence the final assignment of components. For example, larger components will occupy two head spaces and require a different head assignment. The restriction of the component shape is not considered in the current study, which can be examined in future work.

## References

1 S. H. Lee, T. H. Park, B. H. Lee, W. H. Kwon, and W. Kwon: Proc. 1998 IEEE Int. Conf. Robotics and Automation (IEEE, 1998). https://doi.org/10.1109/ROBOT.1998.676377

2 Y. Huang, L. Zhao, and P. Liu: J. Phys. Conf. Ser. **1605** (2020). https://doi.org/10.1088/1742-6596/1605/1/012029

3 M. Ayob and G. Kendall: Proc. IEEE Int. Symp. Assembly and Task Planning (IEEE, 2003). https://doi.org/10.1109/ISATP.2003.1217199

4 A.-R. Oh and T.-H. Park: Proc. SICE 2004 Annu. Conf. (IEEE, 2004).

5 S. Chen and Y. Shen: Proc. 29th Chinese Control Conf. (IEEE, 2010).

6 J. Gao, X. Zhu, A. Liu, Q. Meng, and R. Zhang: Symmetry **10** (2018) 633. https://doi.org/10.3390/sym10110633

7 T. Chen, J. Luo, J. Du, and Y. Hu: Proc. 31st Chinese Control Conf. (IEEE, 2012).

8 T. Chen, J. Luo, and Y. Hu: Proc. 2011 3rd Int. Workshop on Intelligent Systems and Applications (IEEE, 2011). https://doi.org/10.1109/ISA.2011.5873252

9 H. Liu, P. Yuan, J. Luo, and M. Zhang: Proc. 2017 36th Chinese Control Conf. (CCC) (IEEE, 2017). https://doi.org/10.23919/ChiCC.2017.8027802

10 K. Jeevan, A. Parthiban, K. N. Seetharamu, I. A. Azid, and G. A. Quadir: J. Electron. Manuf. **11** (2002) 69. https://doi.org/10.1142/S0960313102000230

11 A. Garcia-Naijera and C. A. Brizuela: Proc. 2005 IEEE Congr. Evolutionary Computation (IEEE, 2005). https://doi.org/10.1109/CEC.2005.1554865

12 P. Neammanee and M. Reodecha: CAIE **56** (2009) 294. https://doi.org/10.1016/j.cie.2008.06.010

13 G. Zhang, Z. Li, and X. Du: Proc. 2010 Int. Conf. Logistics Systems and Intelligent Management (ICLSIM) (IEEE, 2010). https://doi.org/10.1109/ICLSIM.2010.5461358

14 W. C. E. Lim, G. Kanagaraj, and S. G. Ponnambalam: J. Intell. Manuf. **27** (2016) 417. https://doi.org/10.1007/s10845-014-0873-z

15 J. Jiang, X. Chen, M. Zang, Z. Wang, and Z. Tan: Proc. 2010 2nd Int. Conf. Future Computer and Communication (IEEE, 2010). https://doi.org/10.1109/ICFCC.2010.5497291

16 J. Jiang, Z. Du, C. Liu, and K. Zhang: Proc. 2010 6th Int. Conf. Wireless Communications Networking and Mobile Computing (WiCOM) (IEEE, 2010). https://doi.org/10.1109/WICOM.2010.5601320

17 J. Jiang, C. Liu, K. Zhang, Z. Wang, Z. Tan, and X. Chen: Proc. 2010 2nd Int. Workshop on Intelligent Systems and Applications (IEEE, 2010). https://doi.org/10.1109/IWISA.2010.5473437

18 T. He, D. Li, and S. W. Yoon: Adv. Eng. Inf. **37** (2018) 66. https://doi.org/10.1016/j.aei.2018.04.007

19 J. Mumtaz, Z. Guan, L. Yue, L. Zhang, and C. He: Int. J. Prod. Res. **58** (2020) 6252. https://doi.org/10.1080/00207543.2019.1675917

20 Z. Li, X. Yu, J. Qiu, and H. Gao: IEEE Trans. Ind. Inf. (2021) (Early Access). https://doi.org/10.1109/TII.2021.3069459

## About the Authors

**Cheng-Jian Lin** received his B.S. degree in electrical engineering from Ta Tung Institute of Technology, Taipei, Taiwan, R.O.C., in 1986 and his M.S. and Ph.D. degrees in electrical and control engineering from National Chiao-Tung University, Taiwan, R.O.C., in 1991 and 1996, respectively. Currently, he is a chair professor of the Computer Science and Information Engineering Department, National Chin-Yi University of Technology, Taichung, Taiwan, R.O.C., and dean of Intelligence College, National Taichung University of Science and Technology, Taichung, Taiwan, R.O.C. His current research interests are machine learning, pattern recognition, intelligent control, image processing, intelligent manufacturing, and evolutionary robots. (cjlin@ncut.edu.tw)

**Chun-Hui Lin** received her M.S. degree in Computer Science from the University of Texas at Dallas, Texas, USA, in 2017. Currently, she is a Ph.D. student in computer science and information engineering at National Cheng Kung University, Tainan, Taiwan. Her research interests are signal/image processing, ultrasound material characterization, intelligent control, and machine/deep learning.