*Artificial Intelligence for Engineering Design, Analysis and Manufacturing*

**cambridge.org/aie**

**Research Article**

**Cite this article:** Lin C-J, Huang M-L (2019). Efficient hybrid group search optimizer for assembling printed circuit boards. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* **33**, 259–274. https://doi.org/10.1017/S0890060418000240

Received: 18 July 2017
Revised: 16 October 2018
Accepted: 16 October 2018
First published online: 17 December 2018

**Key words:**
Component pick-and-place sequence; group search optimizer; multihead placement machine; PCB assembly; surface mount technology

**Author for correspondence:**
Mei-Ling Huang, E-mail: huangml@ncut.edu.tw



# Efficient hybrid group search optimizer for assembling printed circuit boards

Cheng-Jian Lin[1] and Mei-Ling Huang[2]

[1]Department of Computer Science & Information Engineering, National Chin-Yi University of Technology and [2]Department of Industrial Engineering & Management, National Chin-Yi University of Technology, Taichung, Taiwan

## Abstract

Assembly optimization of printed circuit boards (PCBs) has received considerable research attention because of efforts to improve productivity. Researchers have simplified complexities associated with PCB assembly; however, they have overlooked hardware constraints, such as pick-and-place restrictions and simultaneous pickup restrictions. In this study, a hybrid group search optimizer (HGSO) was proposed. Assembly optimization of PCBs for a multihead placement machine is segmented into three problems: the (1) auto nozzle changer (ANC) assembly problem, (2) nozzle setup problem, and (3) component pick-and-place sequence problem. The proposed HGSO proportionally applies a modified group search optimizer (MGSO), random-key integer programming, and assigned number of nozzles to an ANC to solve the component picking problem and minimize the number of nozzle changes, and the place order is treated as a traveling salesman problem. Nearest neighbor search is used to generate an initial place order, which is then improved using a 2-opt method, where chaos local search and a population manager improve efficiency and population diversity to minimize total assembly time. To evaluate the performance of the proposed HGSO, real-time PCB data from a plant were examined and compared with data obtained by an onsite engineer and from other related studies. The results revealed that the proposed HGSO has the lowest total assembly time, and it can be widely employed in general multihead placement machines.

## Introduction

Multihead placement machines, which can be adapted to process various components, are the most flexible machines for assembling printed circuit boards (PCBs). However, the production efficiency of multihead placement machines is limited. Problems in the related optimization of production efficiency can be classified into three typical subproblems:

(1) Auto nozzle changer (ANC) assembly problem;
(2) Nozzle setup problem;
(3) Component pick-and-place sequence problem.

Various optimization methods have been developed to address these subproblems. For example, Grunow *et al.* (2004) proposed a three-step heuristic algorithm to solve the pick-and-place problem. For multihead placement machines with a specific place sequence, Knuutila *et al.* (2007) employed a greedy algorithm to optimize nozzle assignment and reduce the number of picks. The concept of component batches was employed by Ashayeri *et al.* (2011) to formulate a multihead placement optimization problem for a mixed integer program designed to balance placement head loadings and reduce the number of nozzle changes. Ball and Magazine (1998) defined the place sequence for a multihead machine as a postman problem. Leipälä and Nevalainen (1989) considered a place sequence for a multihead machine to be a traveling salesman problem. Kumar and Li (1995) used linear programming for feeder assignment and place sequence, where the local search was accompanied by 2 and 3 changes. Recently, meta-algorithms, including greedy algorithms, genetic algorithms (GAs), and particle swarm optimization (PSO), have been successfully applied to multihead machine problems. For example, Fu and Su (2000) employed a GA, simulated annealing algorithm, and Tabu search-based algorithm to solve a dynamically combinatorial problem. Moreover, Loh *et al.* (2007) applied a GA to auto nozzle changer (ANC) assignment and place sequence for a placement machine. Jiang *et al.* (2010a, 2010b, 2010c) employed an improved ant colony algorithm to optimize place sequence and minimize nozzle changes. A memetic algorithm was proposed to solve a placement machine scheduling problem (Neammanee and Reodecha, 2009). For PCB assembly, Du *et al.*, focused on feeder assignment and component pick-and-place sequence problems and presented a hybrid GA combining heuristic and genetic algorithms for evaluating the performance of feasible solutions (Du and Li, 2008). In another study of multihead placement machines, a GA with a distinct crossover operation and mutation operators was applied to yield satisfactory results on robot assembly time by reducing movement distance of a feeder carrier and PCB table (Lin and Zhu, 2008). An efficient neural network approach was presented to minimize the cycle time of a schedule for a cyclic job shop problem (Kechadi *et al*., 2013). Al-Anzi and Allahverdi focused on modeling a two-stage multimachine assembly scheduling problem by employing an artificial immune system to minimize total assembly time (Al-Anzi and Allahverdi, 2013). Other related studies include Smed *et al*., 1999; Chen *et al*., 2011; Chen and Shen, 2010; Jiang *et al*., 2010a, 2010b, 2010c; Luo *et al*., 2010; Zhang *et al*., 2010; Alkaya and Duman, 2013; Chen *et al*., 2012a, 2012b; Csaba and Nevalainen, 2015; Jensen, 2003; Chen and Chang, 1995; Park *et al*., 2005.

Kim *et al*., developed an ontology-based model representing morphological characteristics related to assembly joints (Kim *et al*., 2009). A two-phase automatic generation of assembly plans was designed. First, a graph-based procedure with topological and geometric description provided a layout of an assembled product. Second, heuristic knowledge of mechanical components and assembly processes were used to plan selected assembly sequences (Kroll *et al*., 1989). A new GA encoding scheme and selection criteria were proposed for automatic generation of electromechanical engineering designs (Peysakhov and Regli, 2003). Zeng *et al*. developed probability increment-based swarm optimization (PIBSO) through roulette wheel selection and probability updating to solve a PCB assembly optimization problem. PIBSO provides improved tour length and CPU running time (Zeng *et al*., 2014). Chen and Wichman (1993) developed a novel system that integrated neural networks to capture a design concept and instigated a rule-based system for automatically generating an assembly plan. Intelligent selective disassembly with an ant colony algorithm has been used to solve combinatorial optimization problems (Wang *et al*., 2003).

To simplify the investigated problems, the aforementioned studies have ignored some practical limitations of placement machines. This study aimed to overcome these limitations and the following hardware constraints: (1) component height restrictions, (2) time consumption for nozzle changes, (3) picking restrictions, (4) placing restrictions, (5) simultaneous pickup restrictions, and (6) component shape restrictions. Moreover, a hybrid group search optimizer (HGSO) is proposed to generate feasible and efficient scheduling solutions for PCB assembly.

## Machine description

Surface mount technology (SMT) equipment, also termed placement machines, are the most crucial and complex equipment in SMT production. Figure 1 presents the multihead placement machine examined in this study. The machine has eight heads for eight nozzles, and different nozzles can be selected according to the shape of the component to be installed. One ANC accommodates 16 small nozzles and 4 large nozzles. For continuous assembly of components, 90 feeder slots are integrated using tape or stick feeders. The head moves along the X–Y axis to pick components from a feeder slot and place them at appropriate positions on the PCB. A pick-and-place cycle is completed when all the picked components are placed at their appropriate positions.

The primary parts of a placement machine are as follows:

(1) Head: Each head has one nozzle that can pick one component.
(2) Nozzle: A nozzle is installed on a head and used to pick components. A nozzle of a specified shape can only pick up components of the same shape.
(3) ANC: An ANC is designed for holding and exchanging nozzles.
(4) Feeder: A feeder stores and supplies components. Each feeder can store one component.
(5) Slot: A slot is designed for feeder insertion.
(6) Feeder station: A feeder station holds feeders.
(7) PCB table: A PCB table is designed to fix PCBs.
(8) Arm: An arm moves the heads in an X–Y axis direction to appropriate positions for picking and placing components.
(9) Fly vision: This is designed to ensure that the components are correct and defect-free and to relocate X–Y coordinates.

## Problem definition

### *Restrictions of placement machine*

To generate an efficient schedule for the placement machine in this study, the following restrictions were considered:

(1) Component height: The component heights are different. To avoid collisions with already installed components, components with lower heights should be placed with higher priority.
(2) Nozzle setup: Components with different shapes require nozzles with the corresponding shapes, but changing the nozzles increases production time and cost.
(3) Picking restrictions: The moving distance of each head is practically limited. For example, in Figure 2a, b, heads 3–8 cannot reach components in slot c, and heads 1–2 cannot pick components in slot d, respectively.
(4) Placing restrictions: Not all heads can place components in all slots. For example, in Figure 3a, heads 7–8 cannot place components at position a, and in Figure 3b, heads 1–2 cannot place components at position b.
(5) Simultaneous pickup restrictions: Simultaneous pickup is essential for reducing the number of pickups and therefore production time. The following conditions are necessary for simultaneous pickup:
    a. Components are stored in the tape feeder.
    b. One feeder slot between component feeders.
    c. Picking angles for all components are the same.
    d. The same camera is used for fly vision.

In the first run, eight heads, excluding heads #5 and #6, can simultaneously pick components R6, H2, M2, H3, M3, and H5 (Fig. 4a). In the second run, components H2 and M2 can be picked by heads #5 and #6 (Fig. 4b). The increased number of simultaneously picked components reduces the number of pickups required.

(6) Component shape restrictions: When a component is larger than 15 mm, it requires the space of more than one head. In such cases, the adjacent head cannot be used. In our example in Figure 5, the sizes for components V1 and V2 are 36 and 45 mm, respectively; therefore, heads #2, #4, and #6 are idle when heads #3 and #5 are used (Fig. 5).

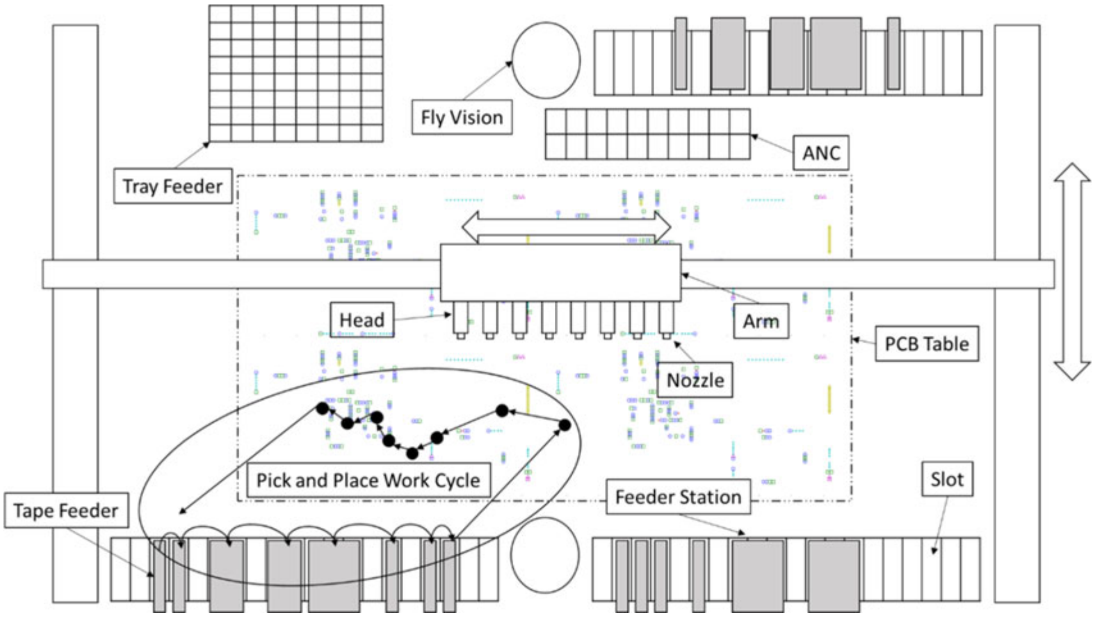


**Fig. 1.** Schematic of the multihead placement machine used in this study

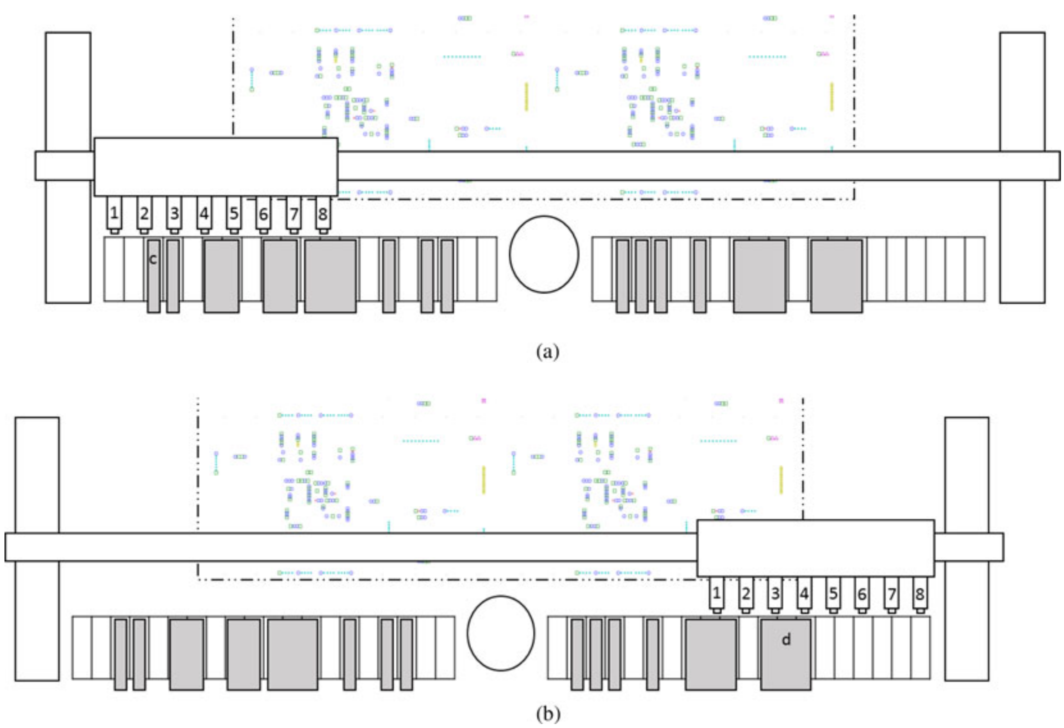


**Fig. 2.** Picking restrictions

## Problem statement

This study aimed to provide efficient scheduling with minimum production time for assembling PCBs. An interview with the site engineer responsible for the placement machine revealed that although the component positions in the feeders are not fixed, the arrangement seldom varies in practice. If the same components are used in different PCBs, their positions in the feeders are generally retained. The following three interacting problems are discussed herein:

(1) ANC assignment: component pick-and-place sequences interact with each other to enable determination of the optimal ANC.

https://doi.org/10.1017/S0890060418000240 Published online by Cambridge University Press

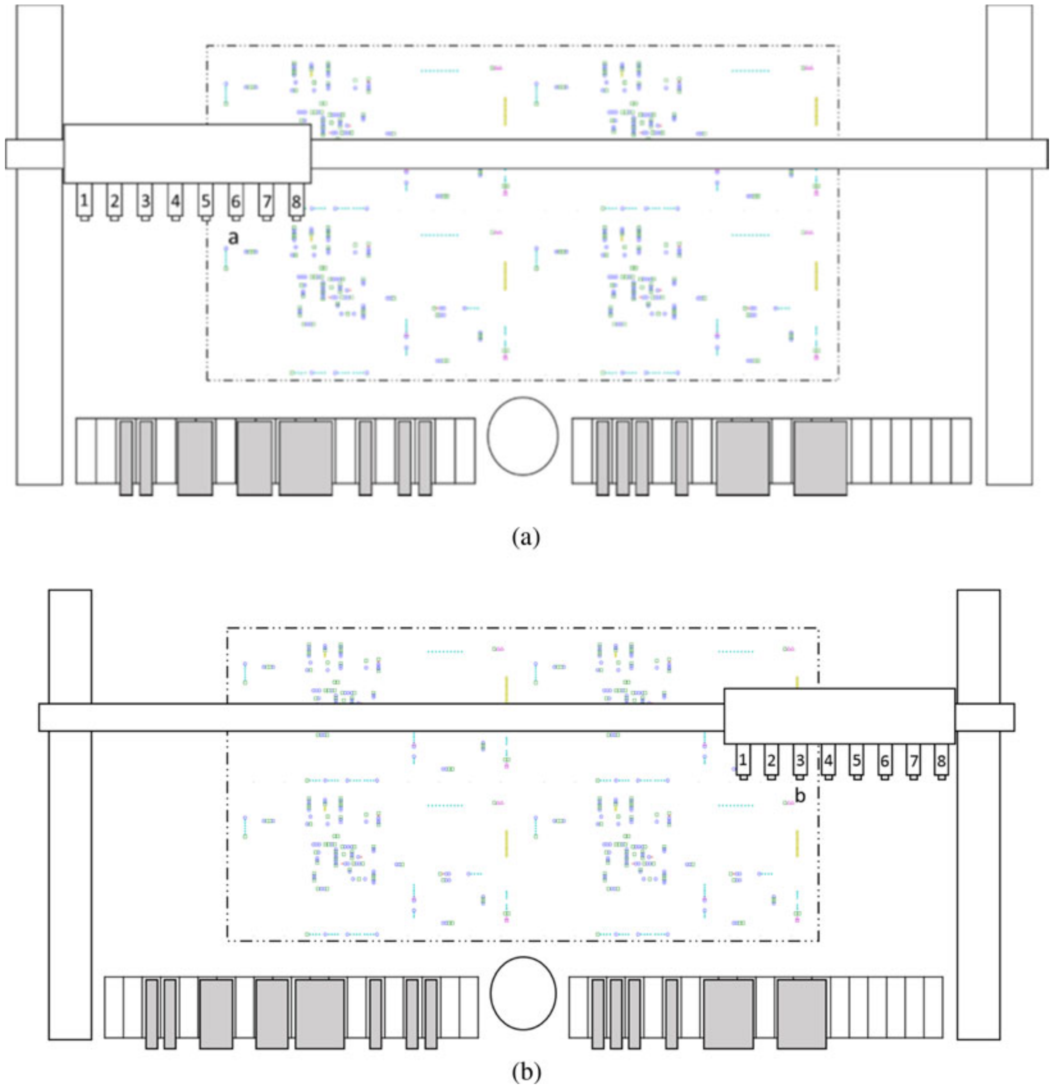


**Fig. 3.** Placing restrictions

(2) Nozzle assignment: The nozzle must be assigned by considering the pick, place, component height, and simultaneous pickup restrictions to reduce the number of pickups and nozzle changes, thereby reducing the total assembly time.
(3) Component pick-and-place sequence: The component pick-and-place sequence must be optimized so that head movement for pick-and-place operations uses the shortest path.

## Problem formulation

Considering limitations in the practical operation of a multihead placement machine, a mathematical model is proposed for optimizing the total assembly time, which includes pick time, place time, and ANC time. Eq. (1) can be applied to minimize the total assembly time for PCB production.

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

where $T_{\text{total}}$, $T_{\text{pick}}$, $T_{\text{place}}$, and $T_{\text{change}}$ are the total production time for PCB assembly, total pick time, total placing time, and total nozzle change time, respectively. $T_{\text{pick}}$, $T_{\text{place}}$, and $T_{\text{change}}$are given as follows:

$$T_{\text{pick}} = \begin{cases} \sum_{i=1}^{NP-1} (T(d(S_i, S_{i+1})) + T_z(\mathrm{i}+1) + \mathrm{TD}_{i+1} + \mathrm{TP}_{i+1}), & c = 1 \\ T(d(\mathrm{PC}_N, S_1)) + T_z(1) + \mathrm{TD}_1 + \mathrm{TP}_1 + \\ \sum_{i=1}^{NP-1} (T(d(S_i, S_{i+1})) + T_z(i+1) + \mathrm{TD}_{i+1} + \mathrm{TP}_{i+1}), & \text{otherwise} \end{cases} \tag{2}$$

$$T_{\text{place}} = T(d(S_N, \mathrm{PC}_1)) + T_z(1) + \mathrm{TB}_1 + \mathrm{TL}_1 + \sum_{j=2}^{N} (T(d(\mathrm{PC}_{j-1}, \mathrm{PC}_j)) + T_z(\mathrm{j}) + \mathrm{TB}_j + \mathrm{TL}_j) \tag{3}$$

https://doi.org/10.1017/S0890060418000240 Published online by Cambridge University Press

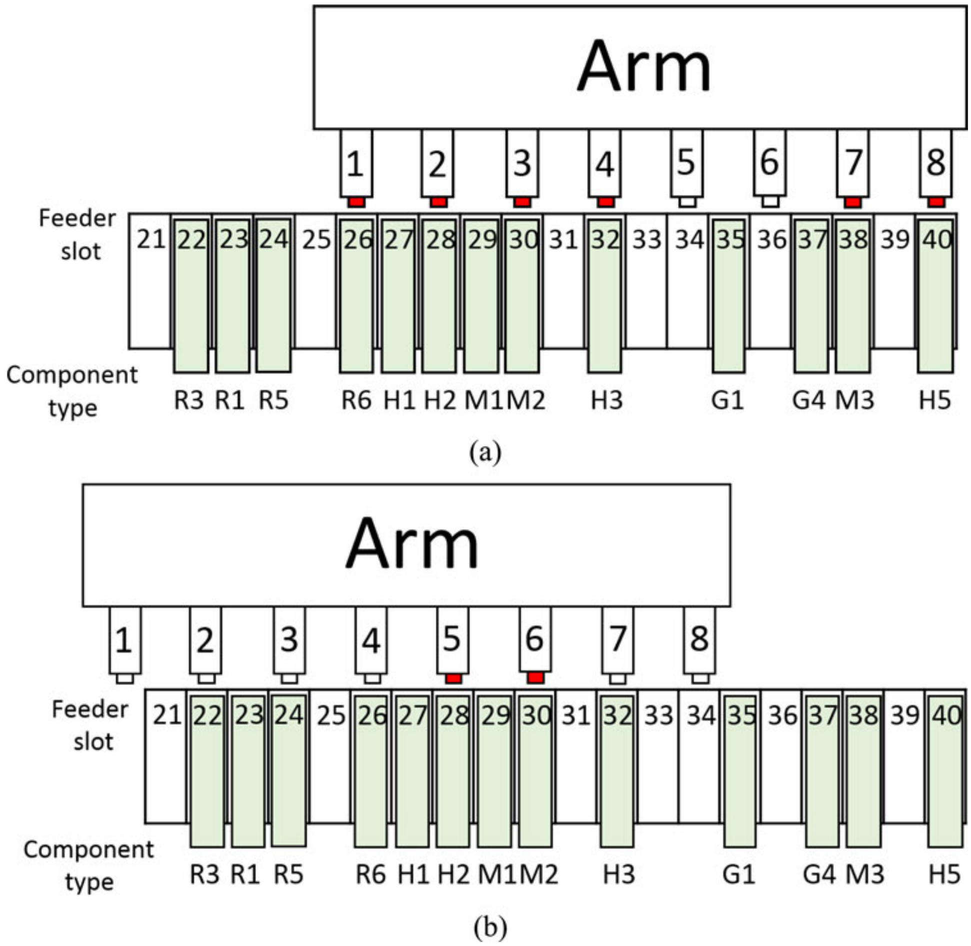


**Fig. 4.** Simultaneous pickups

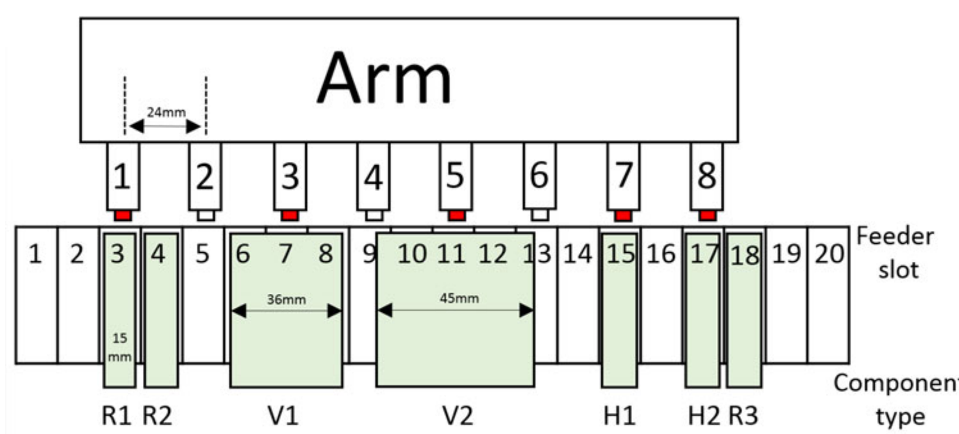


**Fig. 5.** Component shape restrictions

$$T_{\text{change}} = T(\text{d}(PC_N, A_1)) + T(d(A_{CN}, S_1)) + \sum_{k=1}^{\text{CN}} T(d(A_k, A_{k+1})) + \text{TN}_k] \quad (4)$$

where $\text{Min}(T_{\text{total}})$ indicates that the objective is to minimize the total assembly time; NC, $N$, and NP are the total number of cycles in the assembly, heads in the placement machine, and pickups in the assembly, respectively; and TD, TP, TB, and TN are the vacuum on delay time, total waiting time for picking components, total waiting time for placing components, load wait time, and

https://doi.org/10.1017/S0890060418000240 Published online by Cambridge University Press

total time for changing nozzles, respectively. Furthermore, CN is the total number of nozzle changes; PC is the component picking position; $S$ is the position of the active slot; A is the position of nozzle change in the ANC; $d()$ is the movement distance of the head; $T()$ is the time required for head movement; and$T_z(c)$ is the movement time in both directions along the Z axis.

## HGSO for scheduling optimization

The GSO is a heuristic algorithm that was developed in 2009 (He *et al*., 2009). The SO mimics the behavior of animals searching for food, resources, and places. The GSO reaches global optimum through a local search of team members. It involves few control parameters, and the associated calculations are easy. Under most circumstances, only slight adjustments are required to both parameters and team members in order to apply the GSO to different optimization problems.

Based on the GSO, an HGSO algorithm is proposed herein for optimizing PCB assembly time. The proposed HGSO overcomes several hardware limitations, including (1) component height restrictions, (2) time consuming nozzle changes, (3) picking restrictions, (4) placing restrictions, (5) simultaneous pickup restrictions, and (6) component shape restrictions. The proposed HGSO divides PCB assembly into three problems: the (1) ANC assembly problem, (2) nozzle setup problem, and (3) component pick-and-place sequence problem. The first step of the HGSO is to proportionally distribute nozzles to the ANC according to required nozzle usage, which is determined by the component picking sequence. The component heights must be considered to avoid collision (i.e., shorter components must be placed earlier). Then, the HGSO generates an efficient picking schedule by considering picking, simultaneous picking, component shape, and placing restrictions. Chaos local search and a population manager are utilized to increase the diversity of placing routes and reduce total PCB assembly time. Figure 6 presents the procedures of the proposed method.

### *Proportional distribution of nozzles to the ANC*

During production, any head in the placement machine may require nozzle change to pick the subsequent component. The ANC has 20 cells, with 16 small and 4 large cells (Table 1). PCBs are different with different numbers of components. For efficient production, the nozzles must be assigned to the ANC before commencing operation. In this study, the nozzles were distributed to the ANC according to the number of pickups by each nozzle. Table 1 presents nozzle arrangement involving one large nozzle (ANV1) and four small nozzles (AN3, AN5, AN6, and AN7).

Step 1: Ensure that at least one nozzle is assigned in the ANC to each component in Table 2, namely ANV1, AN3, AN5, AN6, and AN7.

Step 2: Subtract the number of cells that have been occupied in step 1 (i.e., $20-5=15$). Compute the number of nozzles required for each nozzle type (Table 3) according to the components to be picked up (Table 6).

Step 3: Ensure that the number of nozzles assigned for each type is not higher than the number of heads. For example, if the number of nozzles assigned for type AN3 exceeds the number of heads, reduce the number of nozzles to the maximum number of heads (Table 4).

Step 4: Repeat steps 2 and 3 until all cells in the ANC are occupied (Tables 5 and 6, Fig. 7).

### *HGSO algorithm*

To avoid collision during component placement, shorter components must be placed earlier. In the following example, components with a height of 0.6–2.6 mm (see Table 7) are grouped together and can be arbitrarily placed if height restriction is 2 mm.

A random number between 0 and 1 is generated and assigned to each member for executing the GSO algorithm (Table 8). $N$ and $x_i$ are the total number of components and the $i^{\text{th}}$ member in the GSO, respectively. Each member represents a scheduling solution.

The traditional GSO cannot be employed to schedule placement machines because this problem is a type of integer optimization. In this study, an MGSO with an added random key (Snyder and Daskin, 2006) was proposed for scheduling optimization of PCB assembly. Tables 9–11 illustrates the procedure. The first random number in Table 9 is 0.95. Component 6 is assigned to the first cell after considering component height, component shape, nozzle assignment, and picking restrictions. In Table 10, the initial solution with random numbers in Table 9 is adjusted by placing the elements in an ascending order: 0.12, 0.15, 0.75, 0.76, 0.88, 0.93, 0.95, and 0.99. Element 0 is assigned to the lowest random number (0.12), element 1 is assigned to the subsequent number (0.15), and so on (Table 11). Table 12 presents the solutions for the large components, where head #3 and #5 are idle in the first cycle because component 9 occupies the space beneath them.

The 2-opt method is employed to the solution of the initial path derived from Table 12. Table 13 presents the improved solution. A mathematical model that calculates the total assembly time is then applied to evaluate the performance of the probable solutions. Table 14 presents the final result.

### *Chaos local search*

Chaos local search (Liu *et al*., 2005), which is a new optimization technique, is applied to improve MGSO evolution performance. Chaos local search has several advantages for obtaining the initial solution, such as sensitivity and dependence, which ensure the diversity and appropriate search behavior of the entire space. The chaos search formula is as follows:

$$x_{n+1} = \mu \times x_n(1 - x_n),\ 0 \le x_0 \le 1 \tag{5}$$

where $\mu$ and $x$ are a control parameter and a random variable, respectively, with $n = 0,\ 1,\ 2,\ \ldots$ However, when $\mu = 4$, $x_0\{0,$ $0.25,\ 0.5,\ 0.75,\ 1\}$ can be applied to Eq. (5). Chaos is sensitive and dependent on initial conditions, and a slight initial difference can generate considerable variances given a long enough timeframe. The MGSO searches the entire space by using particular characteristics of chaos [Eq. (6)]:

$$cx_i^{(k+1)} = 4cx_i^{(k)}(1 - cx_i^{(k)}),\ i = 1,\ 2,\ \ldots\ldots, N \tag{6}$$

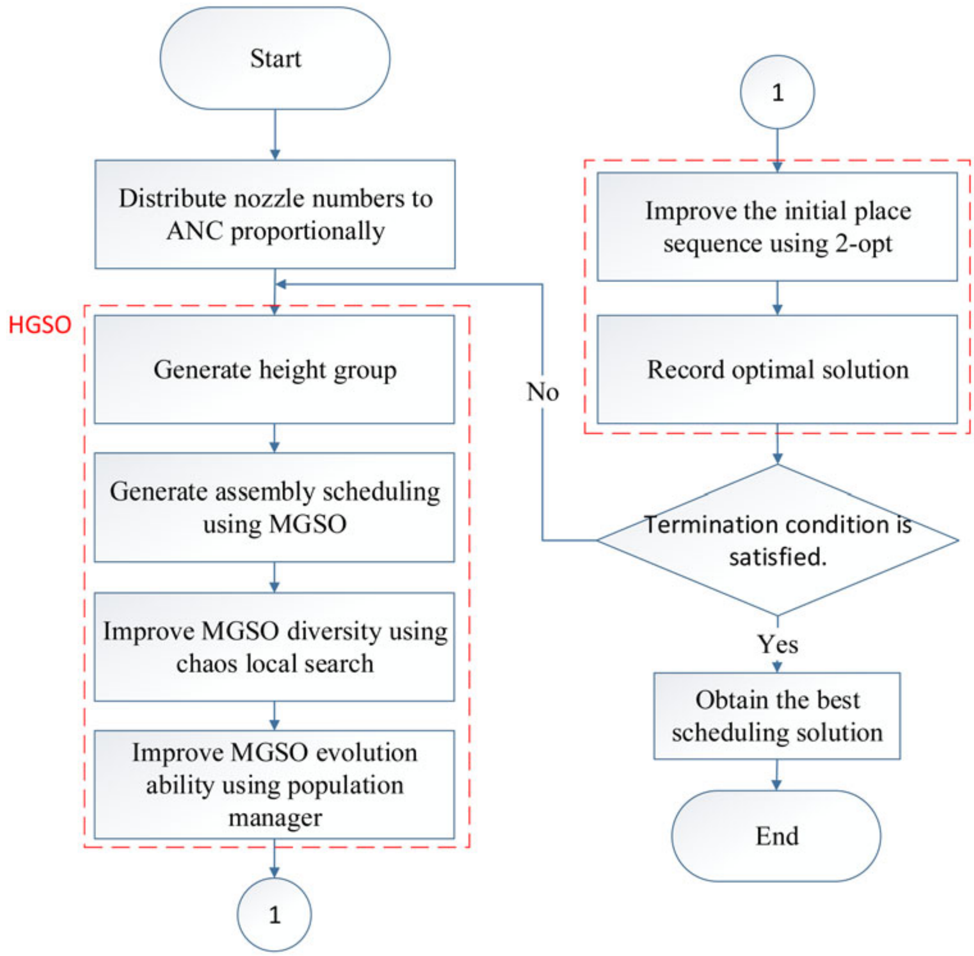


**Fig. 6.** Systematic procedures within HGSO

**Table 1.** Nozzle arrangement in ANC

| Size | Big Nozzle | | Small Nozzle | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Nozzle | ANV1 | ANV1 | AN3 | AN5 | AN6 | AN7 | AN3 | AN3 | AN3 | AN3 |
| | ANV1 | ANV1 | AN3 | AN3 | AN3 | AN5 | AN5 | AN5 | AN5 | AN6 |
| No. | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |

**Table 2.** ANC arrangement step 1

| Size | Big Nozzle | | Small Nozzle | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Nozzle | ANV1 | | AN3 | AN5 | AN6 | AN7 | | | | |
| No. | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |

**Table 3.** Number of nozzles per type

| Nozzle types | ANV1 | AN3 | AN5 | AN6 | AN7 |
|---|---|---|---|---|---|
| Number | 3 | 9 | 2 | 0 | 0 |

where $cx_i$ and $k$ are a chaos variable and the number of iterations, respectively. $cx_i^{(k)}$ has a range of [0, 1] and satisfies $cx_i^{(0)} \in (0, 1)$, $cx_i^{(0)} \not\subset \{0.25, 0.5, 0.75\}$. The procedure of chaos search is as follows:

https://doi.org/10.1017/S0890060418000240 Published online by Cambridge University Press

**Table 4.** Nozzle arrangement checklist

| Nozzle types | | ANV1 | AN3 | | AN5 | | AN6 | | | AN7 |
|---|---|---|---|---|---|---|---|---|---|---|
| Number | | 3 | 7 | | 2 | | 1 | | | 0 |
| Size | Big Nozzle | | Small Nozzle | | | | | | | |
| No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Nozzle | ANV1 | ANV1 | AN3 | AN5 | AN6 | AN7 | AN3 | AN3 | AN3 | AN3 |
| | ANV1 | ANV1 | AN3 | AN3 | AN3 | AN5 | AN5 | AN6 | | |
| No. | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |

**Table 5.** Final nozzle arrangement

| Size | Big Nozzle | | Small Nozzle | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Nozzle | ANV1 | ANV1 | AN3 | AN5 | AN6 | AN7 | AN3 | AN3 | AN3 | AN3 |
| | ANV1 | ANV1 | AN3 | AN3 | AN3 | AN5 | AN5 | AN5 | AN5 | AN6 |
| No. | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |

**Table 6.** Components per nozzle type

| Nozzle | Component shape | Number of component | Proportion of nozzle (%) |
|---|---|---|---|
| AN3 | Small | 330 | 71 |
| AN5 | Small | 90 | 19 |
| AN6 | Small | 30 | 6 |
| AN7 | Small | 18 | 4 |
| ANV1 | large | 18 | 100 |

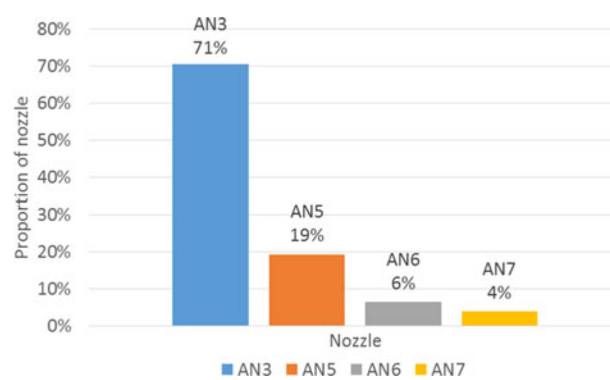


**Fig. 7.** Proportion of small nozzles

Step 1: Use Eq. (6) to produce chaos variables for the subsequent generation.

$$cx_i^{(k+1)}, cx_i^{(k)} = x_i \tag{7}$$

Step 2: Compare the new solution $cx_i^{(k+1)}$ with the original solution $cx_i^{(k)}$. Use the more favorable solution.

$$x_i = cx_i^{(k+1)} \tag{8}$$

Step 3: Repeat steps 1 and 2 until $i = N$.

## *Population manager*

Because similar subjects have similar fitness values and converge to likely positions, the worse subjects must be eliminated to improve efficiency and reduce computation. Similarity is evaluated as follows (Liang and Lee, 2015):

$$\left|\frac{f(x_i) - f(x_j)}{f(x_j)}\right| < \delta_{\mathrm{m}}\ ,\ \ x_{\mathrm{i}} - x_{\mathrm{j}} < r_{\mathrm{m}} \tag{9}$$

where $\delta_{\mathrm{m}}$ and $r_{\mathrm{m}}$ are threshold values, representing the average fitness value and average distance of the entire space, respectively; $\|\cdot\|$ denotes the distance between$x_{\mathrm{i}}$ and $x_{\mathrm{j}}$. Figure 8 illustrates this concept. Subjects Q3 and Q4 have similar fitness values (less than $\delta_{\mathrm{m}}$); however, the distance is larger than $r_{\mathrm{m}}$. The positions for Q5 and Q6 are close (less than $r_{\mathrm{m}}$); however, the fitness values are considerably different. The fitness values and positions

https://doi.org/10.1017/S0890060418000240 Published online by Cambridge University Press

**Table 7.** Component information

| Component height (mm) | Component type | Number of component | Nozzle | Component shape | Slot |
|---|---|---|---|---|---|
| 0.6 | R12 | 132 | AN3 | Small | 35 |
| | R13 | 36 | AN3 | Small | 38 |
| | R6 | 36 | AN3 | Small | 36 |
| | R8 | 18 | AN3 | Small | 34 |
| | R10 | 18 | AN3 | Small | 53 |
| | R5 | 36 | AN3 | Small | 37 |
| | R3 | 18 | AN3 | Small | 54 |
| 1 | D5 | 18 | ANV1 | Large | 5 |
| 1.4 | C2 | 36 | AN3 | Small | 52 |
| 1.5 | D1 | 18 | AN5 | Small | 42 |
| 2.2 | A1 | 72 | AN5 | Small | 56 |
| 3.5 | D2 | 30 | AN6 | Small | 45 |
| 4 | V1 | 18 | AN7 | Small | 58 |

**Table 8.** Coding method for GSO members

| Dimensions | 1 | | | | ……… | | | | $N$ |
|---|---|---|---|---|---|---|---|---|---|
| $x_i$ | 0.95 | 0.99 | 0.15 | 0.76 | 0.75 | 0.93 | 0.12 | 0.88 | …… |

**Table 9.** Initial solution: random numbers

| $x_i$ | 0.95 | 0.99 | 0.15 | 0.76 | 0.75 | 0.93 | 0.12 | 0.88 | … |
|---|---|---|---|---|---|---|---|---|---|
| Element | 6 | 4 | 5 | 8 | 3 | 0 | 7 | 1 | … |

**Table 10.** Members arranged in ascending order

| $x_i$ | 0.93 | 0.76 | 0.88 | 0.99 | 0.75 | 0.12 | 0.95 | 0.15 | … |
|---|---|---|---|---|---|---|---|---|---|
| Element | 6 | 4 | 5 | 8 | 3 | 0 | 7 | 1 | … |

**Table 11.** Final solution of members

| $x_i$ | 0.93 | 0.76 | 0.88 | 0.99 | 0.75 | 0.12 | 0.95 | 0.15 | … |
|---|---|---|---|---|---|---|---|---|---|
| Element | 6 | 4 | 5 | 8 | 3 | 0 | 7 | 1 | … |
| Head No | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | … |
| Number of cycle | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | … |
| Slot No | 34 | 36 | 38 | 34 | 52 | 38 | 56 | 42 | … |
| Pick sequence | 1 | 1 | 1 | 2 | 3 | 2 | 3 | 2 | … |
| Place sequence | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | … |

https://doi.org/10.1017/S0890060418000240 Published online by Cambridge University Press

**Table 12.** MGSO of large components

| $x_i$ | 0.91 | 0.74 | 0.84 | 0.83 | 0.71 | 0.13 | 0.81 | 0.14 | … |
|---|---|---|---|---|---|---|---|---|---|
| Element | 6 | 4 | 9 | 8 | 3 | 0 | 7 | 1 | … |
| Head No | 1 | 2 | 4 | 6 | 7 | 8 | 1 | 2 | … |
| Number of cycle | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | … |
| Slot No | 34 | 36 | 5 | 34 | 52 | 38 | 56 | 42 | … |
| Pick sequence | 1 | 1 | 2 | 3 | 4 | 3 | 1 | 2 | … |
| Place sequence | 1 | 2 | 3 | 4 | 5 | 6 | 1 | 2 | … |

**Table 13.** The 2-opt method

| $x_i$ | 0.93 | 0.76 | 0.88 | 0.99 | 0.75 | 0.12 | 0.95 | 0.15 | … |
|---|---|---|---|---|---|---|---|---|---|
| Element | 6 | 4 | 5 | 8 | 3 | 0 | 7 | 1 | … |
| Head No | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | … |
| Number of cycle | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | … |
| Slot No | 34 | 36 | 38 | 34 | 52 | 38 | 56 | 42 | … |
| Pick sequence | 1 | 1 | 1 | 2 | 3 | 2 | 3 | 2 | … |
| Place sequence | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | … |
| Place sequence 1 | 2 | 1 | 3 | 4 | 5 | 6 | 7 | 8 | … |
| Place sequence 2 | 2 | 3 | 1 | 4 | 5 | 6 | 7 | 8 | … |
| Place sequence 3 | 2 | 3 | 4 | 1 | 5 | 6 | 7 | 8 | … |
| Place sequence…. | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1 | … |

**Table 14.** Final scheduling results

| Cycle | Head | Pick sequence | Place sequence | Component No | Slot | Nozzle |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 2 | 6 | 34 | AN3 |
| 1 | 2 | 1 | 3 | 4 | 36 | AN3 |
| 1 | 3 | 1 | 4 | 5 | 38 | AN3 |
| 1 | 4 | 2 | 5 | 8 | 34 | AN3 |
| 1 | 5 | 3 | 6 | 3 | 52 | AN3 |
| 1 | 6 | 2 | 7 | 0 | 38 | AN3 |
| 1 | 7 | 3 | 8 | 7 | 56 | AN5 |
| 1 | 8 | 2 | 1 | 1 | 42 | AN5 |
| *n* | 1 | 2 | 1 | 40 | 34 | AN3 |
| *n* | 2 | 3 | 2 | 46 | 52 | AN3 |
| *n* | 3 | 2 | 3 | 26 | 38 | AN3 |
| *n* | 4 | 3 | 4 | 8 | 56 | AN5 |
| *n* | 5 | 2 | 5 | 2 | 42 | AN5 |
| *n* | 6 | 1 | 6 | 39 | 34 | AN3 |
| *n* | 7 | 1 | 7 | 44 | 36 | AN3 |
| *n* | 8 | 1 | 8 | 43 | 38 | AN3 |

https://doi.org/10.1017/S0890060418000240 Published online by Cambridge University Press

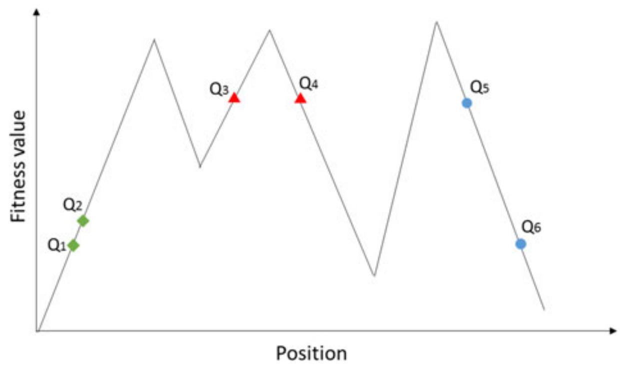


**Fig. 8.** Fitness value and position

**Fig. 9.** Placement machine EM-780

for Q1 and Q2 are similar, which satisfy Eq. (9), and the lower value (Q1) can be deleted.

## Experimental results and comparisons

The real-time data from a placement machine (EM-780, Evest Corporation, Taoyuan, Taiwan; Figure 9) were used to evaluate the feasibility of the proposed method. In this study, 10 PCBs with different numbers and types of components, nozzle types, and feeders (Table 15) were assembled.

According to Section 'GSO for scheduling optimization', the initial settings are as follows: $N = 100$, iterations = 3000, scroungers = 80%, rangers = 20%, and head angle = $0.25\pi$. The maximum iteration is $a = \text{round}\sqrt{n+1}$. If the producer cannot determine the improved area after $a$ iterations, $U_i$ and $L_i$ are equal to 1 and −1. Here, $n$ and $l_{\max} = \sqrt{\sum_{i=1}^{N}(U_i - L_i)^2}$ are the number of components and maximum pursuit distance, respectively. Figure 10 displays the number of pickups, total assembly time, and moving distance obtained using the proposed method.

**Table 15.** PCB information

| PCB ID | Components | Component types | Nozzle types | Enable feeders |
|---|---|---|---|---|
| PCB_1 | 796 | 6 | 1 | 6 |
| PCB_2 | 198 | 3 | 1 | 3 |
| PCB_3 | 396 | 14 | 4 | 14 |
| PCB_4 | 100 | 11 | 3 | 14 |
| PCB_5 | 48 | 3 | 1 | 3 |
| PCB_6 | 59 | 8 | 3 | 8 |
| PCB_7 | 78 | 2 | 1 | 2 |
| PCB_8 | 735 | 7 | 2 | 7 |
| PCB_9 | 151 | 7 | 2 | 7 |
| PCB_10 | 291 | 6 | 1 | 6 |

Yan *et al.* (2011) proposed GSPSO, which integrates high-speed computation in PSO with the favorable high-dimensional performance of GSO. Chen *et al.* (2012*a*, 2012*b*) presented an improved group search optimizer (IGSO) with a quantum-behaved operator for scroungers according to a certain probability of improving GSO convergence performance. In this method, the scroungers are categorized into two types: the scroungers that update their positions with quantum PSO operators and those that search for opportunities to join the resources found by the producer. Nian *et al.* (2013) proposed a differential evolution (DE) GSO, which is a hybrid of DE and GSO. DE prevents traditional GSO from falling into local optimal and enhances its accuracy. Table 16 compares the proposed method and related algorithms. The proposed method required the least total PCB assembly time for the 10 PCBs out of the compared algorithms. Figure 11 shows the learning curve for PCB_7, revealing that the total assembly times of PCB_7 using the proposed method and IGSO are similar [38]. Table 16 shows that the total average assembly times for the 10 PCBs (i.e., PCB_1–PCB_10) were 531.8, 523.3, 517.7, 495.3, and 394.3 s for the GSO (He *et al.*, 2009), GSPSO (Yan and Shi, 2011), IGSO (Chen *et al.*, 2012*a*, 2012*b*), DEGSO (Nian *et al.*, 2013), and proposed method, respectively. Therefore, the total average assembly time of the proposed method is superior to that of other methods (He *et al.*, 2009; Yan and Shi, 2011; Chen *et al.*, 2012*a*, 2012*b*; Nian *et al.*, 2013).

Algorithm performance in this study is determined in terms of the (1) total number of pickups, (2) total head movement distance, and (3) total assembly time. The results of the proposed method were compared with those of related algorithms, such as GAs or the improved shuffled frog leaping algorithm (Loh *et al.*, 2001; Sun *et al.*, 2005; Chang *et al.*, 2007; Ho and Ji, 2010; Zhu and Zhang, 2014). The number of pickups considerably influences PCB assembly, with total assembly time reducing with decreasing number of pickups. The proposed method required the least number of pickups for all experimental PCBs, excluding PCB_2 (Table 17). Figures 12–14 exhibit improvements calculated using the following equation:

$$\text{Improvement} = \frac{\text{Onsite\_Engineer} - \text{Our\_method}}{\text{Onsite\_Engineer}} \times 100\% \quad (10)$$

Assembly time reduces with decreasing total head movement

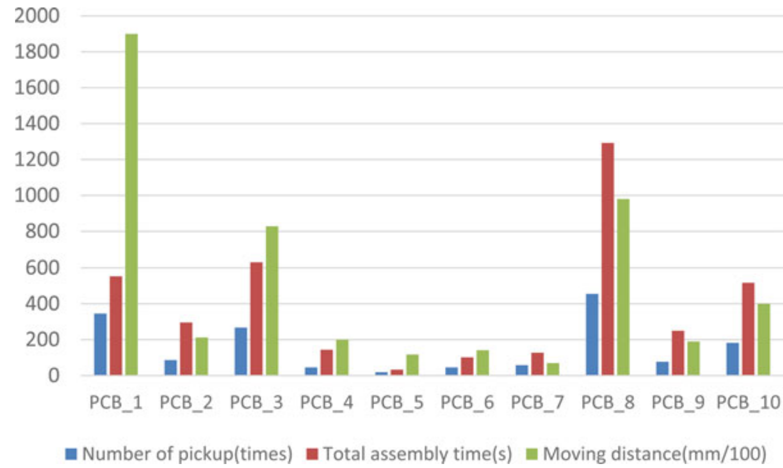


**Fig. 10.** Number of pickups, total assembly time, and moving distance obtained through the proposed method

**Table 16.** Total assembly time in the proposed method and related GSOs

| | Total PCB assembly time (s) | | | | |
|---|---|---|---|---|---|
| PCB ID | GSO [33] | GSPSO[37] | IGSO[38] | DEGSO[39] | Proposed method |
| PCB_1 | 701.41 | 689.781 | 682.031 | 671.252 | 552.925 |
| PCB_2 | 385.43 | 371.9 | 367.25 | 312.35 | 295.796 |
| PCB_3 | 861.31 | 821.63 | 792.12 | 701.33 | 630.982 |
| PCB_4 | 255.296 | 252.988 | 254.596 | 247.52 | 143.316 |
| PCB_5 | 40.4064 | 38.125 | 37.8533 | 37.7147 | 32.8864 |
| PCB_6 | 162.536 | 156.132 | 152.722 | 150.22 | 101.261 |
| PCB_7 | 132.289 | 130.927 | 127.917 | 130.135 | 127.857 |
| PCB_8 | 1710.431 | 1694.912 | 1660.29 | 1672.94 | 1292.1 |
| PCB_9 | 367.913 | 389.897 | 390.124 | 354.163 | 249.459 |
| PCB_10 | 701.416 | 687.198 | 711.79 | 675.61 | 516.653 |
| Average | 531.8 | 523.3 | 517.7 | 495.3 | 394.3 |

distance. Table 18 presents the experimental results, which indicate that the proposed algorithm can effectively reduce total head movement distance. Table 18 outlines the operation for PCB_6, showing that the head movement distance generated by the proposed algorithm is longer than that produced by an onsite engineer. Figure 13 reveals improvement in the total head movement distance achieved using the proposed algorithm.

From Table 19 and Figure 14, the proposed HGSO algorithm outperforms other algorithms and the onsite engineer in terms of assembly time. Although the total number of pickups for PCB_1, PCB_3, and PCB_8 in Table 17 did not differ substantially, the total head movement distances (Table 18) were clearly different. Figure 14 shows the excellent performance of the proposed HGSO in terms of total assembly time. In practice, onsite engineers can appropriately design optimal pickup sequences but not path optimization. The proposed HGSO considers the total number of pickups and the total head movement distance to reduce the total assembly time. In the total assembly time, considerably more time is spent on pickups than it is on head movement. Table 20 presents the percentages of pickup time, placing time, and ANC time. To reduce total assembly time, a higher head movement distance is traded off for fewer pickups. Studies (Loh *et al.*, 2001) and (Jiang *et al.*, 2010*a*, 2010*b*, 2010*c*) have decomposed the scheduling problem into two stages: optimization (minimization) of (1) the number of pickups and (2) the place movement based on the results of stage one. The proposed HGSO is less complicated than the two-stage method.

## Discussion and conclusions

This study aimed to perform scheduling optimization for reducing total PCB assembly time. Scheduling for a placement machine is a highly comprehensive task. Several studies have overlooked machine limitations to simplify the problem. However, the generated schedules produced by these studies cannot be practically applied to production lines. By contrast, this study proposed an HGSO that considers the limitations of placement machines to generate an applicable schedule are summarized as follows.

(1) Component height: The height of each component could be different, and the components are picked and placed in the ascending order of the height to avoid collision, i.e., The component with the lowest height must be placed first at the beginning of the placement machine during work.

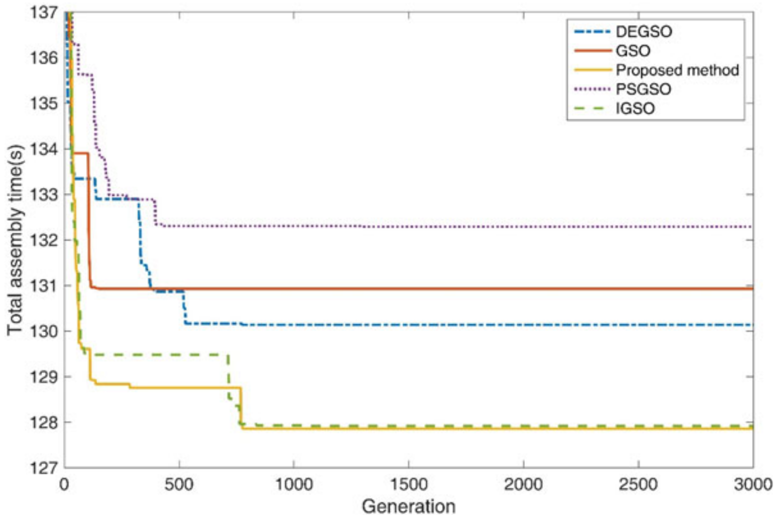


**Fig. 11.** Learning curve of PCB_7

**Table 17.** Comparison of total number of pickups

| PCB ID | Total number of pickups (times) | | | | | | |
|---|---|---|---|---|---|---|---|
| | Onsite engineer | Loh *et al.*, [8] | Sun *et al.*, [18] | Zhu *et al.*, [19] | Ho *et al.*, [20] | Chang *et al.*, [21] | Proposed method |
| PCB_1 | 345 | 358 | 335 | 601 | 557 | 586 | 345 |
| PCB_2 | 82 | 87 | 91 | 90 | 81 | 91 | 86 |
| PCB_3 | 267 | 286 | 254 | 367 | 356 | 374 | 267 |
| PCB_4 | 60 | 85 | 83 | 93 | 87 | 87 | 45 |
| PCB_5 | 19 | 26 | 34 | 35 | 34 | 39 | 19 |
| PCB_6 | 55 | 47 | 43 | 53 | 52 | 54 | 45 |
| PCB_7 | 59 | 58 | 63 | 67 | 61 | 65 | 58 |
| PCB_8 | 455 | 442 | 428 | 585 | 594 | 588 | 455 |
| PCB_9 | 80 | 82 | 91 | 119 | 122 | 115 | 77 |
| PCB_10 | 182 | 215 | 198 | 235 | 232 | 228 | 182 |

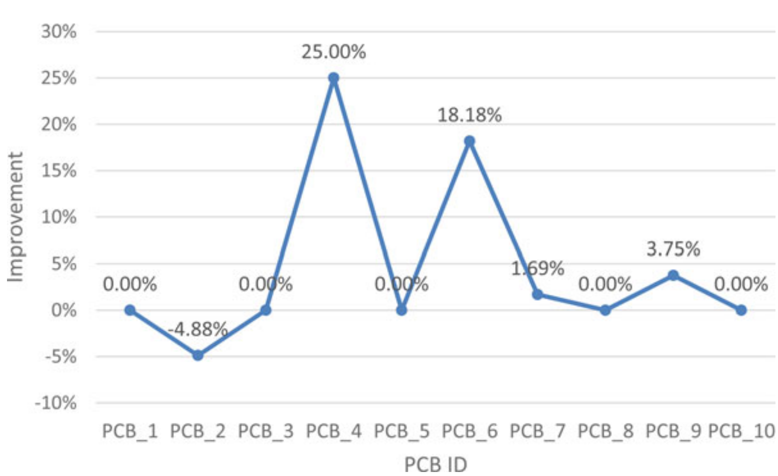


**Fig. 12.** Improvement in total number of pickups

https://doi.org/10.1017/S0890060418000240 Published online by Cambridge University Press

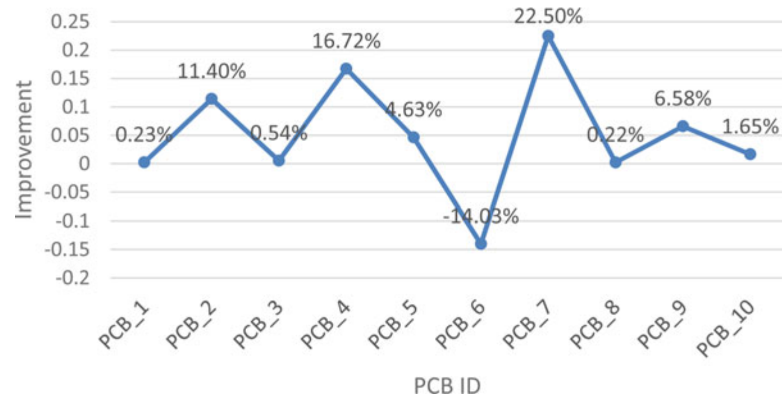


**Fig. 13.** Improvement in total head movement distance

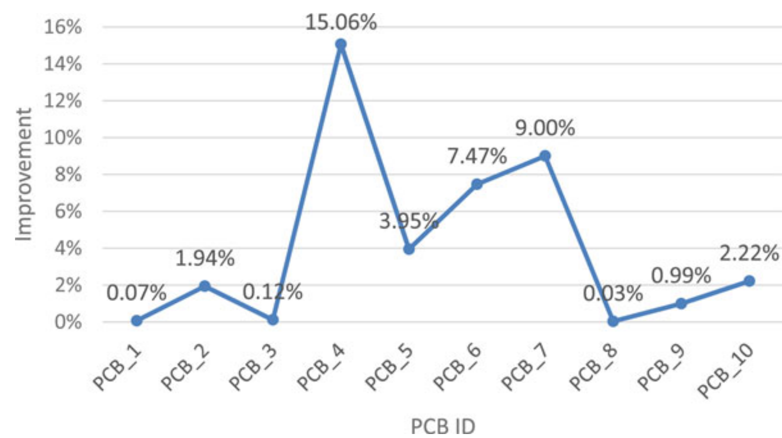


**Fig. 14.** Improvement in total assembly time

**Table 18.** Total head movement distance

| | Total head movement distance (mm) | | | | | | |
|---|---|---|---|---|---|---|---|
| PCB ID | Onsite Engineer | Loh *et al.* [8] | Sun *et al.* [18] | Zhu *et al.* [19] | Ho *et al.* [20] | Chang *et al.* [21] | Proposed method |
| PCB_1 | 190415 | 214840 | 185643 | 217102 | 214007 | 217824 | 189981 |
| PCB_2 | 23980 | 32312.7 | 33045 | 28935.4 | 28629.5 | 30452.2 | 21245.9 |
| PCB_3 | 83275 | 92201 | 86887 | 126815 | 121632 | 138081 | 82828.5 |
| PCB_4 | 23997.5 | 34597.2 | 32089 | 29007.1 | 23741.7 | 33255.7 | 19984.7 |
| PCB_5 | 12255.6 | 14539.5 | 14319.4 | 11945.8 | 11946.4 | 12700.8 | 11688.4 |
| PCB_6 | 12378.1 | 18313.7 | 18153.7 | 17285.2 | 15513 | 18163.6 | 14114.7 |
| PCB_7 | 9716.34 | 9033.89 | 9128.17 | 8986.48 | 8951.44 | 9362.49 | 7530.61 |
| PCB_8 | 98201 | 82474 | 74497 | 113480 | 115105 | 112546 | 97981.5 |
| PCB_9 | 20268.5 | 24712.7 | 26175 | 23234.4 | 22693.3 | 24074.8 | 18934.7 |
| PCB_10 | 40671.4 | 48085 | 46348.9 | 37400.9 | 37606.1 | 37422.9 | 39999.8 |

(2) The Time consuming of nozzle changes: shapes of the components are different, and each matches a unique corresponding nozzle. When the nozzle is not applicable, the head need to exchange the nozzle. The process of nozzle changes takes time. Some scholars ignore the time consuming of nozzle changes.
(3) Picking restrictions: In the placement machine, not every head can pick up component in all feeders because there are some positions that the head cannot reach.
(4) Placing restrictions: As above, there are some destinations that head cannot arrive.
(5) Simultaneous pickup restrictions: Simultaneous pickup is important in the placement machine process. Practically, we want to simultaneously pick up components as many as possible to reduce the time consuming on picking components. The restrictions of the head pitch and the feeder pitch should be under consideration.

https://doi.org/10.1017/S0890060418000240 Published online by Cambridge University Press

**Table 19.** Total assembly time

| | Total PCB assembly time (s) | | | | | | |
|---|---|---|---|---|---|---|---|
| PCB ID | Onsite engineer | Loh *et al.* [8] | Sun *et al.* [18] | Zhu *et al.* [19] | Ho *et al.* [20] | Chang *et al.* [21] | Proposed method |
| PCB_1 | 553.289 | 622.377 | 619.504 | 677.866 | 659.701 | 675.152 | 552.925 |
| PCB_2 | 301.64 | 311.177 | 313.091 | 315.512 | 312.853 | 314.367 | 295.796 |
| PCB_3 | 631.713 | 745.796 | 711.205 | 1278.01 | 1157.9 | 1298.3 | 630.982 |
| PCB_4 | 168.727 | 245.167 | 236.143 | 291.864 | 257.51 | 298.508 | 143.316 |
| PCB_5 | 34.2381 | 38.6468 | 40.1539 | 37.7527 | 36.3122 | 40.3129 | 32.8864 |
| PCB_6 | 109.435 | 133.457 | 132.342 | 173.749 | 147.372 | 157.458 | 101.261 |
| PCB_7 | 140.505 | 138.234 | 141.839 | 143.838 | 138.734 | 143.88 | 127.857 |
| PCB_8 | 1292.53 | 1346.26 | 1341.29 | 1658.94 | 1676.91 | 1663.76 | 1292.1 |
| PCB_9 | 251.964 | 281.621 | 307.634 | 344.778 | 343.124 | 345.683 | 249.459 |
| PCB_10 | 528.366 | 539.662 | 530.19 | 660.545 | 645.079 | 658.863 | 516.653 |

**Table 20.** Total assembly time

| PCB ID | Pick time | | Place time | | ANC change time | | Total assembly time |
|---|---|---|---|---|---|---|---|
| PCB_1 | 208.701 | 37.74% | 344.224 | 62.26% | 0 | 0.00% | 552.925 |
| PCB_2 | 134.002 | 45.30% | 161.794 | 54.70% | 0 | 0.00% | 295.796 |
| PCB_3 | 315.265 | 49.96% | 293.217 | 46.47% | 22.5 | 3.57% | 630.982 |
| PCB_4 | 58.849 | 41.06% | 51.967 | 36.26% | 32.5 | 22.68% | 143.316 |
| PCB_5 | 11.8341 | 35.98% | 21.0523 | 64.02% | 0 | 0.00% | 32.8864 |
| PCB_6 | 38.255 | 37.78% | 30.506 | 30.13% | 32.5 | 32.10% | 101.261 |
| PCB_7 | 47.466 | 37.12% | 80.391 | 62.88% | 0 | 0.00% | 127.857 |
| PCB_8 | 488.989 | 37.84% | 758.111 | 58.67% | 45 | 3.48% | 1292.1 |
| PCB_9 | 101.363 | 40.63% | 143.096 | 57.36% | 5 | 2.00% | 249.459 |
| PCB_10 | 199.212 | 38.56% | 304.941 | 59.02% | 12.5 | 2.42% | 516.653 |

(6) Component-shape restrictions: the components differ in shape and size. If the component shape is larger than the limit, the adjacent head cannot pick up the component.

Compared with the related works in this field, this study considers more restrictions than others. This is the first study includes component heights and component shapes in the scheduling problem of a placement machine. In addition, this study modified the traditional group search optimizer for the possible usage for PCB assembly. The results indicate that the proposed HGSO is superior to related algorithms and an onsite engineer in terms of total assembly time. Although the diversity of placement machines for PCB assembly is many, the key structures of placement machines are the same. The proposed hybrid group search optimizer possesses flexibility on adjusting the parameters according to the head numbers, number of nozzle types, etc. and can be widely applied to general multihead placement machines.

**Acknowledgments.** This study was partially supported by the Ministry of Science and Technology, Taiwan, ROC, under the grant MOST 105-2221-E-167-012.

## References

**Al-Anzi F and Allahverdi A** (2013) An artificial immune system heuristic for two-stage multi-machine assembly scheduling problem to minimize total completion time. *Journal of Manufacturing Systems* **32**, 825–830.

**Alkaya AF and Duman E** (2013) Application of sequence-dependent traveling salesman problem in printed circuit board assembly. *IEEE Transactions on Components, Packaging and Manufacturing Technology* **3**, 1063–1076.

**Ashayeri J, Ma N and Sotirov R** (2011) An aggregated optimization model for multi-head SMD placements. *Computers & Industrial Engineering* **60**, 99–105.

**Ball MO and Magazine MJ** (1998) Sequencing of insertions in printed circuit board assembly. *Operations Research* **36**, 192–201.

**Chang PC, Hsieh JC and Wang CY** (2007) Adaptive multi-objective genetic algorithms for scheduling of drilling operation in printed circuit board industry. *Applied Soft Computing* **7**, 800–806.

**Chen PH and Chang HC** (1995) Large-scale economic dispatch by genetic algorithm. *IEEE Transactions on Power Systems* **10**, 1919–1926.

**Chen S and Shen Y** (2010) An integrated mathematical model for optimizing the component placement process with a multi-heads placement machine *Proc. of the 29th Chinese Control Conf.* (CCC), 1839-1842, Beijing, China, July 29–31.

**Chen CL and Wichman CA** (1993) A systematic approach for design and planning of mechanical assemblies. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* **7**, 19–36.

**Chen T, Luo J and Hu Y** (2011) Component placement process optimization for multi-head surface mounting machine based on tabu search and improved shuffled frog-leaping algorithm. 3rd Int. Workshop on Intelligent Systems and Applications *(ISA),* 1–4, Wuhan, China, May 28–29.

**Chen T, Luo J, Du J and Hu Y** (2012*a*) A modified tabu search algorithm for component placement process optimization of multi-head surface mounting machine. *Proc. of the* 31*st Chinese Control Conf.*, Hefei, China, July 25–27.

**Chen D, Wang J, Zou F, Hou W and Zhao C** (2012*b*) An improved group search optimizer with operation of quantum-behaved swarm and its application. *Applied Soft Computing* **12**, 712–725.

**Csaba BR and Nevalainen OS** (2015) The modular tool switching problem. *European Journal of Operational Research* **242**, 100–106.

**Du X and Li Z** (2008) Placement process optimization of dual-gantry turret placement machine. *Proc. of the* 2008 *IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*1266–1271, Xian, China, July 2–5.

**Fu HP and Su CT** (2000) A comparison of search techniques for minimizing assembly time in printed wiring assembly. *International Journal of Production Economics* **63**, 83–98.

**Grunow M, Günther HO, Schleusener M and Yilmaz IO** (2004) Operations planning for collect-and-place machines in PCB assembly. *Computers & Industrial Engineering* **47**, 409–429.

**He S, Wu H and Saunders JR** (2009) Group search optimizer: an optimization algorithm inspired by animal searching behavior. *IEEE Transactions on Evolutionary Computation* **13**, 1272–1278.

**Ho W and Ji P** (2010) Integrated component scheduling models for chip shooter machines. *International Journal of Production Economics* **123**, 31–41.

**Jensen MT** (2003) Generating robust and flexible job shop schedules using genetic algorithms. *IEEE Transactions on Evolutionary Computation* **7**, 275–288.

**Jiang J, Chen X, Zang M, Wang Z and Tan Z** (2010*a*) Optimization of the surface mount technology based on the max-min ant system. *IEEE Future Computer and Communication (ICFCC)*, **2**, 45–47.

**Jiang J, Du Z, Liu C and Zhang K** (2010*b*). Ant colony algorithms with characteristic of clustering for surface mount technology optimization. *6th Int. Conf. on Wireless Communications Networking and Mobile Computing* (*WiCOM*), 1–4, Chengdu, China, September 23–25.

**Jiang J, Liu C, Zhang K, Wang Z, Tan Z and Chen X** (2010c) Program of route optimization for surface mount. 2*nd International Workshop on Intelligent Systems and Applications* (*ISA*), 1–4, Wuhan, China, May 22–23.

**Kechadi M, Low KS and Goncalves G** (2013) Recurrent neural network approach for cyclic job shop scheduling problem. *Journal of Manufacturing Systems* **32**, 689–699.

**Kim KY, Chin S, Kwon O and Ellis RD** (2009) Ontology-based modeling and integration of morphological characteristics of assembly joints for network-based collaborative assembly design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* **23**, 71–88.

**Knuutila T, Pyöttiälä S and Nevalainen OS** (2007) Minimizing the number of pickups on a multi-head placement machine. *The Journal of the Operational Research Society* **58**, 115–121.

**Kroll E, Lenz E and Wolberg JR** (1989) Rule-based generation of exploded-views and assembly sequences. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* **3**, 143–155.

**Kumar R and Li H** (1995) Integer programming approach to printed circuit board assembly time optimization. *IEEE Transactions on Components, Packaging, and Manufacturing Technology: Part B*, **18**, 720–727.

**Leipälä T and Nevalainen O** (1989) Optimization of the movements of a component placement machine. *European Journal of Operational Research* **38**, 167–177.

**Liang JH and Lee YH** (2015) A modification artificial Bee colony algorithm for optimization problems. *Mathematical Problems in Engineering* **2015**, 581391, 1–14.

**Lin WQ and Zhu GY** (2008) A genetic optimization approach to optimize the multi-head surface mount placement machine. *Lecture Notes in Computer Science* **5315**, 1003–1012.

**Liu B, Wang L, Jin YH, Tang F and Huang DX** (2005) Improved particle swarm optimization combined with chaos. *Chaos, Solitons & Fractals* **25**, 1261–1271.

**Loh TS, Bukkapatnam ST, Medeiros D and Kwon H** (2001) A genetic algorithm for sequential part assignment for PCB assembly. *Computers & Industrial Engineering* **40**, 293–307.

**Luo J, Li X, Liu H and Hu Y** (2010) Mathematical model for the optimization problem in SMT line. *Proc. of the* 29*th Chinese Control Conf.*, 1822–1825, Beijing, China, July 29–31.

**Neammanee P and Reodecha M** (2009) A memetic algorithm-based heuristic for a scheduling problem in printed circuit board assembly. *Computers & Industrial Engineering* **56**, 294–305.

**Nian X, Wang Z and Qian F** (2013) A hybrid algorithm based on differential evolution and group search optimization and Its application on ethylene cracking furnace. *Chinese Journal of Chemical Engineering* **21**, 537–543.

**Park JB, Lee KS, Shin JR and Lee KY** (2005) A particle swarm optimization for economic dispatch with nonsmooth cost functions. *IEEE Transactions on Power Systems* **20**, 34–42.

**Peysakhov M and Regli W** (2003) Using assembly representations to enable evolutionary design of Lego structures. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* **17**, 155–168.

**Smed J, Johnsson M, Puranen M, Leipala T and Nevalainen O** (1999) Job grouping in surface mounted component printing. *Robotics and Computer-Integrated Manufacturing* **15**, 39–49.

**Snyder LV and Daskin MS** (2006) A random-key genetic algorithm for the generalized traveling salesman problem. *European Journal of Operational Research* **174**, 38–53.

**Sun DS, Lee TE and Kim KH** (2005) Component allocation and feeder arrangement for a dual-gantry multi-head surface mounting placement tool. *International Journal of Production Economics* **95**, 245–264.

**Wang JF, Liu JH, Li SQ and Zhong YF** (2003) Intelligent selective disassembly using the ant colony algorithm. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* **17**, 325–333.

**Yan X and Shi H** (2011) A hybrid algorithm based on particle swarm optimization and group search optimization 2011 *Seventh Int. Conf. on Natural Computation* (*ICNC*) **1**, 13–17, Shanghai, China, July 26–28.

**Zeng K, Tan Z, Dong MC and Yang P** (2014) Probability increment based swarm optimization for combinatorial optimization with application to printed circuit board assembly. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* **28**, 429–437.

**Zhang G, Li Z and Du X** (2010) A hybrid genetic algorithm to optimize the printed circuit board assembly process. *2010 Int. Conf. on IEEE Logistics Systems and Intelligent Management* **1**, 563–567, Harbin, China, January 9–10.

**Zhu GY and Zhang WB** (2014) An improved Shuffled Frog-leaping Algorithm to optimize component pick-and-place sequencing optimization problem. *Expert Systems with Applications* **41**, 6818–6829.

**Cheng-Jian Lin** received the Ph.D. degree in electrical and control engineering from the National Chiao-Tung University, Taiwan, R.O.C., in 1996. Currently, he is a Distinguished Professor of Computer Science and Information Engineering Department, National Chin-Yi University of Technology, Taichung County, Taiwan, R.O.C. His current research interests are artificial intelligent, machine learning, intelligent control, evolutionary robotic, image processing, and pattern recognition.

**Mei-Ling Huang** received her M.S. and Ph.D. in Industrial Engineering from University of Wisconsin - Madison and National Chiao-Tung University. Now, she is affiliated with the Department of Industrial Engineering & Management at National Chin-Yi University of Technology. Her research interests include quality management, quality engineering, data mining, and assisted medical diagnosis.

https://doi.org/10.1017/S0890060418000240 Published online by Cambridge University Press