Hyper-Heuristic Optimization Using Multi-Feature Fusion Estimator for PCB Assembly Lines with Linear-Aligned-Heads Surface Mounters

Guangyu Lu, *Graduate Student Member, IEEE*, Huijun Gao, *Fellow, IEEE*, Zhengkai Li, *Member, IEEE*, Xinghu Yu, *Member, IEEE*, Tong Wang, *Senior Member, IEEE*, Jianbin Qiu, *Fellow, IEEE*, and Juan J. Rodríguez-Andina, *Fellow, IEEE*

***Abstract*—Printed circuit board assembly line scheduling (PCBALS) is a difficult task in the electronic industry for assembly lines using surface mounters, which is critical for production efficiency. This is a special type of line optimization problem that uses different allocation techniques, resulting in wide differences in assembly times between machines. This article proposes a hyper-heuristic optimizer embedded with a multi-feature fusion ensemble estimator (HHO-MFFEE) for PCBALS using linear-aligned-heads surface mounters. The objective and constraints of the problem are discussed, and a min-max integer model for small-scale problems is built. At the hyper-heuristic low level, seven data- and target-driven heuristics are presented for allocating components to different machines. Strategies for duplicated conditions with component types and placement points allocation are proposed to improve the applicability of the algorithm and the quality of the solution. An ensemble assembly time estimator that incorporates the coding of multi-features, including estimated sub-objectives, is proposed for evaluating the quality of the solution. Experimental results show that (1) the gaps between the solution from HHO-MFFEE and the optimal solution of the model are 3.44%∼7.28% for small-scale data; (2) the proposed time estimator has higher accuracy than regression and heuristic-based ones, with mean absolute error of 2.01% and 3.43% for training and testing data, respectively; and (3) HHO-MFFEE is better than other state-of-the-art algorithms, with average improvement of 7.21%∼9.47%.**

***Index Terms*—PCBA line optimization, hyper-heuristic, component allocation balance, multi-feature fusion time estimator, linear-aligned-heads surface mounter**

I. Introduction

PRINTED circuit board (PCB) assembly, the process of automatically mounting various electronic components onto bare boards, is an important phase in the manufacturing of electronic products, determining their overall quality. Surface mounters with linear-aligned heads for improving efficiency are widely deployed in PCB assembly lines. Manufacturers tend to use multiple surface mounters in series to increase productivity. However, they face difficulties in both the schedule of a single machine and the optimization of the entire line. The efficiency of single-machine scheduling affects the search process for line optimization, which in turn decides assembly tasks for single machines. Solving these two coupled optimization problems poses a significant challenge.

A PCB assembly line (Fig. 1) consists of automatic equipment, including a loader, screen printer, surface mounters, reflow furnace, automatic optical inspector (AOI), and unloader. Screen printer applies solder paste to the surface of PCBs. Surface mounters pick and place components on the PCB pads. Reflow furnace melts solder paste, which has been already prepositioned on the pads, before cooling it to create a permanent solder. Finally, AOI looks for defects on the PCB to ensure assembly quality. Of all the equipment, the screen printer applies solder paste faster, and the reflow furnace puts PCBs continuously through the oven, which usually does not become a bottleneck as it is not affected by the previous process. Inspection equipment can take pictures of multiple placement points simultaneously, and the computation time for detection is negligible. In contrast, surface mounters, which need to accurately pick and place hundreds of components, have a direct impact on production efficiency. Central to production control is the efficient use of machines, with surface mounters being the bottleneck for assembly efficiency [1–3].

Surface mounters with linear-aligned heads are widely applied in PCB assembly lines. They consist of a stationary platform, two stationary feeder bases, and a moving gantry with multiple heads. The gantry moves between the PCBs and the base and is fitted with heads assembled with suitable nozzles from an automatic nozzle changer for picking and placing different components. The linear-aligned design of the heads is spaced in integer multiples of the slot intervals so that heads can simultaneously pick up components from the feeders on different slots. Compared with the rotary-head type, the mechanical structure of linear-heads is simple and reliable, with higher pickup efficiency, which can achieve high-speed, high-precision assembly. The applicable types of component

This work was supported by the National Key Research and Development Program of China under Grant 2024YFB3409200, the National Natural Science Foundation of China under Grant 62303402, and the New Cornerstone Science Foundation through the XPLORER PRIZE. (*Corresponding author: Huijun Gao.*)

Guangyu Lu, Huijun Gao, Tong Wang, and Jianbin Qiu are with the Research Institute of Intelligent Control and Systems, Harbin Institute of Technology, Harbin 150001, China (e-mail: 20b904007@stu.hit.edu.cn; hjgao@hit.edu.cn; twang@hit.edu.cn, jbqiu@hit.edu.cn).

Zhengkai Li is with the Research Institute of Interdisciplinary Intelligent Science, Ningbo University of Technology, Ningbo 315211, China (e-mail: LZK2024@nbut.edu.cn).

Xinghu Yu is with the Intelligent Control and System Research Center, Yongjiang Laboratory, and also with the Ningbo Institute of Intelligent Equipment Technology Co., Ltd., Ningbo 315201, China (e-mail: 17b304003@stu.hit.edu.cn).

Juan J. Rodríguez-Andina is with the Department of Electronic Technology, University of Vigo, 36310 Vigo, Spain, and also with the Research Institute of Interdisciplinary Intelligent Science, Ningbo University of Technology, Ningbo 315211, China (e-mail: juan.j.r@ieee.org).

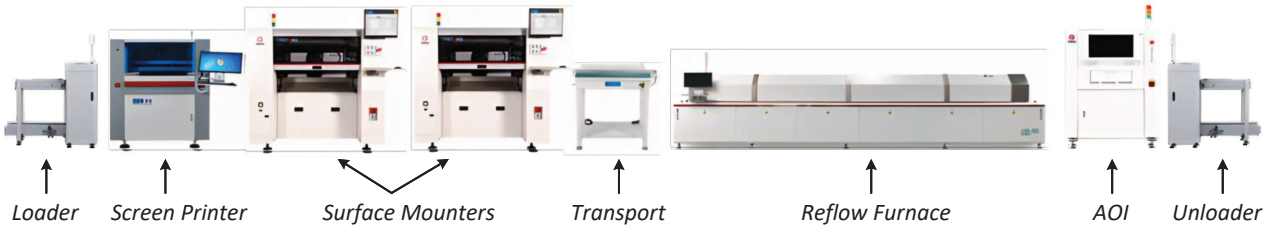

Fig. 1. PCB assembly line.

packages are also more diversified.

PCB assembly line scheduling (PCBALS) focuses on allocating components to multiple surface mounters in a production line to improve assembly efficiency. The search for complex feasible domains, which is an extension of the NP-hard general production line optimization problem, is time-consuming and intricate. The huge solution space requires high-efficiency iterative searching, whereas the long time required for single-machine optimization is inadequate for evaluating each solution. Component allocation for the line and time estimation for a single surface mounter are the main tasks in PCBALS.

Extensive research has been conducted on the PCBALS problem [2–4], and optimization for a single machine has been thoroughly studied [5, 6]. Component allocation has been explored for both model-based [2, 7, 8] and heuristic-based [4, 9–11] algorithms. Most time estimators are fitting-based, which progressively evolved from the number of points to other factors solved by heuristics, such as the number of assembly cycles [2], nozzle changes [8] and feeder utilization [10]. However, most research to date has concentrated on the optimization of lines with rotary-heads surface mounters [2–4, 10], which differs from the structural design with linear-aligned heads.

Heuristic algorithms have been well studied in the field of assembly lines [12], disassembly lines [13] and parallel machines optimization [14]. Hyper-heuristic algorithms are a novel optimization framework that combines the advantages of high- and low-level heuristics to adaptively solve a wide range of complex optimization problems. They have been widely applied for route scheduling [15], truck dispatching [16], or flow shop scheduling [17], among other problems. The estimation of assembly time has been studied with regression fitting approaches [18–20]. Ensemble learning provides strong nonlinear fitting capability, and it can have a high fitting accuracy by designing extracted data features.

In this article, a hyper-heuristic load balancing algorithm with a multi-feature fusion ensemble estimator (HHO-MFFEE) is proposed for PCB assembly lines. Algorithm design is tailored to the structural characteristics of linear-aligned head surface mounters. The hyper-heuristic framework applies techniques with domain knowledge, which results in improved mechanisms for the search and evaluation process, achieving accurate solution evaluation, efficient search process, and balanced allocation results. Compared to state-of-the-art algorithms and industrial solutions, the proposed HHO-MFFEE achieves higher assembly efficiency. The contributions of this article are summarized as follows:

1) A hyper-heuristic optimization method is proposed for linear-aligned-heads surface mounter lines, which can be applied to different scenarios in terms of component-machine constraints, component duplication conditions, or other factors.
2) A set of data- and target-driven low-level heuristics is presented to search the solution space with high-quality results.
3) An extraction method for data features is proposed, and the features are fused within a multi-feature ensemble time estimator, which makes the estimation more accurate.
4) An aggregative grouping algorithm for duplicated component points is proposed to improve the efficiency of PCB assembly lines.

The rest of the article is organized as follows. Section II reviews related work about PCBALS. Section III formulates the mathematical model. The HHO-MFFEE is presented in Section IV. Comparative experiments with other state-of-the-art approaches are presented and discussed in Section V. Finally, Section VI concludes the article.

II. Literature Review

Many studies have contributed to the optimization of PCB assembly lines. This article targets the single-model case [21], where a single PCB type is manufactured without line changeover. This topic has been studied from modeling and heuristic perspectives, with the sub-problems of component allocation and placement sequence. In [22], the former has been proven to be NP-complete, being the main research focus.

Although mathematical modeling can solve problems optimally, it is difficult to obtain mathematical expressions for some real-world applications. Even when this is possible, their implementation may require unacceptably high computational complexity. The integrated model for changeable head configuration and component allocation presented in [2] is linearized and includes a partial relaxation form to speed up the searching process. A min-max approximation integer model with setup and assembly times, as well as an efficient branch-and-bound-based optimal algorithm, are introduced in [7]. As an extension to [7], a mixed integer model with feeder module usage, precedence, and component duplication constraints is proposed in [23]. In [24] and [25], an expected value model and a fuzzy goal model are built to deal with environmental uncertainties, such as demand and machine breakdown, as a tradeoff between optimality and stochasticity.

Meta-heuristics are commonly applied in PCB assembly line optimization. These include genetic algorithms [3, 4] and hybrid spider monkey optimization (HSMO) [26, 27], among others. In [3], a genetic algorithm to identify potential solutions for machine-specific component allocation and placement sequence problems is presented. In [4], a hybrid genetic algorithm is researched, which takes into account a more general scenario of component duplication. The solution is evaluated using a greedy heuristic for assigning nozzles and headsets. An HSMO algorithm is developed in [26] to solve component allocation and placement sequence problems simultaneously. The problem is refined in [27] by incorporating a few extra features to optimize completion time, energy consumption, and maintenance time. A combination of an evolutionary algorithm and mathematical programming to determine the optimal configuration of the type of surface mounters in lines is presented in [28].

In addition, constructive heuristics based on intuition and experience are proposed for PCB line optimization. In [9], line assignment of modular surface mounters is divided into three phases: head to module, component to head, and nozzle to head. Heuristics, including random search, brute force, and evolutionary algorithms, are applied in each phase. In [10], a deterministic hierarchical heuristic is presented to solve the problem at a lower level, allowing component duplication for identical machines. In [29], assembly process decisions are decomposed into four related sub-problems, and list-processing algorithms for lines with dual-head surface mounters are proposed.

Research has also been conducted to optimize the line as part of multi-level production planning, consisting of PCB assignment to the line, component allocation to machines, and surface mounter optimization. An HSMO algorithm to simultaneously solve the multi-level problems is presented in [30]. Hierarchical heuristics are applied in [31] to solve the problem through job partition, selection, grouping, load balancing, and scheduling. In [32], a graph-based divide-and-combine heuristic method is proposed to divide multiple PCBs within a single product, and then sub-problems are solved with standard solvers and meta-heuristics.

Component allocation depends on the assembly time of surface mounters, and most research is based on estimators. In [18], assembly time is estimated from the number of component types and placement points using linear regression. A regularized least-squares regression with a novel feature solved using the nearest neighbor heuristic is proposed in [19]. A supported regression method combined with symbiotic organism search is proposed in [20] to improve estimation accuracy. Neural networks (NNs) have the ability to fit arbitrary nonlinear functions. In [33], a multi-layer perceptron network estimator is presented considering the component shape and the area of the smallest rectangle around the component.

To summarize, the present research focuses more on rotary-head surface mounter line optimization, which inspires us to further optimize a line consisting of surface mounters with special linear-aligned head structures in terms of search capabilities and time estimation accuracy.

III. Problem Formulation and Model

A. Problem Formulation

PCBALS can be regarded as a special type of assembly line optimization, known to be NP-hard. It has a higher decision level and higher complexity compared with single-machine problems. Production optimization of surface mounters can be viewed as a combination of warehouse location, task assignment, and route scheduling problems. There are various combinations of component allocation among different machines. Specifically, each assembly component may be assigned to multi-candidate machines with different processing times, resulting in exponential growth of the number of feasible solutions.

Among the many factors that influence the efficiency of a PCB assembly line, surface mounters take the longest time to process, thus determining the efficiency of the entire line. A variety of interdependent factors influence the assembly efficiency of a single surface mount machine, including the number of cycles, pick-ups, nozzle changes, and placement points [5]. The result of component allocation affects the above multiple sub-objectives. In terms of available resources, assembly tools limit the upper number of assembly machines for each component type, and the priority limits the assembly sequence. Due to resource coupling and conflicting sub-objectives, several local optimal solutions may exist in the feasible domain.

Fig. 2 shows the main tasks and constraints affecting PCB assembly line balancing, namely input, output, constraints, estimator, and optimization tasks. The input is the PCB to be assembled, which includes the component information of the placement point. Constraints can be divided into machine configuration, assembly priority, and available tools. The optimization task consists of two parts: line balancing and assembly process optimization of surface mounter, which have a coupled relationship. The former is generally regarded as the input of the latter, and the latter is used to evaluate the quality of the former solution. In the specific task allocation, the allocation of assembly tools and components for lines determines the head and feeder assignment of each component for surface mounter, which further determines key performance indicators affecting assembly efficiency. Assignment of placement points in load balancing affects the quality of the assembly route scheduling of the surface mounter, which also impacts overall assembly efficiency. The complexity of the assembly tasks makes it difficult to directly get productivity. An estimator evaluates actual assembly time based on the operating process of a single machine and guides line balancing.

Assembly process optimization focuses on performance improvement of individual machines through optimizing feeder configuration, pickup operations, and movement path, among other factors. Meanwhile, assembly line optimization focuses on improving the efficiency of the bottleneck machine. Surface mounter performance directly affects line efficiency, whereas assembly task assignment affects machine utilization rate. The large number of combinations for component allocation makes it difficult to get high-quality solutions, and computing effort increases rapidly as problems scale up, needing massive

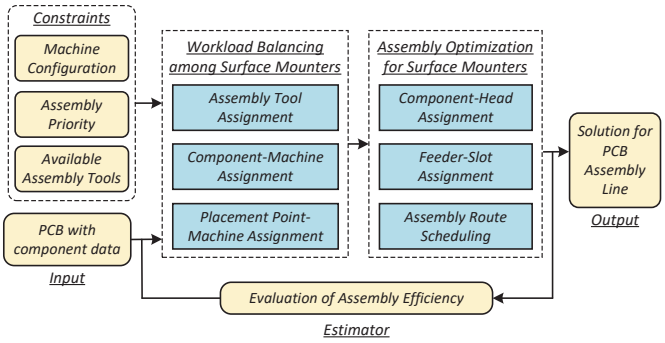

Fig. 2. Main tasks and constraints for PCB assembly line optimization.

resources even for small-scale data. For the unique mechanics of linear-aligned heads, single-machine production simulations with long running time, as well as traditional time estimators with large errors are no longer applicable. In production line optimization, it is necessary to reasonably allocate assembly tasks of each mounter to balance load, which requires accurate and fast estimation of the assembly process of surface mounters.

B. Integer Programming Model

Notations used in the model are listed in Table I. In [6], an integer model for head task assignment, including the major factors that influence assembly efficiency, is proposed. Based on it, a new approximation model is proposed that assesses assembly line efficiency in terms of weighted metrics.

$$\min \max_{m\in M}\left(T_1\cdot\sum_{k\in K}g_{km}+T_2\cdot\sum_{k\in K\setminus\{|K|\}}\sum_{h\in H}n_{khm}+T_3\cdot\sum_{k\in K}w_{km}+T_4\cdot\sum_{s\in S}\sum_{k\in K}e_{skm}+T_5\cdot\sum_{i\in I}\sum_{k\in K}\sum_{h\in H}u_{ikhm}\right) \tag{1}$$

Objective (1) of the model is to minimize the maximum weighted key assembly metrics among all machines, with different weights T_1 for assembly cycle, T_2 for nozzle change, T_3 for pick-up movement, T_4 for pick-up operations, and T_5 for placement operations. As described below, Constraints (2)–(6) are related to the configuration of a single surface mounter, whereas Constraints (7)–(13) incorporate line optimization factors.

$$\sum_{i\in I}u_{ikhm}\le g_{km}\quad\forall k\in K,h\in H,m\in M \tag{2}$$

$$n_{khm}=\sum_{i\in I}\sum_{j\in J}\left|\xi_{ij}\cdot u_{ikhm}-\xi_{ij}\cdot u_{i(k+1)hm}\right| \quad \forall k\in K\setminus\{|K|\},h\in H,m\in M \tag{3}$$

$$e_{skm}\le\sum_{h\in H}v_{[s+(h-1)\cdot\tau]khm}\le N\cdot e_{skm} \quad \forall s\in S,k\in K,m\in M \tag{4}$$

$$w_{km}\ge s\cdot e_{skm}-s'\cdot e_{s'km}+N\cdot(e_{skm}+e_{s'km}-2) \quad \forall k\in K,m\in M,s\in S,s'\in S \tag{5}$$

TABLE I
NOTATIONS OF THE MODEL

Notation	Description
Indices & Sets	
$i\in I$	Index of component type, $I=\{1,2,\cdots\}$
$j\in J$	Index of nozzle type, $J=\{1,2,\cdots\}$
$p\in P$	Index of points, $P=\{1,2,\cdots\}$
$k\in K$	Index of cycle, $K=\{1,2,\cdots\}$
$s\in S$	Index of slot, $S=\{1,2,\cdots\}$
$h\in H$	Index of head, $H=\{1,2,\cdots\}$
$m\in M$	Index of surface mounter machine, $M=\{1,2,\cdots\}$
$q\in Q$	Pair Index of assembly priority , $Q=\{(i,i'),\cdots\}$, $i\in I$, $i'\in I$, which means component type i is assembled before component type i'
Parameters	
ϕ_i	Number of placement points of component type i
θ_i	Number of available feeders of component type i
ζ_j	Number of available nozzles of type j
ξ_{ij}	$=1$ iff. component type i is compatible with nozzle type j (= 0, otherwise)
η_{im}	$=1$ iff. component type i is compatible with machine m (= 0, otherwise)
μ_{ip}	$=1$ iff. component type i is compatible with point p (= 0, otherwise)
τ	Interval ratio between adjacent heads to adjacent slots
$T_1\sim T_5$	Weights for assembly efficiency-related metrics
N	A sufficiently large number
Decision Variables	
g_{km}	Binary variable, $=1$ iff. any point is assembled in cycle k of machine m
u_{ikhm}	Binary variable, $=1$ iff. component type i is assigned to head h in cycle k of machine m
v_{skhm}	Binary variable, $=1$ iff. head h picks up components from slot s in cycle k of machine m
f_{ism}	Binary variable, $=1$ iff. component i is assigned to slot s of machine m
e_{skm}	Binary variable, $=1$ iff. component is picked up when the left-most head aligns to slot s of machine m in cycle k
n_{khm}	Binary variable, $=1$ iff. head h of machine m changes nozzles between cycles k and $k+1$
r_{im}	Binary variable, $=1$ iff. component type i is assembled by machine m
w_{km}	Integer variable, which indicates slots crossed by the gantry during pick-up in cycle k of machine m

$$f_{ism}\le\sum_{k\in K}\sum_{h\in H}u_{ikhm}\cdot v_{skhm}\le N\cdot f_{ism} \quad \forall i\in I,s\in S,m\in M \tag{6}$$

$$\sum_{k\in K}\sum_{h\in H}\sum_{m\in M}x_{ikhm}=\phi_i\quad\forall i\in I \tag{7}$$

$$\sum_{s\in S}\sum_{m\in M}f_{ism}\le\theta_i\quad\forall i\in I \tag{8}$$

$$\sum_{m\in M}\max_{k\in K}\sum_{i\in I}\sum_{h\in H}\xi_{ij}\cdot u_{ikhm}\le\zeta_j\quad\forall j\in j \tag{9}$$

$$r_{im}\le\sum_{k\in K}\sum_{h\in H}x_{ikhm}\le N\cdot r_{im}\quad\forall i\in I,m\in M \tag{10}$$

$$r_{im}\le\eta_{im}\quad\forall i\in I,m\in M \tag{11}$$

$$m-N\cdot(1-r_{im})\le m'+N\cdot(1-r_{i'm'}) \quad \forall q=(i,i')\in Q,m\in M,m'\in M \tag{12}$$

$$\max_{k\in K,h\in H}k\cdot x_{ikhm}+N\cdot(r_{im}+r_{i'm}-2)\le\min_{k\in K,h\in H}\{k\cdot x_{i'khm}+N\cdot(1-x_{i'khm})\}\ \forall q=(i,i')\in Q,m\in M \tag{13}$$

The cycle of each machine with component assignment is defined in Constraint (2). Constraint (3) calculates the number of nozzle changes. Constraint (4) converts pick-up slots to the left-most head-aligned one to get the number of simultaneous pick-ups. Constraint (5) indicates the number of slots through the pick-up movement. Constraint (6) represents the relationship between component and feeder assignment. More details about the relationship between decision variables and tool constraints of a single machine can be found in [6].

Constraint (7) denotes all placement points that are assigned to machines. Constraints (8) and (9) define the maximum number of machines the component can be assigned to, which is limited by the number of feeders and nozzles. Constraint (10) indicates the relationship between machine-assigned and head-assigned components. Constraint (11) restricts the components from being assigned to compatible machines. Constraints (12) and (13) restrict the priority of the assembly process. The former indicates that a component with a high priority cannot be assigned to a machine later than a component with a low priority, whereas the latter restricts the order in which two components are assigned to the same machine.

IV. HHO-MFFEE

A. Solution Framework for the HHO-MFFEE Algorithm

As shown in Fig. 3, the proposed evolutionary-based HHO-MFFEE is built from low-level heuristics and an estimator. Component division and aggregated-based grouping algorithms are designed for component duplication at the beginning and end of the optimization. Multiple populations with varying component allocation sequences iterate separately to avoid allocation sequence limiting efficiency gains while providing multiple high-quality solutions for further evaluation. The combination and execution order of low-level heuristics are specified in the population-generating code. A multi-feature fusion ensemble time estimator based on fully connected NNs is proposed to calculate the fitness value of each individual, fed with data and estimated sub-objectives. In the iterative process, truncated crossover and mutation operations are conducted on individuals. After the evolutionary process is completed, placement points with the same component type are segregated using an aggregated grouping algorithm.

B. Low-Level Heuristics for Component Allocation

Low-level heuristics (LLHs) are basic compositions of hyper-heuristics. They can be divided into data- and target-driven LLHs. The allocation sequence for components is preset and heuristics are selected based on the allocated components.

Data-driven LLHs are connected to the number of points, component type, and nozzle type, as follows: components are allocated to the machine with minimum assigned placement points (*Minimum Points Heuristic*), component types (*Minimum Component Types Heuristic*), nozzle types (*Minimum Nozzle Types Heuristic*), and minimum ratio of number of component types to nozzle types (*Minimum Ratio Heuristic*), respectively.

Target-driven LLHs are related to assembly efficiency, and key sub-objectives are extracted as a basis for component

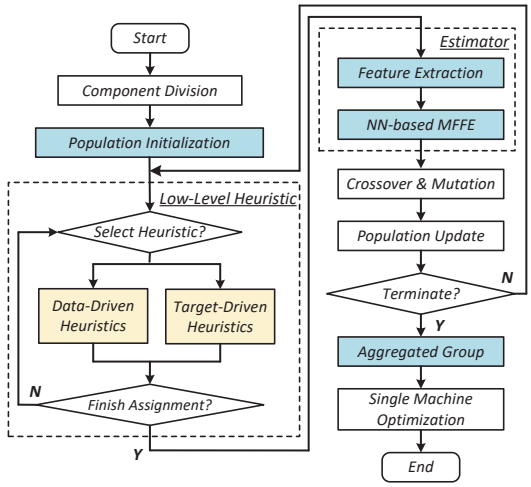

Fig. 3. Flowchart of the proposed HHO-MFFEE algorithm.

allocation. Instead of specific values, they compare relative values of sub-objectives between surface mounters, which can be evaluated without a specialized procedure. The number of heads assigned to nozzle type j of machine m is denoted as γ_{jm}, based on the cascade rounding method proposed in [34]. The target-driven LLHs are:

1) *Minimum Cycle Heuristic*, which allocates components to the machine with the minimum cycle without nozzle change, i.e.,

$$\arg\min_{m\in M}\max_{j\in J}\left(\sum_{i\in I}\sum_{k\in K}\sum_{h\in H}\left(\xi_{ij}\cdot u_{ikhm}\right)/\gamma_{jm}\right) \tag{14}$$

2) *Minimum Nozzle Change Heuristic*, which allocates components to the machine with the minimum probability of nozzle change, reflected in the mean squared error of the points for each head, i.e.

$$\arg\min_{m\in M}\sigma\left(\left\{\overbrace{\sum_{i\in I}\sum_{k\in K}\sum_{h\in H}\frac{\xi_{ij}\cdot u_{ikhm}}{\gamma_{jm}}}^{\gamma_{jm}\text{ terms in total}}\mid j\in J\right\}\right) \tag{15}$$

where $\sigma(\cdot)$ denotes the mean square deviation of a set.

3) *Minimum Pick-up Heuristic*, which allocates components to the machine with minimum pick-up operations.

Algorithm 1 presents a method to estimate the number of pick-ups. A hierarchical greedy heuristic assigns components to heads in decreasing order of the number of points, subject to the number of heads accessible to the nozzle. Assigned points of each component type for machine m are denoted as ϕ'. Assignment of all attachable heads to each component implies the start of a new cycle, and the number of pick-ups equals the maximum number of points assigned to heads in each cycle.

The number of component feeders and machine requirements restricts allocatable machines. All LLHs are based on a set of feasible surface mounters. The set of assigned surface mounters for each component type cannot exceed available

feeders. The feasible set is adjusted based on component-assigned mounters. When the number of assigned mounters equals that of available feeders, the indices of assigned mounters are regarded as the new feasible set. Component prioritization needs to be checked first to see if the loop is closed between constraint relationships and, if so, there is no solution. Otherwise, if during component allocation, a newly allocated component breaks the priority constraint, the assigned components that do not satisfy the constraint relationship are replaced and reallocated with the same strategy. The machine with the fewest points among LLHs with the same evaluation value has the highest priority for assembling components.

Algorithm 1: Hierarchical Greedy Head Assignment

Input : Nozzle heads γ, component points ϕ'
Output: Number of pick-up operations $\mathcal{O}$

Set a $1 \times |J|$ vector $\mathcal{L}$, a $1 \times |J|$ vector $\mathcal{N}$, and a $1 \times \sum_{i \in I} \phi'_i$ vector $\mathcal{K}$ of all zeros;
Sort $i \in I$ decreasingly with ϕ'_i;
for $i \in I$ **do**
  $j \leftarrow \sum_{j' \in J} \xi_{ij'} \cdot j'$; // assign nozzle j compatible with component i
  if $\mathcal{N}_j$ Mod $\gamma_{jm} = 0$ **then**
    $\mathcal{L}_j \leftarrow \mathcal{L}_j + 1$; // nozzle allocation is full and start a new cycle
  end
  /* Update maximum number of allocated points and heads */
  Set cycle index $c \leftarrow \mathcal{L}_j$, $\mathcal{K}_c \leftarrow \max\left(\mathcal{K}_c, \phi'_i\right)$, $\mathcal{N}_j \leftarrow \mathcal{N}_j + 1$
end
$\mathcal{O} \leftarrow \sum_{c=1}^{c=\sum_{i \in I} \phi'_i} \mathcal{K}_c$

C. Hyper Heuristic for Line Optimization

In the evolutionary-based hyper-heuristic, each individual gene correlates to an LLH denoted as a pattern. It operates in a range of populations with various component allocation sequences and individual genes of different lengths, increasing search diversity. The length of genes is limited to the number of component division groups. All individuals are initialized with random lengths and pattern combinations. Cyclic access is applied in individuals with short genes. Each one of the two genes selects a split point and the crossover operator exchanges gene segments. The mutation operator inserts randomly generated patterns at the split point. Truncated procedures are applied to individuals whose length exceeds the limit value. For each solution, the specific algorithm is executed on the machine with the longest estimated time, reducing single-machine optimizations and increasing solving efficiency.

D. Multi-Feature Fusion Ensemble Time Estimator

Ensemble learning with NNs performs well at fitting complex and nonlinear data. Multi-feature of fitting data is related to single-machine optimization. Simulated data are fed to the network to ensure it is sufficiently trained. The complexity of the PCB assembly process makes some properties difficult to uncover. Therefore, a heuristic algorithm is proposed to estimate performance metrics to improve fitting accuracy.

The framework of the estimator is shown in Fig. 4. Input coding consists of three parts. Basic data are the total number of placement points, component types, nozzle types, and board size. The estimated number of cycles and pick-ups of the preceding section, as well as nozzle change, builds the target feature coding. Nozzle and component types are coded in descending order based on the total number of points as extended parts. Grouped components are the object of LLHs allocation, which involves only the type of components and the number of placement points. The independence of point distribution is due to its coding difficulty and relatively small impact on efficiency. A sufficiently long encoding is used to ensure consistency across diverse data inputs to networks, with redundant bits supplemented by zeros.

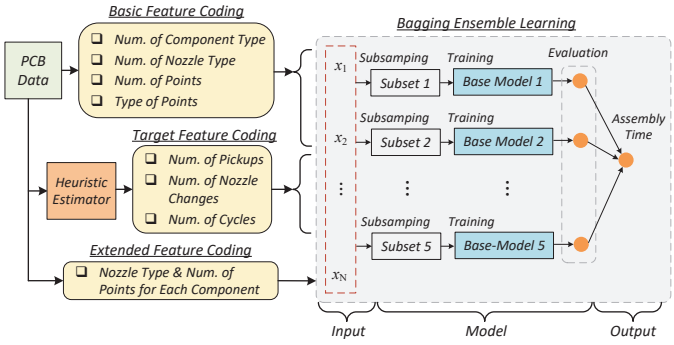

Fig. 4. Framework of the multi-feature fusion ensemble time estimator.

The estimation of nozzle change probability cannot be directly coded. Algorithm 2 proposes a computation heuristic for it. Components with the same nozzle type are grouped according to their respective nozzle heads. The group of nozzle j of machine m is denoted as $\mathcal{G}_{jm}$. Nozzle groups are progressively assigned to heads, starting with empty heads and proceeding sequentially to the heads with the fewest points. When the allocation process is complete, the heads with the most and least points are divided equally, which is accepted if the efficiency gain from reducing the number of cycles after equalization outweighs the efficiency loss from increasing nozzle changes. This process is repeated to increase the number of heads of the nozzle with the most head-averaged points, and the total number of nozzle changes is recorded.

E. Heuristics for Component Duplication

Components with multi-available feeders can be assigned to more than one surface mounter, which is called a duplicated condition. To deal with this case, in this section the algorithm is improved in two ways. Firstly, components are grouped to meet the needs of distributing multiple machines. Equation (16) gives threshold $\hat{\theta}_i$, above which components with more points are to be split.

$$\hat{\theta}_i = \max\left(\varepsilon \cdot \sum_{i' \in I} \theta_{i'} \cdot \phi_i / \sum_{i' \in I} \phi_{i'}, \theta_i\right) \quad \forall i \in I \qquad (16)$$

where ε regulates the number of groups. This grouping strategy balances search efficiency and diversity. Besides, the set of

Algorithm 2: Nozzle Change Computation Heuristic

Input : Nozzle heads γ, component points ϕ
Output: Number of nozzle changes N^*

Set $1 \times |H|$ vector $\mathcal{T}$ of all zeros, $1 \times |H|$ vector $\mathcal{N}$, $V \leftarrow 0$, $V^* \leftarrow \infty$ and $N^* \leftarrow 0$;
while $V \leq V^*$ **do**
  Set $1 \times \gamma_{jm}$ nozzle group $\mathcal{G}_{jm}$ with $\sum_{i \in I} \phi_i \cdot \xi_{ij} / \gamma_{jm}$ points for $j \in J$;
  for $n \in \mathcal{G}_{jm}, j \in J$ **do**
    /* assign nozzle groups to heads */
    $h \leftarrow \arg\min_{h' \in H} \{\mathcal{T}_h\}$, $\mathcal{N}_h \leftarrow j$, $\mathcal{T}_h \leftarrow \mathcal{T}_h + n$
  end
  Set number of cycles $V \leftarrow \max_{h \in H} \mathcal{T}_h$;
  while true do
    /* balance the points of heads */
    $h' \leftarrow \arg\max_{h \in H} \mathcal{T}_h$, $h'' \leftarrow \arg\min_{h \in H} \mathcal{T}_h$;
    if $\mathcal{N}_{h'} = \mathcal{N}_{h''}$ **then**
      break;
    end
    /* compare the weighted metrics */
    $j' \leftarrow \mathcal{N}_{h'}$, $\mathcal{H}_1 \leftarrow \{h \mid \mathcal{N}_h = j', h \in H\}$, $j'' \leftarrow \mathcal{N}_{h''}$, $\mathcal{H}_2 \leftarrow \{h \mid \mathcal{N}_h = j'', h \in H\}$; **if** $T_3 \cdot (\mathcal{T}_{h'} - \mathcal{T}_{h''}) > T_2 \cdot ||\mathcal{H}_2| - |\mathcal{H}_1||$ **then**
      break;
    end
    /* update assignment result */
    $N \leftarrow ||\mathcal{H}_2| - |\mathcal{H}_1||$, $V \leftarrow V - T_3 \cdot (\mathcal{T}_{h'} - \mathcal{T}_{h''}) + T_2 \cdot N$, $\mathcal{T}' \leftarrow \mathcal{T}$;
    for $h \in \mathcal{H}_1 \cup \mathcal{H}_2$ **do**
      $\mathcal{T}_h \leftarrow \sum_{h' \in \mathcal{H}_1 \cup \mathcal{H}_2} \mathcal{T}'_{h'} / (|\mathcal{H}_1| + |\mathcal{H}_2|)$, $\mathcal{N}_h \leftarrow j'$;
    end
  end
  if $V < V^*$ **then**
    $V^* \leftarrow V$, $N^* \leftarrow N$, $\gamma_{j'm} \leftarrow \gamma_{j'm} + 1$; // add nozzle groups and re-allocate
  end
end

Algorithm 3: Duplicated Component Points Allocation Algorithm

Input : Number of feeders θ, component-point compatibility μ, points position (x, y), machine-component assignment u and r
Output: Machine-allocated points $\overline{\mathcal{P}}$

Set machine-assigned sets $\mathcal{P}_m \leftarrow \varnothing$, number of machine-head assigned points $\mathcal{V}_{ihm} \leftarrow 0$ and head-derivation $\overline{h}_{im} \leftarrow 0$, $\forall i \in I, h \in H, m \in M$;
Set ρ_h as the interval distance between head h and left-most head ;
for $m \in M$ **do**
  for $i \in \{i' \mid r_{i'm} > 0, \theta_{i'} = 1, i' \in I\}$ **do**
    $\mathcal{P}_m \leftarrow \mathcal{P}_m \cup \{p \mid \mu_{ip} = 1, p \in P\}$;
  end
  for $i \leftarrow \sum_{i' \in I} i' \cdot u_{i'khm}$, $k \in K$, $h \in H$ **do**
    $\mathcal{V}_{ihm} \leftarrow \mathcal{V}_{ihm} + 1$, $\overline{h}_{im} \leftarrow \left(1 - 1/\sum_{h \in H} \mathcal{V}_{ihm}\right) \cdot \overline{h}_{im} + (h-1) / \sum_{h \in H} \mathcal{V}_{ihm}$
  end
  Set center points of each machine $\mathcal{X}_m \leftarrow \sum_{p \in \mathcal{P}_m}$
  $\left(x_p - \sum_{i \in I} \mu_{ip} \cdot \overline{h}_{im}\right) / |\mathcal{P}_m|$, $\mathcal{Y}_m \leftarrow \sum_{p \in \mathcal{P}_m} y_p / |\mathcal{P}_m|$;
end
while true do
  $\overline{\mathcal{X}} \leftarrow \mathcal{X}$, $\overline{\mathcal{Y}} \leftarrow \mathcal{Y}$, $\overline{\mathcal{V}} \leftarrow \mathcal{V}$, $\overline{\mathcal{P}} \leftarrow \mathcal{P}$;
  for $p \in \{p' \mid p' \in P_i, \theta_i > 1\}, i \in I$ **do**
    Set $\left(\hat{m}, \hat{h}\right) \leftarrow \arg\min_{m \in M, h \in H} \left\{ \left(\overline{\mathcal{X}}_m - x_p + \rho_h\right)^2 + \left(\overline{\mathcal{Y}}_m - y_p\right)^2 \mid \overline{\mathcal{V}}_{ihm} > 0 \right\}$ as the allocated machine,
    $\overline{\mathcal{P}}_{\hat{m}} \leftarrow \overline{\mathcal{P}}_{\hat{m}} \cup \{p\}$, $\overline{\mathcal{V}}_{i\hat{h}\hat{m}} \leftarrow \overline{\mathcal{V}}_{i\hat{h}\hat{m}} - 1$;
    /* update number of assigned points and center of surface mounters */
    $\overline{\mathcal{X}}_{\hat{m}} \leftarrow \overline{\mathcal{X}}_{\hat{m}} + \left(x_p - \overline{\mathcal{X}}_{\hat{m}} - \rho_{\hat{h}}\right) / \left|\overline{\mathcal{P}}_{\hat{m}}\right|$,
    $\overline{\mathcal{Y}}_{\hat{m}} \leftarrow \overline{\mathcal{Y}}_{\hat{m}} + \left(y_p - \overline{\mathcal{Y}}_{\hat{m}}\right) / \left|\overline{\mathcal{P}}_{\hat{m}}\right|$;
  end
  if $\sum_{m \in M} \left(\left|\overline{\mathcal{X}}_m - \mathcal{X}_m\right| + \left|\overline{\mathcal{Y}}_m - \mathcal{Y}_m\right|\right) < 10^{-3}$ **then**
    break;
  end
end

feasible surface mounters is updated synchronously in the allocation process.

Secondly, the distribution of points has an impact on assembly efficiency, and an aggregative grouping is proposed in Algorithm 3. Component allocation determines the upper number of placement points of each type assigned to each surface mounter. The initial aggregated center of each machine is determined by the components with a single feeder. Current research [5, 6] divides surface mounter optimization into head task assignment and pick-and-place sequencing, where the former determines the head-deviation $\overline{h}$ for the alignment of the heads. The adjustment of the group center helps to shorten the moving path of the linear head and improve assembly efficiency.

V. Comparative Experiments

A. Experimental Setup

Experiments have been carried out using a PC with an Intel(R) Core(TM) i5-14600KF. Table II shows the parameter settings of the hyper-heuristic and estimator. Iterations are carried out across populations with ten randomly generated component allocation sequences. The multiplier of component grouping is set to 1.5. The time estimator is a two-middle-layer fully connected NN with 1,000 neurons per layer and ReLu is used as the activation function. Results are compared for PCB assembly lines *L*1, *L*2, and *L*3, equipped with 2, 3, and 4 surface mounters, respectively. Fifteen PCB data from actual manufacturing lines are used to evaluate the assembly efficiency of the algorithm, with the first five being on a smaller scale, as shown in Table III. As meta-heuristic results are random, the average of the five runs is taken as the result.

TABLE II
Hyper-Heuristic and Estimator Parameters

Method	Parameters	Value
Hyper Heuristic	Size of Population	10
	Threshold Parameter	1.5
	Num. of Individuals in Population	20
	Crossover & Mutation Rate	0.6 & 0.1
	Number of Iterations	50
Estimator	Learning Rate	10^{-5}
	Number of Epochs	8000

Training and testing data for time estimation fitting are randomly generated, and assembly times are obtained from the built-in simulator of the surface mounter, which is accurate for performing optimization and full assembly process simulation. The distribution of placement points impacts assembly efficiency. Training data with either sparse or concentrated distribution can reduce the generalization performance of the estimator, which can be refined by fitting randomly generated points with relatively uniform distribution. Table IV shows

TABLE III
STATISTICAL PCB DATA

PCB	1-1	1-2	1-3	1-4	1-5	2-1	2-2	2-3	2-4	2-5	2-6	2-7	2-8	2-9	2-10
Num. of Comp. Type	4	4	5	5	5	16	29	7	24	45	7	47	40	10	40
Num. of Nozzle Type	3	3	3	2	2	3	3	3	3	4	4	4	2	3	4
Number of Points	28	34	34	30	30	78	165	192	236	209	320	390	546	720	1510
Number of Feeders	10	6	8	7	5	19	30	12	28	47	14	54	50	19	40

TABLE IV
PARAMETERS OF TRAINING AND TESTING DATA

Training Sets	# of Samples	Outlier (%)	Mean	Median
	2000	11.25	128.67	130.13
	Minimum	Maximum	Std. Dev	
	2.71	302.94	71.67	
Testing Sets	# of Samples	Outlier (%)	Mean	Median
	400	10.75	126.76	127.11
	Minimum	Maximum	Std. Dev	
	3.80	311.38	72.23	

TABLE V
WEIGHTED KEY METRICS INDICATORS OF THE MATHEMATICAL MODEL AND THE PROPOSED HHO-MFFEE ALGORITHM

Line	*L*1			*L*2			*L*3		
	T_M	T_H	$\delta T(\%)$	T_M	T_H	$\delta T(\%)$	T_M	T_H	$\delta T(\%)$
1-1	2.585	2.626	1.59	1.758	1.837	4.49	1.676	1.813	8.17
1-2	3.286	3.672	11.75	2.785	3.122	12.10	2.473	2.514	1.66
1-3	2.719	2.998	10.26	2.218	2.445	10.23	1.947	2.054	5.50
1-4	2.744	3.017	9.95	2.202	2.314	5.09	2.202	2.243	1.86
1-5	2.933	3.017	2.86	2.432	2.456	0.99	2.432	2.432	0.00
Avg			7.28			6.58			3.44

TABLE VI
ESTIMATED ACCURACY OF THE TESTED ALGORITHMS

Set	Parameters	E_1	E_2	E_3	E_4	E_5
Training	Mean Absolute Error (%)	2.01	5.09	8.75	8.75	45.30
	Max. Absolute Error (%)	18.80	21.28	37.61	37.68	214.94
Testing	Mean Absolute Error (%)	3.43	5.16	9.41	9.44	45.99
	Max. Absolute Error (%)	16.57	18.65	27.65	28.82	183.98

statistical PCB data. Data outliers are detected and removed using the inter-quartile range rule [20] with a multiplier of 0.6. Training and testing data have similar distribution characteristics.

B. Comparison of HHO-MFFEE and Mathematical Model

Mathematical programming is used to find optimal solutions, but only for small-scale data. In this section, the solutions obtained by HHO-MFFEE are compared with the approximated optimal solutions of the model, which is built by extracting key metrics that affect assembly efficiency. The model is validated using the Gurobi solver [35]. To make the model linear and solvable, it is assumed that enough nozzles are available. In addition, placement priority constraints are ignored. The weights of the model are set using a linear fit to the training data, with $T_1 = 0.041$, $T_2 = 0.326$, $T_3 = 0.870$, $T_4 = 0.159$ and $T_5 = 0.015$. The effect of the layout of points on assembly efficiency is ignored. Table V presents the comparison of the first five data. T_M and T_H represent the weighted performance metrics of the model and the proposed algorithm, respectively. Gap $\delta T = (T_H/T_M - 1) \cdot 100\%$ with respect to the optimal solution of the model is 7.28%, 6.58%, and 3.44% on average in 3 assembly lines. Comparison with the model reveals that the proposed algorithm is close to the optimal solution, with a maximum gap of 12.10%. The performance of the hyper-heuristic algorithm is comparable to that of the model solution, and its higher solving efficiency makes it possible to apply it to larger-scale data.

C. Evaluation of the Proposed Time Estimator

The accuracy of the time estimator impacts the search direction for component allocation, as well as the quality of solutions. In this subsection, four different time estimators are compared with the proposed one, which yields E_1. E_2 refers to the ensemble fitting method using basic parameters, without the target related terms, which is another way of encoding. Results for the heuristic estimators proposed in [4] and [9] are denoted as E_3 and E_4, respectively, with coefficients computed using the least squares method. E_5 refers to an ensemble algorithm with symbiotic organism search-based support vector regression (SOS-based SVR) [20].

Mean and maximum absolute errors of training and testing data are listed in Table VI. The performance of the fitting method on the testing set is the basis for evaluating the accuracy of the estimators. The two NN-related methods provide better time estimation. The proposed estimator encoding method reduces the average absolute error on the testing set from 5.16% to 3.43%, compared to simply feeding basic parameters. The number of pick-ups is not incorporated in the two heuristic-based linear regression fittings, resulting in poorly fitted results with mean absolute errors of 9.41% and 9.44%, respectively. Despite being effective in the workshop production line of the PCB assembly process, the SOS-based SVR has the lowest fitting accuracy, as it ignores the distinctive properties of each single PCB.

D. Comparison of Low-level Heuristics

Ten PCBs are used to compare the performances of individual LLHs in *L*2. Table VII shows optimization results of load allocation. A_p, A_n, A_c and A_r are the data-driven LLHs, i.e., Minimum Points, Minimum Nozzle Types, Minimum Component Types, and Minimum Ratio, respectively. A_k, A_g and A_u

TABLE VII
SOLUTIONS OBTAINED BY VARIOUS LLHS

PCB	A_p	A_n	A_c	A_r	A_k	A_g	A_u
2-1	10.06	10.26	9.52	9.88	9.72	9.79	10.70
2-2	15.38	18.28	16.12	19.97	14.75	17.66	16.01
2-3	20.46	20.48	20.00	23.06	20.10	22.81	19.18
2-4	18.98	25.97	19.87	26.24	21.17	27.06	22.84
2-5	22.36	28.25	26.14	29.64	21.13	32.34	22.94
2-6	28.81	35.23	28.79	38.40	28.78	33.84	27.69
2-7	46.21	45.29	40.47	51.13	42.57	45.42	43.27
2-8	52.07	59.06	50.44	59.39	51.73	63.46	49.05
2-9	66.93	67.80	65.82	84.64	65.98	68.50	65.42
2-10	135.68	143.67	143.01	168.83	139.76	146.01	149.82

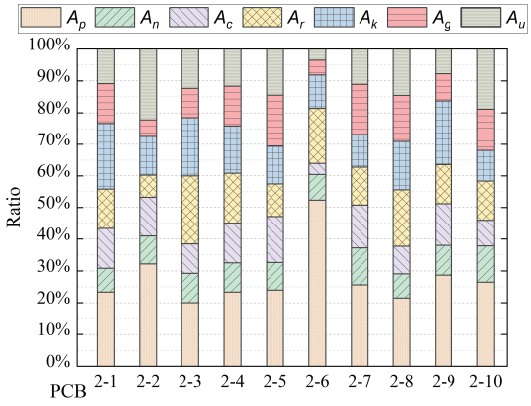

Fig. 5. Ratio of each LLH when using the hyper-heuristic.

are the target-driven LLHs, i.e., Minimum Cycle, Minimum Nozzle Change, and Minimum Pick-up, respectively. Both A_p and A_k achieve higher assembly efficiency by more balanced cycles and placement points. Results of LLHs that indirectly affect efficiency or single-objective related have low efficiency. All single-LLHs fail to achieve the hyper-heuristic effect.

The ratio of each LLH when using the hyper-heuristic is shown in Fig. 5. Balancing the number of placement points among surface mounters is the main task of optimization. The ratio of target-driven operators is higher than that of data-driven ones in the remaining LLHs. Nozzle change-related term A_g occurs less frequently in the assembly process. Thus, for most data, the pickup-related A_u is more relevant in the search process.

E. Comparison with Other Algorithms

The main task of the line optimizer is to allocate components to machines. In this subsection, the proposed algorithm is compared with three state-of-the-art solutions, namely an industrial solver released in 2022 by an advanced manufacturer, the integrated algorithm [4], and the hybrid algorithm [9]. The industrial solver is an optimizer embedded in an integrated production line management tool for surface-mount assembly lines. The integrated algorithm is a genetic-based method that provides solutions for PCB assembly lines by designing operators to search the feasible domain. The hybrid algorithm combines random search, local search, and evolutionary algorithms, among others. Since the spider monkey algorithm has been widely used in PCB assembly line optimization [26, 27, 36], this section further integrates it into the hybrid framework and improves it based on the coding and searching approaches proposed in [27, 36]. The industrial solver provides complete solutions from assembly line balancing to surface mounter optimization, and the rest of the single-machine optimizations are based on the methods proposed in [6].

Table VIII shows the optimization results of the four tested algorithms. The proposed hyper-heuristic algorithm outperforms the industrial solver, and the hybrid and integrated algorithms by 7.21%, 8.67%, and 9.47%, respectively. In addition, the distribution of the optimization results in three assembly lines are shown in Fig. 6. In algorithms with randomized results, the hyper-heuristic produces a more consistent result. In most cases, the results of a single run of the hyper-heuristic outperform those of the other methods. Even if it produces some weaker solutions, the vast majority of them outperform the best solutions from the other methods.

F. Analysis of Solving Efficiency

Solving efficiency is one of the most important performance indicators for large-scale combinatorial optimization problems. Solving times for PCBALS using three of the tested algorithms are shown in Table IX. The industrial solver is not included in the comparison, because it is built into a runtime software package, which includes importing data, optimizing, and outputting results, so its solving times cannot be separated from the rest for fair comparison. The hybrid algorithm consists of relatively basic operators, which allow it to search quickly at the cost of solution quality. The hyper-heuristic and integrated algorithms use a more complex time-fitting approach and account for component duplication, resulting in longer times than that of the hybrid algorithm. The proposed HHO-MFFEE is more efficient than the integrated algorithm, and the quality of the solution it provides is higher. Evaluating the quality of the candidate solutions takes a large part of the solving time of the hyper-heuristic. PCB2-5 and PCB2-10 are more complex. Single-machine optimization takes longer for PCBs with a larger number of components and nozzle types, resulting in relatively poor solving efficiency. By shortening the execution time of surface mounter optimization, efficiency may be further increased.

VI. CONCLUSION

This article presents HHO-MFFEE, a hyper-heuristic optimization method for PCBALS with a multi-feature fusion ensemble time estimator. The hyper-heuristic algorithm is implemented using data- and target-driven LLHs. A min-max mathematical model is built covering the major assembly efficiency metrics. In terms of solution quality, the proposed method has comparable performance to the optimal one obtained by the model when dealing with small-scale data. The strategies for component duplication divide components of the same type, balancing assembly time between machines and improving assembly efficiency. An aggregated grouping algorithm assigns placement points to the specific surface

TABLE VIII
ASSEMBLY TIMES OF THE TESTED ALGORITHMS

	Hyper Heuristic			Industrial Solver				Hybrid Algorithm				Integrated Algorithm			
PCB	*L*1	*L*2	*L*3	*L*1	*L*2	*L*3	δ (%)	*L*1	*L*2	*L*3	δ (%)	*L*1	*L*2	*L*3	δ (%)
2-1	10.14	8.06	6.21	12.91	8.41	6.56	12.46	10.91	9.42	7.17	13.28	14.97	8.41	6.83	20.66
2-2	19.55	14.28	11.61	20.78	14.75	12.95	7.01	19.89	14.93	12.42	4.42	20.61	14.75	12.35	5.01
2-3	21.15	18.06	12.59	21.19	18.77	14.44	6.26	23.39	18.26	14.92	10.06	23.18	18.77	14.72	10.14
2-4	26.10	17.85	13.87	26.29	18.66	13.96	1.95	28.14	19.17	14.38	6.30	29.37	18.66	14.86	8.06
2-5	27.86	19.33	15.35	32.32	19.59	15.79	6.75	33.48	21.29	16.89	13.46	32.57	19.59	16.66	8.95
2-6	38.63	26.53	22.83	44.42	27.91	23.02	7.00	39.11	27.34	22.44	0.86	45.05	27.91	24.22	9.30
2-7	50.15	34.12	26.23	53.91	36.93	26.85	6.04	58.73	40.79	29.72	16.66	57.76	36.93	31.32	14.28
2-8	71.32	48.08	39.42	73.96	51.16	40.18	4.01	72.02	51.91	40.28	3.70	75.09	51.16	42.76	6.72
2-9	85.69	60.91	46.07	91.18	63.91	52.57	8.48	94.66	66.08	53.05	11.36	91.95	63.91	47.10	4.82
2-10	176.99	117.93	90.78	179.94	125.79	116.23	12.12	185.07	128.07	96.73	6.57	188.01	125.79	97.54	6.78
Avg							7.21				8.67				9.47

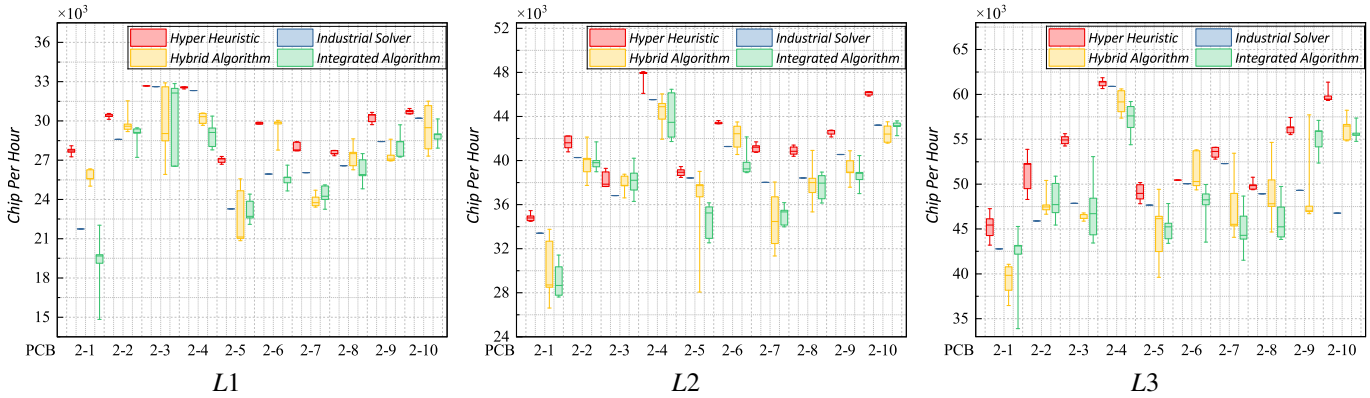

Fig. 6. Distribution of optimization results of the tested algorithms on three PCB assembly lines.

TABLE IX
SOLVING TIMES OF THE TESTED ALGORITHMS

	Hyper Heuristic			Hybrid Algorithm			Integrated Algorithm		
PCB	*L*1	*L*2	*L*3	*L*1	*L*2	*L*3	*L*1	*L*2	*L*3
2-1	17.28	20.95	24.26	15.84	18.97	21.69	54.13	59.35	62.99
2-2	33.98	31.35	30.71	63.45	63.70	68.54	64.05	68.51	75.21
2-3	13.98	15.62	19.64	26.56	32.01	37.27	50.74	54.99	63.95
2-4	21.51	23.73	26.20	9.31	11.08	12.26	64.05	68.17	76.23
2-5	100.22	81.51	87.45	23.49	28.06	32.57	85.59	90.02	96.65
2-6	21.32	18.32	21.74	49.13	56.61	65.24	63.57	67.34	73.92
2-7	93.22	70.93	68.72	12.80	14.17	16.01	100.31	96.79	104.06
2-8	40.19	42.99	38.08	55.18	59.92	65.67	91.20	95.64	104.69
2-9	29.20	27.52	30.12	40.48	48.99	55.56	89.30	92.85	101.16
2-10	135.98	76.67	76.71	25.48	24.94	24.90	144.55	155.60	171.15

mounters. The proposed time estimators have high fitting accuracy, and coding with approximated sub-objectives further enhances fitting accuracy. The combination of the high accuracy of the estimator with the hyper-heuristic search capability for large domains results in high-quality solutions for PCBALS problems. Compared with industrial solutions and other state-of-the-art algorithms, the proposed one achieves higher assembly efficiency and stable results with acceptable solving times.

Future research will focus on load balancing optimization of flexible PCB assembly lines. For high-mix, low-volume PCB production tasks, its efficiency is affected by the configuration adjustments of surface mounters. This involves the optimization of the scheduling of dynamic production tasks, and enhancing the efficiency of the feeder module changeover, among others, which is beneficial to shorten the productive cycle and reduce storage cost, so that small- and medium-batches can achieve profitability comparable to that of mass manufacturing, and improve the efficiency, robustness, and stability of the assembly line.

REFERENCES

[1] M. Ayob and G. Kendall, "The optimisation of the single surface mount device placement machine in printed circuit board assembly: a survey," *Int. J. Syst. Sci.*, vol. 40, no. 6, pp. 553–569, Apr. 2007.

[2] A. Rong, A. Tóth, O. S. Nevalainen, T. Knuutila, and R. Lahdelma, "Modeling the machine configuration and line-balancing problem of a PCB assembly line with modular placement machines," *Int. J. Adv. Manuf. Tech.*, vol. 54, no. 1, pp. 349–360, Apr. 2011.

[3] O. Kulak, I. O. Yilmaz, and H.-O. Günther, "A GA-based solution approach for balancing printed circuit board assembly lines," *OR Spectrum*, vol. 30, no. 3, pp. 469–491, Jun. 2008.

[4] S. Guo, K. Takahashi, K. Morikawa, and Z. Jin, "An integrated allocation method for the PCB assembly line balancing problem with nozzle changes," *Int. J. Adv. Manuf. Tech.*, vol. 62, no. 1, pp. 351–369, Sep. 2012.

[5] H. Gao, Z. Li, X. Yu, and J. Qiu, "Hierarchical multiobjective heuristic for PCB assembly optimization in a beam-head surface mounter," *IEEE Trans. Cybern.*, vol. 52, no. 7, pp. 6911–6924, Jul. 2021.
[6] G. Lu, X. Yu, H. Sun, Z. Li, J. Qiu, and H. Gao, "A scan-based hierarchical heuristic optimization algorithm for PCB assembly process," *IEEE Trans. Industr. Inform.*, vol. 20, no. 3, pp. 3609–3618, 2024.
[7] D. M. Kodek and M. Krisper, "Optimal algorithm for minimizing production cycle time of a printed circuit board assembly line," *Int. J. Prod. Res.*, vol. 42, no. 23, pp. 5031–5048, Dec. 2004.
[8] M. S. Hillier and M. L. Brandeau, "Cost minimization and workload balancing in printed circuit board assembly," *IIE Trans.*, vol. 33, no. 7, pp. 547–557, Jul. 2001.
[9] A. Tóth, T. Knuutila, and O. S. Nevalainen, "Reconfiguring flexible machine modules of a PCB assembly line," *Prod. Eng.*, vol. 4, no. 1, pp. 85–94, Feb. 2010.
[10] T. He, D. Li, and S. W. Yoon, "A heuristic algorithm to balance workloads of high-speed SMT machines in a PCB assembly line," in *FAIM2017*, vol. 11, 2017, pp. 1790–1797.
[11] X. Yan, H. Zuo, C. Hu, W. Gong, and V. S. Sheng, "Load optimization scheduling of chip mounter based on hybrid adaptive optimization algorithm," *Complex Syst. Model. Simul.*, vol. 3, no. 1, pp. 1–11, Dec. 2023.
[12] Z. Zhang, Q. Tang, M. Chica, and Z. Li, "Reinforcement learning-based multiobjective evolutionary algorithm for mixed-model multimanned assembly line balancing under uncertain demand," *IEEE Trans. Cybern.*, vol. 54, no. 5, pp. 2914–2927, 2024.
[13] K. Wang, X. Li, L. Gao, P. Li, and J. W. Sutherland, "A discrete artificial bee colony algorithm for multiobjective disassembly line balancing of end-of-life products," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 7415–7426, 2022.
[14] Z. Pan, D. Lei, and L. Wang, "A knowledge-based two-population optimization algorithm for distributed energy-efficient parallel machines scheduling," *IEEE Trans. Cybern.*, vol. 52, no. 6, pp. 5051–5063, 2022.
[15] S. Wang, Y. Mei, and M. Zhang, "Explaining genetic programming-evolved routing policies for uncertain capacitated arc routing problems," *IEEE Trans. Evol. Computat.*, vol. 28, no. 4, pp. 918–932, 2024.
[16] X. Chen, R. Bai, R. Qu, and H. Dong, "Cooperative double-layer genetic programming hyper-heuristic for online container terminal truck dispatching," *IEEE Trans. Evol. Computat.*, vol. 27, no. 5, pp. 1220–1234, Oct. 2023.
[17] F. Zhao, B. Zhu, and L. Wang, "An estimation of distribution algorithm-based hyper-heuristic for the distributed assembly mixed no-idle permutation flowshop scheduling problem," *IEEE Trans. Syst. Man Cybern, Syst.*, vol. 53, no. 9, pp. 5626–5637, 2024.
[18] Y. Wu and P. Ji, "A solution method for the component allocation problem in printed circuit board assembly," *Assembly Autom.*, vol. 30, no. 2, pp. 155–163, Apr. 2010.
[19] F. Vainio, T. Pahikkala, M. Johnsson, O. S. Nevalainen, and T. Knuutila, "Estimating the production time of a PCB assembly job without solving the optimised machine control," *Int. J. Comput. Integ. M.*, vol. 28, no. 8, pp. 823–835, 2015.
[20] D. Li, S. Chen, R. Chiong, L. Wang, and S. Dhakal, "Predicting the printed circuit board cycle time of surface-mount-technology production lines using a symbiotic organism search-based support vector regression ensemble," *Int. J. Prod. Res.*, vol. 59, no. 23, pp. 7246–7265, 2020.
[21] A. Tóth, T. Knuutila, and O. S. Nevalainen, "Machine configuration and workload balancing of modular placement machines in multi-product PCB assembly," *Int. J. Comput. Integ. M.*, vol. 31, no. 9, pp. 815–830, Sep. 2018.
[22] P. Ji, M. Sze, and W. Lee, "A genetic algorithm of determining cycle time for printed circuit board assembly lines," *Eur. J. Oper. Res.*, vol. 128, no. 1, pp. 175–184, 2001.
[23] S. Emet, T. Knuutila, E. Alhoniemi, M. Maier, M. Johnsson, and O. S. Nevalainen, "Workload balancing in printed circuit board assembly," *Int. J. Adv. Manuf. Tech.*, vol. 50, no. 9, pp. 1175–1182, 2010.
[24] W.-L. Lin and V. Tardif, "Component partitioning under demand and capacity uncertainty in printed circuit board assembly," *Int. J. Flex. Manu. Sys.*, vol. 11, pp. 159–176, 1999.
[25] K.-J. Hu, "Fuzzy goal programming technique for solving flexible assignment problem in PCB assembly line," *J. Inform. Optim. Sci.*, vol. 38, no. 3, pp. 423–442, May. 2017.
[26] J. Mumtaz, Z. Guan, L. Yue, L. Zhang, and C. He, "Hybrid spider monkey optimisation algorithm for multi-level planning and scheduling problems of assembly lines," *Int. J. Prod. Res.*, vol. 58, no. 20, pp. 6252–6267, Oct. 2020.
[27] Y. Chen, J. Zhong, J. Mumtaz, S. Zhou, and L. Zhu, "An improved spider monkey optimization algorithm for multi-objective planning and scheduling problems of PCB assembly line," *Expert Syst. Appl.*, vol. 229, p. 120600, Nov. 2023.
[28] T.-L. Chen, J. C. Chen, Y.-Y. Chen, and Y.-J. Chang, "The optimal configuration for various placement machines in PCB assembly lines," *Ann. Oper. Res.*, 2024.
[29] N. D. Choudhury, W. E. Wilhelm, B. Rao, J. Gott, and N. Khotekar, "Process planning for circuit card assembly on a series of dual head placement machines," *Eur. J. Oper. Res.*, vol. 182, no. 2, pp. 626–639, Oct. 2007.
[30] J. Mumtaz, Z. Guan, L. Yue, Z. Wang, S. Ullah, and M. Rauf, "Multi-level planning and scheduling for parallel PCB assembly lines using hybrid spider monkey optimization approach," *IEEE Access*, vol. 7, pp. 18 685–18 700, Dec. 2019.
[31] J. Koskinen, C. Raduly-Baka, M. Johnsson, and O. S. Nevalainen, "Rolling horizon production scheduling of multi-model PCBs for several assembly lines," *Int. J. Prod. Res.*, vol. 58, pp. 1052–1073, Apr. 2019.
[32] A. Tóth, T. Knuutila, and O. S. Nevalainen, "A divide and combine method for machine configuration and workload balancing problem in multiple product PCB assembly," *Int. J. Adv. Manuf. Technol.*, Mar. 2022.
[33] F. Vainio, M. Maier, T. Knuutila, E. Alhoniemi, M. Johnsson, and O. S. Nevalainen, "Estimating printed circuit board assembly times using neural networks," *Int. J. Prod. Res.*, vol. 48, no. 8, pp. 2201–2218, Aug. 2010.
[34] D. Li and S. W. Yoon, "PCB assembly optimization in a single gantry high-speed rotary-head collect-and-place machine," *Int. J. Adv. Manuf. Tech.*, vol. 88, pp. 2819–2834, 2017.
[35] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2022. [Online]. Available: "https://www.gurobi.com"
[36] J. Zhong, Y. Chen, and J. Mumtaz, "A multi-objective scheduling optimization method for PCB assembly lines based on the improved spider monkey algorithm," in *the 2nd ICAME*. MDPI, 2022, p. 15.

Guangyu Lu (Graduate Student Member, IEEE) was born in Taiyuan, China, in 1996. He received the B.E. degree in automation from Dalian Maritime University, Dalian, China, in 2019.

He is currently working toward the Ph.D. degree in control science and engineering with the Harbin Institute of Technology, Harbin, China. His research interests include production scheduling and process optimization.

Huijun Gao (Fellow, IEEE) received the Ph.D. degree in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2005.

From 2005 to 2007, he was a Postdoctoral Researcher with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. Since 2004, he has been with the Harbin Institute of Technology, where he is currently a Chair Professor and the Director of the Research Institute of Intelligent Control and Systems. His research interests include intelligent and robust control, robotics, mechatronics, and their engineering applications.

Dr. Gao was the Vice President of the IEEE Industrial Electronics Society from 2021 to 2024, and a council member of IFAC from 2017 to 2023. He is/was the Editor-in-Chief of IEEE/ASME TRANSACTIONS ON MECHATRONICS, the Co-Editor-in-Chief of IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, and an Associate Editor of Automatica, IEEE TRANSACTIONS ON CYBERNETICS, and IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS. He was the recipient of the 2022 Dr.-Ing. Eugene Mittelmann Achievement Award and the 2023 Norbert Wiener Award. He is a member of the Academia Europaea and a Distinguished Lecturer of the IEEE Systems, Man and Cybernetics Society. He has been an ESI Highly Cited Researcher since 2014.

Zhengkai Li (Member, IEEE) was born in Jinan, China, in 1991. He received the B.E. degree in detection, guidance, and control technology and the M.E. degree in control engineering from Northwestern Polytechnical University, Xi'an, China, in 2013 and 2016, respectively. He also received the Ph.D. degree in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2022.

He is currently with the Research Institute of Interdisciplinary Intelligent Science, Ningbo University of Technology, Ningbo, China. His current research interests include scheduling and systems optimization.

Xinghu Yu (Member, IEEE) was born in Yantai, China, in 1988. He received the M.M. degree in osteopathic medicine from Jinzhou Medical University, Jinzhou, China, in 2016, and the Ph.D. degree in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2020.

He is currently the Chief Executive Officer of Ningbo Institute of Intelligent Equipment Technology Company Ltd., Ningbo, China. He has authored more than 30 technical papers in refereed international journals and conference proceedings, including IEEE Transactions journals, and holds more than 20 invention patents. His research interests include advanced control, intelligent systems, and biomedical image processing.

Tong Wang (Senior Member, IEEE) received the M.E. degree in control theory and control engineering from the Liaoning University of Technology, Jinzhou, China, in 2013, and the Ph.D. degree in control science and engineering from the Harbin Institute of Technology (HIT), Harbin, China, in 2017.

He is currently an Associate Professor with HIT. His research interests include fuzzy control, stochastic adaptive control, and networked control.

Jianbin Qiu (Fellow, IEEE) received the B.Eng. and Ph.D. degrees in Mechanical and Electrical Engineering from the University of Science and Technology of China, Hefei, China, in 2004 and 2009, respectively. He also received the Ph.D. degree in Mechatronics Engineering from the City University of Hong Kong, Kowloon, Hong Kong, in 2009.

He is currently a Full Professor at the School of Astronautics, Harbin Institute of Technology, Harbin, China. He was an Alexander von Humboldt Research Fellow at the Institute for Automatic Control and Complex Systems, University of Duisburg-Essen, Duisburg, Germany. His current research interests include intelligent and hybrid control systems, signal processing, and robotics.

Prof. Qiu serves as the Chair of the IEEE Industrial Electronics Society Harbin Chapter, China. He is an Associate Editor of IEEE TRANSACTIONS ON FUZZY SYSTEMS, IEEE TRANSACTIONS ON CYBERNETICS, and IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS.

Juan J. Rodríguez-Andina (Fellow, IEEE) received the M.Sc. degree from the Technical University of Madrid, Madrid, Spain, in 1990, and the Ph.D. degree from the University of Vigo, Vigo, Spain, in 1996, both in electrical engineering.

He is currently a Professor with the Department of Electronic Technology, University of Vigo, and also with the Research Institute of Interdisciplinary Intelligent Science, Ningbo University of Technology, Ningbo, China. From 2010 to 2011, he was on sabbatical leave as a Visiting Professor with the Advanced Diagnosis, Automation, and Control Laboratory, North Carolina State University, Raleigh, NC, USA. From 2015 to 2017, he delivered summer courses at Harbin Institute of Technology, Harbin, China. His research interests include the implementation of complex control and processing algorithms and intelligent sensors in embedded platforms. He has authored more than 180 journal and conference articles and holds several Spanish, European, and U.S. patents.

Prof. Rodríguez-Andina is a co-author of the articles awarded the 2023 IEEE Transactions on Industrial Electronics Outstanding Paper Award and the 2017 IEEE Industrial Electronics Magazine Best Paper Award. He received the 2020 Anthony Hornfeck Award from the IEEE Industrial Electronics Society. From 2016 to 2021, he was the Vice President for Conference Activities of the IEEE Industrial Electronics Society. He served as the Editor-in-Chief for IEEE INDUSTRIAL ELECTRONICS MAGAZINE, from 2013 to 2015, and as an Associate Editor for IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, from 2008 to 2018, and IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, from 2011 to 2022. He is currently serving as a Co-Editor-in-Chief for IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS and as an Associate Editor for IEEE OPEN JOURNAL OF THE INDUSTRIAL ELECTRONICS SOCIETY.