Hyper-Heuristic Optimization Using Multi-Feature Fusion Estimator for PCB Assembly Lines

By Guangyu Lu

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

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

***Abstract*—Printed circuit board assembly line scheduling (PCBALS) is critical to production efficiency, which is a major difficulty in the electronic industry for assembly lines using surface mounters. This is a special type of line optimization 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 estimator (HHO-MFFE) for PCBALS using linear-aligned-heads surface mounters. The objective and constraints are discussed, and a min-max mathematical 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 component duplication are proposed to improve the applicability of the algorithm and the quality of the solution. A neural network 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 the proposed time estimator has higher accuracy, with a mean absolute error of 3.43%, compared to both regression and heuristic-based estimators, and that the HHO-MFFE is better than other state-of-the-art algorithms, with average improvement of 4.53%$\sim$12.18%.**

***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 used in PCB assembly. Manufacturers tend to use multiple surface mounters in series to increase productivity. However, they face difficulty 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 loader, screen printer, surface mounters, reflow furnace, automatic optical inspector (AOI), and unloader. The screen printer applies solder paste to the surface of PCBs. Surface mounters pick and place components on the PCB pads. The reflow furnace melts solder paste already pre-positioned on the PCB, before cooling it to create a permanent solder. Finally, the AOI looks for defects on the PCB to ensure assembly quality. Central to production control is the efficient use of machines, with surface mounters been the bottleneck for assembly efficiency.

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 [1], [2], [3], and the optimization for single machine has been thoroughly studied [4], [5]. Component allocation has been explored for both model-based [6], [7], [1] and heuristic-based [8], [9], [3], [10] 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 [1], nozzle changes [7] and feeder utilization [9]. However, most research to date has concentrated on the optimization of lines with rotary surface mounters [2], [3], [1], [9], which differs from the structural design with linearly aligned heads. These optimization methods do not take sufficiently into account the feature of the problem, limiting the productivity of the lines with linearly-aligned-heads surface mounters.

Heuristic algorithms have been well studied in the field of assembly lines [11], disassembly lines [12] and parallel

This work was supported in part by the National Natural Science Foundation of China under Grant U20A20188, Grant 62203141 and Grant 62303402, in part by the Major Scientific and Technological Research Project of Ningbo under Grant 2021Z040, and in part by the New Cornerstone Science Foundation through the XPLORER PRIZE. (*Corresponding author: Huijun Gao.*)

Huijun Gao, Guangyu Lu, and Jianbin Qiu are with the Research Institute of Intelligent Control and Systems, Harbin Institute of Technology, Harbin 150001, China (e-mail: hjgao@hit.edu.cn; 20b004007@stu.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 Ningbo University of Technology, Ningbo 315211, China (e-mail: juan.j.r@ieee.org).

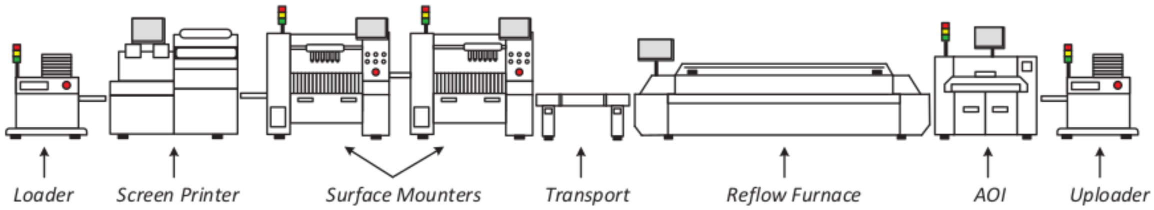

Fig. 1. PCB Assembly Line.

machines optimization [13]. Hyper-heuristic algorithms are a novel optimization framework that combines the advantages of high-level heuristics and low-level heuristics to adaptively solve a wide range of complex optimization problems. They have been widely applied for route scheduling [14], truck dispatching [15], or flow shop scheduling [16], to cite just a few examples. The estimation of assembly time has been studied with regression fitting approaches [17], [18], [19]. Neural networks (NN) provide strong nonlinear fitting capability, and they can have a high fitting accuracy by designing extracted data features. However, related studies [20] target only rotary-head surface mounters.

In this article, a hyper-heuristic optimization algorithm with a multi-feature fusion time estimator for PCB assembly lines is proposed, whose contributions 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 are 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 NN time estimator, which makes the estimation more accurate.
4) An aggregative clustering algorithm for duplicated component points is proposed to improve the efficiency of assembly lines.

The rest of the article is organized as follows. Section II reviews related work about line optimization. Section III formulates the mathematical model. The hyper-heuristic optimization with a multi-feature fusion time estimator is presented in Section IV. Experimental results compared with other approaches are presented and discussed in Section V. Section VI concludes the article.

II. Literature Review

Many studies have contributed to the optimization of PCB assembly lines. In this article, the single-model case [21] is considered, in which 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, which is the main focus of the research. The present work focuses on optimizing search capabilities and time estimation accuracy in PCB assembly lines.

Mathematical modeling can solve problems optimally, yet it is complex and difficult to implement effectively. The integrated model for changeable head configuration and component allocation presented in [1] 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 [6]. As an extension to it, 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, including genetic algorithms [3], [2] and hybrid spider monkey optimization (HSMO) [26], [27], among others. In [2], a genetic algorithm to identify potential solutions for machine-specific component allocation and placement sequence problems is presented. In [3], 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. It 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 used for PCB line optimization. In [8] line assignment of modular surface mounters is divided in 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 [9], 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 [28], 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 state-of-the-art research is based on estimators. Early linear regression research in [17] estimates assembly time from the number of component types and placement points. A regularized least-squares regression with a novel feature that is solved using the nearest neighbor heuristic is proposed in [18]. A supported regression method combined with symbiotic organism search is proposed in [19] to improve estimation accuracy. NNs have the ability to fit arbitrary nonlinear functions. In [20], a multi-layer perceptron network estimator is presented considering component shape and the area of the smallest rectangle around the component.

III. Problem Formulation and Model

A. Problem Formulation

PCB assembly lines have both similarities and differences with regard to general production lines. They both assign components to different machines for processing, and multiple machines can assemble the same type of component to improve efficiency, which is called a duplicated condition. In addition, they are subject to assembly priority requirements and may restrict the types of machines to which components can be assigned. The primary difference is in the computation of assembly time for scheduling, which in PCB manufacturing lines is in general more complex and depends on the optimization of a single machine, as well as machine type, available tools, and the types and number of components to be allocated.

Among the many factors that influence the efficiency of a PCB assembly line, the surface mounter takes the longest time to process in the production line, thus determining the efficiency of the entire line. A variety of interdependent factors influence the assembly efficiency of a single surface mount machine, including number of cycles, pickups, nozzle changes, and placement points [5]. Two primary types of constraints affect assembly line scheduling, namely tool and machine constraints. Tool constraints refer to the limited number of feeders, nozzles, and other devices available, whereas machine constraints refer to the types of parts that must be assembled by a specific machine for high-speed and high-precision surface mounters to operate synergistically in a production line.

Improving search efficiency for high-quality solutions is critical to line optimization. A large number of combinations for component allocation makes it difficult to get high-quality solutions, and computing effort increases rapidly as the problem scales up, needing massive resources even for small-scale data. The unique mechanics of linear-aligned-heads surface mounters must be taken into account when determining assembly time. Traditional point-based fitting procedures are

TABLE I
Notations of the Mathematical Model

Notation	Description
Indices & Sets	
$i \in I$	Index of component type, $I = \{1, 2, \cdots\}$
$j \in J$	Index of nozzle type, $J = \{1, 2, \cdots\}$
$k \in K$	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 of assembly priority , $Q = \{(i, i'), \cdots\}, i \in I, i' \in I$, which means i needs to be assembled before i'
$\tilde{m}_q \in M$	Index of surface mounter machine, which indicates either the last machine to assemble component i or the first machine to assemble component i', $q = (i, i') \in Q$
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
η_{im}	$= 1$ iff. component type i is compatible with machine m
λ_{ip}	$= 1$ iff. component type i is compatible with point p
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}	$= 1$ iff. any point is assembled in cycle k of machine m
u_{ikhm}	$= 1$ iff. component type i is assigned to head h in cycle k of machine m
v_{skhm}	$= 1$ iff. head h picks up components from slot s in cycle k of machine m
f_{ism}	$= 1$ iff. component i is assigned to slot s of machine m
e_{skm}	$= 1$ iff. component is picked up when the left-most head aligns to slot s of machine m in cycle k
n_{khm}	$= 1$ iff. head h of machine m changes nozzles between cycles k and $k+1$
r_{im}	$= 1$ iff. component i is assembled by machine m
w_{km}	Slots crossed by heads during pickup in cycle k of machine m

not applicable, since the type of components assigned to the machine, as well as the number of points of each type, can have a large impact on pickup efficiency. Single-machine optimization takes a long time to obtain the exact time and is not appropriate for line optimization with large solution spaces.

B. Mixed Integer Model

The notations used in this article are listed on Table I.

In [4], an integer model for head task assignment including the major factors that influence assembly efficiency is proposed. Based on this model, a new approximation model is proposed that assesses assembly line efficiency in terms of weighted metrics.

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

The objective (1) of the model is to minimize the maximum assembly time among all machines, using key metrics for assembly cycle, nozzle change, pick up, and placement operations. As described below, Constraints (2)–(6) are related to the configuration of a single surface mounter, whereas Constraints (7)–(13) incorporate the factors for line optimization.

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

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

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

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

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

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

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

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

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

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

$$\begin{aligned} m - N\cdot(1 - r_{im}) \leq \tilde{m}_q \leq m + N\cdot(1 - r_{i'm}) \\ \forall q = (i, i') \in Q, \tilde{m}_q \in M, m \in M \end{aligned} \tag{12}$$

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

The cycle of each machine with component assignment is defined in Constraint (2). Constraint (3) calculates the number of nozzle changes. Constraint (4) converts the 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 [4].

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 components and head-assigned components. Constraint (11) restricts the components to be assigned to compatible machines. Constraints (12) and (13) are restrictions on 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. The model is validated using the Gurobi solver [32].

IV. Hyper-heuristic Optimization with a NN Estimator

A. *Solution Framework for the HHO-MFFE algorithm*

As shown in Fig. 2, the proposed evolutionary-based HHO-MFFE is built from low-level heuristics and an estimator. Component division and cluster-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. The combination and execution order of low-level heuristics are specified in the population-generating code. A multi-feature fusion time estimator based on fully connected NNs is proposed to calculate the fitness value of each individual, which is fed with the data and estimated sub-objectives. In the iterative process, truncated crossover and mutation operations are conducted on the individuals. After the evolutionary process is completed, placement points with the same component type are segregated using an aggregated cluster algorithm. The last phase of line optimization, known as single machine optimization, uses the advanced techniques proposed in [4]. The best solution for each population acts as a candidate solution, which then executes a single machine optimization to evaluate their quality and decrease the impact of estimation errors.

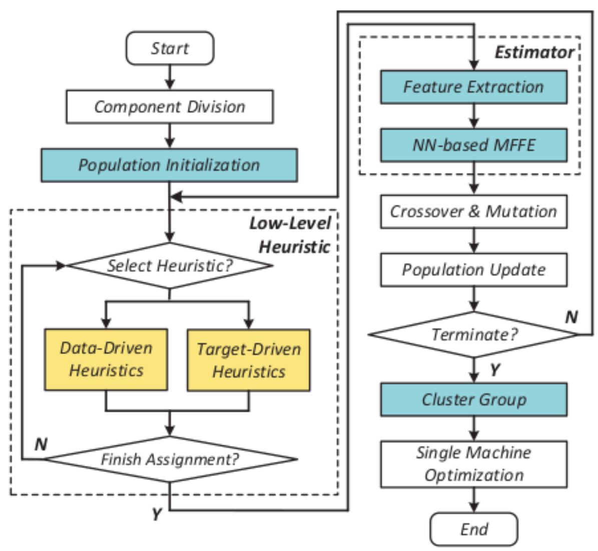

Fig. 2. Flowchart of the proposed HHO-MFFE algorithm.

B. *Low-Level Heuristics for Component Allocation*

Low-Level Heuristics (LLHs) are basic compositions of hyper-heuristics, which can be divided in two types: data- and target-driven. The allocation sequence for components is preset, and heuristics are selected depending on the allocated components.

Data-driven LLHs are connected to the number of points, component type, and nozzle type, as follows: Minimum Points, Minimum Component Types, Minimum Nozzle Types, and Minimum Ratio Heuristics allocate components to the machine with the minimal assigned placement points, component types, nozzle types, and minimal ratio of number of component types to nozzle types, respectively.

Target-driven LLHs are related to assembly efficiency, and the key sub-objectives are extracted as a basis for component allocation. The optimal value of the sub-objective can be estimated without a specialized optimization procedure. The sub-objective estimation based on a cascade rounding method proposed in [33] is used here, and the number of heads assigned to nozzle type j of machine m is denoted as γ_{jm}. The target-driven LLHs are:

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

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

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

3) *Minimum Pickup Heuristic*, which allocates components to the machine with minimal pickup operations.

Algorithm 1 presents a method to estimate the number of pick-ups. A hierarchical greedy head heuristic assigns components to heads in decreasing order, subject to the number of heads that are accessible to the nozzle. Each component that is allocated to the head implies a new cycle, and the number of pick-ups is equal to the maximum number of points that are assigned to the heads in each cycle. The number of component feeders and machine specification restricts the allocatable machines. LLHs take into account the limitations imposed by the allocation of components of the same type. Priority constraints limit the machines that can be allocated, and the component is replaced with one assigned to fulfill the requirement if no machine is allocatable. The machine with fewest points among LLHs with the same evaluation value has the highest priority to assemble components.

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, as well as individual genes of varying lengths, increasing search diversity. The length of genes is limited to the number of component division groups. Cyclic access to individual patterns during component allocation is applied to handle the case when gene length is less than the limit value. All individuals are initialized with random lengths and pattern combinations. Each one of two genes selects a split point and performs a crossover operation to exchange gene

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**
  $j\leftarrow\sum_{j'\in J}\xi_{ij'}\cdot i$;
  if $\mathcal{N}_j \operatorname{Mod} \gamma_j = 0$ **then**
    $\mathcal{L}_j\leftarrow\mathcal{L}_j+1$;
  end
  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
$\mathcal{O}\leftarrow\sum_{c=1}^{c=\sum_{i\in I}\phi_i}\mathcal{K}_c$

segments. The crossover operator inserts randomly generated patterns at a 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, effectively reducing the number of executions of single-machine optimization and increasing solving efficiency.

D. Multi-Feature Fusion Time Estimator

NNs perform 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 that 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 fundamental data consists of 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, comprise the sub-objective coding. Nozzle and component codes are presented in descending order of the total number of points. A sufficiently long encoding is used to ensure consistency across diverse data inputs to networks, with redundant bits supplemented by zeros.

Estimation of nozzle change probability cannot be directly coded, and Algorithm 2 proposes a computation heuristic for that. Components with the same nozzle type are grouped according to their respective nozzle heads. The group of nozzle j is denoted as $\mathcal{G}_j$. Nozzle groups are progressively assigned to heads, starting with empty heads and proceeding sequentially to the heads with fewest points. When the allocation process is complete, the heads with the most and least points are divided equally, which is effective if the efficiency gain from reducing the number of cycles after equalization outweighs the efficiency loss from increasing nozzle change. 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 until there is no overall increase in efficiency.

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 $1 \times \gamma_j$ nozzle group $\mathcal{G}_j$ with $\sum_{i \in I} \phi_i \cdot \xi_{ij} / \gamma_j$ points for $j \in J$;
  for $n \in \mathcal{G}_j, j \in J$ **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 heads with max and min points $h' \leftarrow \arg\max_{h \in H} \mathcal{T}_h$, $h'' \leftarrow \arg\min_{h \in H} \mathcal{T}_h$;
    if $\mathcal{N}_{h'} = \mathcal{N}_{h''}$ **then**
      break;
    end
    $j' \leftarrow \mathcal{N}_{h'}$, $\mathcal{H}_1 \leftarrow \{h \mid \mathcal{N}_h = j', h \in H\}$, $j'' \leftarrow \mathcal{N}_{h''}$, $\mathcal{H}_2 \leftarrow \{h \mid \mathcal{N}_h = j'', h \in H\}$;
    if $T_3 \cdot (\mathcal{T}_{h'} - \mathcal{T}_{h''}) > T_2 \cdot ||\mathcal{H}_2| - |\mathcal{H}_1||$ **then**
      break;
    end
    $N \leftarrow ||\mathcal{H}_2| - |\mathcal{H}_1||$, $V \leftarrow V - T_3 \cdot (\mathcal{T}_{h'} - \mathcal{T}_{h''}) + T_2 \cdot N$, $\mathcal{T}' \leftarrow \mathcal{T}$;
    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**
    $V^* \leftarrow V$, $N^* \leftarrow N$, $\gamma_{j'} \leftarrow \gamma_{j'} + 1$;
  end
end

E. Heuristic for Component Duplication

Components may have multiple available feeders and can be assigned to more than one surface mounter to improve production efficiency. Available feeders are allocated to different machines for the same component type proportionally to the number of points. The specific machine to which each point is assigned needs to be determined. Before executing hyper-heuristic search, the average number of points for each type of component is multiplied by a value that serves as a grouping threshold. Components with points that exceed this threshold are grouped. This grouping strategy balances search efficiency and diversity. The number of available feeders for a component restricts the maximum number of allocated machines, but not as a basis for grouping. The number of machine-allocated points is denoted as

$$U_{im} = \sum_{k \in K} \sum_{h \in H} u_{ikhm} \quad \forall i \in I, m \in M, \tag{16}$$

Algorithm 3 provides an aggregative clustering heuristic. The components with a single feeder have all of their points allocated to one machine, resulting in the center points of each machine. Based on this, the distribution of the duplicated points on heads affects their distance from the center point of the machine. A bias $\mathcal{R}_{im}$ related to the head task assignment is applied in the clustering process, as follows:

$$\mathcal{R}_{im} = \sum_{k \in K} \sum_{h \in H} \left(\sum_{p \in P} x_p \cdot \lambda_{ip} \cdot u_{ikhm} - h \cdot \rho \right) \forall i \in I, m \in M \tag{17}$$

where ρ is the interval distance between adjacent heads.

Algorithm 3: Aggregated Clustering Algorithm for Duplicated Component Points

Input : Available feeder θ, component points set P, machine-assigned points U, points position (x, y)
Output: Machine-allocated points $\overline{\mathcal{P}}$

Set machine-assigned sets $\mathcal{P}_m \leftarrow \varnothing$ and number of machine-assigned points $\mathcal{U}_{im} \leftarrow 0, i \in I, m \in M$;
for $m \in M$ **do**
  for $i \in \{i' \mid U_{i'm} > 0, \theta_{i'} = 1, i' \in I\}$ **do**
    $\mathcal{U}_{im} \leftarrow |P_i|$, $\mathcal{P}_m \leftarrow \mathcal{P}_m \cup P_i$;
  end
  Set center points $\mathcal{X}_m \leftarrow \sum_{p \in \mathcal{P}_m} x_p / |\mathcal{P}_m|$, $\mathcal{Y}_m \leftarrow \sum_{p \in \mathcal{P}_m} y_p / |\mathcal{P}_m|$ of each machine ;
end
while ***true*** **do**
  $\overline{\mathcal{X}} \leftarrow \mathcal{X}$, $\overline{\mathcal{Y}} \leftarrow \mathcal{Y}$, $\overline{\mathcal{U}} \leftarrow \mathcal{U}$, $\overline{\mathcal{P}} \leftarrow \mathcal{P}$;
  for $p \in \{p' \mid p' \in P_i, \theta_i > 1, i \in I\}$ **do**
    $m \leftarrow \arg\min_{m' \in M} \left\{ (\mathcal{X}_{m'} - x_p + \mathcal{R}_{im'})^2 + (\mathcal{Y}_{m'} - y_p)^2 \mid \overline{\mathcal{U}}_{im'} < U_{im'} \right\}$ as the allocated machine, $\overline{\mathcal{P}}_m \leftarrow \overline{\mathcal{P}}_m \cup \{p\}$, $\overline{\mathcal{U}}_{im} \leftarrow \overline{\mathcal{U}}_{im} + 1$;
    $\mathcal{X}_m \leftarrow \mathcal{X}_m + (x_p - \mathcal{X}_m - \mathcal{R}_{im}) / |\overline{\mathcal{P}}_m|$, $\mathcal{Y}_m \leftarrow \mathcal{Y}_m + (y_p - \mathcal{Y}_m) / |\mathcal{P}_m|$;
  end
  if $\overline{\mathcal{X}} = \mathcal{X}$ **and** $\overline{\mathcal{Y}} = \mathcal{Y}$ **then**
    break;
  end
end

V. Comparative Experiments

A. Experimental Setup

Experiments are run using a PC with an Intel(R) Core(TM) i5-14600KF with Gurobi 11.0. Table II shows the experimental parameters of hyper-heuristic and NNs. Iterations are carried out across the 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 activation function. Results are compared for PCB assembly lines $L1$, $L2$, and $L3$, 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

TABLE II
NN AND HYPER-HEURISTIC PARAMETERS

Method	Parameters	Value
Hyper Heuristic	Size of Population Group	10
	Number of Individual	20
	Crossover & Mutation Rate	0.6 & 0.1
	Number of Iterations	50
NN	Learning Rate	10^{-5}
	Number of Epochs	8000

smaller scale, as shown in Table III. As meta-heuristic results are random, the average of the ten runs is taken as the result.

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. A point distribution that is either sparse or concentrated can affect assembly time, reducing the generalization performance of the fitting method. Table IV shows statistical information for PCB data. Data outliers are detected and removed using the inter-quartile range rule [19] with a multiplier of 0.6. Training and testing data have similar distribution characteristics.

B. Comparison of Proposed Algorithm and Mathematical Model

Mathematical programming can be used to find optimal solutions, but only for small-scale data. In this section, the solution using the proposed method is compared with the optimal solution of the model, which is built by extracting key metrics that affect assembly efficiency. 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 a comparison of the first five data. T_M and T_H represent the weighted performance metrics of the model and the proposed algorithm, respectively. The 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 the higher efficiency of the solution makes it possible to apply it to larger-scale data.

C. Evaluation of Proposed Time Estimator

The accuracy of the time estimator impacts the search direction for component allocation, as well as the quality of solutions. Four different time estimators are used for comparison with the proposed one. The proposed estimator yields E_1. E_2 refers to the NN fitting method using basic parameters such as the number of points, number of components, number of nozzles, board size, and so on, which is another way of encoding. The heuristic estimators proposed in [3] and [8] are denoted as E_3 and E_4, respectively, with coefficients computed using the least squares method. E_5 is an ensemble algorithm with symbiotic organism search-based support vector regression [19].

The mean and maximum absolute error of training and testing data are listed in Table VI. The performance of the fitting method on the testing sets the basis for evaluating the accuracy of the estimator. It can be seen that the NN method provides better time estimation. The proposed estimator encoding method reduces the average absolute error on the testing set from 5.09% to 2.01%, in contrast to the encoding method that simply feeds basic parameters. Simultaneous pick-up is not incorporated in two heuristic-based linear regression fittings, resulting in poorly fitted results with mean absolute errors of 28.82% and 27.65%, respectively. Despite being more effective in the workshop production line of the PCB assembly process, the SOS-based SVR has the lowest fitting accuracy, because it ignores the distinctive properties of each single PCB.

D. Comparison of the Component Allocation Algorithm with Other Methods

The main task of the line optimizer is to allocate components to machines. In this section, the proposed algorithm is compared with an industrial solver from an advanced manufacturer released in 2022, the hybrid algorithm [3], and the genetic algorithm [8]. The industrial solver is an optimizer embedded in an integrated production line management tool for surface mount assembly lines. The hybrid and genetic algorithms are both evolutionary-based methods that provide practical and effective solutions for PCB assembly line optimization by designing heuristic operators to search the solution domain. The industrial solver has a built-in surface mount optimization program, and the rest of the single-machine optimizations are based on the methods proposed in [4].

Table VII shows the results of optimization using the four algorithms mentioned above for the remaining 10 data. The proposed hyper-heuristic algorithm outperforms the industrial solver and the hybrid and genetic algorithms by 4.53%, 8.47%, and 12.18%, respectively. In addition, data distribution of the optimization results in three PCB assembly lines are shown in Fig. 3. In algorithms with randomized results, the hyper-heuristic produces a more consistent result. In most cases, 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 them outperform the best solutions from the other methods.

E. Analysis of Solving Efficiency

Solving efficiency is one of the most important indicators of algorithmic performance. Solving times are shown in Table VIII. 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. As a consequence, in the industrial solver optimization time cannot be separated from the rest and the comparison would not be fair for it. The genetic algorithm consists of relatively basic operators, which allow it to search quickly at the cost of solution quality. The hyper-heuristic and hybrid algorithms

TABLE III
PARAMETERS OF 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
Num. of Points	28	34	34	30	30	78	165	192	236	209	320	390	546	720	1510
Num. of Feeders	10	6	8	7	5	16	30	12	24	46	12	53	48	18	40

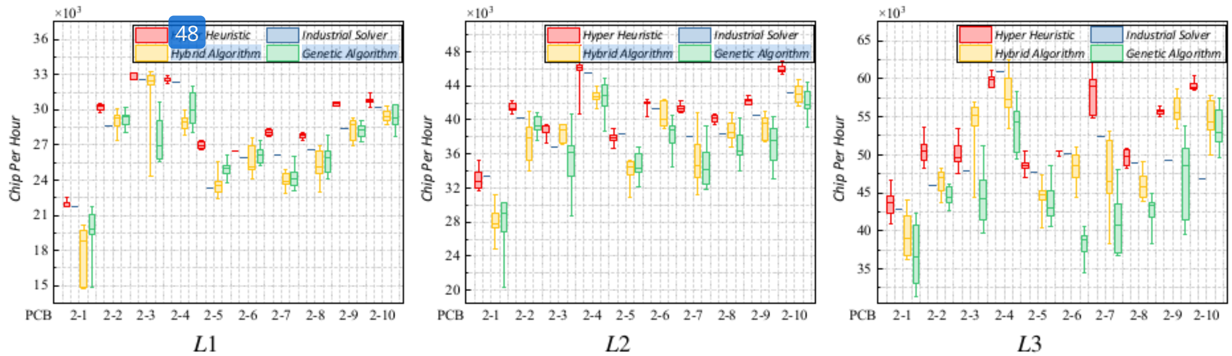

Fig. 3. Comparison of the distribution of assembly efficiency optimization results of different methods on three PCB assembly lines.

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
	Maximum	Minimum	Std. Dev	
	3.80	311.38	72.23	

TABLE V
COMPARISON OF THE WEIGHTED KEY METRICS INDICATORS MATHEMATICAL MODEL AND PROPOSED ALGORITHM

Line	L1			L2			L3		
	T_M	T_H	δT	T_M	T_H	δT	T_M	T_H	δT
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
COMPARISON OF ESTIMATED ACCURACY BETWEEN THE NN AND OTHER 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
Test	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

use a more complex time-fitting approach and account for component duplication, resulting in longer times than that of the genetic algorithm. The proposed hyper-heuristic is more efficient than the hybrid 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. By shortening the execution time of surface mounter optimization, efficiency may be further increased.

VI. CONCLUSION

This article presents a hyper-heuristic optimization method for PCBALS with an NN-based time estimator. The hyper-heuristic algorithm is implemented using data- and target-driven LLHs. A min-max mathematical model has been 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 strategy for component duplication divides components of the same type, balancing assembly time between machines and improving assembly efficiency. An aggregated clustering algorithm assigns placement points to the specific surface mounters. NN-based time estimators have high fitting accuracy, and the proposed coding with approximated sub-objectives further enhances fitting accuracy. The combination of the high accuracy of the estimator, along with the search capability of the hyper-heuristic for large domains, results in high-quality solutions for PCBALS problems. Compared with industrial solutions and other state-of-the-art solutions, the proposed algorithm has higher assembly efficiency and relatively stable results with acceptable solving times.

REFERENCES

[1] A. Rong, A. Tóth, O. S. Nevalainen, T. Knuutila, and R. Lahdelma, "Modeling the machine configuration and line-balancing problem of a

TABLE VII
COMPARISON OF THE ASSEMBLY TIME OF THE PROPOSED LINE OPTIMIZER WITH STATE-OF-THE-ART ALGORITHMS

	Hyper Heuristic			Industrial Solver				Hybrid Algorithm				Genetic 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	**12.72**	8.50	**6.45**	12.91	**8.41**	6.56	0.68%	16.23	10.06	7.17	19.03%	14.34	10.28	7.73	17.78%
2-2	**19.64**	**14.33**	**11.80**	20.78	14.75	12.95	6.13%	20.41	15.09	12.78	5.85%	20.35	15.92	13.36	9.31%
2-3	**21.10**	**17.81**	13.82	21.19	18.77	14.44	3.42%	22.37	17.95	**12.86**	-0.06%	25.23	19.70	15.70	14.58%
2-4	**26.10**	**18.60**	14.21	26.29	18.66	**13.96**	-0.26%	29.43	20.72	14.67	9.13%	28.32	20.08	15.85	9.34%
2-5	**27.90**	19.87	**15.46**	32.32	**19.59**	15.79	5.51%	31.91	22.19	16.94	11.88%	30.22	21.81	17.31	10.01%
2-6	**43.53**	**27.56**	**22.90**	44.42	27.91	23.02	1.28%	44.96	28.50	23.90	3.69%	44.09	30.18	24.21	5.51%
2-7	**50.23**	**34.00**	**26.74**	53.91	36.93	26.85	5.44%	58.71	40.14	30.27	16.03%	57.96	40.92	34.44	21.50%
2-8	**70.90**	**48.96**	**39.57**	73.96	51.16	40.18	3.45%	78.18	50.97	42.65	7.39%	76.26	53.67	46.02	11.17%
2-9	**84.95**	**61.40**	46.54	91.18	63.91	52.57	8.13%	91.66	66.19	**46.50**	5.21%	92.11	70.23	55.60	14.09%
2-10	**176.68**	**118.11**	**92.01**	179.94	125.79	116.23	11.56%	184.37	126.30	99.60	6.51%	185.38	129.72	101.87	8.49%
Avg	53.38	36.91	28.95	55.69	38.59	32.25	**4.53%**	57.82	39.81	30.74	**8.47%**	57.42	41.25	33.21	**12.18%**

TABLE VIII
COMPARISON OF THE SOLVING TIME OF THE PROPOSED LINE OPTIMIZER WITH STATE-OF-THE-ART ALGORITHMS

	Hyper Heuristic			Hybrid Algorithm			Genetic Algorithm		
PCB	*L*1	*L*2	*L*3	*L*1	*L*2	*L*3	*L*1	*L*2	*L*3
2-1	17.51	20.75	24.19	53.88	58.69	62.98	3.54	4.12	4.94
2-2	32.86	31.14	31.73	63.41	67.96	75.27	6.84	7.31	8.33
2-3	13.74	15.86	19.67	50.73	54.72	63.88	2.28	2.64	3.05
2-4	21.56	22.78	25.83	63.85	68.38	75.79	5.13	6.35	7.58
2-5	101.63	85.46	86.58	85.81	89.39	95.90	19.12	18.97	39.73
2-6	21.14	18.02	22.12	63.55	67.24	74.09	2.99	3.15	3.63
2-7	89.65	72.00	67.60	97.14	95.66	105.41	21.20	19.31	19.22
2-8	40.26	43.13	37.50	90.67	95.49	105.47	9.19	10.98	11.22
2-9	29.55	26.83	29.94	88.91	92.60	101.81	5.51	5.06	5.49
2-10	131.74	83.84	75.50	145.04	156.13	170.96	21.29	18.75	17.86

PCB assembly line with modular placement machines," *Int. J. Adv. Manuf. Tech.*, vol. 54, no. 1, pp. 349–360, Apr. 2011.

[2] 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.

[3] 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.

[4] 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.

[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] 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.

[7] 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.

[8] 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.

[9] 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, pp. 1790–1797, 2017.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] N. R. Sabar, M. Ayob, G. Kendall and R. Qu, "A Dynamic Multiarmed Bandit-Gene Expression Programming Hyper-Heuristic for Combinatorial Optimization Problems,," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 217-228, Feb. 2015.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[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.*, Feb. 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] L. Gurobi Optimization, “Gurobi optimizer reference manual,” 2022. [Online]. Available: ”https://www.gurobi.com”

[33] 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.

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

From 2005 to 2007, he was 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 is the Vice President of the IEEE Industrial Electronics Society and a Council Member of the International Federation of Automatic Control. 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 is a Member of the Academia Europaea and a Distinguished Lecturer of the IEEE SYSTEMS, MAN,AND CYBERNETICS SOCIETY.

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 Harbin Institute of Technology, Harbin, China.

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 2021.

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

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 School of Electronic and Information Engineering, 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 with 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.

Hyper-Heuristic Optimization Using Multi-Feature Fusion Estimator for PCB Assembly Lines

ORIGINALITY REPORT

11%
SIMILARITY INDEX

PRIMARY SOURCES

1	Huijun Gao, Yanbin Liu, Weichao Sun, Xinghu Yu. "Adaptive Wavelet Tracking Control of Dual-Linear-Motor-Driven Gantry Stage With Suppression of Crossbeam Rotation", IEEE/ASME Transactions on Mechatronics, 2024 Crossref	71 words — 1%
2	Zhitai Liu, Huijun Gao, Xinghu Yu, Weiyang Lin, Jianbin Qiu, Juan J. Rodríguez-Andina, Dongsheng Qu. "B-Spline Wavelet Neural-Network-Based Adaptive Control for Linear-Motor-Driven Systems via a Novel Gradient Descent Algorithm", IEEE Transactions on Industrial Electronics, 2024 Crossref	56 words — 1%
3	Guangyu Lu, Xinghu Yu, Hao Sun, Zhengkai Li, Jianbin Qiu, Huijun Gao. "A Scan-Based Hierarchical Heuristic Optimization Algorithm for PCB Assembly Process", IEEE Transactions on Industrial Informatics, 2024 Crossref	50 words — 1%
4	www.researchgate.net Internet	32 words — < 1%
5	www.tandfonline.com Internet	28 words — < 1%
6	Shujuan Guo, Fei Geng, Katsuhiko Takahashi, Xiaohan Wang, Zhihong Jin. "A MCVRP-based	25 words — < 1%

model for PCB assembly optimisation on the beam-type placement machine", International Journal of Production Research, 2018
Crossref

7 Proceedings in Adaptation Learning and Optimization, 2015.
Crossref
19 words — < 1%

8 Xin Gong, Masaki Ogura, Jun Shen, Tingwen Huang, Yukang Cui. "Optimal Epidemics Policy Seeking on Networks-of-Networks Under Malicious Attacks by Geometric Programming", IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2023
Crossref
19 words — < 1%

9 Shujuan Guo, Katsuhiko Takahashi, Katsumi Morikawa, Zhihong Jin. "An integrated allocation method for the PCB assembly line balancing problem with nozzle changes", The International Journal of Advanced Manufacturing Technology, 2011
Crossref
18 words — < 1%

10 acris.aalto.fi
Internet
18 words — < 1%

11 wiki.control.fel.cvut.cz
Internet
18 words — < 1%

12 Teen-Hang Meen, Stephen Prior, Artde Lam. "Innovation in Design, Communication and Engineering - Proceedings of the 2014 3rd International Conference on Innovation, Communication and Engineering (ICICE 2014), Guiyang, Guizhou, P.R. China, October 17-22, 2014", CRC Press, 2019
Publications
16 words — < 1%

13	Zhengkai Li, Xinghu Yu, Jianbin Qiu, Huijun Gao. "Cell Division Genetic Algorithm for Component Allocation Optimization in Multi-Functional Placers", IEEE Transactions on Industrial Informatics, 2021 Crossref	16 words — < 1%
14	salford-repository.worktribe.com Internet	15 words — < 1%
15	Jabir Mumtaz, Zailin Guan, Lei Yue, Zhengya Wang, Saif Ullah, Mudassar Rauf. "Multi-Level Planning and Scheduling for Parallel PCB Assembly Lines Using Hybrid Spider Monkey Optimization Approach", IEEE Access, 2019 Crossref	13 words — < 1%
16	Sharples, J.J.. "A simple index for assessing fuel moisture content", Environmental Modelling and Software, 200905 Crossref	12 words — < 1%
17	Guanghui Zhang, Keyi Xing, Guangyun Zhang, Zhenxue He. "Memetic Algorithm With Meta-Lamarckian Learning and Simplex Search for Distributed Flexible Assembly Permutation Flowshop Scheduling Problem", IEEE Access, 2020 Crossref	11 words — < 1%
18	hrcak.srce.hr Internet	11 words — < 1%
19	core.ac.uk Internet	10 words — < 1%
20	Assembly Automation, Volume 31, Issue 3 (2011-08-06) Publications	9 words — < 1%

21	Fuqing Zhao, Yuebao Liu, Ningning Zhu, Tianpeng Xu, Jonrinaldi. "A selection hyper-heuristic algorithm with Q-learning mechanism", Applied Soft Computing, 2023 Crossref	9 words — < 1%
22	Junyong Liang, Shunsheng Guo, Baigang Du, Wenfang Liu, Yunfei Zhang. "Restart genetic flatworm algorithm for two-sided disassembly line balancing problem considering negative impact of destructive disassembly", Journal of Cleaner Production, 2022 Crossref	9 words — < 1%
23	Kang, Qinma, Hong He, and Rong Deng. "Bi-objective task assignment in heterogeneous distributed systems using honeybee mating optimization", Applied Mathematics and Computation, 2012. Crossref	9 words — < 1%
24	Xuesong Yan, Hao Zuo, Chengyu Hu, Wenyin Gong, Victor S. Sheng. "Load Optimization Scheduling of Chip Mounter Based on Hybrid Adaptive Optimization Algorithm", Complex System Modeling and Simulation, 2023 Crossref	9 words — < 1%
25	Yarong Chen, Jingyan Zhong, Jabir Mumtaz, Shengwei Zhou, Lixia Zhu. "An improved spider monkey optimization algorithm for multi-objective planning and scheduling problems of PCB assembly line", Expert Systems with Applications, 2023 Crossref	9 words — < 1%
26	archive.org Internet	9 words — < 1%
27	scholars.ttu.edu	

Internet

9 words — < 1%

28 vtechworks.lib.vt.edu
Internet

9 words — < 1%

29 www.hindawi.com
Internet

9 words — < 1%

30 "Bio-Inspired Computing: Theories and Applications", Springer Science and Business Media LLC, 2024
Crossref

8 words — < 1%

31 "Handbook of Heuristics", Springer Science and Business Media LLC, 2018
Crossref

8 words — < 1%

32 "Simulated Evolution and Learning", Springer Science and Business Media LLC, 2017
Crossref

8 words — < 1%

33 Debiao Li, Liting Wang, Qingxian Huang. "A case study of SOS-SVR model for PCB throughput estimation in SMT production lines", 2019 International Conference on Industrial Engineering and Systems Management (IESM), 2019
Crossref

8 words — < 1%

34 Huijun Gao, Chao Ye, Weiyang Lin, Jianbin Qiu. "Complex Workpiece Positioning System With Nonrigid Registration Method for 6-DoFs Automatic Spray Painting Robot", IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2020
Crossref

8 words — < 1%

35 Jin Yu, Xiaoming You, Sheng Liu. "Dynamically induced clustering ant colony algorithm based on

8 words — < 1%

a coevolutionary chain", Knowledge-Based Systems, 2022
Crossref

36 Konstantinos P. Anagnostopoulos, Georgios K. Koulinas. "A genetic hyperheuristic algorithm for the resource constrained project scheduling problem", IEEE Congress on Evolutionary Computation, 2010 — 8 words — < 1%
Crossref

37 Like Zhang, Qianwang Deng, Zhen Wang, Guiliang Gong, Xiaoyu Wen, Xiaoyan Liu. "Collaborative scheduling of production resources and spare parts inventory for distributed equipment with feedback guidance and minimum capacity loss", Swarm and Evolutionary Computation, 2022 — 8 words — < 1%
Crossref

38 Osman Kulak. "A GA-based solution approach for balancing printed circuit board assembly lines", OR Spectrum, 06/2008 — 8 words — < 1%
Crossref

39 Shanshan Zha, Hao Shao, Wenzhong Zou, Luyang Li, Yiqun Chu. "Surface Mounted Production Line Capacity Prediction Based on Data Driven", Journal of Physics: Conference Series, 2023 — 8 words — < 1%
Crossref

40 Tansel Dokeroglu, Tayfun Kucukyilmaz, El-Ghazali Talbi. "Hyper-heuristics: A survey and taxonomy", Computers & Industrial Engineering, 2024 — 8 words — < 1%
Crossref

41 Yi-Jian Wang, Juan Li, Gai-Ge Wang. "Fuzzy correlation entropy-based NSGA-II for energy-efficient hybrid flow-shop scheduling problem", Knowledge-Based Systems, 2023 — 8 words — < 1%
Crossref

42	Ying Hou, Yilin Wu, Honggui Han. "Multi-Objective Differential Evolution Algorithm Balancing Multiple Stakeholders for Low-Carbon Order Scheduling in E-Waste Recycling", IEEE Transactions on Evolutionary Computation, 2023 Crossref	8 words — < 1%
43	Zeliang Ju, Yan Wang, Zhen Quan, Xiang Liu, Zhicheng Ji. "Bottleneck alleviation and scheduling optimization of flexible manufacturing system based on information-energy flow model", Swarm and Evolutionary Computation, 2024 Crossref	8 words — < 1%
44	april.zju.edu.cn Internet	8 words — < 1%
45	bura.brunel.ac.uk Internet	8 words — < 1%
46	irep.ntu.ac.uk Internet	8 words — < 1%
47	mdpi-res.com Internet	8 words — < 1%
48	pdffox.com Internet	8 words — < 1%
49	pdfs.semanticscholar.org Internet	8 words — < 1%
50	www.scilit.net Internet	8 words — < 1%
51	Lecture Notes in Computer Science, 2016. Crossref	7 words — < 1%

52 Shujuan Guo, Katsuhiko Takahashi, Katsumi Morikawa. "PCB assembly scheduling with alternative nozzle types for one component type", Flexible Services and Manufacturing Journal, 2011
Crossref
7 words — < 1%

53 Debiao Li, Siping Chen, Raymond Chiong, Liting Wang, Sandeep Dhakal. "Predicting the printed circuit board cycle time of surface-mount-technology production lines using a symbiotic organism search-based support vector regression ensemble", International Journal of Production Research, 2020
Crossref
6 words — < 1%

54 www.utupub.fi
Internet
6 words — < 1%

EXCLUDE QUOTES	OFF	EXCLUDE SOURCES	OFF
EXCLUDE BIBLIOGRAPHY	ON	EXCLUDE MATCHES	OFF