Flexible worker allocation in aircraft final assembly line using multi-objective evolutionary algorithms

Pengcheng Fang, Jianjun Yang, Qingmiao Liao, Ray Y Zhong, *Member, IEEE*, and Yuchen Jiang

***Abstract*—In a paced aircraft final assembly line, some disturbances can be collected timely on the basis of the Cyber-Physical production system (CPPS). In order to reduce the execution deviation, some workers need to switch among stations after a fixed period. Thus, a worker allocation problem with the multi-stage workstation is introduced firstly. Then an integer programming formulation is presented to formulate the problem with the objective of shortest workstation cycle and the workload balance of both stations and workers. Moreover, a modified non-dominated sorting genetic algorithm (NSGA-IV) is proposed to solve it, which trades off the convergence and the population diversity in the decision space. Finally, the NSGA-IV algorithm compares with five multi-objective evolutionary algorithms (MOEA) in a real-world case. Compared to manual allocation, the takt time of an aircraft final assembly line is reduced by 20.86% by using the NSGA-IV algorithm.**

***Index Terms*—Worker allocation, aircraft final assembly line, multi-stage workstation, multi-objective evolutionary algorithm, Cyber-Physical production system**

NOTATIONS

Indices

i	the time grid index, $i \in I$
I	set of time grids
j	the detailed task, $j \in J$
J	set of detailed tasks
k	the stage index, $k \in K$
K	set of stages
m	the station index, $m \in M$
M	set of stations
w	the worker type index, $w \in W$
W	set of worker types

Manuscript received October 17, 2020; revised November 21, 2020; accepted January 5, 2021. This work was supported in part by the National High Technology Research and Development Programme (863) of China under Grant 2012AA040907. Paper no. TII-20-4794. (Corresponding author: Pengcheng Fang.)

P.C. Fang, J.J. Yang are with the School of Mechanical Engineering & Automation, Beihang University,100191, China. (e-mail: by1507118@buaa.edu.cn; jjyang@buaa.edu.cn).

Q. M. Liao is with the AVIC digital corporation LTD., Beijing 100028, China (e-mail: liaoqingmiao@gmail.com).

R. Y. Zhong is with the Department of Industrial and Manufacturing Systems Engineering, The University of Hong Kong, Hong Kong, China (e-mail: zhongzry@hku.hk).

Yuchen Jiang is with the School of Cyber Security, University of Chinese Academy of Sciences; State Key Laboratory of Information Security, Institute of Information Engineering, Beijing, China (e-mail: jiangyuchen@iie.ac.cn).

Decision variables

s_j	the plan start time of task j
f_j	the plan finish time of task j
x_{w,m_k}	the number of worker type w assigned to stage k of station m

Parameters

a_{jw}	1, if task j needs to be processed by workers w
b_{jm}	1, if task j needs to be processed in station m
e_{iw}	1,if the workers w is at work on the time grid i
n_j	the number of workers used for processing task j
n_w	the total number of worker type w
p_j	the processing time of task j
t_i	the start time of time grid i
u_{ik}	1, if the time grid i is in the stage k
v_{ij}	1, if $t_i \in [s_j\ ,\ f_j)$
r^i_{mw}	the remaining number of worker type w assigned to station m at time grid i, $r^i_{mw} \in R_{mw}$
Δt	the time interval of each grid
DS_q	the decision set of the q-th iteration
N	population size
NFE	the number of fitness evaluations
PT_j	the predecessor task set of a task j
SS_q	the scheduled set of the q-th iteration
ST_j	the successor task set of a task j

I. INTRODUCTION

TYPICAL features of aircraft final assembly line are mixed-model production, paced line with fixed takt time, and multi-manned assembly at every workstation. A mixed-model line means manufacturing several types of aircraft from a basic product family simultaneously. Each aircraft is slightly different in some attributes and optional features compared to the basic product. Thus, the production process of every aircraft is quite similar. Furthermore, each aircraft visits a series of workstations in sequence and is simultaneously moved to the next stations at the end of each takt time, considering the spatial constraints. The takt time is equal to a maximum cycle for executing all tasks processed by workers assigned to every station [1]. Moreover, the simultaneous operation of more than one worker at every workstation describes as the multi-manned assembly line [2]. In labor-intensive manufacturing, these workers assigned to each station perform a larger number of tasks for every cycle. However, there exist a lot of disparate tasks in a station for different cycles.

A limited amount of human resources needs to be assigned among the stations under satisfying the precedence relations and human resources constraints [3], called the worker allocation problem. In order to flexibly allocate workers, each workstation is divided into multiple stages on the time dimension. In every stage, every worker is assigned to a station. Thus, employees can switch among the stations at the end of each stage, which can contribute to resisting some disturbances that occurred in the final assembly line. In addition, the multi-stage worker allocation strategy is more practical, and the performance is better than that of the single-stage worker allocation strategy. Specifically, this study illustrates that all workers are flexibly assigned to multi-stage stations.

The multi-objective evolutionary algorithms (MOEA) have gradually been employed to deal with the scheduling problem with several conflicting objectives [4]. Due to their outstanding performance, the improved non-dominated sorting genetic algorithm (NSGA-II) and NSGA-III are widely used [5]. Deb et al. proposed NSGA-II according to the non-dominated sorting and crowding distance sorting. Subsequently, they put forward the NSGA-III [6] using the reference-point-based non-dominated sorting to solve a many-objective problem. But there exist two flaws in the NSGA-II and the NSGA-III. On one hand, they are easy to become stuck in a locally optimal solution and trap in premature convergence. The two MOEAs focus more on the convergence and rarely consider the population diversity, especially when the number of solutions in the last front is far less than the population size (N). On the other hand, it is difficult to find diversified solutions while some solutions associated with different decision variables have almost the same objective function value. It's mainly because crowded-comparison and reference-point-based methods rank the solutions of the last front in objective space rather than in decision space. To deal with these defects, a modified non-dominated sorting genetic algorithm (NSGA-IV) is proposed.

The innovative of this paper are as follows. Firstly, a systematic framework on the basis of the Cyber-Physical Production Systems (CPPS) is put forward to ensure the production of a real aircraft final assembly line with a stable takt time. Secondly, an integer programming formulation is presented to formulate the worker allocation problem with the multi-stage workstation. Thirdly, an NSGA-IV is proposed to trade off the convergence and the population diversity in the decision space. As a result, the on-site continuous optimization of the human resource allocation ensures a stable production process.

The rest of the paper is organized as follows. Section II gives a literature review about the CPPS, the worker allocation problem, and the solution methods. Section III briefly describes the system architecture and the formulation of the problem models. The proposed NSGA-IV algorithm is introduced in Section IV. In Section V, a case study is given to validate the effectiveness of the proposed algorithm. Section VI concludes the paper and introduces the future work.

II. Literature Review

A. Cyber-Physical Production Systems

The CPPS is an emerging paradigm for addressing the requirements of future production systems [7]. The tasks scheduling scheme and the worker allocation scheme need to be generated and published. The CPPS can streamline the decision-making process, allowing flexible production lines [8]. Additionally, multiple kinds of disruptions often occur in the final assembly line, including tasks tardiness, equipment fault, assembly failure, staff absenteeism, and instability of a global supply chain [9]. According to our previous study [10], the CPPS can capture these disturbances in time. To address them, the final assembly line needs to be re-balanced periodically after the reality production deviates from the plan. Thus, the running final assembly line can keep rhythm with a takt time.

There are three types of adjustment strategies for processing uncertain events, namely completely reactive scheduling, predictive-reactive scheduling, and robust pro-active scheduling [11]. Completely reactive scheduling is made locally in real-time, which is applied to the production with a high automation level. The predictive-reactive scheduling revises schedules in response to real-time events [11]. But this method may lead to a significant deviation between the original schedule and the new schedule. Besides, robust pro-active scheduling builds predictive schedules under satisfactory performance requirements in a dynamic environment with several uncertainties. Yet it is difficult to determine the predictability measures.

B. Worker allocation problem

All tasks for a new aircraft are allotted to each workstation while satisfying the precedence constraints [12], which is the assembly line balancing problem. A large number of tasks often shifts among the whole stations, resulting in the workers' movement in the multi-manned assembly line frequently. Thus, workers need to be assigned to every station at the subsequent decision stage [13]. It is known to be NP-hard in the strong sense.

Multiple variants of worker allocation problems have been studied in many works, as shown in Table I. Sikora et al. [15] developed a mixed-integer linear programming model to solve the traveling worker assembly line balancing problem. Thus, all tasks and these workers are assigned to stations considering movement times. But the order of performing tasks in every station is neglected. Battaïa et al. [1] proposed eight heuristics methods to balance the workload among workstations in a mixed-model assembly line, considering the workers' movement between the stations at the end of each task. Yet, it is difficult to frequently swap workers among stations in an aircraft final assembly line. Heike et al. [14] developed a linear and two nonlinear programs for evaluating worker allocation scheme in low-volume aircraft manufacturing environments. However, optimization algorithms are not discussed. Subsequently, Biele et al. [16] continued to solve the worker assignment problem by a heuristics that hybridize the mathematical formulations with variable neighborhood search techniques in the aircraft manufacturing domain. But both objective functions are aggregated into a single objective to simplify the problem. By using a proposed meta-heuristic algorithm based on NSGA-II, Lian et al. [17] dealt with the

TABLE I
RESEARCH HIGHLIGHTS ON WORKER ASSIGNMENT PROBLEM.

Year	Production environment	Proposed approach	Strengths	Weaknesses	Success rate	Ref
2001	Mixed model assembly in aerospace industry	One linear and two non-linear programs	Crew is flexibly allocated among stations.	Optimization methods were not used.	Using flexible crews can reduce overtime by as much as 24%.	[14]
2015	Automotive-assembly line	Eight constructive heuristic methods	Workers can switch between the stations at the end of each task.	Too often swap workers among stations.	The maximal relative error gap is 66.7% by using sequential-station heuristic.	[1]
2015	Aircraft assembly lines	Adaptive binary particle swarm optimization algorithm	The cooperation of multi-skilled workers and their skill level are considered.	The past positions are neglected while updating the position now.	The maximum human cost value is 5.6 per cent lower than that of particle swarm optimization.	[3]
2017	Mixed-model assembly line	Mixed-integer linear programming algorithm	Sparse sets are utilized to reduce search space.	The order of performing tasks is ignored.	Improvements of 11.3%, 9.4%, and 12.7% in cycle times.	[15]
2018	Aircraft assembly lines	Variable neighborhood search techniques	Two mathematical formulations are proposed.	Both objective functions are aggregated.	Improvements of up to 11% are possible in a real case.	[16]
2018	Seru production systems	Meta-heuristic algorithm based on NSGA-II	Multi-skilled worker is considered.	Without considering workers' maturity	Heterogeneous workers perform well in balancing inter-seru workload.	[17]

multi-skilled worker assignment problem without considering workers' maturity. In addition, Xin et al. [3] proposed an adaptive binary particle swarm optimization algorithm to investigate the multi-skilled worker assignment problem.

C. Solution methods

As a combinatory optimization problem, the worker allocation problem is solved by using exact and approximate methods [18], as illustrated in Fig. 1. The exact methods contain mathematical programming and enumerative algorithms. For the large-sized instances, exact solutions cannot be provided in a reasonable time [19]. In addition, the approximate methods could be classified as heuristic algorithms, meta-heuristics algorithms, and artificial intelligence algorithms. The heuristic methods only get a feasible solution with general performance in a short time period [20], which are widely utilized in practical production. However, the meta-heuristics algorithms have better results than the heuristic algorithms by sacrificing computing time [21]. For solving the multi-objective problem, the meta-heuristics algorithms are often used by constructing Pareto frontier [22]. For instance, Zhang et al. proposed an improved Nondominated Sorting Genetic Algorithm-II algorithm to optimize the manufacturing service composition [23]. Besides, some artificial intelligence algorithms are a very promising area. One of them is the neural network algorithm. Although the solution quantity is between that of heuristics and meta-heuristics, neural network algorithm takes a polynomial time in terms of problem size [24]. Thus it satisfies the requirement of some real-time scheduling problems.

In summary, there remains a considerable gap between the production practice and the academic research in aircraft final assembly line, as follows.

- The investigation on the worker allocation considering their movements in the multi-manned assembly line is limited. However, frequent switching workers among stations leads to management difficulties and production chaos on one hand, and no switching workers results in the low utilization efficiency of workers on the other hand. Thus, allocating workers to the multi-stage workstation is worthy of further research.

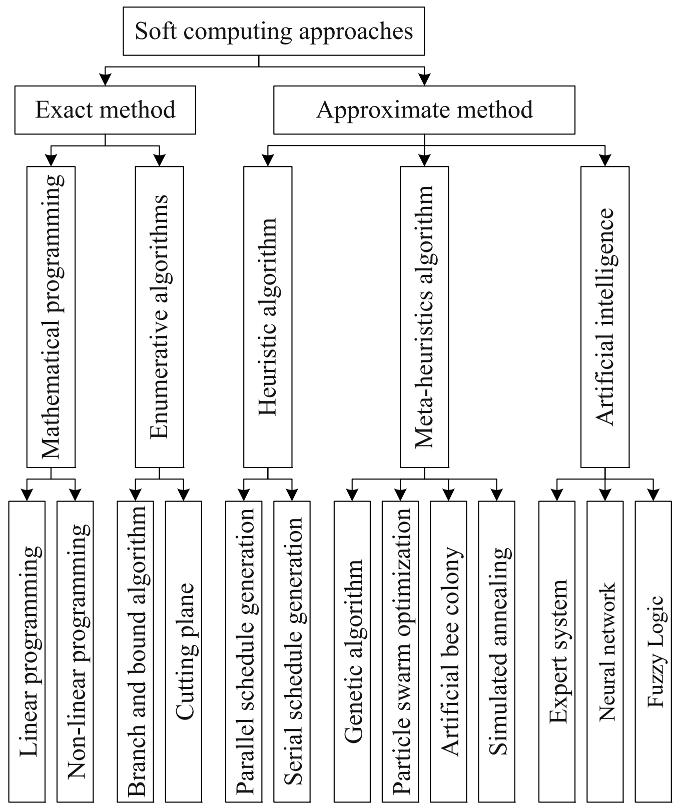

Fig. 1. Classification of soft computing approaches.

- Although the application of NSGA-II is widely used, it focuses too much on convergence and not enough on population diversity. Thus, it makes sense for studying the trade-off between both sides in the decision space.

III. PROBLEM DESCRIPTION AND FORMULATION

A. Systematic architecture on the basis of CPPS

A systematic framework on the basis of CPPS is introduced to realize flexible worker allocation in an aircraft final assembly line, as shown in Fig. 2. The top two layers are deployed to cyberspace in the cloud platform, whereas the bottom two layers are installed in the physical world. A sequential workflow is presented to clearly defines how to construct a closed-loop scheduling system [25]. The detailed CPPS architecture is outlined as follows:

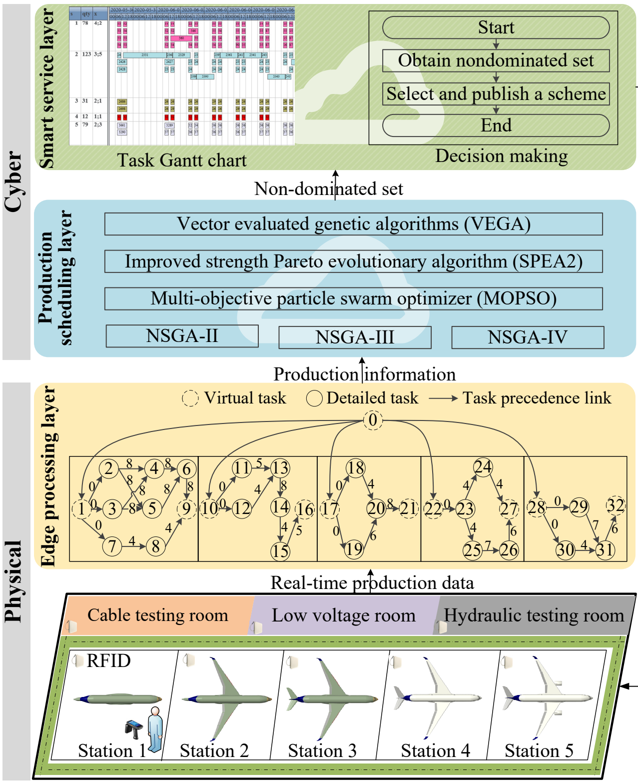

Fig. 2. Systematic architecture.

1) Production assembly and smart connection: The build-up of the aircraft is completed in a streamlined workflow that moves in steps through five stations. Each station installs several specific components, such as joining three fuselage sections in the first workstation. Besides, instead of manual recording, the workshop deploys variable kinds of sensors such as the RFID readers and the mobile terminal equipment. Thus, real-time production data can be captured on-site.
2) Edge processing: In this layer, a large quantity of raw data is captured and addressed into the production information. Therefore, managers can monitor the status of production tasks in time. The deviation between the real world and the virtual space can detect.
3) Production scheduling: By leveraging some multi-objective evolutionary algorithms, some work allocation schemes from the Pareto frontier are obtained. In addition, the procedure of the NSGA-IV is detailed in section IV-A.
4) Smart service: Production monitor and decision making are implemented. Some visual interfaces are shown, such as the task Gantt chart. Thus, every worker clearly learns the production status and prospective tasks. Besides, the manager selects and publishes a worker allocation scheme. All unfinished tasks are executed according to the scheme.

B. Problem statement

We focus on assigning workers to the multi-stage stations, yet the task assignment to workstations is ignored. An aircraft final assembly line consists of five stations $m(m \in M)$, each of which has multiple stages in terms of the time dimension. There are multiple types of workers in the final line, and the total number of each worker type $w(w \in W)$ is constant, denoted with n_w. Each type of worker is identical who has a single skill. For every stage $k(k \in K)$, all workers are partitioned into five subsets associated with stations. These workers can stay in a station for an appropriate duration to assemble tasks. After finishing a stage, some workers exchange among stations without considering movement times.

Assembling an aircraft requires partitioning the total amount of work into a set of elementary tasks. The assembly tasks and the precedence between them in each station are known and cannot be modified. There are two types of tasks, i.e., virtual tasks and detailed tasks, which are visualized in a precedence graph (Fig. 2). The processing time of each virtual tasks is set to 0 without any worker. Performing a detailed task $j(j \in J)$ takes a deterministic processing time p_j and requires n_j special w-type of workers in the station m, but regardless of the transportation time. Additionally, the start time s_0 of the initial task0 is set to the scheduling start time. The task precedence constraints are represented by the task precedence links. Thus, all direct predecessor tasks of a detailed task j are grouped into the predecessor task set PT_j, and so do the immediate successor task set ST_j.

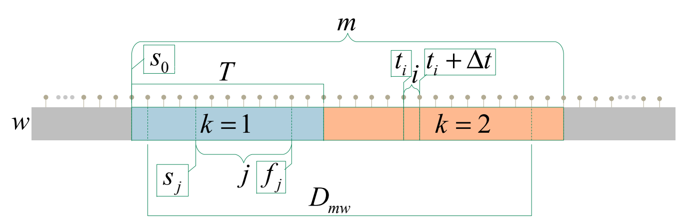

Fig. 3. A discrete time-grid model of worker type w on station m.

As can be seen in Fig. 3, the timeline of each worker type on every station is divided into many discrete-time grids with a fixed interval of Δt. For example, the 500-hours period is split into 500 grids with an interval of $\Delta t = 1$ hour. A time-grid $i(i \in I)$ has a start time of t_i. In addition, each timeline of every workstation is split into multi-stage. Except for the last stage, the time length of k stages are equal. For instance, there are two stages in this study and the first stage $k = 1$ contains 200 time-grids, while another stage includes 300 time-grids. There are three decision variables, namely x_{w,m_k}, the plan start time s_j and the plan finish time f_j of each task j. The first variable represents the number of worker type w assigned to the stage k of station m.

C. Mathematical model

The worker allocation problem with the multi-stage workstation can be formulated as the following integer programming model, as follows:

$$\min f_1 = \max_{m\in M, w\in W} D_{mw} \tag{1}$$

$$\min f_2 = \sqrt{\frac{1}{|M|}\sum_{m\in M}\left(\max_{w\in W} D_{mw} - \frac{1}{|M|}\sum_{m\in M}\max_{w\in W} D_{mw}\right)^2} \tag{2}$$

$$\min f_3 = \max_{w\in W}\sqrt{\frac{1}{|M|}\sum_{m\in M}\left(D_{mw} - \frac{1}{|M|}\sum_{m\in M} D_{mw}\right)^2} \tag{3}$$

Subject to:

$$D_{mw} = \max \left\{a_{jw}\cdot b_{jm}\cdot f_j \,|j\in J\right\} - \min \left\{a_{jw}\cdot b_{jm}\cdot s_j \,|j\in J\backslash 0\right\}, \forall m\in M, \forall w\in W \tag{4}$$

$$t_i = s_0 + (i-1)\cdot \Delta t, \forall i\in I \tag{5}$$

$$p_j = \sum_{i\in I}\sum_{w\in W} a_{jw}\cdot e_{iw}\cdot v_{ij}\cdot \Delta t, \forall j\in J \tag{6}$$

$$n_w = \sum_{m\in M} x_{w,m_k}, \forall w\in W, \forall k\in K \tag{7}$$

$$x_{w,m_k} \geq \max_{j\in J} a_{jw}\cdot b_{jm}\cdot n_j, \forall w\in W, \forall m\in M, \forall k\in K \tag{8}$$

$$s_j \geq \max_{j^*\in PT_j} \left\{f_{j^*}\right\}, \forall j\in J \tag{9}$$

$$\sum_{w\in W}\sum_{m\in M} a_{jw}\cdot\ b_{jm} = 1, \forall j\in J \tag{10}$$

$$|v_{i-1,j} - v_{ij}| = \begin{cases} 1, & \text{if } \ (t_i = f_j || t_i = s_j) \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

$$a_{jw}\in\{0,1\}, \forall w\in W, \forall j\in J \tag{12}$$

$$b_{jm}\in\{0,1\}, \forall m\in M, \forall j\in J \tag{13}$$

$$e_{iw}\in\{0,1\}, \forall w\in W, \forall i\in I \tag{14}$$

$$u_{ik}\in\{0,1\}, \forall k\in K, \forall i\in I \tag{15}$$

$$v_{ij}\in\{0,1\}, \forall i\in I, \forall j\in J \tag{16}$$

Formulas (1) is minimizing the maximal workstation cycle (MWC). It directly affects the takt time, which determines the efficiency of the assembly line [13]. Formulas (2) expresses that the deviation of the workstation cycle (DWC) is minimized. The DWC balances stations by calculating the standard deviation of workstation processing times. Formulas (3) denotes minimizing the maximal deviation of worker workload (MDPW). The worker workload is set forth to establish the total processing time for each worker type. Thus, the last two objectives aim at reducing the imbalance among stations and work types separately [18]. Equation (4) states the duration of all tasks processed by worker type w in the station m. Equation (5) computes that the start time of every time grid i. Equation (6) ensures that the assembly time of task j is fully processed by specified workers at the time interval. Equation (7) denotes that all w-type workers are allocated to five workstations on each stage k. Constraint (8) is applied to enforce human resource constraints where a lower bound on each type of available worker is imposed on every station [16]. Constraint (9) follows task precedence constraints that the start time of task j is later than the finish time of any direct predecessor task [26]. Constraint (10) makes sure that each task can only be processed by a type of worker in a fixed station. Constraint (11) guarantees that preemption is not allowed. Constraint (12)- (16) illustrate five binary variables. Constraint (12) determines whether task j needs to be processed by workers w. If so, the value of a_{jw} is 1. Otherwise, a value of 0 is returned. Constraint (13) expresses that the value b_{jm} is 1 if task j needs to be processed in station m, otherwise it is 0 [15]. Constraint (14) defines whether the worker type w is at work on the time grid i. If so, the value e_{iw} is 1. The value is calculated in terms of the working calendar. Constraint (15) implies whether the time grid i is in the stage k. If $(k < |K| \& t_i \in [s_0 + T(k-1), s_0 + T\cdot k))$ or $(k = |K| \& t_i \in [s_0 + T(k-1), +\infty))$ is satisfied, the value of u_{ik} is 1. Constraint (16) describes whether the time grid i falls within the range of plan assembly time of task j. The value of v_{ij} is 1 while $t_i \in [s_j, f_j)$.

IV. Algorithm Design

In this section, an NSGA-IV is introduced to solve the worker allocation problem. First, the key steps of the algorithm are also detailed. Second, the encoding scheme is described. The final part of this chapter describes the fitness evaluation procedure.

A. Algorithm framework

Fig. 4 shows a graphical interpretation of the NSGA-IV. There are three main steps to select the next generation in NSGA-IV. Firstly, the parents and offsprings are divided into three groups in terms of the results of the non-dominant sorting, namely Q1, Q2, and Q3. The number of individuals in Q1 is not more than half of the population size, whereas the quantity of solutions in Q2 is not less than the population size. Secondly, hierarchical clustering is utilized for the group Q2. The closest individuals are removed from Q2 until the number of Q1 and Q2 is the same as the population size. Finally, all the individuals in Q1 are directly incorporated into the offspring to ensure fast convergence. The remaining individuals in Q2 are accommodated in the next population to guarantee population diversity. Besides, the rest of the individuals, belonging to Q3, are removed directly.

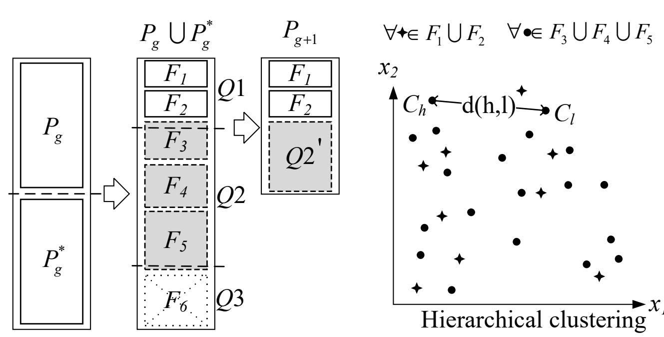

Fig. 4. NSGA-IV's schematic.

The main procedure of the NSGA-IV is detailed in Algorithm 1.

1) The diversified initial population P_0 is randomly generated to start the evolution.

Algorithm 1 Pseudo-code of NSGA-IV

Input: P_0

Output: $P_{(NFE/N)}$

```
1: while g = 1 to NFE/N do
2:     P*_g = Make-new-population(P_g)
3:     Fitness-evaluation (P*_g)
4:     {F_1, F_2, ..., F_q, ...} = Nondominated-sort(P_g ∪ P*_g)
5:     q = 1, Q1_g = ∅, and Q2_g = ∅
6:     while |Q1_g| + |F_q| ≤ 0.5N do
7:         Q1_g = Q1_g ∪ F_q
8:         q++
9:     end while
10:    do
11:        Q2_g = Q2_g ∪ F_q
12:        q++
13:    while |Q1_g| + |Q2_g| ≤ 1.5N
14:    P_{g+1} = Hierarchical-clustering(Q1_g ∪ Q2_g)
15:    g++
16: end while
```

2) After performing the evolution process of binary tournament selection, the simulated binary crossover (SBX) and the polynomial mutation (PM), some new offspring P_g^* are generated at line 2.
3) At line 3, every individual in the offspring is decoded. Then, the fitness evaluation is executed to obtain the multi-objectives of each individual.
4) All individuals in the P_g and P_g^* is divided into multiple sets on the basis of the non-dominant sorting at line 4.
5) All individual in every set F_q is allocated to three groups in sequence from line 5 to line 13. The third group is deleted directly.
6) At line 14, the next population P_{g+1} is generated by removing the worst member in the group Q2 on the basis of the hierarchical clustering.
7) The algorithm continues with step 2 until the number of iterations exceeds the maximum number of iterations (NFE/N). The NFE represents the number of fitness evaluations.

B. Hierarchical clustering

In this study, a modified bottom-up hierarchical clustering is presented to remove redundant individuals in group Q2. The pseudo-code in Algorithm 2 illustrates the procedure.

$$d\left(C_h, C_l\right)=\sqrt{\sum_{k,m,w}\left(x_{w,m_k}^{(h)}-x_{w,m_k}^{(l)}\right)^2}, \forall C_h, C_l \in C \quad (17)$$

1) Line 1 reports that each individual in Q1 and Q2 is divided into a cluster.
2) Line 2 represents the Euclidean distance matrix in which any value represents a Euclidean distance of two individuals in set C. The Euclidean distance is calculated by Formula 17.
3) On the basis of the Euclidean distance matrix, two nearest clusters in which at least one cluster belongs to $Q2_g$ are found in line 4.

Algorithm 2 Pseudo-code of hierarchical clustering

Input: $Q1_g, Q2_g$

Output: $Q1_g \cup Q2_g\left[1:\left(N-\left|Q1_g\right|\right)\right]$

```
1: C = {C_1, C_2, ..., C_q, ...}, ∀C_q ∈ Q1_g ∪ Q2_g
2: Calculate-Euclidean-Distance-Matrix (ED_{|C|×|C|})
3: while |C| > N do
4:     Find-nearest-clusters(C_h, C_l), C_h ∈ C, C_l ∈ Q2_g
5:     if C_h ∈ Q1_g & C_l ∈ Q2_g then
6:         remove C_l from C
7:     else if C_h ∈ Q2_g & C_l ∈ Q2_g then
8:         o = argmin_{o∈{h,l}, q∈C\{C_h∪C_l}} d(C_o, C_q)
9:         remove C_o from C
10:    end if
11:    Update ED_{|C|×|C|}
12: end while
```

4) Algorithm 5-10 lines show that a cluster in $Q2_g$ is removed. If only one of the two nearest clusters belongs to set Q1, another is deleted. Once both clusters are from set Q2, the cluster that is closer to other clusters will be remove.
5) The Euclidean distance matrix is updated. The hierarchical clustering method goes back to step 3 until the remaining quantity of all clusters is equal to the population size.

The non-dominated sorting (line 4 in Algorithm 1) of a population size 2N having M-dimensional objective vectors require $O\left(N \log^{M-2} N\right)$ computations [6]. In the worst case scenario, the number of the last non-dominated set exceeds a half of the population size; that is $|Q3|=0$. Therefore the time complexity of a hierarchical clustering is $O\left(N^3\right)$. Thus, the overall worst-case complexity of one generation of NSGA-IV is $O\left(N^4\right)$.

C. Solution Encoding

An integer encoding scheme is presented in a series of the integer decomposition codes. Each code $c_{w,k}$ denotes the allocation scheme of worker type w in station k. The length of an individual is equal to $|W| \times |K|$.

13	121	165	162	1	1	...	$c_{w,k}$	...	993	555

Fig. 5. Solution encoding.

In order to satisfy Equation (7), the number of each worker type is decomposed into five (M=5) integers, which represent the allocation result of the worker type in five stations successively. Besides, the allocation schemes of each worker type are encoded from 1 to the number of decomposition solutions. Thus, the number of the worker type w in stage k of every station can be retrieved by the unique code $c_{w,k}$. For example, the value of the first gene is 13 in Fig. 5. By retrieving in the Table II, the third station and the fourth station assign two workers and six workers belonging to the first worker type, respectively. Yet each of the other stations allocates a worker of the first type.

TABLE II
INTEGER DECOMPOSITION TABLE OF FIRST WORKER TYPE($\boldsymbol{w}=\mathbf{1}$).

$c_{1,k}$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	...	210
$x_{1,1_k}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	...	7
$x_{1,2_k}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	...	1
$x_{1,3_k}$	1	1	1	1	1	1	1	2	2	2	2	2	2	3	3	3	3	...	1
$x_{1,4_k}$	1	2	3	4	5	6	7	1	2	3	4	5	6	1	2	3	4	...	1
$x_{1,5_k}$	7	6	5	4	3	2	1	6	5	4	3	2	1	5	4	3	2	...	1

Workstations in the final assembly line are aligned as shown in Fig. 6 [13]. Each aircraft is moved from the first station to the fifth workstation in sequence. Additionally, all kinds of assembly parts are delivered to the corresponding station continuously. Moreover, all workers are assigned to every multi-stage workstation. Thus, the worker allocation scheme depicted in Fig. 5 is decoded and shown in Fig. 6. For each worker allocation scheme, each line signifies workers assigned to the corresponding stage, and each column represents the corresponding worker type with different colors. For example, the number of the first worker type assigned to the first stage can be shown on the top left corner in every station.

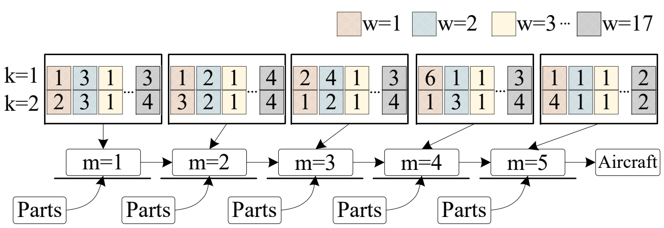

Fig. 6. Worker allocation in multi-stage workstation.

D. A serial schedule generation scheme with working calendar

For solving the evaluation problem, a serial schedule generation scheme (SSGS) with a working calendar is introduced. Only one task is assigned for each iteration under satisfying task precedence constraint and human resource constraint [26]. Associated with each iteration are two disjoint task sets, namely the scheduled set SS_q and the decision set DS_q. The scheduled set comprises all tasks that have been already scheduled, whereas the decision set contains all tasks, each of which can be assigned at this iteration. In addition, let r^i_{mw} be the remaining capacity of worker type w in station m at time-grid i. And R_{mw} represents the set of r^i_{mw} in the whole-time grids. The procedure of the fitness evaluation is shown in Algorithm 3.

1) At line 1, the initialization puts the virtual assembly task0 into the scheduled set SS_0. The remaining capacity R_{mw} is initialized by setting each r^i_{mw}.
2) The decision set DS_q is calculated according to the scheduled tasks and the task precedence constraints.
3) At line 4, one task j is selected from the decision set based on six priority rules, which are maximum total successors, longest processing time, most immediate successors, shortest processing time, maximum critical, and random rule, in sequence.

Algorithm 3 Pseudo-code of SSGS with working calendar.

Input: $s_0, \{x_{w,m_k} | w \in W, m \in M, k \in K\}$
Output: $\{(s_j, f_j) | j \in J\}$
1: **Initialization** $SS_0 = \{0\}, DS_0 = \varnothing$, $R_{mw} = \left\{r^i_{mw} \middle| r^i_{mw} = \sum_{k \in K} x_{w,m_k} \cdot e_{iw} \cdot u_{ik}, i \in I, w \in W, m \in M\right\}$
2: **while** $q = 1$ to $|J|$ **do**
3: Calculate $DS_q = \{j | (j \notin SS_{q-1}) \cup (PT_j \subset SS_{q-1})\}$
4: Select $j \in DS_q$ according to priority rules
5: Calculate $s_j = \max_{j^* \in PT_j} \{f_{j^*}\}$
6: Calculate f_j according to Equation (6)
7: Update $R_{mw} = \left\{r^i_{mw} \middle| r^i_{mw} -= a_{jw} \cdot b_{jm} \cdot v_{ij} \cdot n_j, i \in I\right\}$, $\forall w \in W, \forall m \in M$
8: Update $SS_{q+1} = SS_q \cup \{j\}$
9: **end while**

4) The plan start time of the task j is set to the latest time of the direct predecessor tasks.
5) At line 6, the plan finish time of the task j is calculated by backward accumulation from the plan start time until the total time of the worker-feasible grid equals processing time.
6) The remaining capacity R_{mw} is updated by subtracting the used number of workers in each grid.
7) The task j is added to the scheduled set. The procedure continues to go to step 2 until all tasks are scheduled.

V. CASE STUDY

The overall experiment process is described as follows: Firstly, the production data from a real aircraft final assembly line is preprocessed. Secondly, the basic parameters of algorithms are determined. Thirdly, three performance metrics and the Wilcoxon signed-rank test are utilized to compare the NSGA-IV with five MOEAs. Fourthly, it describes how to apply the process in practice. Finally, we discuss and analyze the application of the NSGA-IV in the worker allocation problem. The experiments were implemented in Java 8.0 based on the Eclipse platform with 6 threads and run on a computer with an Intel Core i5-8500 3.0 GHz processor and 16 GB RAM.

A. Preprocessing production information

The case illustrates that five aircrafts are assembled on five different stations, drawn from an aeronautical manufacturing plant. The production information was preprocessed as follows.

There are 17 types of workers in the aircraft final assembly line, as shown in Table III. The total number n_w of worker type w is listed in the second row. Furthermore, the normal working calendar of each worker is 08:00–12:00 and 14:00-18:00 from Monday to Saturday.

The number n_j of a specific worker type w and the processing time p_j is already known to process a task j. Moreover, the processing state of each task is acquired on the basis of CPPS. Thus, all finished tasks are removed from the precedence graph. The unfinished assembly tasks are listed

TABLE III
WORKER TYPE INFORMATION.

w	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
n_w	11	11	5	16	12	14	16	7	11	15	16	10	15	10	15	12	16

in Table IV partially and the total amount of detailed tasks is $|J| = 3787$.

TABLE IV
ASSEMBLY TASKS INFORMATION.

j	p_j	m	w	n_j	Type	j	p_j	m	w	n_j	Type
0	0	1		0	virtual	5	8	1	17	1	detail
1	0	1		0	virtual	6	8	1	17	1	detail
2	8	1	6	1	detailed	7	4	1	7	1	detailed
3	8	1	6	1	detailed	8	4	1	7	1	detailed
4	8	1	17	1	detailed	9	0	1		0	virtual
...	...	...	...	...	...	...	...	...	...	...	...

Task precedence constraints in the real final assembly line are shown in Table V. For example, any task in the immediate successor set $ST_2 = \{task4, task5\}$ cannot be assembled before task2 is completed.

TABLE V
TASK PRECEDENCE CONSTRAINTS.

id	1	2	3	4	5	6	7	8	9	10	11	12	13	...
Predecessor	0	1	1	1	2	2	3	3	7	4	5	8	6	...
Successor	1	2	3	7	4	5	4	5	8	6	6	9	9	...

B. Parameter settings

All evolution algorithms apply the same parameters for a fair comparison, as shown in Table VI. Each population has N=100 individuals and the number of fitness evaluations is fixed to NFE=10,000. Additionally, the crossover rate and the mutation rate are configured as 0.92 and 0.03, respectively. Both of the maximum cognitive factor $\Phi_{1,max}$ and the maximum social factor $\Phi_{2,max}$ are set to 0.8 for running multi-objective particle swarm optimizer (MOPSO).

TABLE VI
THE PARAMETER SETTINGS.

N	NFE	Stage	SBX	PM	$\Phi_{1,max}$	$\Phi_{2,max}$
100	10,000	2	0.92	0.03	0.8	0.8

The solution for the worker allocation problem with the multi-stage workstation outperforms that with a single-stage. To prove it, the proposed NSGA-IV algorithm was run separately 30 times for worker allocation problem with a single-stage station and with two stages station. This study set two stages instead of many stages since too many stations can easily cause production chaos. The Pareto set of two stages compares with the Pareto frontier of a single-stage on the dimension of the three objectives (MWC, DWC, and MDPW), as shown in Fig. 7. It can be observed that the three objectives are conflicting with each other, except for the connection between DWC and MDPW. In addition, it is obvious that the Pareto set of the two-stages dominates that of a single-stage under all objectives.

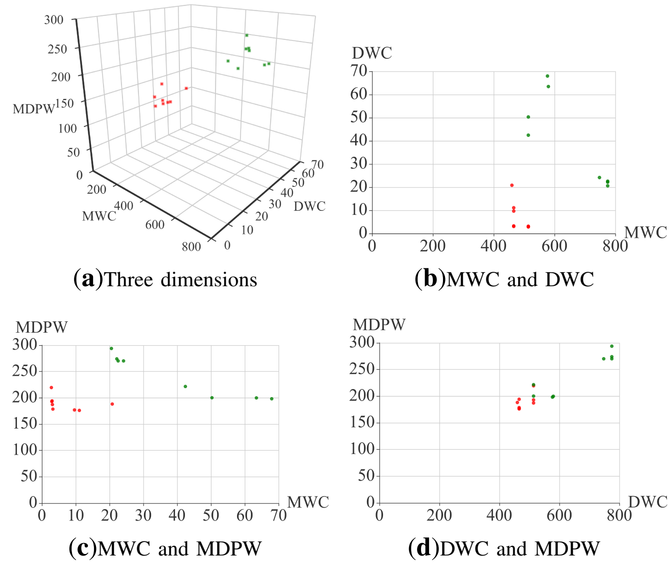

Fig. 7. Pareto Frontier of the worker allocation problem with two-stages (red point) compared to that with single stage (green point).

C. Comparison with five MOEAs

Five MOEAs algorithms, namely vector evaluated genetic algorithms (VEGA), improved strength Pareto evolutionary algorithm (SPEA2), MOPSO, NSGA-II, and NSGA-III, were employed to compare with the proposed NSGA-IV algorithm. Each of them was run independently 30 times to ensure generalization performance. Then, the hypervolume ratio (HVR), the inverted generational distance (IGD), and the additive epsilon indicator (AEI) were used for measuring diversity, proximity, and consistency of solution, respectively [27]. Furthermore, the Wilcoxon signed-rank test was utilized to compare the significant differences between NSGA-IV and five MOEAs.

The three subplots on the left of Fig. 8 show the average performance metrics of six MOEAs across generations. These trend plots represent that all algorithms can converge quickly, except VEGA and MOPSO. It is clear that the three performances (HVR, IGD, and AEI) of SPEA2, NSGA-II, NSGA-III, and NSGA-IV are similar improvements between generations 1 and 50. However, after generation 50, the three performances of NSGA-IV are slightly better than those of other MOEAs. Besides, the average AEI of NSGA-III drops slightly faster than that of NSGA-IV over the top 20 generations, whereas the trend has reversed after that.

In addition, each algorithm can obtain 90 performance metrics by running 30 times independently. The same metrics get together to draw three box plots on the right of Fig. 8. The solution obtained by NSGA-IV has the largest median of HVR as well as has the smallest median of IGD and AEI in all algorithms. The NSGA-IV has the shortest interquartile range in the IGD and AEI metric. It indicates that the NSGA-IV is not susceptible to random factors comparing with other algorithms.

The Wilcoxon signed-rank test with the significance level 0.05 is utilized to calculate the significant difference between NSGA-IV and other MOEAs. The results are listed in Table

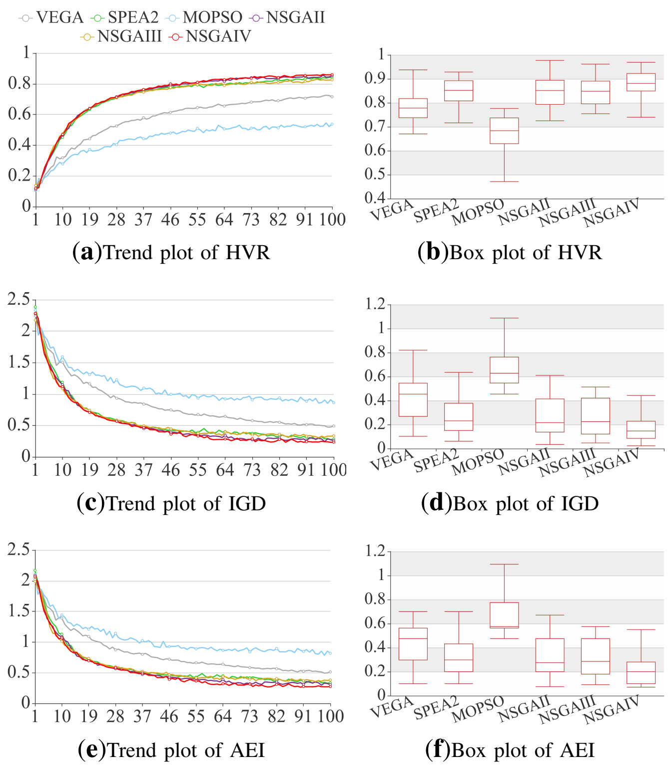

Fig. 8. Trend plots and box plots of average performance metrics.

VII in terms of the HVR, IGD, AEI. The results illustrate that the NSGA-IV can dominate any other algorithm under the IGD, and AEI. Besides, the NSGA-IV is better than VEGA, SPEA2, and MOPSO on the metric of HVR, but there is no significant difference between the proposed algorithm and NSGA-II, NSGA-III.

TABLE VII
COMPARISON WITH FIVE ALGORITHMS IN TERM OF THE WILCOXON SIGNED-RANK TEST.

NSGA-IV	VEGA	SPEA2	MOPSO	NSGA-II	NSGA-III
HVR	0.00(+)	0.019(+)	0.00(+)	0.139(≡)	0.079(≡)
IGD	0.00(+)	0.003(+)	0.00(+)	0.043(+)	0.032(+)
AEI	0.00(+)	0.030(+)	0.00(+)	0.025(+)	0.024(+)

+ the NSGA-IV is significantly better than others according to the metric considered.
≡ there is no significant difference between NSGA-IV and others.

D. Practical application

This study aims to address the worker allocation problem with the multi-stage workstation. The proposed NSGA-IV is embedded in the manufacturing execution system and successfully applied on the shop floor. In the past, manual allocation is based on a heuristic rule that assigning workers is proportional to the tasks remaining time. Thus, its three objective values (MWC, DWC, MDPW) are 580.0, 82.05, 241.47, respectively.

This study utilizes the predictive-reactive scheduling strategy that is widely used in practice. All finished tasks remove from the production tasks set and the remaining processing times of all unfinished tasks are calculated. Besides, the production managers manually update the product information in terms of uncertain events. For example, if one staff is absent, the working calendar needs to be changed. Once an assembly task fails, the task will be added to the production task set again. Thus, the manager needs to make a new worker allocation scheme every weekend. Then, one solution is selected on the basis of Lexicographic resolution [13]. According to the impact importance extent, these objective functions are utilized in sequence. Finally, the solution is published in the workshop. However, some critical events inevitably have a strong impact on the execution of the scheme. Thus, the worker allocation scheme can be locally adjusted by managers. Once the event is tackled, these workers will return to the original stations.

We selected a Pareto solution according three objective, i.e., MWC, DWC, MDPW (values 459.0, 19.20, 195.55), respectively. Compared to manual allocation, the takt time of the aircraft final assembly line is reduced by 20.86%. As space is limited, this paper only displays its task Gantt graph of critical worker type $w = 16$, as illustrated in Fig. 9. The Gantt bars in each workstation are represented by different colors. In the header table, the first three columns signify the station number, the total number of processing tasks, and the worker allocation results, respectively. As shown in Fig. 9, the two stages in the first station assign four workers and two workers of the sixteen type in sequence. Moreover, the detailed information of each task can be viewed by clicking the Gantt bar. The processing time of task2450 is 4 hours, yet the interval time from plan start time to plan finish time is 18 hours which includes the off-duty time. Besides, the constraint (8) results in that the fourth station is not less than one worker belonging to the sixteenth type. Therefore, although there is no task in the second stage, one worker still is allocated in the fourth station. The idle worker either take a holiday or assist other workers in a bottleneck station, making a robust production system.

E. Discussions and analysis

The worker allocation problem with the multi-stage workstation comes from a real aircraft factory, which is an effective complement to the assembly line balancing problem. It can help reduce takt time and improve production efficiency. However, there are still three issues that need to be discussed and further analyzed.

Firstly, it is necessary to study how long the worker allocation scheme is modified. The planning scheme is easy to lag behind the real production situation as a result of the uncertain events. On one hand, if the deviation cannot be eliminated timely, an aggravation of the condition will occur. On the other hand, the production arrangement will be chaotic if the adjustment is frequent.

Secondly, there are fewer bottleneck resources in the aircraft final assembly line, which has a great impact on the scheduling results. Therefore, these bottleneck resources should have been concerned to generate the next generation. For example, the weighted Euclidean distance may be employed in the NSGA-IV, whereas it is difficult to determine the bottleneck resources and their weight.

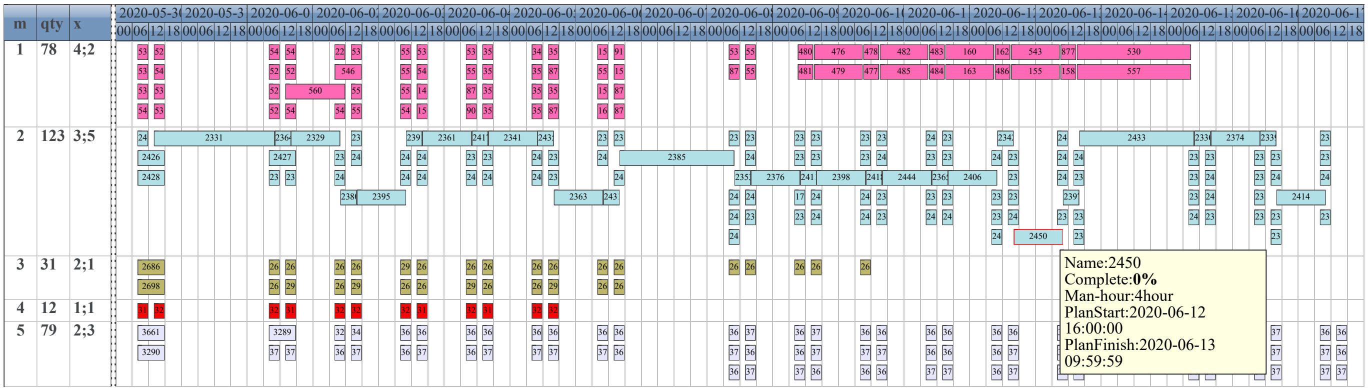

Fig. 9. Task Gantt graph of sixteen worker type (w=16).

Thirdly, the SSGS is used for the fitness evaluation to find a solution within reasonable computation time. However, the heuristic method is usually defeated by the meta-heuristic algorithm in terms of the solution quality. So it is meaningful to encode the worker allocation and the operations assignment simultaneously. But the encoding scheme needs to consume huge amounts of computing resources to converge to the optimal solutions.

VI. Conclusions and future work

This paper proposes an NSGA-IV algorithm to solve the worker allocation problem with the multi-stage workstation. The major contributions of this work are summarized as follows: 1) A systematic framework is put forward on the basis of the application of Cyber-Physical Production Systems. It has the capability of capturing and tackling uncertain events quickly, eventually forming a closed-loop production system. 2) An integer programming formulation is developed to formulate a worker allocation problem with the multi-stage workstation in an aircraft final assembly line, implementing effective arrangements for the scarce human resource. And 3) an NSGA-IV by leveraging the non-dominant sorting and the hierarchical clustering is proposed to improve the population diversity in the decision space.

The multi-stage workstation is effective to improve the productivity of a paced aircraft final assembly line. There is not the assignment of lots of tasks among stations, making a controllable production process. Moreover, the Pareto solutions of the proposed NSGA-IV dominate those of the VEGA, SPEA2, and MOPSO in terms of HVR in a real-world case. Additionally, the Pareto solutions of the NSGA-IV dominate those of five algorithms consisting of VEGA, SPEA2, MOPSO, NSGA-II, and NSGA-III under IGD and AEI. Therefore, it is clear that NSGA-IV is an outstanding algorithm for solving the worker allocation problem. The algorithm has been applied in production to enable manufacturing aircraft with a stable takt time.

Our future work will mainly focus on three aspects. Firstly, we consider the influence of multi-skill workers and their skill levels in real-world applications. Secondly, some hybrid evolutionary algorithms can be exploited to improve the strengths of multiple meta-heuristics. Finally, the fuzzy logic scheduling algorithm can be employed to solve robust proactive scheduling.

References

[1] O. Battaïa, X. Delorme, A. Dolgui, J. Hagemann, A. Horlemann, S. Kovalev, and S. Malyutin, "Workforce minimization for a mixed-model assembly line in the automotive industry," *International Journal of Production Economics*, vol. 170, 06 2015, doi:10.1016/j.ijpe.2015.05.038.

[2] M. Şahin and T. Kellegöz, "Balancing multi-manned assembly lines with walking workers: Problem definition, mathematical formulation, and an electromagnetic field optimisation algorithm," *International Journal of Production Research*, vol. 57, no. 20, pp. 6487–6505, Oct. 2019, doi:10.1080/00207543.2019.1566672.

[3] B. Xin, Y. Li, J. Yu, and J. Zhang, "An adaptive bpso algorithm for multi-skilled workers assignment problem in aircraft assembly lines," *Assembly Automation*, 2015, doi:10.1108/AA-06-2015-051.

[4] W. Du, Y. Tang, S. Y. S. Leung, L. Tong, A. V. Vasilakos, and F. Qian, "Robust order scheduling in the discrete manufacturing industry: A multiobjective optimization approach," *Ieee transactions on industrial informatics*, vol. 14, no. 1, pp. 253–264, 2017, doi:10.1109/TII.2017.2664080.

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002, doi:10.1109/4235.996017.

[6] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints," *IEEE transactions on evolutionary computation*, vol. 18, no. 4, pp. 577–601, 2013, doi:10.1109/TEVC.2013.2281535.

[7] J. Otto, B. Vogel-Heuser, and O. Niggemann, "Automatic parameter estimation for reusable software components of modular and reconfigurable cyber-physical production systems in the domain of discrete manufacturing," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 1, pp. 275–282, Jan. 2018, doi:10.1109/TII.2017.2718729.

[8] C. Cimini, F. Pirola, R. Pinto, and S. Cavalieri, "A human-in-the-loop manufacturing control architecture for the next generation of production systems," *Journal of Manufacturing Systems*, vol. 54, pp. 258–271, 2020, doi:10.1016/j.jmsy.2020.01.002.

[9] W. Du, W. Zhong, Y. Tang, W. Du, and Y. Jin, "High-dimensional robust multi-objective optimization for order scheduling: A decision variable classification approach," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 293–304, Jan. 2019, doi:10.1109/TII.2018.2836189.

[10] P. Fang, J. Yang, L. Zheng, R. Y. Zhong, and Y. Jiang, "Data analytics-enable production visibility for cyber-physical production systems," *Journal of Manufacturing Systems*, vol. 57, pp. 242 – 253, Oct. 2020, doi:10.1016/j.jmsy.2020.09.002.

[11] D. A. Rossit, F. Tohmé, and M. Frutos, "Production planning and scheduling in cyber-physical production systems: a review," *International journal of computer integrated manufacturing*, vol. 32, no. 4-5, pp. 385–395, 2019, doi:10.1080/0951192X.2019.1605199.

[12] A. Scholl and C. Becker, "State-of-the-art exact and heuristic solution procedures for simple assembly line balancing," *Eur. J. Oper. Res.*, vol. 168, no. 3, pp. 666–693, Feb. 2006, doi:10.1016/j.ejor.2004.07.022.

[13] O. Battaïa and A. Dolgui, "A taxonomy of line balancing problems and their solution approaches," *International Journal of Production Economics*, vol. 142, no. 2, pp. 259–277, Apr. 2013, doi:10.1016/j.ijpe.2012.10.020.

[14] G. Heike, M. Ramulu, E. Sorenson, P. Shanahan, and K. Moinzadeh, "Mixed model assembly alternatives for low-volume manufacturing: the case of the aerospace industry," *International Journal of Production Economics*, vol. 72, no. 2, pp. 103–120, Jul. 2001, doi:10.1016/s0925-5273(00)00089-x.

[15] C. G. S. Sikora, T. C. Lopes, and L. Magatão, "Traveling worker assembly line (re)balancing problem: Model, reduction techniques, and real case studies," *European Journal of Operational Research*, vol. 259, no. 3, pp. 949–971, Jun. 2017, doi:10.1016/j.ejor.2016.11.027.

[16] A. Biele and L. Mönch, "Hybrid approaches to optimize mixed-model assembly lines in low-volume manufacturing," *Journal of Heuristics*, vol. 24, no. 1, pp. 49–81, Feb. 2018, doi:10.1007/s10732-017-9357-6.

[17] J. Lian, C. Liu, W. Li, and Y. Yin, "A multi-skilled worker assignment problem in seru production systems considering the worker heterogeneity," *Computers & Industrial Engineering*, vol. 118, pp. 366–382, Apr. 2018, doi:10.1016/j.cie.2018.02.035.

[18] S. O. Tasan and S. Tunali, "A review of the current applications of genetic algorithms in assembly line balancing," *Journal of intelligent manufacturing*, vol. 19, no. 1, pp. 49–69, Feb. 2008, doi:10.1007/s10845-007-0045-5.

[19] C. Becker and A. Scholl, "A survey on problems and methods in generalized assembly line balancing," *European journal of operational research*, vol. 168, no. 3, pp. 694–715, 2006, doi:10.1016/j.ejor.2004.07.023.

[20] M. C. O. Moreira, M. Ritt, A. M. Costa, and A. A. Chaves, "Simple heuristics for the assembly line worker assignment and balancing problem," *Journal of heuristics*, vol. 18, no. 3, pp. 505–524, Jun. 2012, doi:10.1007/s10732-012-9195-5.

[21] S. K. Mishra, D. Puthal, J. J. Rodrigues, B. Sahoo, and E. Dutkiewicz, "Sustainable service allocation using a metaheuristic technique in a fog server for industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4497–4506, Oct. 2018, doi:10.1109/TII.2018.2791619.

[22] X. Li, C. Lu, L. Gao, S. Xiao, and L. Wen, "An effective multiobjective algorithm for energy-efficient scheduling in a real-life welding shop," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 12, pp. 5400–5409, Dec. 2018, doi:10.1109/TII.2018.2843441.

[23] Y. Zhang, F. Tao, Y. Liu, P. Zhang, Y. Cheng, and Y. Zuo, "Long/short-term utility aware optimal selection of manufacturing service composition toward industrial internet platforms," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3712–3722, 2019, doi:10.1109/TII.2019.2892777.

[24] C.-X. Wu, M.-H. Liao, M. Karatas, S.-Y. Chen, and Y.-J. Zheng, "Real-time neural network scheduling of emergency medical mask production during covid-19," *Applied Soft Computing*, vol. 97, p. 106790, 2020, doi:10.1016/j.asoc.2020.106790.

[25] L. Mo, P. You, X. Cao, Y. Song, and A. Kritikakou, "Event-driven joint mobile actuators scheduling and control in cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 11, pp. 5877–5891, Nov. 2019, doi:10.1109/TII.2019.2906061.

[26] J. Weglarz, *Project scheduling: recent models, algorithms and applications*. Springer Science & Business Media, 2012, vol. 14, doi:10.1007/978-1-4615-5533-9.

[27] C. A. C. Coello, G. B. Lamont, D. A. Van Veldhuizen *et al.*, *Evolutionary algorithms for solving multi-objective problems*. Springer, 2007, vol. 5, doi:10.1007/978-0-387-36797-2.

Jianjun Yang was born in Jiangxi Province of China, in 1960. He received the B.S. and M.S. degrees in manufacturing engineering from the Beijing University of Aeronautics and Astronautics BUAA), in 1989.

From 1993 to 2002, he was a associate professor with the Institute of manufacturing systems, Since 2003, he has been an Professor with the School of Mechanical Engineering and Automation, Beihang University (BUAA). He served as vice president of the college and director of the Department of industrial and manufacturing systems engineering. He is the author of one books, more than 100 articles, His research interests include manufacturing resource planning and intelligent optimization, production scheduling and hyper-heuristic algorithm, manufacturing process information integration and control technology.

Qingmiao Liao was born in Jingzhou, Hubei, China, in 1978. He received the B.S. degree, M.S. degree, and Ph.D. degree in manufacturing engineering from Beihang University, Beijing, China,in 1999,2002 and 2019, respectively.

He is currently a senior business consultant of AVIC digital Co., Ltd. His research interests include product planning and scheduling, big data ,cloud computing and intelligent manufacturing system.

Ray Y. Zhong (M13) received the Ph.D. degree in industrial and manufacturing systems engineering from the University of Hong Kong, China, in 2013, M.S. degree in signal and information processing from Guangdong University of Technology, China, in 2009 and B.S. degree in mathematics and computer science and technology from Gannan Normal University, China, in 2004.

Dr. Zhong is current an Assistant Professor in Department of Industrial and Manufacturing Systems Engineering from the University of Hong Kong. He was a lecturer at Department of Mechanical Engineering, The University of Auckland from June 2016 to January 2019. From December 2015 to May 2016, he is a Research Assistant Professor with the Department of Industrial and Manufacturing Systems Engineering, The University of Hong Kong. His research interest includes IoT-enabled manufacturing and Big Data in manufacturing.

Dr. Zhong is member of IEEE, CIRP RA (2017-2020), IET, and ASME. He was a recipient of the Best Conference Paper Award in the 2018 and 2014 IEEE International Conference on Networking, Sensing and Control (ICNSC2018 and 2014), 2018 Best Paper of International Journal of Production Research and Certificate of Merit in the Hong Kong U-21 RFID Awards 2011, GS1 HK.

Pengcheng Fang was born in Inner Mongolia, China, in 1991. He received the B.S. degree in mechanical engineering from the Inner Mongolia University of Science and Technology, Baotou, China, in 2012, and the M.S. degree in mechanical engineering from the University of Science and Technology Beijing, Beijing, China, in 2015. He is currently working toward the Ph.D. degree in Mechanical Manufacture and Automation from the Beihang University, Beijing, China.

His research interests include Cyber Physical Production System, production scheduling and control, data analysis, and intelligent manufacturing system.

Yuchen Jiang was born in Tianjin, China, in 1990. He received the B.S. degree and M.S. degree in School of Mechanical Engineering and Automation Beihang University, Beijing, China, in 2013 and 2018, respectively. He is currently working toward the Ph.D. degree in State Key Laboratory of Information Security, Institute of Information Engineering, University of Chinese Academy of Sciences, Beijing, China.

His research interests include Maintenance Operations Management Systems, predictive maintenance of industrial equipment, deep learning, and intelligent manufacturing system.