

ORIGINAL ARTICLE

Open Access



A Discrete Multi-Objective Artificial Bee Colony Algorithm for a Real-World Electronic Device Testing Machine Allocation Problem

Jin Xie, Xinyu Li and Liang Gao*

Abstract

With the continuous development of science and technology, electronic devices have begun to enter all aspects of human life, becoming increasingly closely related to human life. Users have higher quality requirements for electronic devices. Electronic device testing has gradually become an irreplaceable engineering process in modern manufacturing enterprises to guarantee the quality of products while preventing inferior products from entering the market. Considering the large output of electronic devices, improving the testing efficiency while reducing the testing cost has become an urgent problem to be solved. This study investigates the electronic device testing machine allocation problem (EDTMAP), aiming to improve the production of electronic devices and reduce the scheduling distance among testing machines through reasonable machine allocation. First, a mathematical model was formulated for the EDTMAP to maximize both production and the scheduling distance among testing machines. Second, we developed a discrete multi-objective artificial bee colony (DMOABC) algorithm to solve EDTMAP. A crossover operator and local search operator were designed to improve the exploration and exploitation of the algorithm, respectively. Numerical experiments were conducted to evaluate the performance of the proposed algorithm. The experimental results demonstrate the superiority of the proposed algorithm compared with the non-dominated sorting genetic algorithm II (NSGA-II) and strength Pareto evolutionary algorithm 2 (SPEA2). Finally, the mathematical model and DMOABC algorithm were applied to a real-world factory that tests radio-frequency modules. The results verify that our method can significantly improve production and reduce the scheduling distance among testing machines.

Keywords: Electronic device, Machine allocation, Multi-objective optimization, Artificial bee colony algorithm

1 Introduction

The production process of an electronic device is as follows: (1) purchase raw materials and reshape the components, (2) install electronic components on a printed circuit board using surface-mounted technology, (3) assemble the printed circuit board into parts of the electronic device, (4) assemble the pieces into a complete electronic device, and (5) test the finished electronic device. As the last step in electronic device production, testing determines any quality defects and prevents

inferior products from entering the market [1]. A reasonable allocation of testing machines can help improve production efficiency by reducing the blocking time associated with electronic devices at each testing stage. Consequently, the electronic device testing machine allocation problem (EDTMAP) plays a crucial role in improving the production efficiency of enterprises, thereby increasing their profits.

Existing studies on electronic device testing have focused more on developing new testing technologies and systems to improve productivity. However, few studies have considered the reasonable allocation of testing machines to improve productivity. Singh et al. [2] proposed a classification-based scanning technique for

*Correspondence: gaoliang@mail.hust.edu.cn

State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

testing electronic devices. The technology efficiently tested the electromagnetic compatibility of an electronic device by only testing its regions of interest. Orlov and Vasilchenko [3] presented an intelligent measurement system to test electronic devices. The system efficiently tested the electronic device by measuring its surface temperature. Stoyanov et al. [4] developed a smart qualification testing approach for electronic devices. The method combines data analytics and data-driven predictive modeling to reduce the testing time and cost. Gallo et al. [5] designed an automated testing system for use in electronic devices. Metrology was introduced into the system to improve the testing efficiency. Most research improves productivity by developing new technologies, and this study focuses on the reasonable allocation of testing machines to improve productivity.

In addition, most studies on the machine allocation problem (MAP) consider only one evaluation index, e.g., productivity or cost, but ignore the combined effects of both. Even when considering multiple indices, most studies solve them using a linear weighted sum method instead of obtaining the Pareto optimal solution set. Thouin and Coates [6] formulated a mathematical model of the MAP to minimize scheduling costs and developed a heuristic to solve the model. Jaramillo and McKendall [7] presented a mathematical model of the MAP intended to minimize material handling costs and used a tabu search heuristic and a memetic algorithm to solve this model. Poon et al. [8] presented a genetic algorithm for the MAP to minimize the waiting time of production workstations and improve productivity. Chehade et al. [9] proposed a multi-objective algorithm based on the non-dominated sorting genetic algorithm II (NSGA-II) with a local search to solve the MAP by considering productivity and cost. Mohtashami [10] proposed a multi-objective mathematical model and hybrid genetic algorithm for the MAP. The objectives were to maximize the production rate and minimize the total cost. Based on the above analysis, balancing the productivity and cost of the MAP is the focus of and difficulty associated with this study. Because a shorter scheduling distance among machines leads to a lower cost for machine allocation, this study considers the two objectives of maximizing production and the inverse of the scheduling distance among machines.

Recently, an increasing number of researchers have used meta-heuristic algorithms to solve multi-objective optimization problems (MOPs) [11]. The most commonly used meta-heuristics are NSGA-II [12] and the strength Pareto evolutionary algorithm 2 (SPEA2) [13]. Recently, many researchers have focused on the artificial bee colony (ABC) algorithm because of its specific global search ability and suitable local searchability. Karaboga

[14] first proposed the original ABC algorithm in 2005. Karaboga and Basturk [15] compared ABC with other classic algorithms to verify its performance. Hedayatzadeh et al. [16] proposed a multi-objective ABC (MOABC) algorithm. Akbari et al. [17] verified its superiority by comparing it with other traditional algorithms. Currently, the ABC algorithm is used in a variety of fields, such as shop scheduling [18–20], network planning [21–24], nurse rostering [25–27], and economic scheduling [28–30]. Because MOABC has been successfully applied in the above fields, this paper proposes a discrete MOABC algorithm to solve the EDTMAP.

The main contributions of this study are as follows. (1) We designed an electronic device testing machine allocation model. This can be easily extended to other types of machine allocation models. (2) We developed a discrete multi-objective artificial bee colony (DMOABC) algorithm to solve the EDTMAP. Three key operators, i.e., feasible solution generator (FSG), local search operator (LSO), and crossover operator (XO), were designed for use in the algorithm. FSG is used to generate a feasible machine allocation scheme, and LSO and XO are used to balance both local and global searches. (3) We demonstrate that DMOABC is robust and highly competitive with NSGA-II and SPEA2 for solving the EDTMAP by analyzing and comparing the results of numerical experiments. (4) We applied DMOABC to real-world electronic device testing workshops. The results show that our method can guide actual electronic device testing machine allocation.

The remainder of this paper is organized as follows. Section 2 introduces the problem description and mathematical model of the EDTMAP. Section 3 presents a discrete multi-objective optimization algorithm based on ABC for the EDTMAP. Section 4 presents numerical experiments to verify the superiority of our proposed algorithm. In Sect. 5, we describe the application of the proposed DMOABC to a real-world testing workstation. Finally, Sect. 6 presents the conclusions and highlights future research directions.

2 Problem Formulation

2.1 Problem Description

In this study, the EDTMAP was investigated based on a real-world electronic device production factory with multiple testing workshops. Each workshop tests different types of products to determine their reliability. Each product is sequentially tested at various stages, and the testing times at the different stages are different. Multiple machines can be used to test the products at the same stage. Compatibility and quantity constraints exist for machine interfaces at each stage. If the interfaces of the machine match a stage and satisfy the quantity

requirement for the stage, the machine can be scheduled for the stage. The total number of products tested at each stage per week is calculated based on the testing time and the number of machines at each stage. The minimum amount of products tested in the series of stages is the weekly production. In addition, all the testing machines are initially distributed in different workshops. These machines therefore must be scheduled for multiple workshops. We attempt to keep the total scheduling distance among the machines as short as possible, while keeping the total production of the factory as large as possible.

Briefly, the EDTMAP can be described as follows: there are m testing workshops in a factory, and each workshop produces n types of product. In addition, there are different types of testing machines in the factory that are initially distributed in various workshops. The number of machines of the same type in each workshop differs. Currently, it is necessary to schedule machines for different stages in multiple workshops to test products to maximize weekly production. Meanwhile, the total scheduling distance among the machines with respect to multiple workshops should be as short as possible. The main characteristic of the EDTMAP is that multiple machines that

meet the adaptability constraint of interfaces can be allocated at the same stage.

There are some necessary assumptions for the proposed mathematical model of the EDTMAP:

- 1) The same type of product follows the same testing route.
- 2) There can be more than one machine at a stage, and there is no capacity limit.
- 3) Interruption and preemption are not allowed.
- 4) Each machine can be scheduled for only one stage every week and is not allowed to be re-mobilized in that week.
- 5) Each machine can only test one product at a time.

2.2 Mathematical Modeling of the EDTMAP

The notations and decision variables for the mathematical model of the EDTMAP are given as Table 1.

Based on the above notations, a multi-objective mathematical model of the EDTMAP is designed to simultaneously maximize the total production and the inverse

Table 1 Notations and decision variables for the mathematical model of the EDTMAP

Variable	Notation
W	Set of workshops
$i (i')$	Index of the workshop, $i \in W$
J_i	Set of the types of products tested in the i th workshop
j	Index of the product type, $j \in J_i$
$S_{i,j}$	Set of stages where the j th type of product in the i th workshop is tested
k	Index of the stage, $k \in S_{i,j}$
T	Set of the types of testing machines
p	Index of the type of testing machines, $p \in T$
$R_{p,i'}$	Set of the p th type of testing machines in the i' th workshop
q	Index of the p th type of testing machines in the i' th workshop, $q \in R_{p,i'}$
D_{\max}	The maximum scheduling distance among all machines
$w_{i,j}$	Production weight of the j th type of product in the i th workshop
$t_{i,j,k}$	Time of the j th type of product in the i th workshop tested at the k th stage
$P_{i,j,k}$	Production of the j th type of product in the i th workshop tested by one machine at the k th stage
$l_{i,j,k}$	Interface type constraint for machines testing the j th type of product in the i th workshop at the k th stage
l'_p	Interface type of the p th type of testing machines
$N_{i,j,k}$	Interface quantity requirement for machines testing the j th type of product in the i th workshop at the k th stage
N'_p	Interface number of the p th type of testing machines
$d_{i,i'}$	Distance from the i th workshop to the i' th workshop
t_{minute}	Minutes that the testing machine runs per hour
t_{hour}	Hours that the testing machine runs per day
t_{day}	Days that the testing machine runs per week
η	Production efficiency of testing machines
$X_{i,j,k,p,i',q}$	Decision variable. If the q th of the p th type of testing machines in the i' th workshop is scheduled to the k th stage where the j th type of product in the i th workshop tested $X_{i,j,k,p,i',q} = 1$; otherwise, $X_{i,j,k,p,i',q} = 0$

of the total scheduling distance among machines. The mathematical model is as follows:

$$\max f_1 = \sum_{i \in I} \sum_{j \in J_i} w_{ij} \cdot \min_{k \in S_{ij}} \left\{ \sum_{p \in T} \sum_{i' \in I} \sum_{q \in R_{p,i'}} P_{i,j,k} \cdot x_{i,j,k,p,i',q} \right\}, \quad (1)$$

$$\max f_2 = - \sum_{i \in I} \sum_{j \in J_i} \sum_{k \in S_{ij}} \sum_{p \in T} \sum_{i' \in I} \sum_{q \in R_{p,i'}} d_{i,i'} \cdot x_{i,j,k,p,i',q}, \quad (2)$$

where f_1 denotes the weighted sum of production in all workshops and f_2 represents the inverse of the total scheduling distance among all testing machines. The constraints of this problem are as follows:

$$P_{i,j,k} = \frac{t_{\min}}{t_{i,j,k}} \cdot t_{\text{hour}} \cdot t_{\text{day}} \cdot \eta, \quad (3)$$

$$D_{\max} = \sum_{p \in T} \sum_{i' \in I} \sum_{q \in R_{p,i'}} d_{i,i'}, \quad i = s : \arg \max_{s \in I} d_{s,i'}, \quad (4)$$

$$(I_{i,j,k} - I'_p) \cdot x_{i,j,k,p,i',q} = 0, \quad \forall i, i' \in I, \forall j \in J_i, \forall k \in S_{ij}, \forall p \in T, \forall q \in R_{p,i'}, \quad (5)$$

$$(N_{p,i',q} - N'_p) \cdot x_{i,j,k,p,i',q} \geq 0, \quad \forall i, i' \in I, \forall j \in J_i, \forall k \in S_{ij}, \forall p \in T, \forall q \in R_{p,i'}, \quad (6)$$

$$\sum_{i \in I} \sum_{j \in J_i} \sum_{k \in S_{ij}} \sum_{p \in T} \sum_{i' \in I} x_{i,j,k,p,i',q} \leq 1, \quad \forall q \in R_{p,i'}, \quad (7)$$

$$x_{i,j,k,p,i',q} \in \{0, 1\}, \quad \forall i, i' \in I, \forall j \in J_i, \forall k \in S_{ij}, \forall p \in T, \forall q \in R_{p,i'}. \quad (8)$$

Constraint (3) computes the weekly production of the testing machine at each stage. Constraint (4) calculates the maximum total scheduling distance among all machines. Constraint (5) ensures that only machines with interfaces that match a stage can be scheduled for the stage. Constraint (6) ensures that only machines that meet the interface quantity requirement of a stage can be scheduled for the stage. Constraint (7) guarantees that any machine can be scheduled for only one stage every week at most. Constraint (8) describes the range of the decision variables.

2.3 Mathematical Modeling of the EDTMAP

Based on the above model, there are some differences between the EDTMAP and the traditional MAP. The classic MAP only considers machine scheduling between multiple production lines in a single workshop, but not

that among various workshops. Conversely, the EDTMAP considers machine scheduling between numerous production lines as well as between multiple workshops. Scheduling machines for a different workshop may increase productivity, but the scheduling distance among machines will increase accordingly, resulting in increased machine damage probability and cost. Balancing these two objectives is the focus of our algorithm design, as introduced in the next section.

In addition, the traditional MAP does not need to consider the interface compatibility of machines. However, the EDTMAP should consider compatibility and quantity constraints for machine interfaces at each stage. The machine can be scheduled for the stage only if the interfaces of the machine match the stage and satisfy the quantity requirement for the stage.

Table 2a–c describes an example of the EDTMAP. Table 2a describes the two testing workshops in a factory. Each workshop tests two types of products, and each type of product is tested sequentially in two stages. The testing times and required interface quantities of the products at each stage are listed in Table 2a. Table 2b lists the types and number of machines in each workshop. The matching stage and interface quantity of each type of machine are listed in Table 2b. Table 2c shows the distance between workshops. If a machine is not in use, then the scheduling distance is 0. Handling distances exist in the same workshop. Therefore, if the initial workshop of a machine is the same as the workshop where it is scheduled, then the scheduling distance is 0.5. Table 3 shows the allocation scheme for the testing machines in this example. “M3W1 (2)” denotes that two third-type machines (M3) are initially in the first workshop (W1). In the allocation scheme, the two third-type machines are scheduled to test product A2T2R at Stage 2 in W1. “Null” denotes that no machine is scheduled for the stage.

3 Proposed DMOABC for EDTMAP

3.1 Main Framework of the Proposed Algorithm

In this paper, a DMOABC algorithm to obtain non-dominated solutions for balancing the production and scheduling distances is presented. Considering the compatibility and quantity constraints for the interfaces of machines, we designed three operators, i.e., FSG, LSO, and XO, to ensure the feasibility of solutions and cause the population to evolve in the correct direction. FSG is designed to ensure that the initial solution is a feasible machine-allocation scheme. LSO is used as a solution to execute a local search. This effectively improves the local search ability of the DMOABC algorithm. A slight change in the solutions helps the algorithm to jump out of the local optimum. XO implements a crossover operator that allows individuals

Table 2 Product and testing machine information

Workshop	Product	Testing time (and interface quantity constraint) at stage 1	Testing time (and interface quantity constraint) at stage 2	
(a) Testing time (and interface quantity constraint) of products at various stages				
W1	A2T2R	5 (1)	10 (1)	
	A2T4R	3 (2)	3 (2)	
W2	B2T2R	2 (1)	4 (1)	
	B2T4R	3 (2)	3 (2)	
Machine	Matching stage	Interface quantity	Workshop	Quantity
(b) Available machines in workshops				
M1	S1	1	W1	1
			W2	1
M2	S1	2	W1	0
			W2	2
M3	S2	1	W1	2
			W2	2
M4	S2	2	W1	2
			W2	4
Workshop	Distance from W1 (km)	Distance from W2 (km)		
(c) Distance between workshops				
W1	0.5	1		
W2	1	0.5		

Table 3 Testing machine allocation scheme

Workshop	Product	Machine allocation at Stage 1	Machine allocation at Stage 2
W1	A2T2R	M1W1 (1)	M3W1 (2)
	A2T4R	Null	Null
W2	B2T2R	M1W2 (1)	M4W1 (2)
		M2W2 (1)	M4W2 (2)
	B2T4R	M3W2 (2)	M4W2 (2)

in the population to exchange information with each other. This effectively improves the global search ability of the DMOABC algorithm. Hence, LSO and XO can balance the local search and global search abilities of DMOABC.

The original ABC algorithm was designed to solve continuous optimization problems. Considering that the EDTMAP is a discrete optimization problem, we modified the encoding to make the solution a feasible allocation scheme. The proposed DMOABC employs a fast non-dominated sorter and a crowding distance sorter [12] to store the non-dominated individuals from each generation to the next. The procedure for the proposed DMOABC is illustrated in Figure 1.

Step 1: Initialize the population size P , number of workshops N_w , type number of products N_p , and limit of abandoning solution L_a . Randomly generate an initial population P_G ;

Step 2: Employ the bee phase. For each individual x_i ($i = 1, 2, \dots, P$) in the population, the following sub-steps are used to generate a new population E_G :

1) Execute LSO for x_i to generate a newly employed bee;

2) x_i becomes a scout bee in the case of having not been improved for L_a generations. The procedure goes to step 6;

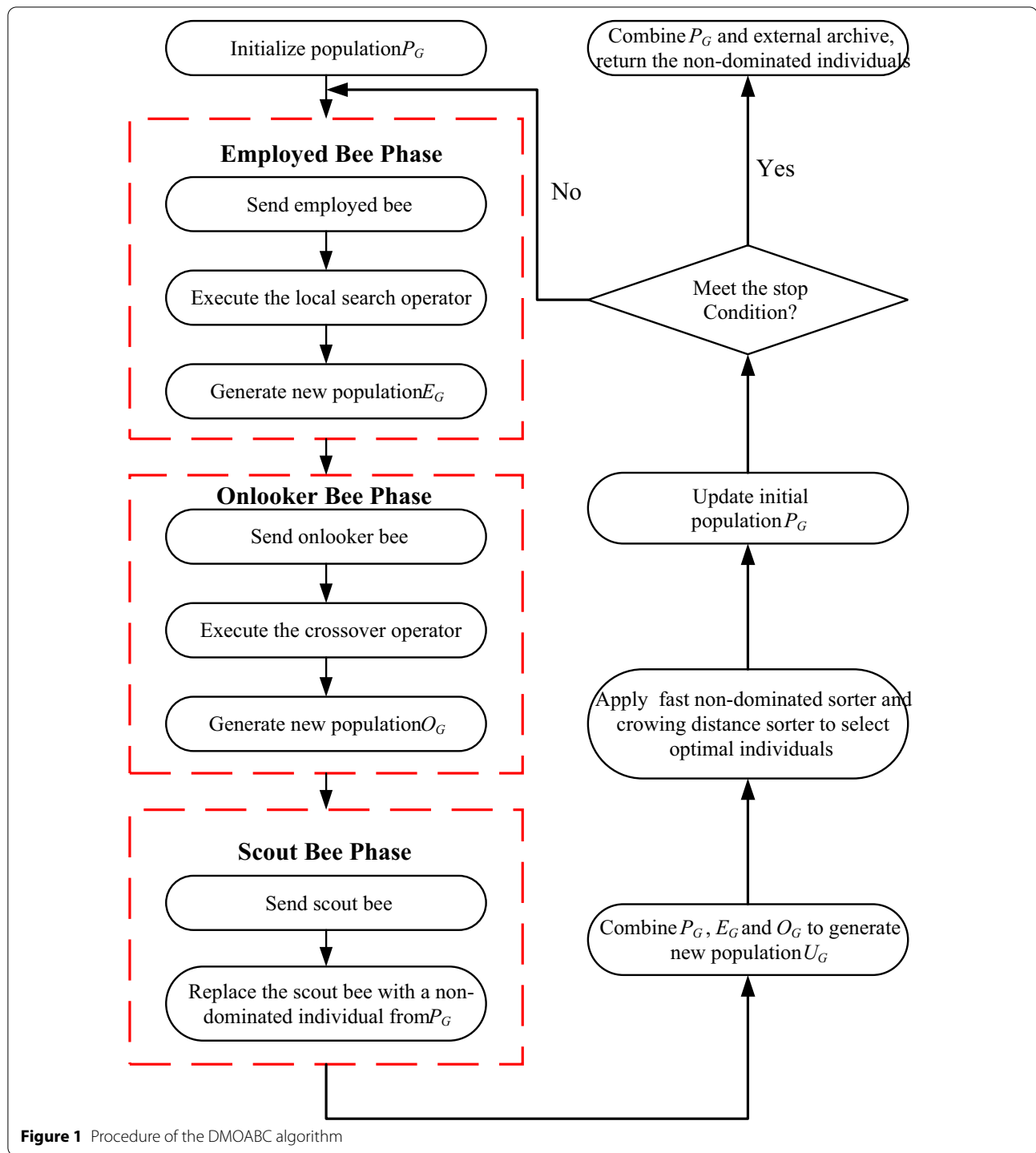
Step 3: Employ the fast non-dominated sorter to rank the new population E_G ;

Step 4: Onlooker bee phase. For $i = 1, 2, \dots, P$, circle the following sub-steps to generate a new population O_G :

1) Use the binary tournament selection for E_G . The better one is selected as the onlooker bee;

2) Execute XO for the onlooker bee and a randomly selected employed bee to generate a new individual;

Step 5: Combine P_G , E_G , and O_G as population U_G ; use the fast non-dominated sorter and crowding distance sorter to sort U_G ; and finally choose the first P individuals to update P_G ;



Step 6: Scout bee phase. If $\max(NIG) > L_a$ (NIG is the generation number in which any of the two objective function values of an individual have not been improved), it is selected as the scout bee and repeated:

1) The scout bee is placed into an external archive. Then, we apply the non-dominated sorter to the external archive and finally choose the non-dominated individuals to update the external archive;

- 2) Randomly select a non-dominated individual from P_G and employ LSO for it to generate a new one;
 - 3) Replace the scout bee with the new individual;
- Step 7: If the termination criterion is satisfied, combine the non-dominated individuals in the last generation with the external archive and use the non-dominated sorter for this combined population to obtain the Pareto solution set; otherwise, proceed to step 2.

The critical elements of DMOABC are the FSG, LSO, XO, fast non-dominated sorter, and crowding distance sorter, which are described in detail below.

3.2 Feasible Solution Generator

A feasible solution for the EDTMAP is a machine-allocation scheme that satisfies the given allocation constraints. First, we define a permutation A_k that includes all stages of n types of products in m workshops. Then, for each stage $k \in A_k$, the unallocated machine that matches stage k is sorted into a set S_k . For each machine $l \in S_k$, a number r in the range $[0, 1]$ is randomly generated. If $r < 0.5$, machine l is allocated to stage k ; otherwise, the machine is not assigned. Finally, a feasible machine-allocation scheme is generated. The pseudo-code of the FSG is outlined in **Operator: FSG**.

Operator: FSG

Input: Interface requirement of each stage and machine information

Output: a feasible machine allocation scheme

Define a permutation A_k including all stages of n types of product in m workshops

For each stage $k \in A_k$

For each machine $l_1 \in$ the set of unallocated machines L

If l_1 meets the interface requirement of stage k **then**

 Put l_1 into set S_k

End If

End For

For each machine $l_2 \in S_k$

 Randomly generate a number r in the range of $[0, 1]$

If $r < 0.5$ **then**

 Allocate machine l_2 to stage k ;

End If

End For

End For

3.3 Local Search Operator

The LSO is used by employed bees to execute a local search, enabling them to look for nectar separately. First, we define a permutation A_k that includes the stages of n types of products in m workshops. Subsequently, in each stage, $k \in A_k$ is selected to execute the local search. There are four neighborhood structures available: (1) add one of the machines that has not been allocated and matches stage k to stage k . (2) Reduce one of the existing machines

at stage k . (3) Select one of the machines that has not been allocated and matches stage k with less scheduling distance or interface quantity to replace one of the existing machines at stage k . (4) Nothing has changed at stage k . Finally, a new machine allocation scheme is generated after the local search has been completed for each stage. The pseudo-code of the LSO is shown in **Operator: LSO**.

Operator: LSO

Input: an original machine allocation scheme

Output: a new machine allocation scheme

Define a permutation A_k including the stages of n types of products in m workshops.

For each stage $k \in A_k$

 Randomly generate a number r ($r \in \{1, 2, 3, 4\}$)

Case $r = 1$:

 Add one of the machines, matching stage k and not having been allocated, to stage k

Case $r = 2$:

 Reduce one of the existing machines at stage k

Case $r = 3$:

 Choose one of the machines that match stage k and is not allocated with less scheduling distance or less interface quantity to replace one of the existing machines at stage k

Case $r = 4$:

 Nothing has changed at stage k

End For

An illustrative example of the LSO is shown in Table 4, based on the case described in Table 2a–c above. For the sake of brevity, we use “W1, A2T2R, S1” to represent stage 1 of product A2T2R in workshop 1. “M3W2N1” denotes the first of the third type of machines in the second workshop. The number r generated for each stage is $[4, 1, 1, 1, 2, 2, 4, 3]$. Therefore, “W1, A2T2R, S1” and “W2, B2T4R, S1” have not changed. “W1, A2T2R, S2”, “W1, A2T4R, S1” and “W1, A2T4R, S2” add one machine. “W2, B2T2R, S1” and “W2, B2T2R, S2” reduce one machine. “W2, B2T4R, S2” exchanges one machine.

3.4 Crossover Operator

The XO is designed to allow onlooker bees to exchange information so that the bee colony can move in the nectar direction. The crossover operation is as follows: (1) randomly choose two stages as the cross points. (2) The machine allocation scheme is copied at stages between the two cross points from the onlooker bee to a newly employed bee. (3) Copy the machine allocation scheme at stages outside the two cross points from the initially employed bee to the newly employed bee. (4) Remove repetitive machines from the stages outside the two cross points for the newly employed bee. The pseudo-code for the XO is shown in **Operator: XO**.

Table 4 Illustrative example of the LSO

Workshop	Product	Stage	Machine allocation (Before LSO)	Machine allocation (After LSO)
W1	A2T2R	S1	M1W1N1	M1W1N1
		S2	M3W1N1	M3W1N1 M3W1N2
	A2T4R	S1	Null	M2W2N2
		S2	Null	M4W2N3
W2	B2T2R	S1	M1W2N1 M2W2N1	M1W2N1
		S2	M4W1N1 M4W1N2	M4W1N1
	B2T4R	S1	M3W2N1	M3W2N1
		S2	M4W2N2	M4W1N2

Operator: XO**Input:** an onlooker bee and an employed bee**Output:** a new employed bee

Randomly choose two stages as cross points

Copy the machine allocation scheme at stages between the two cross points from the onlooker bee to the newly employed bee

Copy the machine allocation scheme at stages outside the two cross points from the original employed bee to the newly employed bee

For each machine $l \in$ the set of all machines L **For** each stage k outside the two cross points**If** l has been allocated to both stage k and a stage between the two cross points for the newly employed beeRemove l from stage k for the newly employed bee**End if****End for****End For**

An illustrative example of the XO is given in Table 5a–c, whose symbolic representation is the same as that describing the LSO. First, two stages are randomly selected as the cross points: “W1, A2T2R, S2” and “W1, A2T4R, S2”. The green shades represent the machine allocation within the crossover points. Second, the machine allocation scheme is copied at the stages between the cross points from the onlooker bee to a newly employed bee. Then, the machine allocation scheme is copied at the stages outside the cross points from the employed bee to the newly employed bee. Finally, the duplicate machines “M2W2N1” at stage “W2, B2T2R, S1” and “M4W1N1” and “M4W1N2” at stage “W2, B2T2R, S2” are removed from the initially employed bee to obtain a newly employed bee. Red lines indicate the removal operations.

Table 5 Illustrative example of the XO

(a) Machine allocation scheme for the onlooker bee

Workshop	Product	Stage	Machine allocation
W1	A2T2R	S1	M1W1N1, M1W2N1
		S2	M3W1N1, M3W1N2
	A2T4R	S1	M2W2N1, M2W2N2
		S2	M4W1N1, M4W1N2
W2	B2T2R	S1	M3W2N1
		S2	M4W2N3
	B2T4R	S1	M3W2N2
		S2	M4W2N4

(b) Machine allocation scheme for the original employed bee

Workshop	Product	Stage	Machine allocation
W1	A2T2R	S1	M1W1N1
		S2	M3W1N1
	A2T4R	S1	Null
		S2	Null
W2	B2T2R	S1	M1W2N1, M2W2N1
		S2	M4W1N1, M4W1N2
	B2T4R	S1	M3W2N1
		S2	M4W2N2

(c) Machine allocation scheme for the newly employed bee

Workshop	Product	Stage	Machine allocation
W1	A2T2R	S1	M1W1N1
		S2	M3W1N1, M3W1N2
	A2T4R	S1	M2W2N1, M2W2N2
		S2	M4W1N1, M4W1N2
W2	B2T2R	S1	M1W2N1, M2W2N1
		S2	M4W1N1, M4W1N2
	B2T4R	S1	M3W2N1
		S2	M4W2N2

3.5 Non-Dominated Sorter

The design of the non-dominated sorter is based on NSGA-II [12], which is used to divide all solutions into different levels. The specific procedure of the sorter is as follows: (1) for each solution p , store each solution dominated by p in a solution set S_p and calculate n_p , the total number of solutions dominating p . (2) Put the solutions that have not been dominated by any solution (i.e., $n_p = 0$) into level 1. (3) For each solution p belonging to level 1, visit each solution q in S_p and set $n_q = n_q - 1$. $n_q = 0$ implies that q is dominated by solutions belonging to level 1, so put q into level 2. (4) As in Step 3, all solutions can be assigned to different levels. The

pseudo-code of the non-dominated sorter is presented in **Algorithm: Non-dominated sorter**.

Algorithm: Non-dominated sorter

Input: Population solutions P
Output: Pareto sorting solution set

For each solution $p \in P$
 $S_p = \emptyset$ // Store the solutions dominated by p
 $n_p = 0$ // The total number of solutions dominating p
For each solution $q \in P$
 If $p < q$ **then** // If p is dominated by q
 $S_p = S_p \cup \{q\}$ // Store q dominating p in S_p
 Else if $p > q$ **then** // If p dominates q
 $n_q = n_q + 1$ // Increase the domination count of q
 End if
End for
If $n_p = 0$ **then** // Store p in Pareto level 1
 $p_{\text{rank}} = 1$
 $\Gamma_1 = \Gamma_1 \cup \{p\}$
End if
End for
 $i = 1$
While $\Gamma_i \neq \emptyset$ **do**
 $Q = \emptyset$ // Store solutions in the next Pareto level
 For each $p \in \Gamma_i$
 For each $q \in S_p$
 $n_q = n_q - 1$ // Decrease the domination count of q
 If $n_q = 0$ **then** // Store q in the next Pareto level
 $q_{\text{rank}} = i + 1$
 $Q = Q \cup \{q\}$
 End if
 End for
 $i = i + 1$
 $\Gamma_i = Q$
End while

3.6 Crowding Distance Sorter

A crowding distance sorter based on NSGA-II [12] is employed. The specific procedure for the sorter is as follows: (1) the crowding distance of each solution is set to 0. (2) For each objective m , the solution set is sorted according to the objective function value in ascending order. We then set the crowding distances of the first and last solutions to infinity. Finally, other solutions are assigned a crowding distance value equal to the absolute normalized difference in the function values of two adjacent solutions. The corresponding procedure for the crowding distance sorter is described in **Algorithm: Crowding distance sorter**.

Algorithm: Crowding distance sorter

Input: solution set $L = \{L_1, L_2, \dots, L_n\}$
Output: the crowding distance $dist(i)$ for each solution

For each solution $i \in L$, set $dist(i) = 0$, **end if**

For each objective m
 $L = \text{sort}(L, m)$
 $dist(1) = dist(n) = \infty$
 $d = f_m^{\max} - f_m^{\min}$
 For $i = 2$ to $(n - 1)$
 $dist(i) = dist(i) + (f_m(i+1) - f_m(i-1)) / d$
 End for
End for

4 Numerical Experiments and Comparisons

4.1 Experimental Instances and Parameter Settings

In this section, the performance of DMOABC in solving the EDTMAP is evaluated. Numerical experiments of size $n \times m$ were designed. The number of workshops $n \in \{2, 3, 4\}$ and the number of product types $m \in \{2, 3, 4\}$ in each workshop constitute the problem $n \times m$. Other parameters include the runtime of each machine, and the production efficiency was set as follows: $t_{\text{minute}} = 60$, $t_{\text{hour}} = 24$, $t_{\text{day}} = 7$, $\eta = 0.85$. In addition, we assumed that the production weight of each type of product is equal.

Two classic multi-objective optimization algorithms, NSGA-II and SPEA2, were selected for comparison with DMOABC. To guarantee comparison fairness, all the parameters in the various algorithms were set consistently. The number of function evaluations (NFEs) is used as the control variable. In DMOABC, NSGA-II, and SPEA2, the NFEs were all set to $n \times m \times 10^5$, and the population size was 100. In NSGA-II and SPEA2, the crossover probability was 0.9, and the mutation probability was 0.2. All of the algorithms were coded in Java and implemented on an Intel Core 3.0-GHz PC with 8-GB memory.

4.2 Performance Evaluation Indices

Because the result of a multi-objective optimization problem is a Pareto solution set, the performance of our proposed multi-objective optimization algorithm cannot be evaluated merely by the values of the objective functions. The evaluation indices should indicate the dominance between the solutions, how close the Pareto solution set is to the true Pareto front (PF), and how uniformly the Pareto solution set is distributed. Therefore, this study uses two indices to measure the performance of DMOABC for solving the EDTMAP.

- Set coverage (C -metric) [31]: this metric can directly indicate the dominance between two Pareto solution sets. A and B denote two approximations of the PF,

and $C(A, B)$ represents the percentage of the solutions in B that are dominated by at least one solution in A , i.e.,

$$C(A, B) = \frac{|\{u \in B | \exists v \in A : v \text{ dominate } u\}|}{|B|}, \quad (9)$$

$C(A, B)$ is not necessarily equal to $1 - C(B, A)$. $C(A, B) = 1$ indicates that all solutions in B are dominated by some solutions in A , and $C(A, B) = 0$ implies that no solution in B is dominated by a solution in A . Hence, $C(A, B) > C(B, A)$ implies that the Pareto solution set of A is better than that of B .

- Distance from representatives in the PF (D -metric) [31]: this metric is a comprehensive index reflecting both the diversity and convergence of a Pareto solution set. P^* denotes the true PF. A represents an approximation of the PF. The average distance from P^* to A is defined as follows:

$$D(A, P^*) = \frac{\sum_{v \in P^*} d(v, A)}{|P^*|}, \quad (10)$$

where $d(v, A)$ is the minimum Euclidean distance between v and the points in A . $|P^*|$ is the number of solutions in P^* . A smaller $D(A, P^*)$ denotes a better solution set. Because the true PF is unknown, in this study, P^* was used as the non-dominated solution set found by all algorithms for each instance.

4.3 Experimental Results and Analysis

In this section, a statistical analysis of the comparison results for the three algorithms is presented. Normalization was used to eliminate the influence of the various magnitudes of the objective value. Each scale of the problem contained three other test cases, each of which ran ten times independently. The mean of the C -metric is shown in Table 6, and the mean of the D -metric is analyzed in Table 7. The optimal values are set in bold font among these three algorithms. A testing instance with a size of 3×2 was selected to compare the PF of these algorithms, as shown in Figure 2. The horizontal axis represents the inverse of the machine scheduling distance, whereas the vertical axis denotes production. The red circle points “○” denote the PF of DMOABC, the green rectangular points “□” indicate the PF of NSGA-II, and the blue triangle points “△” are the PF of SPEA2.

These statistical results demonstrate that DMOABC performs better than the other algorithms on all test instances in terms of the C -metric and D -metric. It can be observed from Table 6 that the C -metric values

Table 6 C -metric comparison for DMOABC, NSGA-II, and SPEA2

Problem	$C(\text{DMOABC}, \text{NSGA-C})$	$C(\text{NSGA-C}, \text{DMOABC})$	$C(\text{DMOABC}, \text{SPEA2})$	$C(\text{SPEA2}, \text{DMOABC})$
2×2	0.9033	0.0138	0.9470	0.0215
2×3	0.8875	0.0396	0.9264	0.0257
2×4	0.9604	0.0221	0.9335	0.0338
3×2	0.9535	0.0184	0.7621	0.1065
3×3	0.9182	0.0397	0.5731	0.1794
3×4	0.9259	0.0333	0.4342	0.2216
4×2	0.9022	0.0507	0.4766	0.2449
4×3	0.9164	0.0366	0.4633	0.2324
4×4	0.8631	0.0555	0.3987	0.2703
Mean	0.9145	0.0344	0.6572	0.1485

The optimal values are set in bold font among these three algorithms

Table 7 D -metric comparison for DMOABC, NSGA-II, and SPEA2

Problem	DMOABC	NSGA-S	SPEA2
2×2	0.0107	0.0560	0.0656
2×3	0.0117	0.0661	0.0868
2×4	0.0156	0.0812	0.1109
3×2	0.0166	0.0704	0.0949
3×3	0.0243	0.0712	0.1266
3×4	0.0265	0.0818	0.1355
4×2	0.0236	0.0725	0.1107
4×3	0.0289	0.0787	0.1334
4×4	0.0386	0.0924	0.1415
Mean	0.0218	0.0745	0.1118

The optimal values are set in bold font among these three algorithms

obtained for DMOABC are more extensive than those obtained for the other two algorithms. This indicates that those obtained for DMOABC dominate most of the solutions obtained for NSGA-II and SPEA2. In Table 7, the statistical results for the D -metric reveal that the solutions obtained by DMOABC are closer to the ideal PFs than those obtained by the other two algorithms. Figure 2 shows that those obtained by DMOABC dominate most of the solutions obtained by NSGA-II and SPEA2.

The superiority of DMOABC is attributed to the following facts. First, in the employed bee phase, the LSO can improve the quality of each solution in the population. Second, in the onlooker bee phase, the XO can improve population quality because employed bees and onlooker bees share nectar information. Finally, in the scout bee phase, each scout bee can abandon a solution that has not been improved during the limited number of generations to avoid falling into a local optimum. In summary, according to the above analysis, we can conclude that DMOABC is effective for solving the multi-objective EDTMAP.

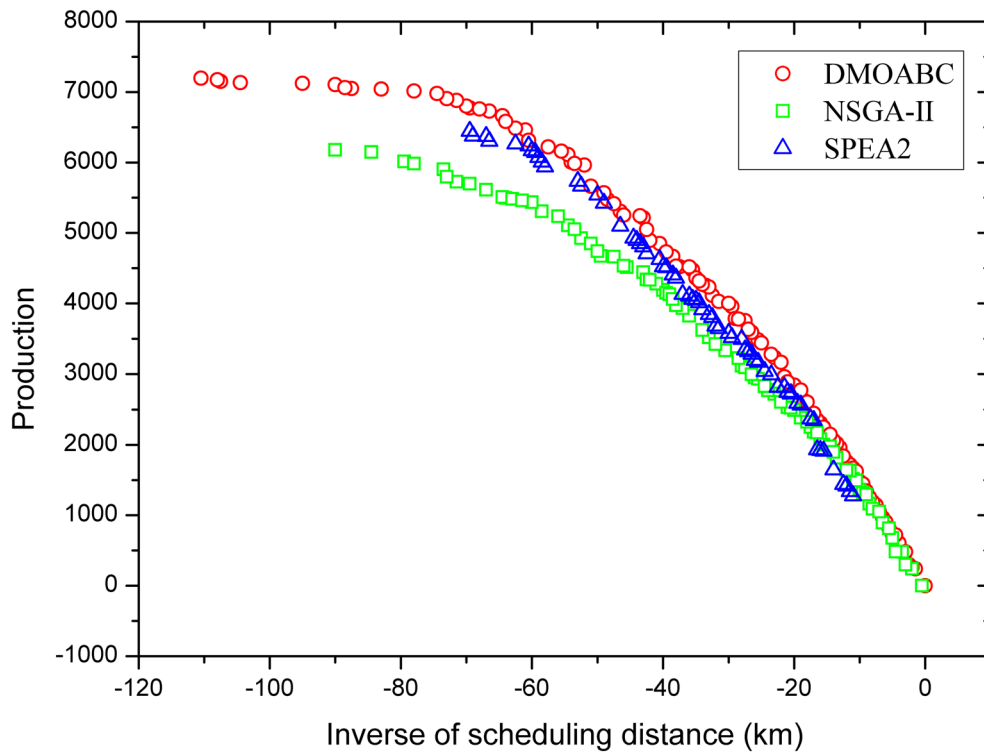


Figure 2 Pareto front comparison for DMOABC, NSGA-e, and SPEA2

5 Real-World Case Study and Comparisons

In this section, the proposed method is applied to a real-world case involving electronic device testing workshops belonging to a Chinese company that produces wireless radio-frequency modules (WRFMs).

5.1 Description of a Real-World Case

A WRFM is a modular digital radio product, a high-performance professional radio module for data transmission implemented through digital signal processing and radio technologies. WRFMs have many applications, such as in wireless remote controllers, remote garage doors, remote control lighting, communication, and security. A diagram of a WRFM is shown in Figure 3.

In this real-world case study, there are three workshops in the factory. Each workshop tests three types of WRFM. The actual WRFM production line for each workshop is shown in Figure 4. In actual production, each type of WRFM is tested in three stages. There are various types of testing machines with other interfaces in each workshop. All of these machines can be allocated to any workshop. To maximize both the total production and the inverse of the total scheduling distance of the machines, a reasonable testing machine allocation scheme is required. The testing time and

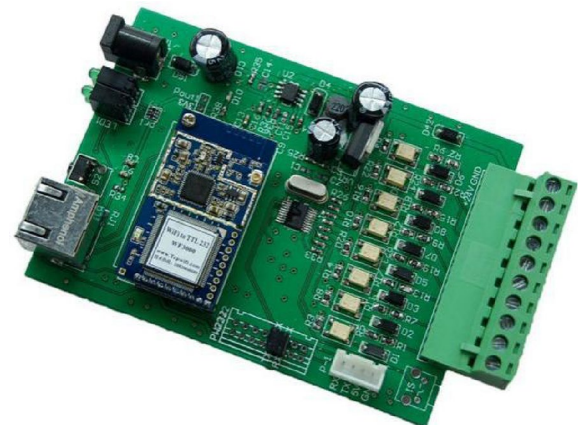


Figure 3 Diagram of a WRFM

requirements associated with 16 compatibilities and quantities for interfaces of each type of WRFM at the various stages are listed in Table 8a, information concerning available testing machines in each workshop is provided in Table 8b, and the distances among the three workshops are listed in Table 8c.



Figure 4 Production line of WRFM

5.2 Solving the Real-World Case Using the Proposed DMOABC

DMOABC and its comparison algorithms (i.e., NSGA-II and SPEA2) were applied to this real-world case. All the parameter settings were the same as those in Section 4. Each algorithm was run independently 30 times. The statistical results for the C -metric are shown in Table 9, and the statistical results for the D -metric are summarized in Table 10. In terms of the C -metric, the values obtained for DMOABC were significantly larger than those obtained for the other algorithms. This shows that the solutions obtained by DMOABC dominate most of the PF solutions obtained by NSGA-II and SPEA2. In terms of the D -metric, the values obtained by DMOABC were smaller than those obtained by the other algorithms. This implies that the PF accepted by the DMOABC is closer to the ideal PF than that obtained by NSGA-II and SPEA2. The above results show that DMOABC outperforms the comparison algorithms in solving this case.

Figure 5 shows the PFs obtained using the different multi-objective optimization algorithms. In this figure, the horizontal axis represents the inverse of the scheduling distance of the machines, while the vertical axis represents production. The red circles “○” denote the PF obtained by DMOABC, the green rectangles “□” indicate those obtained by NSGA-II and the blue triangles “△” are those obtained by SPEA2. It is easily observed that most of the solutions obtained by DMOABC can dominate those obtained by NSGA-II and SPEA2. Based on this figure, we can conclude that the DMOABC is superior to its comparison algorithms in terms of convergence performance.

Points A and B in Figure 5 correspond to the two extreme objectives of this case. More specifically, Point A is the machine allocation scheme with the maximum production on the PF for DMOABC. Point B is the machine allocation scheme with the minimum scheduling distance of machines

on the PF for DMOABC. Point A is $(-130.5, 6686.33)$, and Point B is $(0, 0)$. The maximum production values on the PF for NSGA-II and SPEA2 are 5985.04 and 6013.92, respectively. Therefore, compared with NSGA-II and SPEA2, DMOABC improved the production capacity by 11.72% and 11.18%, respectively. Decision makers can select an appropriate testing machine allocation scheme from PF solutions according to real-world production needs. For instance, the machine allocation scheme at Point A might be selected if the executors wish to have more production without considering the increase in the machine scheduling cost. When the production target is given, the executors expect the machine scheduling costs to be as high as possible. In this case, the solution on the PF can be selected to obtain the machine allocation scheme that both satisfies the production target and minimizes the machine scheduling cost. The allocation scheme of Point A is shown in Table 11, and Point B indicates that no machine is scheduled. From the allocation scheme of Point A, we can observe that six M1F1, two M1F2, and two M2F3 are scheduled for stage 1 of product C2T2R in workshop 3. This allocation scheme demonstrates one characteristic of the EDTMAP: that machines are compatible. Additionally, more machines are scheduled to increase the production capacity, but the total scheduling distance increases; thus, the scheduling cost increases. Hence, there is a clear conflict between the production and scheduling distance, which confirms that the EDTMAP is a typical multi-objective problem.

6 Conclusions and Future Works

In this study, the real-world electronic device testing machine allocation problem (EDTMAP) was investigated from both theoretical and actual production perspectives. First, we formulated a multi-objective mathematical model that considers the total production and machine-scheduling distance. We developed a discrete multi-objective artificial bee colony algorithm (DMOABC) to address this problem. Furthermore, we compared the proposed DMOABC with two state-of-the-art multi-objective algorithms: NSGA-II and SPEA2. The experimental results show that DMOABC solves the EDTMAP more effectively compared with the other algorithms. Finally, the proposed mathematical model and algorithm were successfully applied to a real-world EDTMAP. The results demonstrate that our method can effectively guide actual machine allocation.

The contributions of this research can be summarized as follows:

- 1) An electronic device testing machine allocation model was designed and extended to other types of machine allocation models.
- 2) A discrete multi-objective artificial bee colony algorithm was developed to deal with the EDTMAP.

Table 8 Product and testing machine information

Workshop	Product	Testing time (interface quantity) at stage 1	Testing time (interface quantity) at stage 2	Testing time (interface quantity) at stage 3
(a) Testing time (and interface quantity constraint) of WRFMs at various stages				
W1	A2T2R	7.54 (2)	9.45 (2)	8.12 (2)
	A2T4R	10.18 (4)	9.53 (4)	11.24 (2)
	A4T4R	6.89 (4)	5.90 (4)	7.51 (4)
W2	B2T2R	9.25 (2)	8.85 (2)	7.39 (2)
	B2T4R	8.87 (4)	6.71 (4)	9.41 (2)
	B4T4R	5.91 (4)	7.37 (4)	9.40 (4)
W3	C2T2R	8.54 (2)	8.18 (2)	9.07 (2)
	C2T4R	7.60 (4)	8.59 (4)	11.61 (2)
	C4T4R	8.54 (4)	7.80 (4)	8.07 (4)
Machine	Matching stage	Interface quantity	Workshop	Quantity
(b) Available machines in workshop				
M1	S1	2	W1	8
			W2	6
			W3	0
M2	S1	4	W1	8
			W2	4
			W3	9
M3	S2	2	W1	1
			W2	8
			W3	0
M4	S2	4	W1	8
			W2	4
			W3	5
M5	S3	2	W1	8
			W2	11
			W3	7
M6	S3	4	W1	3
			W2	7
			W3	3
Workshop	Distance from W1 (km)	Distance from W2 (km)	Distance from W3 (km)	
(c) Distance between workshops				
W1	0.5	1.5	3	
W2	1.5	0.5	2	
W3	3	2	0.5	

Table 9 C-metric comparison for DMOABC, NSGA-II, and SPEA2

Problem	C(DMOABC, NSGA-C)	C(NSGA-C, DMOABC)	C(DMOABC, SPEA2)	C(SPEA2, DMOABC)
3×3	0.9395	0.0316	0.5688	0.2128

The optimal values are set in bold font among these three algorithms

Table 10 C-metric comparison for DMOABC, NSGA-II, and SPEA2

Problem	DMOABC	NSGA-S	SPEA2
3×3	0.0319	0.0743	0.1121

The optimal values are set in bold font among these three algorithms

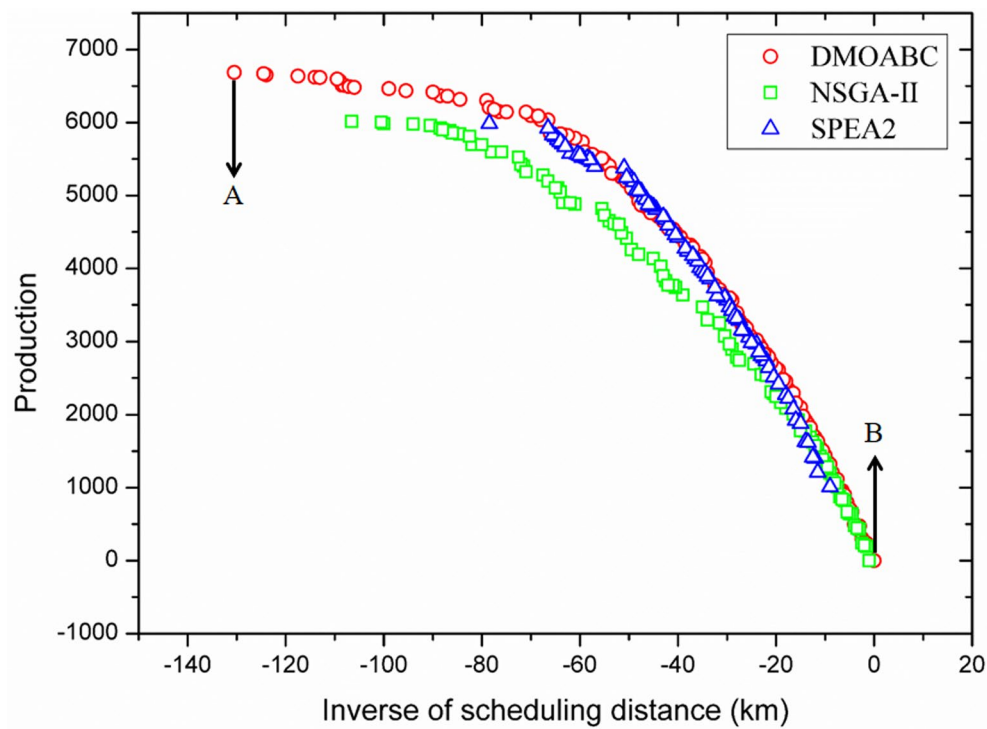


Figure 5 Pareto front solutions of the case

Table 11 Allocation scheme at Point A

Workshop	Product	Stage 1	Stage 2	Stage 3
W1	A2T2R	Null	Null	Null
	A2T4R	Null	Null	Null
	A4T4R	M2F1 (6)	M4F1 (5)	M6F1 (3)
		M2F2 (2)	M4F2 (3)	M6F2 (7)
W2	B2T2R	M2F3 (3)	M4F3 (1)	M6F3 (2)
		M1F1 (1)	M3F2 (2)	M5F1 (1)
		M1F2 (3)		M5F2 (2)
	B2T4R	M2F1 (2)	M4F1 (3)	M5F1 (2)
W3		M2F2 (2)	M4F3 (3)	M5F2 (7)
		M2F3 (4)		
	B4T4R	Null	Null	Null
	C2T2R	M1F1 (6)	M3F1 (1)	M5F1 (2)
		M1F2 (2)	M3F2 (6)	M5F2 (1)
		M2F3 (2)	M4F2 (1)	M5F3 (7)
			M4F3 (1)	
	C2T4R	Null	Null	Null
	C4T4R	Null	Null	Null

Three key operators, i.e., FSG, LSO, and XO, were designed for and implemented in the algorithm. FSG is used to generate a feasible machine allocation scheme, and LSO and XO are used to balance local and global searches.

- 3) By analyzing the comparison results of the numerical experiments, we demonstrated that DMOABC is robust and highly competitive with NSGA-II and SPEA2 for solving the EDTMAP.
- 4) We applied γ DMOABC to real-world electronic device testing workshops. The results demonstrate that our method can guide the allocation of an actual electronic device testing machine.

Although our proposed method can effectively solve the EDTMAP, there are still some limitations to this study. One limitation is that our proposed mathematical model only considers the constraints of compatibility and number of interfaces. Therefore, additional conditions should be considered. Another limitation is that the model only considers two indices: productivity and scheduling distance. With increasing awareness of the need for environmental protection, environmental pollution indices need to be introduced into our model. In the future, the following studies can be considered: one research direction is to modify our mathematical model to align it with the actual production environment. Another exciting research direction is to introduce new evaluation indices, such as environmental pollution indices, to better guide the allocation of electronic device testing machines.

Authors' Contributions

JX was in charge of the entire trial, XL assisted with the problem analyses, and LG wrote the manuscript. All authors read and approved the final manuscript.

Authors' Information

Jin Xie, born in 1992, is currently a master's candidate at *State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, China*. He received his bachelor's degree from *China University of Geosciences, China*, in 2015. His main research interests include shop scheduling and algorithm optimization.

Xinyu Li, born in 1985, is currently a professor at *State Key Laboratory of Digital Manufacturing Equipment and Technology at Huazhong University of Science and Technology, China*. He received his Ph.D. degree in industrial engineering from *Huazhong University of Science and Technology, China*, in 2009. His main research interests include intelligent manufacturing systems, shop scheduling, intelligent optimization, and machine learning.

Liang Gao, born in 1974, is currently a professor at *State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, China*. He received his Ph.D. degree in mechatronic engineering from *Huazhong University of Science and Technology, China*, in 2002. His main research interests include intelligent optimization methods and their applications in design and manufacturing.

Funding

Supported by National Key R&D Program of China (Grant No. 2019YFB1704600), National Natural Science Foundation of China (Grant Nos. 51825502, 51775216), and Program for HUST Academic Frontier Youth Team of China (Grant No. 2017QYTD04).

Competing interests

The authors declare no competing financial interests.

Received: 13 January 2021 Revised: 28 September 2022 Accepted: 29 September 2022

Published online: 27 October 2022

References

- [1] B Sarkar, S Saren. Product inspection policy for an imperfect production system with inspection errors and warranty cost. *European Journal of Operational Research*, 2016, 248(1): 263-271.
- [2] P Singh, D Deschrijver, D Pissort, et al. Adaptive classification algorithm for EMC-compliance testing of electronic devices. *Electronics Letters*, 2013, 49(24): 1526-1528.
- [3] S P Orlov, A N Vasilchenko. Intelligent measuring system for testing and failure analysis of electronic devices. *Proceedings of the XIX IEEE International Conference on Soft Computing & Measurements*, Saint-Petersburg, Russia, 2016: 401-403.
- [4] S Stoyanov, M Ahsan, C Bailey, et al. Predictive analytics methodology for smart qualification testing of electronic components. *Journal of Intelligent Manufacturing*, 2019, 30(3): 1497-1514.
- [5] D Gallo, R Langella, M Luiso, et al. A new test procedure to measure power electronic devices' frequency coupling admittance. *IEEE Transactions on Instrumentation and Measurement*, 2018, 67(10): 2401-2409.
- [6] F Thouin, M Coates. Equipment allocation in video-on-demand network deployments. *Proceedings of the ACM Transaction on Multimedia Computing, Communications and Applications*, Augsburg, Germany, 2008, 5(1): 1-22.
- [7] J R Jaramillo, A R Mckendall. Metaheuristics for the integrated machine allocation and layout problem. *International Journal of Operational Research*, 2010, 7(1): 74-89.
- [8] T C Poon, K L Choy, C K Cheng, et al. Effective selection and allocation of material handling equipment for stochastic production material demand problems using genetic algorithm. *Expert Systems with Applications*, 2011, 38(10): 12497-12505.
- [9] H Chehade, A Dolgui, F Dugardin, et al. Multi-objective approach for production line equipment selection. *Management and Production Engineering Review*, 2012, 3(1): 4-17.
- [10] A Mohtashami. A new hybrid method for buffer sizing and machine allocation in unreliable production and assembly lines with general distribution time-dependent parameters. *The International Journal of Advanced Manufacturing Technology*, 2014, 74(9): 1577-1593.
- [11] W Du, Y Tang, S Y Leung, et al. Robust order scheduling in the discrete manufacturing industry: a multiobjective optimization approach. *IEEE Transactions on Industrial Informatics*, 2018, 14(1): 253-264.
- [12] K Deb, A Pratap, S Agarwal, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182-197.
- [13] E Zitzler, M Laumanns, L Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm. *Proceedings of Proc. Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, Athens, Greece, 2002, 95-100.
- [14] D Karaboga. An idea based on honey bee swarm for numerical optimization. *Technical Report-TR06*, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
- [15] D Karaboga, B Basturk. On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing*, 2008, 8(1): 687-697.
- [16] R Hedayatzadeh, B Hasanizadeh, R Akbari, et al. A multi-objective artificial bee colony for optimizing multi-objective problems. *Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering*, Chengdu, China, 2010: V5-277-V5-281.
- [17] R Akbari, R Hedayatzadeh, K Ziarati, et al. A multi-objective artificial bee colony algorithm. *Swarm and Evolutionary Computation*, 2012, 2: 39-52.
- [18] X Li, C Lu, L Gao, et al. An effective multiobjective algorithm for energy-efficient scheduling in a real-life welding shop. *IEEE Transactions on Industrial Informatics*, 2018, 14(12): 5400-5409.
- [19] X Li, S Xiao, C Wang, et al. Mathematical modeling and a discrete artificial bee colony algorithm for the welding shop scheduling problem. *Memetic Computing*, 2019, 11(4): 371-389.
- [20] G Gong, R Chiong, Q Deng, et al. A hybrid artificial bee colony algorithm for flexible job shop scheduling with worker flexibility. *International Journal of Production Research*, 2019, 58(14): 4406-4420.
- [21] C Öztürk, D Karaboga, B Gorkemli. Probabilistic dynamic deployment of wireless sensor networks by artificial bee colony algorithm. *Sensors*, 2011, 11(6): 6056-6065.
- [22] C Öztürk, D Karaboğa, B GÖRKEMLİ. Artificial bee colony algorithm for dynamic deployment of wireless sensor networks. *Turkish Journal of Electrical Engineering & Computer Sciences*, 2012, 20(2): 255-262.
- [23] S Mini, S K Udgata, S L Sabat. Sensor deployment and scheduling for target coverage problem in wireless sensor networks. *IEEE Sensors Journal*, 2014, 14(3): 636-644.
- [24] L Ma, X Wang, M Huang, et al. Two-level master-slave RFID networks planning via hybrid multiobjective artificial bee colony optimizer. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017, 99: 1-20.
- [25] I N Todorovic, S Petrovic. Bee colony optimization algorithm for nurse rostering. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2012, 43(2): 467-473.
- [26] M A Awadallah, A L Bolaji, M A Albetar, et al. A hybrid artificial bee colony for a nurse rostering problem. *Applied Soft Computing*, 2015, 35: 726-739.
- [27] K Buyukozkan, A Sarucan. Applicability of artificial bee colony algorithm for nurse scheduling problems. *International Journal of Computational Intelligence Systems*, 2014, 7(1): 121-136.
- [28] S C Hornig. Combining artificial bee colony with ordinal optimization for stochastic economic lot scheduling problem. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2014, 45(3): 373-384.
- [29] O Bulut, M F Tasgetiren. An artificial bee colony algorithm for the economic lot scheduling problem. *International Journal of Production Research*, 2014, 52(4): 1150-1170.
- [30] O Bulut, M F Tasgetiren. A discrete artificial bee colony algorithm for the economic lot scheduling problem with returns. *International Journal of Production Research*, 2014, 52(4): 551-557.
- [31] Q Zhang, H Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 2007, 11(6): 712-731.