

Article

Sustainable Scheduling of an Automatic Pallet Changer System by Multi-Objective Evolutionary Algorithm with First Piece Inspection

Qingmiao Liao *, Jianjun Yang and Yong Zhou

School of Mechanical Engineering and Automation, Beihang University, Beijing 100191, China; jjyang@buaa.edu.cn (J.Y.); zhoyong@buaa.edu.cn (Y.Z.)

* Correspondence: liaoqingmiao@gmail.com; Tel.: +86-1362-119-7201

Received: 30 January 2019; Accepted: 7 March 2019; Published: 12 March 2019



Abstract: In this study, the machining center with the Automated Pallet Changer (APC) scheduling problem considering the disturbance of the first piece inspection is presented. The APC is frequently used in industry practice; it is useful in terms of sustainability and robustness because it increases the machine utilization rate and enhances the responsiveness to uncertainties in dynamic environments. An enhanced evolutionary algorithm for APC scheduling (APCEA) is developed by combining the multi-objective evolutionary algorithm with APC simulation. The dynamic factors in the simulation model include the pass rate of the first piece inspection (FPI) and the adjusted time when the FPI is unpassed. The proposed APCEA defines the non-robust gene based on the risk combination of the first piece inspection, and screens the non-robust gene in the genetic operation, thus improving the solution quality under the same computation times. Compared with the other three multi-objective evolutionary algorithms (MOEAs), it is demonstrated that the proposed APCEA produces the best result among the four methods. The proposed APCEA has been embedded into the manufacturing execution system (MES) and successfully applied in a manufacturing plant. The application value of the proposed method is verified by a practical example.

Keywords: automatic pallet changer system; first piece inspection; multi-objective evolutionary algorithm; multi-phase scheduling

1. Introduction

With the advent of the era of industrial 4.0, machining centers with an Automated Pallet Changer (APC) are becoming more and more common in modern factories. Because complex parts are often processed in aerospace factories, the first piece inspection (FPI) is generally undertaken to prevent the whole batch having quality problems. The FPI checks the processing quality of the first workpiece in each operation of the machining process. After the first workpiece of task Lot1 on pallet A has been sent for FPI, the APC system can immediately switch to another pallet B to process task Lot2. If the FPI of Lot1 is passed, pallet A can be switched back to continue processing task Lot1. Due to the use of APC, a machine can have two or more pallets, which can effectively reduce the idle time of the machine and improve the overall utilization rate.

Due to the complex characteristics of the multi-variety and small batch production organization, the FPI has a low pass rate in aerospace manufacturing plants. Therefore, the APC scheduling system is the most important part of shop floor scheduling. In addition, the dynamic factors considered in this study include the pass rate of the FPI and the adjusted time when the FPI is unpassed.

Recently, many researchers have focused on sustainable scheduling. Giret et al. [1] pointed out that sustainability must urgently be taken into account at all levels, not only at the strategic level, but

also at the tactic and operational levels to have a complete transmission belt from global policies to operational activities. Kleindorfer [2] pointed out that operational management, whether research or practice, can and should contribute to sustainable development. Mouzon et al. [3] addressed the energy consumption of a production facility by minimizing the expended energy of manufacturing equipment using operational methods. Lee and Ha [4] proposed a sustainable integrated process planning and scheduling optimization using a genetic algorithm with an integrated chromosome representation. Zhang [5] proposed the sustainable scheduling of cloth production processes by the multi-objective genetic algorithm with tabu-enhanced local search. These studies show that operational management plays an important role on sustainable development.

Because dynamic events occur frequently in the shop floor, many optimization methods have been proposed to solve the dynamic scheduling problem. Vieira et al. [6] presented definitions appropriate for most applications of rescheduling manufacturing systems and described a framework for understanding rescheduling strategies, policies, and methods. Ouelhadj et al. [7] defined the problem of dynamic scheduling and provided a review of the state-of-the-art of currently developing research on dynamic scheduling. Alcaide et al. [8,9] developed a heuristic approach to solve the stochastic open-shop scheduling problems where random machine breakdowns can happen. Goren et al. [10] addressed the issue of finding robust and stable schedules with respect to random disruptions by surrogate measures. Siedlak et al. [11] focused on the development, implementation, and evaluation of a robust simulation-based planning and optimization methodology which attempts to reduce the risk of schedule overruns by identifying plans that are robust to various process delays and disruptions.

Current task scheduling mainly concerns the availability of machining resources, rather than the potential errors after scheduling. Sortrakul et al. [12] developed a heuristics method based on genetic algorithms to solve an integrated optimization model for production scheduling and preventive maintenance planning. Zhang et al. [13] proposed a goal-guided neighborhood function to solve a robust schedule for a flexible job-shop scheduling problem with flexible workdays.

The actual production scheduling problem needs not only to optimize one goal, but also to optimize multiple conflicting goals. Multi-objective evolutionary algorithms (MOEAs) have attracted attention from researchers because of their intelligence, parallelism, robustness, and good adaptability, and their capability for global search.

To this end, many researchers have used various MOEAs to solve the multi-objective scheduling problem. Yulan et al. [14] proposed a multi-objective integrated optimization method on preventive maintenance planning and production scheduling for a single machine. Coello et al. [15] introduced coevolutionary multi-objective evolutionary algorithms (MOEAs) and hybridizations of MOEAs with local search procedures. Zhou et al. [16] proposed three hyper-heuristic methods for coevolution of the machine assignment rules and job sequencing rules to solve the multi-objective dynamic flexible job shop scheduling problem. Liao et al. [17] proposed a SVM-based intelligent dispatcher for parallel machine scheduling with sequence-dependent setup times. Gen and Lin [18,19] gave a research review for the multi-objective evolutionary algorithm for manufacturing scheduling problems. Fang et al. [20] achieved real-time monitoring of workshop production information and logistics information, and laid a solid foundation for workshop dynamic scheduling.

Godinho et al. [21] reviewed the literature regarding genetic algorithms (GAs) applied to flexible manufacturing system (FMS) scheduling. They found that most studies have focused on complex environments concerning both the routing flexibility and the job complexity, but a minority of papers simultaneously considered the pallets' capacity constraint. However, a very small amount of existing research is relevant to this subject. Diabat et al. [22] presented a heuristics algorithm for scheduling jobs in a machining center equipped with multiple-pallet Automated Pallet Changers.

Although many studies have been carried out on the sustainable scheduling problem, which have focused on equipment failure maintenance, inaccurate working hours, and unstable material supply, there is still a lack of research focus on sustainable scheduling of APC systems. Additionally, the risk

combination of the first piece inspection has not been considered before. Since machining centers equipped with APC in aerospace factories have been frequently used in recent years, it is important to investigate this type of scheduling problem. The main factor affecting the scheduling stability is the product processing quality because Computerized Numerical Control(CNC) machines have a low failure rate.

The motivation of this study is to increase the scheduling stability of the machining center with APC, which is mainly influenced by poor-quality processing products. To the best of our knowledge, our study serves as the first attempt to use the multi-objective evolutionary algorithm with non-robust gene avoidance to tackle the APC scheduling problem. Compared to previous studies, our contributions are as follows:

- (1) A multi-phase scheduling of machining center with APC considering the risk combination of the first piece inspection (RCFPI) is presented. The mathematical model of the problem is established, and an APCEA method is proposed to solve this type of scheduling problem.
- (2) By referring to the idea of tabu search, a genetic operation that can effectively avoid non-robust gene is studied. Additionally, a biased roulette method is proposed, which can effectively increase the mutation probability of the RCFPI. The APCEA defines the RCFPI in advance through domain knowledge of APC scheduling and screens the non-robust gene in the genetic operation, thus improving the solution quality under the same number of fitness evaluations (NFE).
- (3) Two kinds of APC scheduling model are proposed: the original simulation model and the simplified model. The experimental results show that the two models have a very high correlation, so the simplified model is in the evolutionary process to obtain the Pareto solution set. After the evolution has been completed, the original simulation model is used to evaluate the individual in the final Pareto set to make a good balance between the solving speed and the solution quality. The simplified model can be used as a surrogate model for evolutionary algorithm.
- (4) The proposed algorithm makes full use of the automatic pallet changer to reduce the downtime of the spindle of the machining center, reduce the energy consumption (EC) caused by idle and frequent off-on, maintain the stable running of the equipment and prolong the service life of the equipment, thus realizing a sustainable scheduling of APC system.

The rest of the paper is structured as follows. Section 2 describes the problem model investigated in this study. Detailed descriptions of the proposed APCEA are given in Section 3. Section 4 presents the design of experiments and the results. A case study is presented in Section 5. Finally, Section 6 concludes this study followed by a discussion on future research.

2. Problem Model

This study considers multiple phase of the process's execution. A task will go through three phases in sequence, which are the setup phase, the first piece inspection phase, and the normal phase. A scene diagram of the APC system with first piece inspection is shown in Figure 1. There are two tasks on the machining center with APC, task O_i is assigned to the pallet A while task O_j on the pallet B. Firstly, the machine spindle works on pallet A because task O_i has a higher priority than task O_j . When task O_i is submitted to inspection, the spindle switches to pallet B to process task O_j . Secondly, the FPI of O_j is also submitted after the first piece has been completed. Since the FPI result of task O_j is qualified, task O_j enters the normal processing phase and begins to process the remaining pieces. Task O_i requires an adjustment because the first piece is unpassed. After the adjustment of O_i has been completed, the machine spindle is switched back to pallet A to process another first piece. Task O_i also enters the normal processing phase because the second FPI result passed. Finally, while both tasks are in the normal phase, the executions of task O_i and task O_j can be staggered until both tasks finished to reduce spindle downtime due to work piece changing. However, in some case pallets may not be switched, for example, the task on one pallet is urgent. When a task finished, the pallet assigned to it can be assigned to other tasks. The red line represents the running process of the machining

center spindle. The red line between the two pallets indicates pallet switching, and the red dotted line indicates that the switching is optional. The white line indicates that the spindle is idle, because both pallets are involved in the FPI activities.

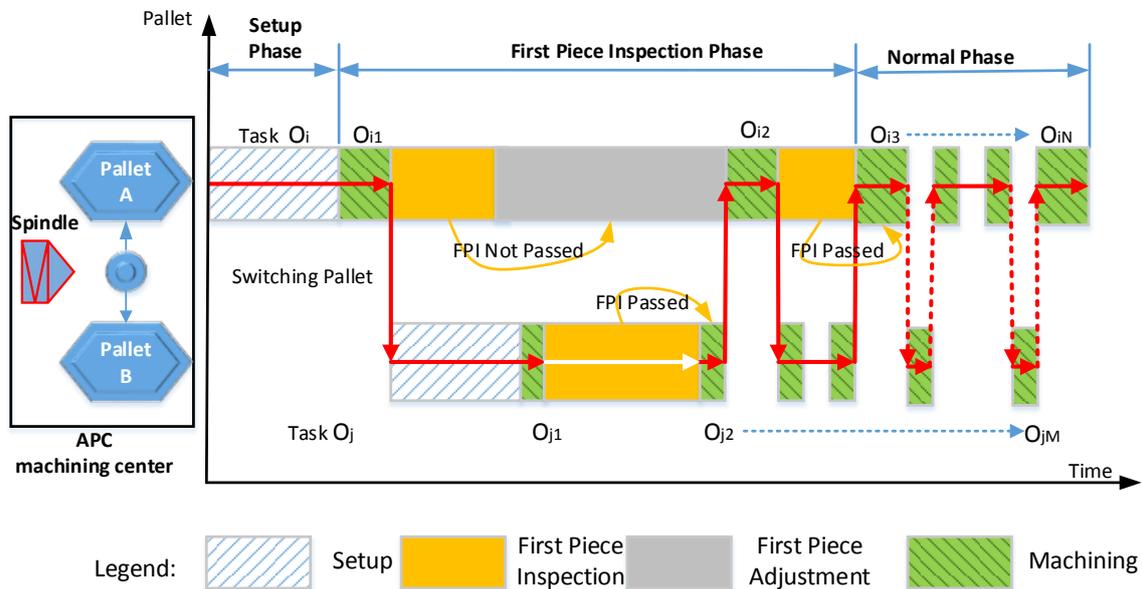


Figure 1. Scene diagram of the Automated Pallet Changer machining center with first piece inspection.

As can be seen from Figure 1, if both tasks O_i and O_j have a low FPI pass rate, the risk of machine spindle shutdown will be greatly increased. In this case, we call task O_i and O_j a risk combination of the first piece inspection (RCFPI). RCFPI will undoubtedly affect the robust performance of the scheduling solution. To identify RCFPI, this study defines the risk degree of combination (RDFPI):

$$RDFPI_{ij} = (1 - fp_i)(1 - fp_j) \quad (1)$$

When RDFPI is greater than a threshold coefficient c , these two tasks will be considered as a RCFPI. Threshold coefficient c is set by the actual experience and is normally set to 0.2.

To easily establish the model, six assumptions are made as follows:

- (1) A machine spindle can only process one workpiece on one pallet at a time.
- (2) A task requires at least one first piece inspection, which is no longer required after passing it.
- (3) To reduce the cost of fixture adjustments, all pieces of a task should be assigned to only one pallet.
- (4) When the FPI is unpassed, no spindle assistance is required for adjustment.
- (5) The adjustment times are independent of each other and follow a uniform distribution.
- (6) This study does not consider the impact of random inspection and rework during the process.

In this study, the scheduling problem of two pallets is considered. The machining center with APC scheduling problem considering the first piece inspection in this study can be expressed as follows: There is a machining center with APC and the number of pallets is m . There are n part tasks [23]. Part task is a batch task varied with part item id and batch number. Each part task has multiple work pieces and contains some operations according to its process route, but only one operation passes through the machining center with APC. This study does not consider the problem of reentrant. To choose the optimal robust sequence for the tasks, three different objectives representing the general performance are considered in this study. The makespan (C_{max}) is equivalent to the completion time of the last task. The total flow time (TFT) gives a measure of the lean level. The total weighted tardiness (TWT) represents the goal of on-time delivery [24].

The objective of the machining center with APC scheduling problem is to achieve the minimum of the above three indicators. According to the standard expression ($\alpha | \beta | \gamma$) of the scheduling problem

proposed by Graham et al. [25], the APC scheduling problem considering first piece inspection can be expressed as $1|APC, FPI|TWT, C_{max}, TFT$. Because the scheduling problem $1||\sum w_j T_j$ is already a strong NP-hard problem, the problem studied in this research is also a NP-hard problem. The mathematical model of APCEA problem is as follows:

MIN

$$TWT = \sum_{j=1}^n w_j \times \max(C_j - d_j, 0). \quad (2)$$

$$C_{max} = \max(C_j) \quad (3)$$

$$TFT = \sum_{j=1}^n (C_j - r_j) \quad (4)$$

S.t.

$$r_j, s_j, p_j, q_j, f_j > 0, \forall j \in \{1, 2, \dots, n\} \quad (5)$$

$$x_{j,k} = \begin{cases} 1, & \text{if pallet } k \text{ is selected for task } o_j \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

$$\sum_{k=1}^m x_{j,k} = 1, \forall j \in \{1, 2, \dots, n\} \quad (7)$$

$$fa_{j,g} \sim U(5p_j, 10p_j), \forall j \in \{1, 2, \dots, n\}; \forall g \in \{1, 2, \dots, q_j\} \quad (8)$$

$$fp_{j,g} \sim B(1, p), \forall j \in \{1, 2, \dots, n\}; \forall g \in \{1, 2, \dots, q_j\} \quad (9)$$

$$fp_{j,g} = \begin{cases} 1, & \text{if the } g\text{th first piece inspection of task } O_j \text{ is unpassed} \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

$$fp_{j,g+1} \leq fp_{j,g}, \forall j \in \{1, 2, \dots, n\}; \forall g \in \{1, 2, \dots, q_j - 1\} \quad (11)$$

$$b_j = s_j + q_j \cdot p_j + f_j + \sum_{g=2}^{q_j} fp_{j,g-1} \cdot f_j + \sum_{g=1}^{q_j-1} fp_{j,g} \cdot fa_{j,g}, \forall j \in \{1, 2, \dots, n\} \quad (12)$$

$$c_j - r_j \geq b_j, \forall j \in \{1, 2, \dots, n\} \quad (13)$$

$$(c_j - c_i)(c_j - c_i - b_j) \cdot x_{j,k} \cdot x_{i,k} \geq 0, \forall i, j \in \{1, 2, \dots, n\}; \forall k \in \{1, 2, \dots, m\} \quad (14)$$

$$c_{j,g+1} - c_{j,g} \geq p_j + f_j \cdot fp_{j,g} + fa_j \cdot fp_{j,g}, \forall j \in \{1, 2, \dots, n\}; \forall g \in \{1, 2, \dots, q_j - 1\} \quad (15)$$

$$(c_{j,g} - c_{i,l})(c_{j,g} - c_{i,l} - p_j) \geq 0, \forall i, j \in \{1, 2, \dots, n\}; \forall g \in \{1, 2, \dots, q_j\}; \forall l \in \{1, 2, \dots, q_i\} \quad (16)$$

$$c_{j,q_j} = c_j, \forall j \in \{1, 2, \dots, n\} \quad (17)$$

Equations (2)–(4) define the three objective functions to be minimized, i.e., the total weighted tardiness TWT, the makespan C_{max} and the total flow time TFT. Equation (5) is variable restriction to define a task. Equations (6) and (7) state one task can only be processed on one pallet. Equation (8) shows the distribution range of the adjustment time when fail the FPI. Equation (9) states the FPI pass rate follows a 0–1 distribution. As is shown in Equation (10), the $fp_{j,g}$ is 1 means that the first piece inspection is unpassed. Equation (11) guarantees that once the first piece inspection of a task passes, the task does not need another inspection. Equation (12) defines a temp variable to indicate the total processing time of a task. Equation (13) ensures that enough time is given to complete the task. Equation (14) ensures that two tasks are not overlapping if both of them are assigned on the same pallet. Equation (15) guarantees that enough time is given when replaced with next piece task. For the cross operation of multiple pallets, Equation (16) guarantees that two pieces of two tasks are not overlapping because the spindle of the machining center can only work on one pallet at the same time. Equation (17) states the completion time of last piece task is the same as the completion time of the task.

3. Algorithm Design

In this section, a multi-objective evolutionary algorithm with non-robust gene avoidance is presented to solve the studied scheduling problem. In the following subsections, we introduce the key components of our algorithm, which are designed to adapt to the characteristics of APC scheduling problem.

3.1. The Overall Design of the Proposed Algorithm

Firstly, the APCEA generates an initial population of N . After the evolution process of tournament selection, PMX crossover, and SWAP mutation, a new offspring is generated. If the offspring individual contains a RCFPI, the genetic operation need to be re-executed with a certain probability. When the number of offspring is equal to N , all offspring individuals need to be evaluated in terms of fitness. The parent population and the offspring population are merged to form a new population. Then, the non-dominant rank of each individual is obtained by the Fast Non-Dominated Sort [26] of the new population, and then the crowding degree of each individual is calculated. Finally, the individuals of the new population are sorted according to the non-dominance degree and the crowding degree, and the top N individuals are archived and will continue to be involved in the next generation. If the number of generations of the evolution is less than the maximum number of generations, the genetic operation continues to produce new offspring. Otherwise, the Pareto frontier is acquired, and the case needs to be re-scheduled using the APC simulation model to obtain the final scheduling target value.

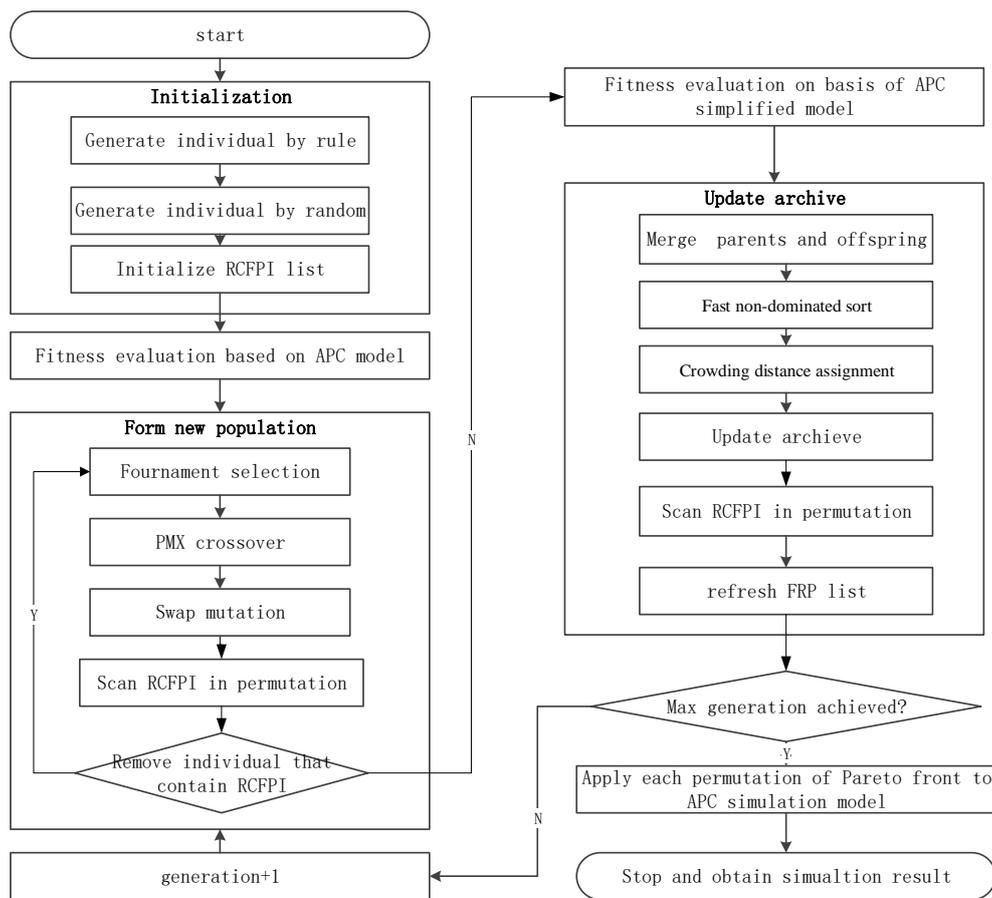


Figure 2. The flow chart of the enhanced evolutionary algorithm for APC scheduling (APCEA).

In this study, the coding consistent with RCFPI is considered the non-robust gene, which will be suppressed as much as possible in the evolutionary process. During the evolution process of APCEA,

the RCFPI list plays an important role in inhibiting non-robust gene. After each genetic operation, APCEA uses the RCFPI list to screen new offspring. Once a RCFPI has been found in a chromosome, APCEA will only allow it to pass with a certain probability. We call this probability the RCFPI survival rate (rcRate). The APCEA flow is shown in Figure 2.

3.2. Encoding and Decoding

Permutation coding is adopted in the gene coding process of the APCEA. A natural number between 1 and n is used to indicate the priority of each task, where n is the number of tasks. Figure 3 shows an example of chromosome coding with 8 tasks.

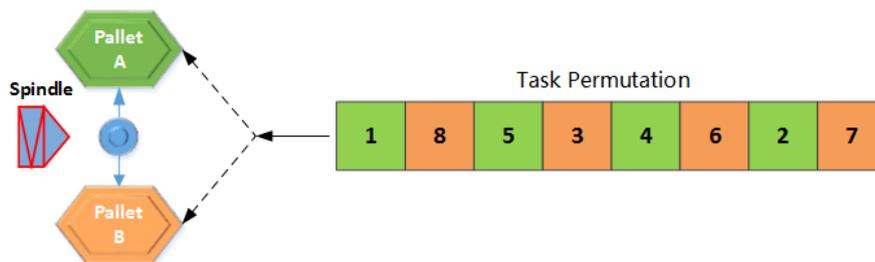


Figure 3. The permutation encoding of APCEA.

During the decoding phase, tasks are dynamically assigned to the pallets according to the priority. When a pallet is idle, the highest priority task of the remaining tasks is assigned to it. From the two tasks that have been allocated pallets, the machining center spindle selects the task that has arrived. If both tasks have been arrived, then break the tie by the priority.

3.3. Initialization

Dispatch rules are very effective for scheduling a single target, Panwalkar [27] summarized 113 scheduling rules, to improve the quality of the initial population. This study chooses six rules to generate individuals, as shown in Table 1. Other individuals are randomly generated by the algorithm until the total number of individuals reaches the initial population size(popSize). The algorithm must also initialize the RCFPI list during the Initialization.

Table 1. The used heuristic rules.

SN	Rule Name	Rule Description
1	EDD	Earliest due date
2	FIFO	First Input First Output
3	SPT	Shortest processing time
4	LPT	Longest processing time
5	CR	Critical ratio
6	ATC	Apparent tardiness cost

3.4. Fitness Evaluation

The scheduling result of APC is affected by two uncertain factors, which are the FPI pass rate and the adjustment time of the unpassed first workpiece. The pass rate of the first workpiece is subject to the 0–1 distribution. The adjustment time of the unqualified first workpiece is subject to the uniform distribution $f_{a_j} \in [5p_j, 10p_j]$, which is related to the processing time of a single piece. This research proposes two models to evaluate the fitness of each individual.

3.4.1. The APC Simulation Evaluation Model

The input conditions are the task list formed after gene decoding and the number of simulation runs, K . In this study, the value of K is set to 100, which means that each fitness evaluation needs

to run the simulation 100 times. The output data of the APC simulation is the object value of each optimization objective function. The simulation algorithm, Algorithm 1, can be explained as follows:

- (1) The remain task set O is initialized, and the task priority is set according to the gene coding sequence.
- (2) If there are no assigned task on all pallets, each task in O is assigned to pallets according to the priority order.
- (3) The tasks that have arrived in the pallet are selected at time t . If all tasks have arrived at pallets, the task O_j which has the highest priority is selected.
- (4) If a task is in the SETUP phase, the current time is set to $t = t + s_j$ and then the task phase is set to FPI.
- (5) If a task is in the FPI phase, the inspection result is carried out by the pass rate of the first piece inspection. If a task passes the first piece inspection, the task is set to the NORMAL phase, and the ready time of the task is $r_j = t + p_j + f_j$. Otherwise, the adjustment time fa of the unqualified task is randomly generated according to uniform distribution. The ready time of the task is $r_j = t + p_j + f_j + fa$.
- (6) If a task is in the NORMAL phase, the number of completed tasks is increase by 1. If the task is completed, task will be removed from the remain task set O . If O is empty, each scheduling target value needs to be calculated, and then the schedule process is exited; otherwise, the schedule process returns to 2.
- (7) repeat step 2–6 K times, then get the average of TWT, C_{max} and TFT.

The algorithm pseudo code is as follows.

Algorithm 1 The simulation process of APC scheduling

Input: Task List, Number of simulation runs K

Output: TWT, C_{max}, TFT

```

1: for  $k \leftarrow 0$  to  $K$  do
2:   initialize remain tasks collection  $O$ 
3:   let initial simulation clock  $t$  be 0
4:   while  $O \neq \phi$  do
5:     if all pallets is free then
6:       add tasks onto pallets order by list index
7:     end if
8:     select the first task  $o_j$  on pallets ordered by release time and list index
9:      $t \leftarrow \max(t, r_j)$ 
10:    if the phase of  $o_j$  is "SETUP" then
11:       $t \leftarrow t + s_j$ 
12:      set task  $o_j$  to "FPI" phase
13:    else if the phase of  $o_j$  is "FPI" then
14:      generate FPI result randomly
15:      if FPI result is passed then
16:        set task  $o_j$  to "NORMAL" phase
17:        set task  $o_j$  ready at  $t + p_j + f_j$ 
18:      else
19:        the adjustment time of the first workpiece  $fa$  by random
20:        set task  $o_j$  ready at  $t + p_j + f_j + fa$ 
21:      end if
22:    else if the phase of  $o_j$  is "NORMAL" then
23:       $t \leftarrow t + p_j$ 
24:      if  $o_j$  is complete then
25:        remove task  $o_j$  from  $O, O \leftarrow O \setminus \{o_j\}$ 
26:      end if
27:    end if
28:  end while
29: end for
30: return averageTWT,  $C_{max}$ , TFT

```

3.4.2. The Simplified APC Evaluation Model

The simplified evaluation model assumes that the four steps need to be executed in sequence to complete a task O_j , i.e., setup, first piece inspection, first piece adjustment and remaining pieces processing. The durations of each step are estimated as follows:

- (1) Time of setup, ts_j . It contains the setup time s_j and processing time of the first workpiece p_j .

$$ts_j = s_j + p_j \quad (18)$$

- (2) Time of first piece inspection, $E(f_j)$. The expectation time of the first piece inspection $E(f_j)$ can be evaluated by Equation (19).

$$E(f_j) = \frac{f_j}{fp_j} \quad (19)$$

- (3) Time of first piece adjustment, $E(fa_j)$. The single adjustment time of the unqualified first workpiece in this research follows the uniform distribution $fa_j \in [5p_j, 10p_j]$, so the average single adjust time value can be estimated as $7.5p_j$. The expectation time of the first piece adjustment $E(fa_j)$ can be evaluated by Equation (20).

$$E(fa_j) = 7.5p_j \cdot \frac{1 - fp_j}{fp_j} \quad (20)$$

- (4) Time of remaining pieces processing, tr_j . It can be evaluated by Equation (21).

$$tr_j = (q_j - 1) \cdot p_j \quad (21)$$

According to the above simplified model, the scheduling target values are obtained by implementing the process of task assignments in Algorithm 2.

Algorithm 2 The simplified APC scheduling algorithm

Input: Task List

Output: TWT, C_{max}, TFT

```

1: initialize remain tasks collection O
2: initialize all tasks phase be "SETUP"
3: let initial simulation clock t be 0
4: while O ≠ ∅ do
5:   add task to pallet p order by list index
6:   select first task oj on pallets ordered by release time and list index
7:   t ← max(t, rj)
8:   if the phase of oj is "SETUP" then
9:     t ← t + Sj + pj
10:    set task oj to "NORMAL" phase
11:    set task oj ready at t + E(faj) + E(fj)
12:   else if the phase of oj is "NORMAL" then
13:     t ← t + pj
14:     if oj is complete then
15:       remove task oj from O, O ← O \ {oj}
16:     end if
17:   end if
18: end while
19: return TWT, Cmax, TFT

```

- (1) The remain task set O is initialized, and the task priority is set according to the gene coding sequence.
- (2) If there are no assigned task on all pallets, each task in O is assigned to pallets according to the priority order.

- (3) The tasks that have arrived in the pallet are selected at time t . If all tasks have arrived at pallets, the task O_j which has the highest priority is selected.
- (4) If a task is in the SETUP phase, the current time of the task is set to $t = t + s_j + p_j$. If at least one workpiece has passed the first piece inspection, the next phase of the task O_j is set to NORMAL, and the expected arrival time of the next workpieces is $r_j = t + E(fa_j) + E(f_j)$
- (5) If a task is in the NORMAL phase, the number of completed tasks is increase by 1. If the task is completed, task will be removed from the remain task set O . If O is empty, each scheduling target value needs to be calculated, and then the schedule process is exited; otherwise, the schedule process returns to 2.

3.5. Crossover

This research uses the tournament on the basis of selection method to select individuals with high fitness values that enter the next generation of computing. Partial-Mapped Crossover (PMX) operators are used for cross-operation, and the method is as follows.

- (1) Several genes at the same position in a pair of parent chromosomes are selected randomly.
- (2) The locations of these two gene sets are exchanged.
- (3) A mapping relationship on the basis of the two genes groups needs to be exchanged, and conflict detection is executed.
- (4) Finally, all conflicting genes are mapped to ensure that there is no conflict between the new pair of offspring genes.

As shown in Figure 4, the mapping relationship of genes 1-6-3 is established. In step 3, there are two gene 1s in offspring 1, and they are converted into gene 3 by the mapping relationship until there is no conflict.

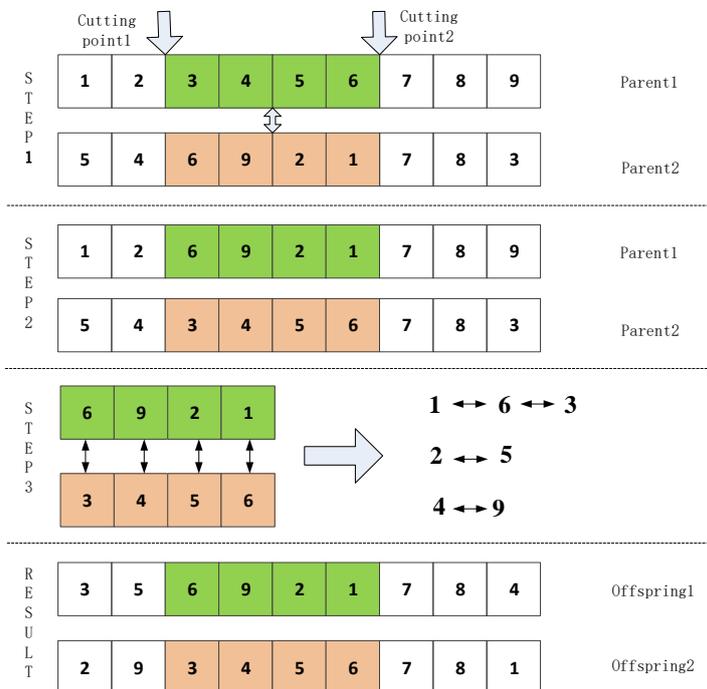


Figure 4. Partial-Mapped Crossover (PMX) crossover.

3.6. Mutation

The mutation operation is implemented by the switching method SWAP. Two points of permutation are exchanged to generate a new permutation. This study proposes a roulette to increase the mutation probability of RCFPI. The method is as follows.

- (1) Select two adjacent tasks as a combination and calculate their RDFPI. Use the RDFPI as the selected probability of each position.
- (2) Calculate the cumulative probability of each position and store it in array Q.
- (3) A uniformly distributed pseudo-random number R is generated in the interval [0,1]: if $R < Q [1]$, then the first node is at position 1. otherwise, the first node is at position k, so that $Q [k-1] < R < Q [k]$.
- (4) Randomly select another node as the second node.
- (5) Swap the values of the first node and the second node.

As shown in Figure 5, the first node is selected by roulette and the second node is selected randomly.

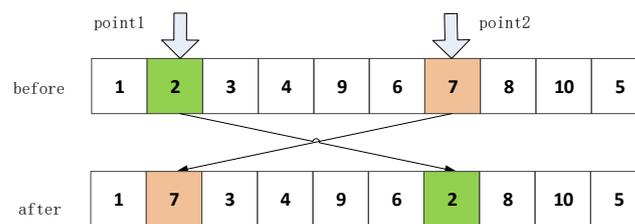


Figure 5. SWAP mutation.

3.7. Update Archive

In the process of updating the archive set, the algorithm firstly merges the parent generation and the child generation to form a new population. To ensure each individual F in the population can obtain two attributes, that is, the non-dominant order F_{rank} and the crowding degree F_d , the rank order of the non-dominant solution set and the crowding degree calculation are carried out for the new population, respectively. The individuals in the new population are sorted in order according to F_{rank} and F_d , and the pioneer $popSize$ individuals are selected as the new elite solution set.

After updating the archive set, the algorithm also needs to refresh the RCFPI list. The RCFPI list is a dynamic list, Each RCFPI has its own survival generations, and the initial values are determined by the parameter RCFPI Life (rcLife). After each generation of evolution, the APC algorithm scans the individuals of its archived elite solution set. If the RCFPI does not exist, its life value will be increased by 1. Otherwise, its life value will be reduced by 1. Once the RCFPI life value equals 0, it will be removed from the RCFPI list. However, this does not necessarily mean permanent removal. Once the life value is greater than 0 again, the algorithm will add it to the RCFPI list.

3.8. The Terminating Condition

The termination condition of the APCEA algorithm is as follows.

- (1) The fitness of the best individual in the population has not improved for 20 consecutive generation populations.
- (2) The total number of evaluations reaches 10,000 times.

3.9. The Parameter Setting of APCEA

The APCEA contains five parameters that need to be set: the population size ($popSize$), the PMX crossover rate ($pmxRate$), the SWAP mutation rate ($swapRate$), $rcRate$, and $rcLife$. Five factors with five levels are given in Table 2.

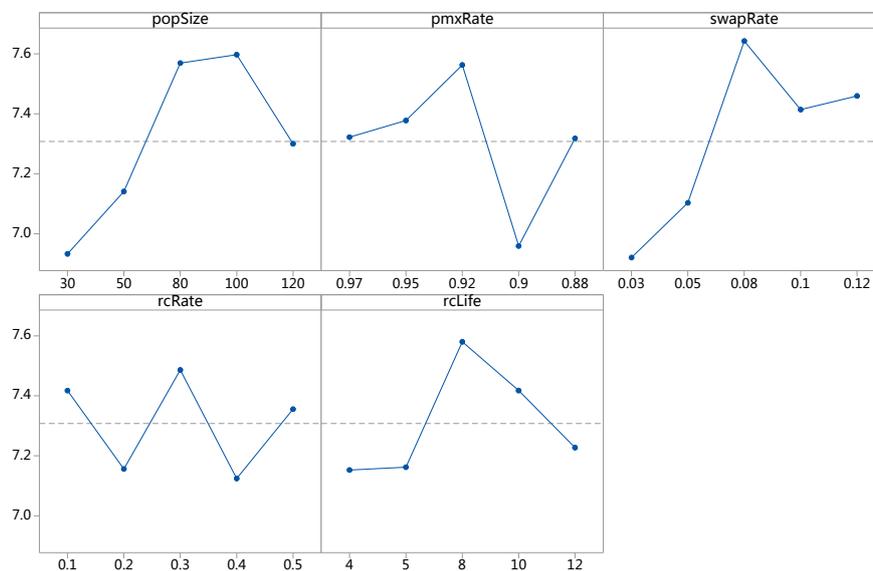
Table 2. The factor levels of the parameters.

Parameter	Factor Level				
	1	2	3	4	5
1 popSize	30	50	80	100	120
2 pmxRate	0.97	0.95	0.92	0.9	0.88
3 swapRate	0.03	0.05	0.08	0.1	0.12
4 rcRate	0.1	0.2	0.3	0.4	0.5
5 rcLife	4	5	8	10	12

According to the number of parameters of APCEA and the levels of the factors shown in Table 2, we chose $L_{25}(5^5)$ for the orthogonal experiment table of parameter analysis of APCEA. Each orthogonal experiment needs to test 25 factor level combinations in the orthogonal experiment table. For each combination of factor level, the average contribution rate (ARV) is used to define the performance of the combination. The calculation of perimeter is shown in Equation (22).

$$ARV(i) = \frac{1}{10} \sum_{j=1}^{10} \frac{|RS_j \cap PA_j^i|}{|RS_j|} \times 100\%. \quad (22)$$

PA_j^i is the non-dominated solution set which is obtained by selecting the j th sample under the i th combination of the factor level, RS_j is the global non-dominated solution set obtained by selecting the j th sample under all the combinations of factor level. The larger the value of ARV is, the higher the proportion of the number of non-dominated solutions obtained by the corresponding factor-level combination divided by the overall factor-level combination, and the better the effect. Figure 6 shows the trend of APCEA parameters.

**Figure 6.** The level trend of APCEA parameters.

From Figure 6, it can be concluded that the preferred parameters of APCEA are as follows: population size $popSize = 100$, crossover probability $P_c = 0.92$, mutation probability $P_m = 0.08$, RCFPI survival rate $rcRate = 0.3$, RCFPI life $rcLife = 8$.

4. The Experiment Results

In this study, the APCEA is compared to other three MOEAs with performance metrics. Through the analysis of experimental data, it is proved that APCEA has a better performance in solving the problems studied in this study.

4.1. Design of Experiments

According to Lee and Bhaskaran [28], the characteristics of the APC scheduling problem are as follows:

- (1) The number of tasks is n .
- (2) The due date tightness factor is $\tau = 1 - \bar{d}/C_{max}$. In this study, C_{max} is estimated according to Equation (23)

$$C_{max} = \sum_{j=1}^n (s_j + p_j \times q_j) \quad (23)$$

- (3) The due date range factor is $R = (d_{max} - d_{min}) / C_{max}$.

The range of the parameters is as follows: $n = (10, 20, 30, 50)$, $\tau = (0.2, 0.4, 0.6, 0.8)$, and $R = (0.2, 0.4, 0.6, 0.8)$. The properties of tasks generated by each parameter combinations followed a uniformly distributed pattern. The range of setup times is $s_j \in [2, 3]$ hours. The range of processing times of the single piece is $p_j \in [2, 4]$. The range of the first piece inspection is $f_j \in [2, 4]$. The number of tasks is $q_j \in [5, 30]$. The range of pass rates of the first piece inspection is $fp_j \in [0.4, 1]$. The range of the weight of a task is $w_j \in [1, 3]$. The total processing time of a task o_j is estimated by $b_j = s_j + q_j \times p_j$. The range of the due date of a task follows $d_j \in \left[r_j + \left(\sum_{j=1}^n b_j - r_j \right) \left(1 - \tau - \frac{R}{2} \right), r_j + \left(\sum_{j=1}^n b_j - r_j \right) \left(1 - \tau + \frac{R}{2} \right) \right]$.

There are $4 \times 4 \times 4 = 64$ levels of combined (n, τ, R) . In this study, 10 instances were made by each level. Therefore, there were a total of $64 \times 10 = 640$ instances covering the experimental samples.

4.2. Comparison of APC Original and Simplified Models

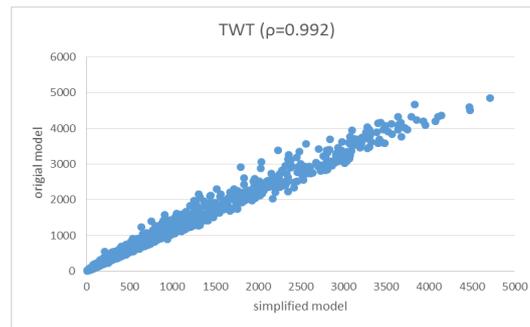
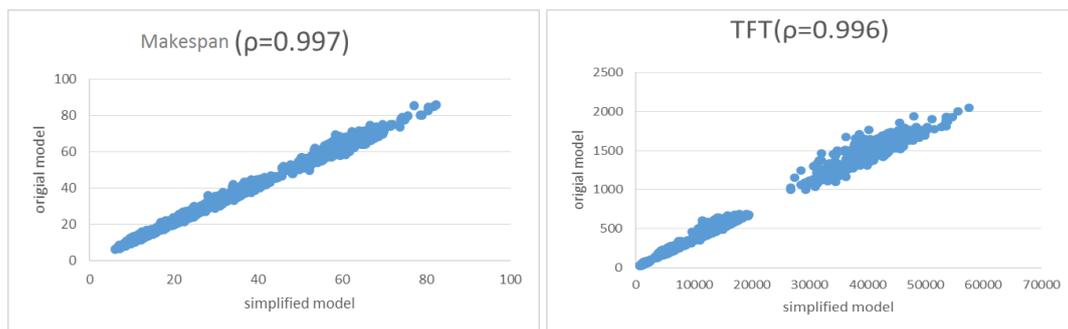
From the above 640 instances, 5 task sequences were randomly generated from each instance, then we obtained a total of 3200 sequences. According to the APC simulation model, the TWT values of these task sequences were obtained, and then the TWT vector $(X_1, X_2, \dots, X_{3200})$ was formed. Then, according to the simplified model, the TWT values were obtained and the TWT vector $(Y_1, Y_2, \dots, Y_{3200})$ was formed. The Pearson Equation (24) was used to calculate the correlation between the two vectors.

$$\rho = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}} \quad (24)$$

The Pearson correlation coefficient of TWT between the original model vector and the simplified model vector was about 0.992. The Pearson correlation coefficients of the other two objects were obtained by the same method: the correlation coefficient of C_{max} vectors was 0.997, and the correlation coefficient of TFT vector was 0.996. The correlation scatter plot shown in Figure 7 proves that the APC simplified model is very relevant to the APC original model.

Table 3. The time comparison between the simplified model and the original models.

Task Count	Simplified Model	Original Model		
		1 Thread	10 Threads	20 Threads
	Time (s)	Time (s)	Time (s)	Time (s)
10	0.83	33.48	16.98	17.74
20	0.74	64.58	35.44	35.80
30	1.05	94.23	49.95	48.52
50	1.85	158.26	81.02	83.19

**(a)** The scatter plot of the correlation of the total weighted tardiness (TWT)**(b)** The scatter plot of the correlation of the makespan (C_{max}) **(c)** The scatter plot of the correlation of the total flow time (TFT)**Figure 7.** The scatter plot of the correlation of each objective.

In this study, the running time comparison between the simplified model and the original model was also made, and two models were used to test the above instances in the APC algorithm. For the original model, because it needs to be simulated for many times, this research adopts JAVA multi-thread parallel technology to carry out the simulation test of single threads, 10 threads and 20 threads respectively. The proposed tests are all implemented in Java 8.0 and run on an Intel Core i5-7200U 2.50 GHz processor with 4 GB of RAM. Table 3 lists the average time statistics of each model under different task count. As can be seen from Table 3, The average time of APCEA algorithm with the original model is many times more than the time of the simplified model, so the simplified model has a great time performance advantage. Therefore, this research used the simplified model to calculate the fitness and the simulation model was only used to calculate the final result.

4.3. MOEAs for Comparison

We selected the following three multi-objective evolutionary algorithms from the MOEA framework [29]: the non-dominated sorting genetic algorithm II (NSGA-II), the strength Pareto evolutionary algorithm (SPEA2), and the epsilon-dominated multi-objective evolutionary algorithm

(ε -MOEA). The non-dominated sorting genetic algorithm II proposed by Deb [26] in 2002 is a kind of multi-objective evolutionary algorithm. The NSGA-II algorithm ranks the populations by the level of non-dominated solutions, and preserves the population of high-level non-dominated to approximate the Pareto frontier in each iteration and obtains the population of Pareto frontier which is uniformly distributed by preserving the non-dominated solutions of the lower crowding degree. Zitzler et al. proposed the SPEA2 algorithm [30] which introduces a fine-grained fitness assignment strategy, a density estimation technique, and an enhanced archive truncation method on basis of the strength Pareto evolutionary algorithm. Deb K et al. [31] proposed the steady-state MOEA based on the ε -dominance concept and efficient parent and archive update strategies for the purpose of developing a compromised algorithm for achieving a well-distributed set of solutions quickly.

4.4. Performance Metrics

The following three performance metrics were adopted to compare the performance of APCEA and three MOEAs: NSGA-II, SPEA2, and ε -MOEA.

4.4.1. Hypervolume Ratio

The hypervolume [32] represents the size of the objective space dominated by the obtained non-dominated front PF_{known} , and the calculation formula is shown in Equation (25). The higher the hypervolume is, the better the performance is.

$$HV = \text{volume} \left(\bigcup_{i=1}^{n_{PF}} v_i \right). \quad (25)$$

where n_{PF} is the number of members in the obtained non-dominated front PF_{known} , v_i is the hypercube constructed with a reference point, and the member i is the diagonal of the hypercube (HVR) [32]. The HVR is the ratio of the hypervolume of PF_{known} and the hypervolume of the reference Pareto front PF_{ref} . The hypervolume represents the covered level of the hyperspace, and the HVR calculation formula is shown in Equation (26).

$$HVR = \frac{HV_{known}}{HV_{ref}}. \quad (26)$$

4.4.2. Inverted Generational Distance

The generational distance [33], which was introduced by Stadler et al., is a way of estimating how far away the points in the Pareto front PF_{known} produced by our algorithm are from those in the true Pareto front of the problem PF_{ref} . The IGD metric is defined in Equation (27):

$$IGD = \frac{\left(\sum_{i=1}^n d_i^p \right)^{1/p}}{n} \quad (27)$$

where n is the number of all elements in PF_{ref} , p is set to 2, and d_i is the minimum Euclidean distance between each of the elements PF_{ref} and the nearest element PF_{known} .

4.4.3. Spacing

Spacing (SP) [34] is used to measure the distribution of the neighboring points in PF_{known} . The calculation formula is defined in Equation (28).

$$SP = \sqrt{\frac{1}{n-1} \sum_{i=1}^n \left(\bar{d} - d_i \right)^2} \quad (28)$$

where n is the number of members in the obtained Pareto front PF_{known} , d_i is the minimum distance between the member and its nearest member in PF_{known} , and \bar{d} is the average value of all d_i .

4.5. Comparison Results Analysis

In this section, we compare the results of our APCEA algorithm with other three Multi-objective evolutionary algorithms. The parameters used in the APCEA processing were experimentally selected to yield a good solution quality within a reasonable time.

A set of 64 instances is prepared to verify the APCEA algorithm for solving the scheduling problem. The proposed algorithms are all implemented in Java 8.0 and run on an Intel Core i5-7200U 2.50 GHz processor with 4 GB of RAM.

Table 4 presents the Comparison of the four algorithms in terms of the average hypercube (HVR), Inverted Generational Distance (IGD) and Spacing (SP) values over 30 independent runs for all 64 instances. For each instance and each performance metric (HVR, IGD and SP), those better than the others are marked in bold.

As can be seen from Table 4, there are 54 instances in which the solution obtained by APCEA has a larger HVR than that obtained by other algorithms, which indicates that APCEA dominates NSGA-II, SPEA2, and ϵ -MOEA in the HVR domain. In 48 instances, the solution obtained by APCEA has a smaller IGD than that obtained by other algorithms, indicating that any other algorithm is dominated by APCEA in the IGD domain. For the SP metric, the APCEA yields the best results in 28 out of 64 instances; the NSGA-II yields the best results in 7 out of 64 instances; the SPEA2 yields the best results in 15 out of 64 instances; and the ϵ -MOEA yields the best results in 14 out of 64 instances. In particular, APCEA outperform other algorithms in large-scale complex instances. When the number of tasks exceeds 20, APCEA is almost 100 percent better than the other three algorithms in all metrics.

In addition, this study shows four boxplots in Figure 8 to visually represent Table 4.

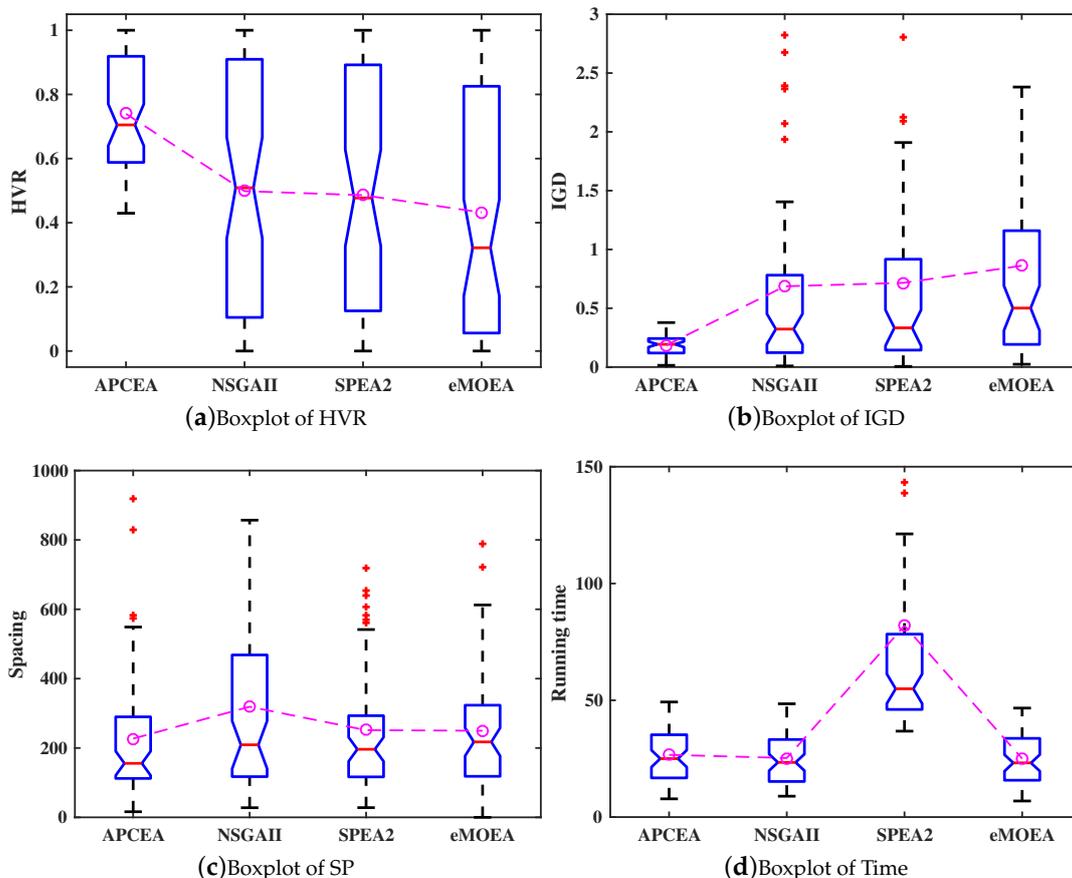


Figure 8. Comparison results of the proposed APCEA with other MOEAs (HVR to be maximized, Spacing and IGD to be minimized).

Table 4. Comparison of the four algorithms in terms of the average hypercube (HVR), Inverted Generational Distance (IGD) and Spacing (SP) values over 30 independent runs for all 64 instances (HVR to be maximized, Spacing and IGD to be minimized).

Instance $n \times \tau \times R$	APCEA				NSGA-II				SPEA2				ϵ -MOEA			
	HVR	IGD	SP	TIME	HVR	IGD	SP	TIME	HVR	IGD	SP	TIME	HVR	IGD	SP	TIME
10 × 0.2 × 0.2	0.99	0.08	90.67	15.97	1.00	0.04	75.69	13.77	1.00	0.07	71.21	183.53	1.00	0.07	77.37	12.33
10 × 0.2 × 0.4	1.00	0.32	310.73	13.77	0.99	0.24	308.97	13.23	0.98	0.33	117.30	324.00	0.99	0.34	48.18	13.67
10 × 0.2 × 0.6	1.00	0.12	304.47	13.80	1.00	0.10	298.32	12.97	1.00	0.07	292.17	404.27	1.00	0.24	335.23	12.07
10 × 0.2 × 0.8	0.98	0.02	95.23	10.48	0.94	0.08	106.39	9.92	0.96	0.06	88.25	54.98	0.85	0.14	122.77	6.88
10 × 0.4 × 0.2	1.00	0.27	16.17	10.22	1.00	0.10	56.60	8.93	1.00	0.17	40.43	186.27	1.00	0.33	0.00	9.72
10 × 0.4 × 0.4	1.00	0.04	112.89	13.97	1.00	0.03	118.47	13.30	1.00	0.02	115.91	190.87	0.99	0.10	113.13	12.40
10 × 0.4 × 0.6	0.99	0.02	111.47	14.43	0.97	0.05	136.42	12.37	0.97	0.05	117.51	121.23	0.94	0.10	119.49	11.73
10 × 0.4 × 0.8	1.00	0.01	115.64	10.10	1.00	0.01	122.08	9.47	1.00	0.01	125.45	61.65	1.00	0.02	150.02	9.37
10 × 0.6 × 0.2	1.00	0.02	143.74	13.53	0.98	0.06	115.55	13.23	0.98	0.02	129.71	182.00	0.99	0.06	117.05	15.00
10 × 0.6 × 0.4	0.97	0.10	101.82	9.63	0.96	0.09	99.75	9.60	0.98	0.03	101.87	61.00	0.97	0.10	79.40	8.98
10 × 0.6 × 0.6	1.00	0.04	37.79	22.80	1.00	0.02	40.31	14.50	1.00	0.03	38.40	201.03	0.99	0.04	37.77	13.93
10 × 0.6 × 0.8	0.98	0.04	101.02	15.30	0.94	0.05	91.35	13.90	1.00	0.01	89.66	143.30	0.99	0.04	106.52	12.53
10 × 0.8 × 0.2	1.00	0.02	30.91	16.00	0.98	0.03	27.74	12.47	0.99	0.01	27.99	222.37	0.96	0.05	27.16	11.90
10 × 0.8 × 0.4	1.00	0.16	36.29	14.17	1.00	0.11	30.53	12.57	1.00	0.16	31.85	182.97	1.00	0.18	28.15	11.90
10 × 0.8 × 0.6	1.00	0.06	240.14	14.30	0.99	0.07	202.57	13.07	1.00	0.06	186.08	138.70	0.99	0.09	229.20	11.57
10 × 0.8 × 0.8	0.99	0.04	239.88	7.80	0.95	0.05	205.73	9.02	0.99	0.02	202.69	47.55	0.96	0.08	208.37	8.33
20 × 0.2 × 0.2	0.85	0.20	106.86	23.80	0.82	0.20	101.73	22.33	0.83	0.22	104.43	88.60	0.72	0.25	104.62	22.27
20 × 0.2 × 0.4	0.83	0.13	98.68	16.65	0.80	0.13	85.00	15.97	0.81	0.14	81.51	37.63	0.68	0.18	102.78	17.23
20 × 0.2 × 0.6	0.81	0.11	190.78	18.30	0.64	0.20	122.86	16.58	0.61	0.18	168.10	38.98	0.47	0.29	125.11	16.53
20 × 0.2 × 0.8	0.77	0.14	65.62	18.17	0.71	0.14	115.51	17.43	0.58	0.22	76.73	38.50	0.65	0.20	104.23	17.65
20 × 0.4 × 0.2	0.86	0.12	213.43	24.50	0.88	0.10	249.43	17.07	0.81	0.13	189.78	42.63	0.80	0.18	248.84	16.83
20 × 0.4 × 0.4	0.79	0.15	86.90	18.33	0.74	0.19	153.88	17.50	0.78	0.22	115.06	85.53	0.67	0.28	175.19	16.73
20 × 0.4 × 0.6	0.86	0.13	100.60	19.37	0.83	0.15	86.31	19.55	0.79	0.15	94.34	39.33	0.75	0.18	86.38	18.97
20 × 0.4 × 0.8	0.86	0.08	97.27	19.80	0.67	0.15	113.44	18.20	0.73	0.12	138.26	36.78	0.40	0.35	108.17	18.28
20 × 0.6 × 0.2	0.67	0.26	129.21	18.10	0.43	0.51	160.06	17.03	0.57	0.43	274.91	89.40	0.41	0.54	457.06	16.57
20 × 0.6 × 0.4	0.75	0.14	176.43	19.50	0.71	0.16	158.41	18.40	0.53	0.23	165.85	49.68	0.44	0.29	139.45	18.12
20 × 0.6 × 0.6	0.73	0.15	123.15	18.72	0.72	0.16	182.72	19.12	0.65	0.19	145.59	36.98	0.61	0.21	172.83	18.67
20 × 0.6 × 0.8	0.62	0.25	140.32	18.97	0.72	0.18	117.83	18.35	0.72	0.18	104.83	40.45	0.56	0.29	121.41	19.22
20 × 0.8 × 0.2	0.81	0.13	132.92	18.30	0.53	0.25	106.56	17.37	0.37	0.38	72.87	71.10	0.26	0.53	105.50	16.73
20 × 0.8 × 0.4	0.69	0.19	79.93	18.43	0.52	0.38	116.88	17.20	0.47	0.36	160.83	86.80	0.56	0.33	164.41	16.93
20 × 0.8 × 0.6	0.80	0.12	164.26	19.03	0.66	0.25	316.33	17.97	0.61	0.25	168.30	48.42	0.48	0.36	497.40	19.33
20 × 0.8 × 0.8	0.76	0.16	147.78	16.87	0.63	0.22	308.01	18.02	0.62	0.23	188.74	49.23	0.58	0.26	206.65	18.88

Table 4. Cont.

Instance $n \times \tau \times R$	APCEA				NSGA-II				SPEA2				ϵ -MOEA			
	HVR	IGD	SP	TIME	HVR	IGD	SP	TIME	HVR	IGD	SP	TIME	HVR	IGD	SP	TIME
30 × 0.2 × 0.2	0.56	0.32	297.71	27.40	0.02	1.00	260.96	25.83	0.15	0.96	255.41	51.80	0.07	1.21	254.90	25.40
30 × 0.2 × 0.4	0.73	0.19	155.86	27.37	0.37	0.36	161.40	26.73	0.38	0.37	291.13	48.83	0.19	0.63	211.54	25.67
30 × 0.2 × 0.6	0.71	0.24	119.26	27.60	0.25	0.57	170.14	26.57	0.49	0.33	147.92	54.00	0.16	0.69	139.23	25.57
30 × 0.2 × 0.8	0.69	0.18	161.70	28.73	0.50	0.31	233.73	26.83	0.50	0.33	206.17	44.60	0.39	0.40	261.87	25.80
30 × 0.4 × 0.2	0.70	0.37	199.40	28.73	0.30	0.76	294.72	27.90	0.14	1.01	213.98	49.45	0.07	1.03	261.51	27.58
30 × 0.4 × 0.4	0.72	0.20	155.53	27.20	0.26	0.45	167.13	26.07	0.17	0.66	149.73	51.63	0.09	0.83	248.95	25.73
30 × 0.4 × 0.6	0.63	0.19	155.07	27.40	0.06	0.80	212.61	27.10	0.17	0.73	261.19	49.03	0.11	1.07	273.48	25.83
30 × 0.4 × 0.8	0.67	0.20	225.83	27.03	0.54	0.27	313.46	27.32	0.46	0.33	271.84	42.40	0.23	0.54	195.16	28.55
30 × 0.6 × 0.2	0.61	0.22	147.79	27.33	0.07	0.77	215.36	26.97	0.08	0.92	283.20	51.20	0.04	1.13	247.75	26.00
30 × 0.6 × 0.4	0.60	0.21	261.77	27.30	0.22	0.54	238.37	25.60	0.12	0.83	228.96	41.67	0.12	0.95	311.52	24.12
30 × 0.6 × 0.6	0.53	0.23	212.72	26.47	0.31	0.41	282.58	25.45	0.19	0.60	164.71	43.73	0.25	0.50	173.36	25.90
30 × 0.6 × 0.8	0.64	0.17	230.41	26.75	0.27	0.53	189.01	25.55	0.16	0.76	246.77	42.33	0.13	0.73	223.36	25.73
30 × 0.8 × 0.2	0.62	0.21	149.37	27.18	0.04	0.90	139.59	25.87	0.02	0.92	207.34	39.53	0.01	1.40	201.11	26.47
30 × 0.8 × 0.4	0.57	0.23	229.58	26.32	0.22	0.71	378.51	24.48	0.13	0.69	294.08	39.98	0.09	0.84	231.57	24.60
30 × 0.8 × 0.6	0.51	0.27	151.82	25.55	0.39	0.34	366.71	24.72	0.26	0.47	241.64	40.18	0.24	0.50	226.55	25.08
30 × 0.8 × 0.8	0.69	0.22	115.18	27.47	0.42	0.34	463.75	25.90	0.45	0.33	274.09	54.87	0.22	0.75	256.63	26.30
50 × 0.2 × 0.2	0.50	0.33	128.28	47.20	0.00	7.71	547.48	45.37	0.00	7.03	261.16	58.87	0.00	7.76	248.07	45.33
50 × 0.2 × 0.4	0.54	0.23	247.72	47.47	0.00	2.68	472.83	45.80	0.00	3.02	332.91	58.77	0.00	3.58	291.75	46.10
50 × 0.2 × 0.6	0.55	0.29	829.24	47.80	0.18	0.97	527.12	45.93	0.15	0.80	718.59	59.23	0.04	1.54	721.49	46.43
50 × 0.2 × 0.8	0.66	0.17	418.29	49.27	0.00	2.82	481.60	46.53	0.00	3.03	537.02	56.87	0.00	3.33	599.68	45.67
50 × 0.4 × 0.2	0.55	0.36	430.11	43.50	0.01	1.94	757.61	41.92	0.00	2.80	445.42	49.20	0.00	2.28	411.22	42.28
50 × 0.4 × 0.4	0.61	0.21	364.87	47.60	0.00	1.25	554.78	45.50	0.04	1.38	569.72	59.73	0.04	1.55	254.76	45.93
50 × 0.4 × 0.6	0.50	0.30	475.32	47.80	0.17	0.56	1081.15	45.83	0.25	0.48	519.66	58.27	0.19	0.64	407.47	46.40
50 × 0.4 × 0.8	0.64	0.21	573.80	45.65	0.02	1.07	1157.12	40.65	0.01	1.21	560.80	49.58	0.04	1.41	788.65	44.48
50 × 0.6 × 0.2	0.56	0.28	281.91	41.75	0.00	2.39	544.03	38.53	0.00	2.09	383.73	51.22	0.00	2.38	371.27	38.85
50 × 0.6 × 0.4	0.53	0.23	377.52	41.88	0.00	2.07	656.00	42.28	0.00	2.12	582.72	49.57	0.00	2.27	448.20	40.32
50 × 0.6 × 0.6	0.56	0.22	548.75	47.90	0.02	1.40	610.16	45.80	0.06	1.43	525.89	59.03	0.00	1.68	360.61	46.67
50 × 0.6 × 0.8	0.68	0.21	582.83	47.67	0.31	0.50	1163.51	45.73	0.31	0.54	653.87	60.00	0.24	0.55	428.07	46.17
50 × 0.8 × 0.2	0.58	0.27	384.76	47.57	0.01	0.97	801.55	46.00	0.04	0.91	606.72	61.07	0.00	1.23	598.38	45.90
50 × 0.8 × 0.4	0.46	0.35	435.23	47.47	0.14	0.64	857.02	45.93	0.02	1.03	639.67	59.47	0.03	1.19	514.84	45.87
50 × 0.8 × 0.6	0.43	0.34	419.73	47.73	0.00	2.37	850.12	48.17	0.00	1.91	450.08	59.20	0.00	2.15	486.53	45.43
50 × 0.8 × 0.8	0.43	0.38	918.85	48.23	0.07	1.06	732.63	48.50	0.03	1.04	541.56	60.37	0.00	1.69	612.48	44.23
count	54	48	28		5	7	7		9	9	15		3	0	14	

Figure 8 shows the comparison results of the four algorithms (APCEA, NSGA-II, SPEA2, ϵ -MOEA). For each performance metric, a Wilcoxon signed-rank test with the significance level of 0.05 was carried out on the results obtained by 30 independent runs of each method. The comparison results illustrate that the proposed APCEA performs significantly better than other three MOEAs in terms of HVR and IGD indicators. However, the APCEA is only significantly better than NSGA-II in three MOEAs for the metric of spacing, and it is only significantly better than SPEA2 in three MOEAs on the metric of running time. It can be concluded that the comprehensive performance of the proposed APCEA is still the best among the four algorithms.

To demonstrate the detailed evolution process of the four algorithms, Figure 9 shows the average performance of HVR, IGD, and SP across generations, respectively. The detailed results from Figure 9a–c show that the proposed APCEA can achieve good metrics of HVR and IGD much faster than other three MOEAs. It is clear that the APCEA method shows the fastest convergence speed and obtains the best fitness finally. As shown in Figure 9d, although the SP performance is not stable across generations, APCEA still achieves a similar performance to other MOEAs at the end of evolution.

Through the above analysis, we can observe that the performance of the APCEA is outstanding in terms of the quality of the results. Therefore, both the effectiveness and the efficiency of the APCEA are verified by comparison with other three MOEAs on 64 scheduling scenarios.

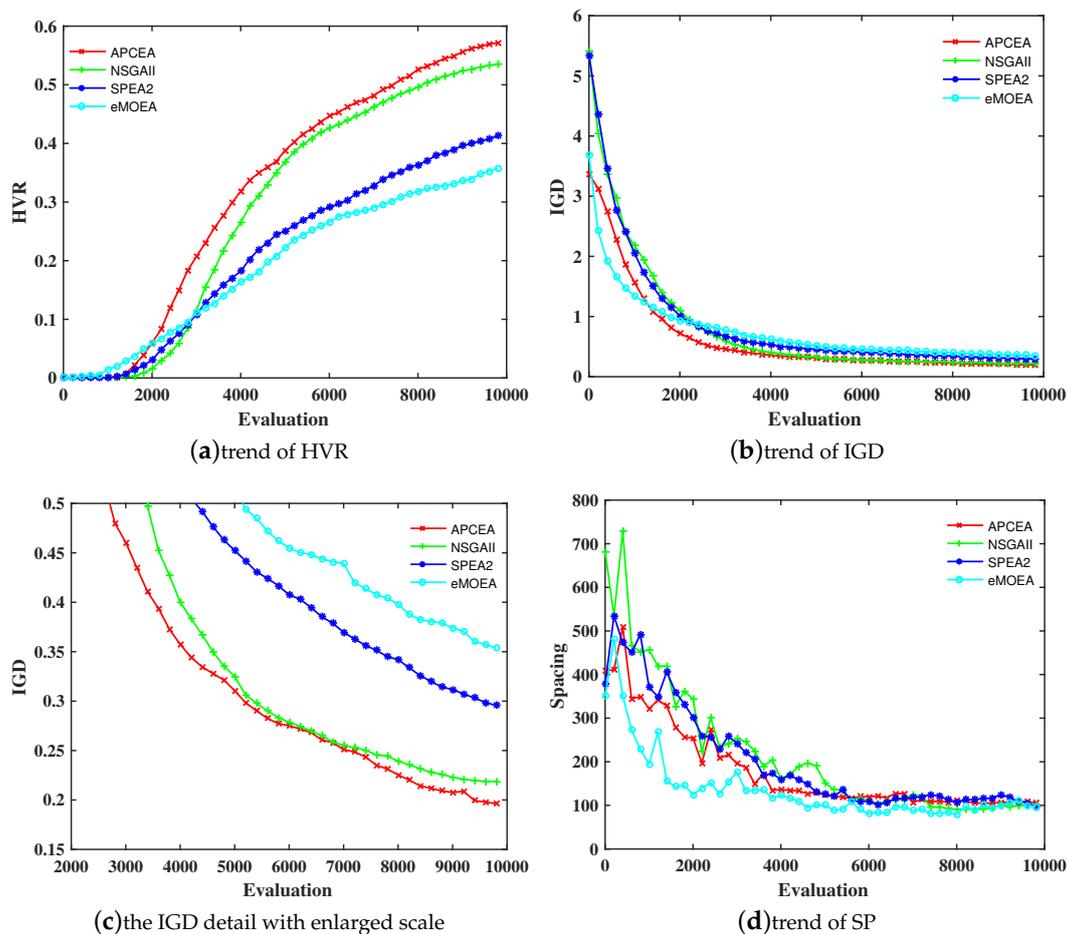


Figure 9. Average performance of the proposed APCEA and other multi-objective evolutionary algorithms (MOEAs) across generations.

5. A Case Study

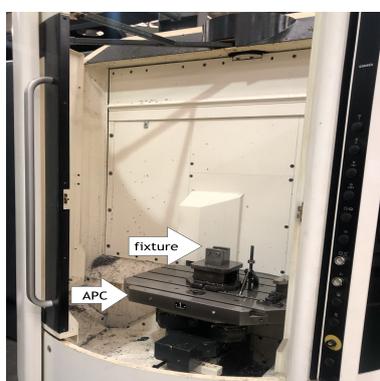
The industrial application addressed in this study is drawn from a real-world aeronautical manufacturing plant. This company is a tier-one supplier to various aeronautical manufacturing

companies in China. It produces over 300 different aluminum alloy shell parts for aeronautics products to order, including batch products and trial products. The ratio of trial products to batch products is generally 6:4. The new trial products are generally used for scientific research and have a low FPI pass rate because of immature processing technology. The batch quantity of each part task is small, usually between 5 and 20. The production mode of this company is characterized by being high mix-low volume. To prevent the whole batch of products from having quality problems, first piece inspection is usually applied at each operation of the machining process. The factory has used a large number of machining centers with APC to improve the use of machines. However, the low FPI pass rate of a large number of new scientific research products leads to low use of machines. This study aims to tackle this problem, and the proposed APCEA has been embedded into the Manufacturing Execution System (MES) and successfully applied in the shop floor.

As shown in Figure 10, Figure 10a shows the panorama of a machining center with an APC that is processing the first workpiece. It contains an idle pallet and a running pallet. A portable android device (PAD) installed with the mobile application of the MES is called device Kanban. It is placed next to the machining center in the workshop and displays the current task list of the device. It can be seen from Figure 10b that the first workpiece of a task has been withdrawal from the idle pallet, and it has been sent to the inspection room in Figure 10c. There is a coordinate measuring machine (CMM) that is doing the first piece inspection in Figure 10c. Figure 10d is a workshop device Kanban, which shows a Gantt chart of the two pallets.



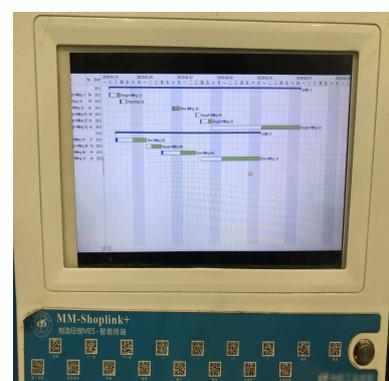
(a) An APC machine panorama



(b) An idle pallet



(c) First piece inspection



(d) Device Kanban

Figure 10. The practical application on the shop floor.

The device Kanban in the workstation and the computers in the inspection room are connected to the MES. The information from the first piece inspection can be transferred to the MES in real time. Hence, the workers can receive the real-time information of the first piece inspection to prepare for the next task, and feed back the current status of tasks by the device Kanban.

To visually demonstrate the application value of the proposed method, we applied the proposed APCEA on a practical case drawn from industry practice. The dispatching rule FIFO (first in and first out) which is frequently used in production scheduling was also applied on the same case. These two methods were simultaneously implemented in the MES on the real case for application. Figure 11 shows the resource Gantt chart generated by the proposed APCEA, and the result values (TWT, C_{max} , and TFT) are 4269, 1819, and 8595, respectively. Figure 12 illustrates the resource Gantt chart generated by the FIFO dispatching rule, and the resulting values (TWT, C_{max} , and TFT) are 6132, 1987, and 10,631, respectively. Each Gantt bar represents a task, and there are three colors in each Gantt bar. The blue represents the setup phase of a task, the white indicates the first piece inspection phase of the task, and the green shows the normal phase of the task. Because there are many solutions in the Pareto set produced by the proposed APCEA method, we randomly selected a solution for illustration.

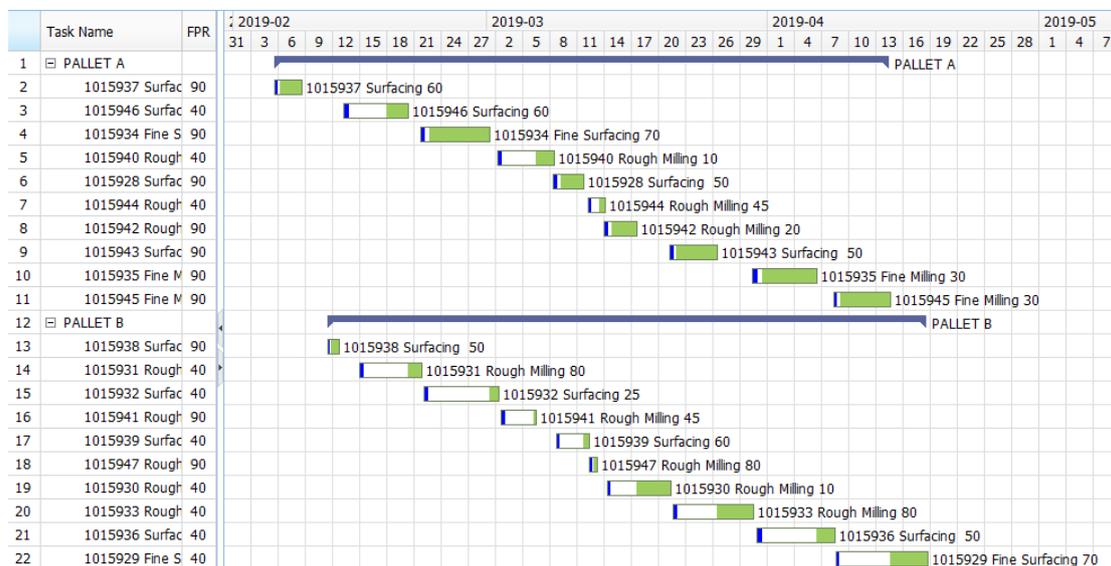


Figure 11. Resource Gantt Chart of the solution generated by the APCEA.

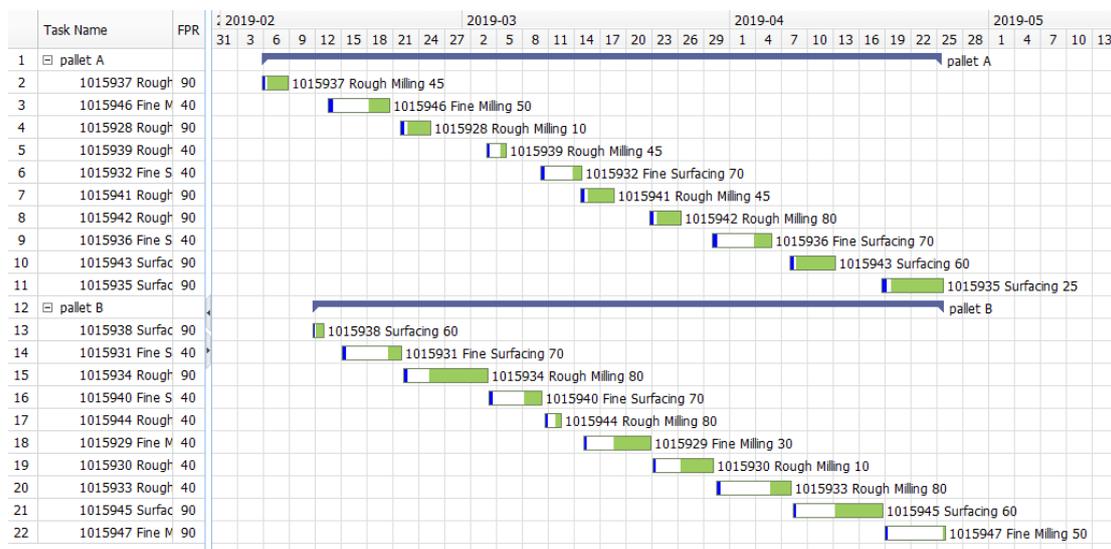


Figure 12. Resource Gantt Chart of the solution generated by the FIFO (first in and first out) rule.

The factory planner thinks that the FPI pass rate of a new trial product being developed is 0.4, while that of other products that have been mass-produced is 0.9. It can be seen from Figure 12 that the tasks are loosely arranged, and the latest completion time of all tasks is 25 April 2019. As shown in Figure 11, the APC system has a tight task arrangement, and the latest completion time of all tasks is 15 April 2019, which is about 10 days earlier than that of Figure 12. It should be noted that column “FPR” denotes the first piece inspection pass rate. The scheduler usually tries to avoid RCFPI in industry practice. As shown in Figure 11, there are only one RCFPI, which are 3–14. On the contrary, there are four RCFPIs which are 3–14, 5–16, 6–17, and 9–20 in Figure 12. These detailed results confirm the above conclusions and verify the practical application value of the proposed method in industry practice.

6. Conclusions

The aim of this study is to investigate the sustainable scheduling of a machining center equipped with APC; the first piece inspection is also considered in the scheduling model. Three objectives are simultaneously optimized by the proposed APCEA algorithm. The main conclusions are as follows.

- (1) The APCEA defines the non-robust gene based on the RCFPI, and screens the non-robust gene in the genetic operation, thus improving the solution quality under the same computation times.
- (2) The comparison results show that the proposed APCEA performs better than the other MOEAs (NSGA-II, SPEA2, ϵ -MOEA) under the same NFE. It is clear that the proposed APCEA method shows the fastest convergence speed and obtains the best performance in terms of HVR, IGD, and spacing indicators among the four algorithms.
- (3) Using the simplified model in the evolution process can greatly reduce the computational expense, because the simplified model has a high correlation with the original simulation model. After the evolution has been completed, the original model was used to evaluate the individuals in the final Pareto set to make a good balance between the solving speed and the solution quality. The proposed APCEA method was successfully integrated into the intelligent terminal of the MES, and the application value of the proposed method was demonstrated by a practical example.

In the future, the authors will carry out the factory’s ongoing intelligent manufacturing project to realize the networking of the APC system. The operation data from machine tools and the information from the first piece inspection will be collected by automatic data acquisition equipment. These will gradually accumulate to form the big data of the APC system. APCEA2.0 will focus on genetic programming, reinforcement learning, and other intelligent algorithms through the analysis and learning of historical data to realize a data-driven scheduling APC system.

Author Contributions: Q.L. contributed to the design of the concept and algorithms; J.Y. provided fund support and thesis guidance; Y.Z. assisted Q.L. in analyzing the experiment results and designing the surrogate model.

Funding: This work was supported mainly by the Beijing Key Laboratory of Digital Design and Manufacturing, and the National High Technology Research and Development Programme (863) of China under Grant 2012AA040907.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviation

g	the index of pieces in a task
i, j	the index of scheduling tasks
k, l	the index of pallets
m	the number of pallets for APC machining center
n	the number of scheduling tasks
fp_j	the pass rate of the first piece inspection of the j th task
$fp_{j,g}$	the g th first piece inspection result of task j
o_j	the j th task

q_j	the quantity of the j th task
w_j	the weighted tardiness of the j th task
b_j	the total processing time of the j th task
C_j	the completion time of the j th task
$C_{j,g}$	the completion time of the g th piece of task o_j
d_j	the due date of the j th task
f_j	the first piece inspection time of the j th task
$fa_{j,g}$	the g th first piece adjustment time of task o_j
p_j	the single piece processing time of the j th task
r_j	the release time of the j th task
s_j	the setup time of the j th task

References

- Giret, A.; Trentesaux, D.; Prabhu, V. Sustainability in manufacturing operations scheduling: A state of the art review. *J. Manuf. Syst.* **2015**, *37*, 126–140. [[CrossRef](#)]
- Kleindorfer, P.R.; Singhal, K.; Wassenhove, L.N.V. Sustainable Operations Management. *J. Prod. Oper. Manag.* **2005**, *14*, 482–492. [[CrossRef](#)]
- Mouzon, G.; Yildirim, M.B.; Twomey, J. Operational methods for minimization of energy consumption of manufacturing equipment. *Int. J. Prod. Res.* **2007**, *45*, 4247–4271. [[CrossRef](#)]
- Lee, H.C.; Ha, C. Sustainable Integrated Process Planning and Scheduling Optimization Using a Genetic Algorithm with an Integrated Chromosome Representation. *Sustainability* **2019**, *11*, 502. [[CrossRef](#)]
- Zhang, R. Sustainable scheduling of cloth production processes by multi-objective genetic algorithm with Tabu-enhanced local search. *Sustainability* **2017**, *9*, 1754. [[CrossRef](#)]
- Vieira, G.E.; Herrmann, J.W.; Lin, E. Rescheduling manufacturing systems: A framework of strategies, policies, and methods. *J. Sched.* **2003**, *6*, 39–62. [[CrossRef](#)]
- Ouelhadj, D.; Petrovic, S. A survey of dynamic scheduling in manufacturing systems. *J. Sched.* **2009**, *14*, 417. [[CrossRef](#)]
- Alcaide, D.; Rodriguez-Gonzalez, A.; Sicilia, J. A heuristic approach to minimize expected makespan in open shops subject to stochastic processing times and failures. *Int. J. Flex. Manuf. Syst.* **2005**, *17*, 201–226. [[CrossRef](#)]
- Alcaide, D.; Rodriguez-Gonzalez, A.; Sicilia, J. An approach to solve the minimum expected makespan flow-shop problem subject to breakdowns. *Eur. J. Oper. Res.* **2002**, *140*, 384–398. [[CrossRef](#)]
- Goren, S.; Sabuncuoglu, I. Robustness and stability measures for scheduling: Single-machine environment. *IIE Trans.* **2008**, *40*, 66–83. [[CrossRef](#)]
- Siedlak, D.J.; Pinon, O.J.; Robertson, B.E.; Mavris, D.N. Robust simulation-based scheduling methodology to reduce the impact of manual installation tasks on low-volume aerospace production flows. *J. Manuf. Syst.* **2018**, *46*, 193–207. [[CrossRef](#)]
- Sortrakul, N.; Nachtmann, H.L.; Cassady, C.R. Genetic algorithms for integrated preventive maintenance planning and production scheduling for a single machine. *Comput. Ind.* **2005**, *56*, 161–168. [[CrossRef](#)]
- Zhang, J.; Yang, J.; Zhou, Y. Robust scheduling for multi-objective flexible job-shop problems with flexible workdays. *Eng. Optim.* **2016**, *48*, 1973–1989. [[CrossRef](#)]
- Jin, Y.; Jiang, Z.; Hou, W. Multi-objective integrated optimization research on preventive maintenance planning and production scheduling for a single machine. *Int. J. Adv. Manuf. Technol.* **2008**, *39*, 954–964.
- Coello, C.A.C.; Lamont, G.B.; Van Veldhuizen, D.A. *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed.; Springer: New York, NY, USA, 2007; pp. 131–276.
- Zhou, Y.; Yang, J.; Zheng, L. Hyper-heuristic coevolution of machine assignment and job sequencing rules for multi-objective dynamic flexible job shop scheduling. *IEEE Access* **2019**, *7*, 68–88. [[CrossRef](#)]
- Liao, Q. Study of SVM-based Intelligent Dispatcher for Parallel Machines Scheduling with Sequence-dependent Setup Times. In Proceedings of the 2018 6th IEEE International Conference on Mechanical, Automotive and Materials Engineering (CMAME), Hong Kong, China, 10–12 August 2018; pp. 46–50.
- Gen, M.; Lin, L. Multiobjective evolutionary algorithm for manufacturing scheduling problems: State-of-the-art survey. *J. Intell. Manuf.* **2014**, *25*, 849–866. [[CrossRef](#)]

19. Lin, L.; Gen, M. Hybrid evolutionary optimisation with learning for production scheduling: State-of-the-art survey on algorithms and applications. *Int. J. Prod. Res.* **2014**, *25*, 849–866. [[CrossRef](#)]
20. Fang, P.; Jiang, Y.; Zhong, R.Y. Real-time monitoring of workshop status based on internet of things. In Proceedings of the 2018 48th IEEE International Conference on Computers and Industrial Engineering (CIE 48), Auckland, New Zealand, 2–5 December 2018.
21. Godinho Filho, M.; Barco, C.F.; Neto, R.F.T. Using Genetic Algorithms to solve scheduling problems on flexible manufacturing systems (FMS): A literature survey, classification and analysis. *Flex. Serv. Manuf. J.* **2014**, *26*, 408–431. [[CrossRef](#)]
22. Diabat, A.; Al-Araidah, O.; Alsyof, I.; Duh, C. A heuristic approach to scheduling jobs in machining centres equipped with automated pallet changers. *Int. J. Adv. Oper. Manag.* **2011**, *3*, 326–336. [[CrossRef](#)]
23. Zhou, Y.; Yang, J.J.; Zheng, L.Y. Multi-agent based hyper-heuristics for multi-objective flexible job shop scheduling: A case study in an aero-engine blade manufacturing plant. *IEEE Access* **2019**, *7*, 21147–21176. [[CrossRef](#)]
24. Pinedo, M.L. *Scheduling: Theory, Algorithms, and Systems*, 5th ed.; Springer: New York, NY, USA, 2016; pp. 13–25.
25. Graham, R.L.; Lawler, E.L.; Lenstra, J.K.; Rinnooy Kan, A.H.G. Optimization and approximation in deterministic sequencing and scheduling: A survey. *Ann. Oper. Res.* **1979**, *5*, 287–326.
26. Deb, K.; Amrit, P.; Sameer, A.; Meyarivan, T.A.M.T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [[CrossRef](#)]
27. Panwalkar, S.S.; Iskander, W. A survey of scheduling rules. *Oper. Res.* **1977**, *25*, 45–61. [[CrossRef](#)]
28. Lee, Y.H.; Bhaskaran, K.; Pinedo, M. A heuristic to minimize the total weighted tardiness with sequence-dependent setups. *IIE Trans.* **1997**, *29*, 45–52. [[CrossRef](#)]
29. Hadka, D.; Reed, P. Diagnostic assessment of search controls and failure modes in many-objective evolutionary optimization. *Evol. Comput.* **2012**, *20*, 423–452. [[CrossRef](#)] [[PubMed](#)]
30. Zitzler, E.; Laumanns, M.; Thiele, L. *SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multi-Objective Optimization*; ETH: Zurich, Switzerland, 2001. doi:10.3929/ethz-a-004284029.
31. Deb, K.; Mohan, M.; Mishra, S. *A Fast Multi-Objective Evolutionary Algorithm for Finding Well-Spread Pareto-Optimal Solutions*; KanGAL Report 2003002; KanGAL: Kanpur, India, 2003.
32. Van Veldhuizen, D.A.; Lamont, G.B. Multi-objective evolutionary algorithm test suites. In Proceedings of the 1999 ACM Symposium on Applied Computing, San Antonio, TX, USA, 28 February–2 March 1999; pp. 351–357.
33. Coello, C.A.C.; Cortés, N.C. Solving multi-objective optimization problems using an artificial immune system. *Genet. Program. Evol. Mach.* **2005**, *6*, 163–190. [[CrossRef](#)]
34. Schott, J.R. Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization. Master's Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1995.

