Multiple Precast Component Orders Acceptance and Scheduling

Wen Jiang, Lanjun Wu, and Yunzhong Cao

College of Architecture and Urban-Rural Planning, Sichuan Agricultural University, Chengdu 611830, China

Correspondence should be addressed to Wen Jiang; xuezhongsha_wen@163.com and Yunzhong Cao; caoyz@sicau.edu.cn

Received 15 June 2019; Accepted 17 January 2020; Published 29 February 2020

Academic Editor: Benjamin Ivorra

Copyright © 2020 Wen Jiang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Precast components manufacturer generally operates under limited production capacity and produces products of one order which may delay another. This paper develops a precast component order acceptance and scheduling model that aims to maximize the total profit in a stochastic multiple orders environment. In that model, the increasing of the overall profit of the precast component manufacturer is achieved by using a heuristic algorithm and a dynamic order acceptance heuristic. Results of numerical examples indicate the proposed model realizes the increasing total profit in most cases comparing to accept all of the orders. Besides, this study tested three order acceptance criteria and found that the profit-based criterion is to be more stable in terms of maximum total profit. This approach is anticipated to provide support to precast component manufacturers when faced with multiple orders in long-term production.

1. Introduction

Precast construction refers to the practice of transferring producing construction components to a manufacturer factory in an industrialized way. Comparing to the conventional cast-in-suit construction, precast construction provides controlled conditions for weather and quality, facilitates the compression of project schedules by changing the sequencing of the workflow, and reduces the material waste [1]. From a life cycle perspective, precast components such as fundamental can achieve lower environmental impacts compared to cast-in-suit fundamentals [2]. Successful evidence from Japan, Netherlands, and Sweden indicates that prefabricated housing has been routinely promoted as a means to improve the efficiency, quality, and environmental performance of house construction, usage, and demobilization [3]. Governments mandate the adoption of a certain amount of prefabricated products in all its residential projects, such as Singapore and China. The State Council of China announced that the proportion of the precast component (PC) in new constructions should reach 30% among some megacities in ten years [4]. This proportion of China is reported to be 4.9% in 2016 [5]. Also, the National Plan on New Urbanization 2014–2020 pointed out that industrialization of construction is one of the critical issues in the path of energy efficient urbanization in China [6]. Nowadays, many construction project developers have realized the irresistible trend and the potential benefit of adopting prefabrication buildings, which will lead to the uprising demand for PCs.

Proper management of PC production is critical to construction project when PCs contribute to a majority of the project progress [7, 8]. The time of delivery impacts significantly to the performance of a construction project [9]. Late delivery of PCs could prolong the construction period and cause unnecessary labor costs. On the other hand, early delivery demands extra space for storage and causes several non-value-adding operations such as double handling of materials [10]. Stationary flow shop and make-to-order are two significant characteristics of PCs production. Stationary flow shop means that most production operations are performed at a fix location while labor shifting [11]. Make-to-order refers that products are designed by customers with different specifications and no finished inventory is carried by the manufacturer for this type of product [12]. In that production system, the PC manufacturer is faced with a stream of orders which are arriving stochastic over time. The manufacturer gains a full payment if a particular PC order is delivered to the construction site just in time. Since the production capacity is
limited, produce one order may delay another one, and both early or late delivery will incur penalties. Extreme late delivery can lead to low profit and construction project progress prolonged which will damage the benefit of both the PC manufacturer and the construction contractor. Would it be better for if the manufacturer turns down that order in the beginning? How would the manufacturer plan the production schedule to ensure just in time delivery? This type of problem is known as the PC order acceptance and scheduling (OAS) problem.

Abundant studies had been conducted on the PC production scheduling problem. Such as a static single order flow shop scheduling model [13, 14] and job permutation rescheduling with demand variability [15]. Previous studies in the PC production scheduling area often focus on single order scheduling and aim to optimize the operating costs in various aspects. However, little attention has been paid to consider the operational decisions of the PC manufacturer in multiple orders environment. Some key questions are discussed in this study. (1) How could the manufacturer develop an appropriate strategy for order acceptance decision? (2) What are the feasible scheduling approaches of the new accepted PC orders to satisfy the on-time delivery under consideration of limited production capacity? The objective of this study is to develop a PC order acceptance and scheduling model (PC_OASM) to maximize the total profit of the PC manufacturer. To cope with the multiple orders arriving environment, this study modified the conventional PC production model to ensure the production time and production resource constraints between orders. The production scheduling of the model is solved by a genetic algorithm (GA) that aims to maximize the profit. Later, this model is integrated into a dynamic order acceptance heuristic to help determine whether to receive new incoming orders.

The rest of this article is organized as follows. Section 2 is a brief review of the related research. Section 3 formulates the multiple PC production model and describes the order acceptance decision. Section 4 presents the GA used to solve the optimal production sequence. Section 5 is the computational example to examine the proposed multiple PC order acceptance and scheduling model. Finally, the conclusion and future extensions are discussed in Section 6.

2. Literature Review

To provide the research background and highlight our contributions, we mainly review two relevant research streams that are representative and particularly relevant to our research. One is the production scheduling problem of PC production, the other is the order and scheduling problem of the general industry.

2.1. Precast Component Production Scheduling. The production schedule of PC is known to be a nondeterministic polynomial hard problem. It is a tough question for human decision-makers. Still, there are plenty of research studies conducted by computational methods. For example, Zhai et al. consider a schedule model for make-to-order precast production based on simulation technology and GA to minimize total costs [12]. Benjaoran et al. formulate a six-step PC production model with a flow shop scheduling technique and adopt GA to optimize a multiobjective function, which includes minimizing the total flow time, total machine idle time, and total tardiness and earliness [16]: Namely, the six steps are mold assembling, reinforcement and embedded parts placing, concrete casting, concrete curing, mold stripping, and production finishing. Later, Benjaoran and Dawood showed that the proposed model is statistically 25% total time shorter than the traditional earliest due date rule [17]. Yang et al. proposed a flow shop scheduling model of multiple production lines for PC manufacture and applied GA optimization to minimize the changing costs of PCs during production [18]. Anvari et al. considered the manufacturing, transportation, and assembly process and optimized time, cost, and safety with GA [19].

However, these studies all neglected the negative effects caused by early delivery. Chan and Hu proposed a flow-shop sequencing model to minimize the tardiness and earliness penalty and made a distinction between working and off duty time, and they also divided operations into preemptive and non-preemptive tasks [20]. Ko and Wang took buffer sizes into consideration and proposed a multiobjective GA to solve the precast production schedule model with minimal total production time and delay penalty [13]. Ko and Wang [21] considered production resources and buffer size between stations and proposed a multiple objective PC production scheduling model. Kong et al. studied precast construction as a whole sector and proposed a dynamic program that aims to search for solutions that entail maximum production efficiency while under constraints of changeover costs and production deadlines [22]. Wang and Hu make a step further by extending the PC production stages from six to nine, by involving mold manufacture, PCs’ storage, and transportation activities [14]. Later, they considered the PC production rescheduling [15], coordination of resources, and production scheduling [23]. Ma et al. studied the PC scheduling with multiple production lines with GA [24].

However, the literature on this stream concentrates on the production scheduling on the single order environment, and less attention has been paid on the multiple orders with dynamic arriving.

2.2. Multiple Orders Acceptance and Scheduling. Efficient and scientific order decision-making is an important issue for the MTO manufacturer. Slotnick presented a comprehensive review of the literature about OAS [25]. Order selection at the entry stage can lead to the increase of the profit and market share of the company significantly, and accepting or rejecting orders require capacity estimation of all existing orders [26]. Furthermore, using knowledge of the current production schedule when generating new schedules would perform better [27]. Nobibon and Leus studied a problem where a company has to select orders from a pool of firms...
planned orders using Branch and Bound algorithms [28]. Wang et al. developed a modified artificial bee colony algorithm to tackle the OAS problems in a two-stage MTO production system [29]. Ögüz et al. presented mixed-integer linear programming that is capable of decision OAS simultaneously in a single machine situation [30]. Some research studies study the OAS problem in a flow shop environment. Xiao et al. studied the permutation flow shop scheduling problem with order acceptance and weighted tardiness and developed simulated annealing (SA) based partial optimization to solve the problem [31]. Lin and Ying proposed a multi-initiator SA for the same problem [32]. Later, Xiao et al. studied the nonpermutation flow shop scheduling, developing a mixed-integer programming model and using a two-stage GA to derive solutions [33]. Wu et al. concentrated on the OAS with identical parallel machines flow shop and developed a water flow-like algorithm to solve the problem [34]. However, these research studies were carried out based on order selection from known arrived orders and neglected the dynamic of order arriving.

A small amount of research considered the dynamic order arriving circumstance. Nandi and Rogers applied a look-ahead simulation method to evaluate the dynamic order arrival, and a measurement rule based on the ratio of actual to maximal possible revenue was established [35]. Moreira and Alves used simulation to investigate multi-decision-making, which includes order acceptance, due date adjustment, order releasing, and scheduling in a flow shop to improve tardiness penalty and workload performance [36]. Xu et al. studied the OAS problem with sequence-dependent setup time based on opportunity cost [37]. The abovementioned literature focused on the order level scheduling, and some studies assumed that their existed several jobs are included in an order. Rahman et al. proposed two different order schedule strategies (right shift strategy and real-time strategy) for flow shop manufacturers and also used extended due date acceptance rule to estimate the order acceptance decision [38].

Although PC production scheduling and multiple orders acceptance scheduling are studied comprehensively in the past, as far as our understanding, few studies have pay attention to the influence of OAS of multiple orders in the PC production area. On the other hand, current research studies of the OAS problem tend to focus on the simple flow-shop environment, single machine, parallel machine, etc., whereas the parallel machine, time constraints included in the flow shop of PC production makes the OAS problem complex. It is not clear how the PC manufacturer tackles the OAS problem in the dynamic order arriving environment.

3. Precast Component Order Acceptance and Scheduling

In a multiple PC orders environment, the manufacturer is faced with two decisions in order to gain maximal profit among a lot of arriving orders: First, how to schedule the accepted new orders to minimize the delivery penalty and second, how to decide which orders are to be accepted or rejected. The order acceptance or rejection decision and scheduling are made separately and sequentially. To make the decision more reasonable, a trial schedule will be performed before determining order acceptance or rejection decision. The workflow of this model is illustrated in Figure 1. In the proposed PC-OAS model, an arriving order will be considered immediately using GA to generate the near-optimal schedule, then evaluated by the order acceptance rule. The production system will be updated by adding new order into a queue if it is accepted.

Moreover, few assumptions below are employed in this study:

(1) The order arrives randomly (no perceived arriving time)
(2) The composition of each order is unknown in advance
(3) The processing time of each PC on each process is known in advance
(4) Orders are independent of one another and rejection, or acceptance will not affect any other orders
(5) The due date of PCs among an order is counted individually since order arrival
(6) The sharing of common mold is allowed and no maximal quantity limits to mold production
(7) Preemptive orders are not considered in this study

3.1. The Multiple PC Orders Production Model. The PC production scheduling model has been developed by previous researchers [14, 17]. This paper inherits the constraints of working and nonworking hours and the interruptible parallel features among nine steps of the PC production process. However, the differences in the model are that the introduction of multiple orders production environment and the objective function of the maximum profit. The reason for setting maximizing profit rather than minimizing cost as the object is because of a special situation in OAS problem. When none of the orders is accepted by the manufacturer, then the total cost equals to zero. Moreover, this study proposes a novel process of handling the production mold for long-term production.

Contrary to set the maximal mold quantity as constraint, this study considers whether a new model can be produced in order to save production time. The parameters, decision, objective function, and constraints of multiple orders PC-OASM are formulated below.

Parameters list:

- \( l \): the number of orders, a very large positive constant
- \( n \): the number of PCs among an order
- \( m \): the number of production workstations
- \( k \): the index of order, \( k \in (1, 2, \ldots, l) \)
- \( i \): the index of job (PC), \( i \in (1, 2, \ldots, n) \)
\[ j: \text{the index of machine, } j \in (1, 2, \ldots, m) \]
\[ u_i: \text{PC type number} \]
\[ H_{W}: \text{normal working hour} \]
\[ H_{A}: \text{overtime working hour} \]
\[ H_{N}: \text{nonworking hour} \]
\[ w_1: \text{tardiness penalty cost per hour} \]
\[ w_2: \text{earliness penalty cost per hour} \]

Variable:

\[ at^k: \text{order arriving time} \]
\[ d_i: \text{allowed time period for production PC } i \text{ after order arrived} \]
\[ d^k_i: \text{due date of PC } i \text{ in order } k, \quad d^k_i = at^k + d_i \]
\[ r^k_i: \text{price for producing PC } i \text{ in order } k \]
\[ S^k_{ij}: \text{starting time of PC } i \text{ of order } k \text{ on workstation } j \]
\[ T_c^k: \text{completion time without the working time constraint} \]
\[ D: \text{accumulated working days, } D = \text{Integer}(T/24) \]
\[ p^k_{ij}: \text{processing time of PC } i \text{ in order } k \text{ on workstation } j \]
\[ u_i: \text{type number of PC } i \]
\[ Q_{ui}: \text{mold quantity of PC } i \text{ of type } u_i, \text{initial as } 0 \]
\[ \tau_{ui}: \text{mold free time of type } u_i \]
\[ p^k_i: \text{Produced type of PC } i \text{ in order } k \]
\[ \gamma: \text{binary number to ensure mold constrains, } \gamma \in (1, 0) \]

Decision variables:

\[ X^k \text{ the production sequence of order } k \]

Objective function:

\[
\max f(X^k) = \sum_{i=1}^{n} r^k_i - w_1 \max(0, T^k_{i,m} - d^k_i) - w_2 \max(0, d^k_i - T^k_{i,m}).
\]
The model is a maximization problem composed of one objective function and 12 constraints. The objective function (1) maximizes the profit of accepted orders and subtracts order early delivery or tardiness delivery penalty costs. The terms of this function represent the revenue, tardy delivery penalty, and early delivery penalty, respectively.

Constraint (2) calculates the starting time of PC of order k on workstation j. The first term ensures it initially equals to arriving time in the beginning. The second term ensures the last PC i − 1 is completed and the last process of PC i is ready. The third term ensures the availability of machines at the start of subsequent orders and other constraints identical to previous terms.

The production processes of PC are divided into 9 steps: ① mold manufacture; ② mold assembling; ③ reinforcement setting; ④ concrete casting; ⑤ curing; ⑥ mold removal; ⑦ finishing and repairing; ⑧ storing; ⑨ transportation. The work stations of these productions are represented with index j.

Constraints (3)–(8) calculate the completion time of order k. Constraint (3) calculates the completion time of PCs without considering the working and nonworking time constraint, which is similar to the situation of normal flow shop calculation. The completion time equals to start time plus processing time. Constraint (4) ensures the constraint of interruptible and sequential processes, including mold manufacture, mold assembling, reinforcement setting, mold
removal, finishing, and repairing, are satisfied. Interruptible and sequential mean that processes can be paused for workers who are off duty and resumed on the next day. Constraint (5) calculates the completion time of the concrete casting process. Because concrete must be cast into mold once, or there will be a potential weakening of strength. If it is not completed in working hour, the process will be removed to the next day. Constraints (6)–(8) calculate the completion time of uninterruptible and parallel processes. Once the production of the last process is finished, the parallel process, curing, storing, and transportation starts automatically. Constraint (6) calculates the completion time of parallel processes. Constraint (7) calculates the completion time of curing and storage processes. These processes require no labor and can be processed during nonworking hours, where \( D^* = \text{Integer}(T^*/24) \). Constraint (8) calculates the completion time of the transportation process because it is capable of processing a batch of PCs in one time and human labors are required. If the work lasts beyond the working hour, then remove it to the next day.

Constraints (9) and (13) calculate problems involving mold quantity and sharing constraints. Constraint (9) calculates the quantity of mold for PC type \( u_t \). Equation (10) records the produced PC type accordingly. Equation (11) records the mold removed time of the corresponding quantity of mold type \( u_t \). Constraint (13) ensures the production time relates to mold production calculate correctly. There is a total of four situations under mold quantity constraints. Firstly, each new encountered type of mold will be produced and its quantity increased by one. For others, if a type \( u_t \) is produced before, that is, \( u_t \in \rho \), then check the condition of time. Secondly, if a mold of the type \( u_t \) was removed from the previous PC and ready for another PC, that is, \( \min(\tau_{u_t}) \leq S_{u_t}^i \), then skip the mold production process and \( y = 0 \). Constraint (12) represents the third situation that if the mold was not removed from the previous PC and the waiting time \( \min(\tau_{u_t}) - S_t^k \) is less than the production process time then production will be paused until the mold is ready, that is, \( S_{u_t}^i = \min(\tau_{u_t}) \). For the last condition, if the waiting time for mold is longer than mold production time, a new mold of the same type will be produced, that is, \( Q_{u_t} + 1 \) and \( y = 1 \).

### 3.2. Order Acceptance

The decision about acceptance or rejection is made when the order arrives at the PC manufacturer. In the dynamic order acceptance problem, accepted or rejected decisions are made by a threshold [39]. The heuristic for order acceptance or rejection is provided below.

**Step 1: initialization**

1. The production starts and 1st order arrives with \( a^k \), \( d_{l,m}^k, r_i^k \)
2. Set \( k = 1 \)
3. Use the genetic algorithm in Section 4 to generate a solution and calculate the completion time \( T_{l,m}^k \) and profit.

**Step 2: while \( k \leq l \)**

1. Evaluate \( y_k \) of order \( k \)
2. If \( y_k = 1 \)
   
   - Accept order \( k \) and add into the accepted order list
   
   - Update the status of the PC mold and available time of machines
   
   Else, reject order \( k \)
3. Set \( k = k + 1 \)
4. End of step 2
5. End of algorithm

There are some acceptance evaluation criteria in the previous literature, such as accepting or rejecting by comparing the profit of new arriving order [35], rules based on workload and negotiable due date [36], and extend the acceptable due date with a proportion of total completion time of order [38]. This decision depends on the start time, completion time, and the due date of each order. Orders will be accepted if it can be done before a specified due date or slightly delayed with an acceptable penalty.

The profit acceptance (PA) adopted from Nandi and Rogers [35] is stated as follows. If the contribution for acceptance exceeds that for rejection by a proportion, denote as a coefficient \( \mu_1 \in [0, 1] \), of maximal possible contribution, and the order is accepted. In our case, the contribution is objective of the model \( \sum_{i=1}^{n} r_i^k - u_k \max(0, T_{l,m}^k - d_{l,m}^k) - w_0 \max(0, d_{l,m}^k - T_{l,m}^k) \). The maximal contribution \( R_k^\text{max} = \sum_{i=1}^{n} r_i^k \). Then, the PA decision is formulated in the following equation:

\[
y_k = \begin{cases} 
1, & f(X_k) \geq \mu_1 P_k^\text{max} \\
0, & f(X_k) < \mu_1 P_k^\text{max} 
\end{cases}
\]  

The extended due date acceptance (EDDA) rule by Rahman [38] is described below. The additional time is denoted as \( \mu_2 T_k^\text{max} \), of maximal possible contribution, represents the coefficient of extension for EDDA, \( T_k^\text{max} \) is the makespan of order \( k \) and equals to \( T_{k,n,n}^k - a^k \). If any PC is completed before the extended due date \( d_{l,m}^k + \mu_2 T_k^\text{max} \), the order \( k \) is accepted, otherwise rejected. The EDDDA is formulated in the following equation:

\[
y_k = \begin{cases} 
1, & T_{l,m}^k - d_{l,m}^k - \mu_2 T_k^\text{max} \leq 0 \\
0, & T_{l,m}^k - d_{l,m}^k - \mu_2 T_k^\text{max} > 0 
\end{cases}
\]  

This study proposed an order acceptance criterion by measuring the total delayed time of each component and production makespan of order, namely, total delayed acceptance (TDA). A coefficient \( \mu_3 \) is added to measure the difference between total delayed time and production makespan. The total delayed accepted time denoted as \( T^d = \sum_{i=1}^{n} \max(0, T_{l,m}^k - d_{l,m}^k) \). Then, the level of tardiness is defined as the additional time \( \mu_3 T_k^\text{max} \), and the decision about acceptance or rejection is described in the following equation:

\[
y_k = \begin{cases} 
1, & T^d - \mu_3 T_k^\text{max} \leq 0 \\
0, & T^d - \mu_3 T_k^\text{max} > 0 
\end{cases}
\]
4. Genetic Algorithm

The permutation flow shop scheduling problem (PFSP) is a NP-hard problem [40]. A majority of research is aiming to find the optimal solution to minimize the makespan [41]. However, the objective of PC production flow shop scheduling is to minimize the total tardiness and earliness. According to the recent literature about PC production scheduling [18, 19, 23, 24], GA has been wildly applied to find acceptable solutions at a reasonable time cost. Similar to the previous research studies, this study adopts a GA to search for the near-optimal solution. Figure 2 provides a flowchart of the GA.

4.1. Individual Representation and Initialization. A chromosome denotes the sequence of jobs to be considered for scheduling. The job-permutation-based representation has been widely applied in the literature for scheduling problems. A problem with $n$ job could be presented as a list with no repeated numbers from 1 to $n$. For example, a six job can be represented as [2, 3, 5, 1, 6, 4], which means the job-2 is produced first, follows job-3, then is job-5, and so on.

As for the initialization, the initial population size is set to $8 \times n$, where $n$ represents the job amount in the order. First, we apply NEHedd heuristic to generate an initial solution. Then, we use swap operation on the solution to create certain random individuals. Figure 3 shows an example of a chromosome before and after swap operation. At last, the rest of the population are generated randomly. In the NEHedd heuristic [42], jobs are sorted in a non-decreasing order of due date, adding at each step a new job in that order, and then inserting it in the best place by evaluating the lowest total tardiness. An extensive computational evaluation of heuristics for minimizing the total flow shop tardiness problem carried out by [43] shows that NEHedd is a key constructive heuristic for the problem. Note that, in the process of NEHedd, not any constraints were considered.

4.2. Evaluation and Selection. The evaluation is the same as the propose of the model, using equations (1)–(13) to evaluate the best individual that leads to maximal profit for that order.

The selection operation is carried out by the tournament selection scheme [44]. Tournament selection randomly samples some individuals repeatedly, and the one with the best fitness value is selected as parents until the size of the parent pool equals population size.

4.3. Crossover and Mutation. By using integer representation, the offspring after ordinary crossover could be illegitimate. This study adopted the operation called "similar job order crossover," according to [45]. The process of this operation is presented in Figure 4. First, identical jobs at the same position are copied over to both offspring. Then, each offspring inherits all jobs from one parent before a randomly chosen point. Finally, the missing numbers of each offspring are copied in the relative order from the other parent. To add variability to the population, the shift method is adopted. Figure 5 describes a job permutation before and after the shift of an example. A randomly picked position is relocated to another randomly picked position, and the jobs between these two positions are moved along. After crossover and mutation, the offspring will directly replace the current population, and duplicated individuals will be deleted from the population.

4.4. Stopping Criterion. The terminate conditions provide the criterion for stopping the evolutionary process, which in this study, is terminated by iterations.

5. Numerical Example

This study validated the performance of the proposed PC_OASM via two experiments. The first experiment demonstrates how PC_OASM works. The second one analyzes the performance of three different order acceptance criteria.

The input date of PCs’ type, process time, and due date are inherited from the previous research [14] and revenue is randomly created. Assume a PC manufacturer produces eight types of PCs. The corresponding processing time,
revenue, and time period are shown in Table 1. Eight orders are created by random select PCs from Table 1. Table 2 shows the PCs’ quantity and PCs’ types in each order. For example, the first order consists of a total 12 PCs, including two of type 4, three of type 3, three of type 8, and two of type 5. Nine cases of the order arriving are created and shown in Table 3. Other parameters are set as follows, delivery penalty \( w_1 = 10 \) and \( w_2 = 2 \), normal working hour \( H_W = 10 \), overtime working hour \( H_A = 0 \), and nonworking hour \( H_N = 14 \).

The code is programmed with Python 3 and implemented on a personal computer with Intel i5 processor (3.1 GHz) and 8 GB RAM. Python is a relatively slow language for mathematical computation comparing to C. We managed to speed up the performance using Numba [46], a JIT compiler, to compile to a native machine instruction and cut the runtime down. Calculation of a single order takes about 12 seconds. The parameters used in the genetic algorithm are set as follows:

- Crossover rate = 0.9
- Mutation rate = 0.3
- Tournament size = 5
- Population size = 8 * n

Figure 4: Similar job crossover.

Figure 5: Shift operation.
Experiment 1. This part of the experiment considers order 1 and order 2 from Case 6 as input to demonstrate the multiple PC orders scheduling method. Order 1 arrives at the manufacturer at the beginning, and the schedule starts. Based on the production time, due date, and revenue of PCs, GA calculates the near-optimal scheduling sequence that maximizes the profit. Results of order 1 are shown in Table 4. The production sequence of order 1 is job-4, job-10, job-5, etc., the total delivery time is 55 hours of earliness and 366.5 hours of tardiness, and results in a profit of 6387 for manufacture order 1. To make it more clearly, the Gantt chart of production time of order 1 is drawn in Figure 6. Each color in the top represents a certain PC type. The mold quantity of each type and its corresponding production time was calculated. Take job-4 and job-3, for example, both belong to type 8, job-4 is the first time that mold type 3 is encountered, mold was produced, and removed later. When job 3 was scheduled, mold type 8 was ready for another assembling, thus, the production time of job 3 on M1 is zero. Other process time is similar to that in a traditional flow shop environment, except that there are some overlaps in the parallel machines, such as M5, M8, and M9.

NEHedd population proportion = 0.3
Maximal iteration = 100

Table 1: PC information.

<table>
<thead>
<tr>
<th>PC types</th>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>M4</th>
<th>M5</th>
<th>M6</th>
<th>M7</th>
<th>M8</th>
<th>M9</th>
<th>Allowed production time</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11</td>
<td>1.5</td>
<td>2</td>
<td>0.5</td>
<td>8</td>
<td>1</td>
<td>0.5</td>
<td>10</td>
<td>1.5</td>
<td>74</td>
<td>819</td>
</tr>
<tr>
<td>2</td>
<td>11</td>
<td>1</td>
<td>2</td>
<td>0.4</td>
<td>8</td>
<td>1</td>
<td>0.5</td>
<td>10</td>
<td>1</td>
<td>80</td>
<td>998</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>1</td>
<td>1.5</td>
<td>0.5</td>
<td>8</td>
<td>0.5</td>
<td>0.5</td>
<td>10</td>
<td>1</td>
<td>78</td>
<td>923</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>0.5</td>
<td>1</td>
<td>0.3</td>
<td>8</td>
<td>0.3</td>
<td>0.5</td>
<td>10</td>
<td>1.5</td>
<td>90</td>
<td>920</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>1</td>
<td>0.8</td>
<td>1</td>
<td>8</td>
<td>1.5</td>
<td>0.5</td>
<td>10</td>
<td>1.5</td>
<td>70</td>
<td>702</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>0.5</td>
<td>2</td>
<td>0.4</td>
<td>8</td>
<td>0.5</td>
<td>0.5</td>
<td>10</td>
<td>1.5</td>
<td>94</td>
<td>736</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>1.5</td>
<td>2</td>
<td>0.5</td>
<td>8</td>
<td>1</td>
<td>0.4</td>
<td>10</td>
<td>0.5</td>
<td>84</td>
<td>740</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>0.5</td>
<td>2</td>
<td>0.3</td>
<td>8</td>
<td>0.6</td>
<td>0.3</td>
<td>10</td>
<td>1.5</td>
<td>78</td>
<td>656</td>
</tr>
</tbody>
</table>

Note. M \(i = 1, 2, 3, 4, 5, 6, 7, 8, 9\) represents the nine production processes, including mold manufacture, mold assembling, reinforcement setting, concrete casting, curing, mold removal, finishing and repairing, storing, and transportation respectively.

Table 2: Order information.

<table>
<thead>
<tr>
<th>Order number</th>
<th>Component quantity</th>
<th>Types contained PCs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>[4, 3, 8, 8, 5, 3, 2, 5, 4, 7, 8, 3]</td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td>[2, 8, 1, 5, 1, 1, 4, 5, 3, 8, 2, 3, 1]</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>[6, 5, 5, 1, 5, 7, 4, 5, 1]</td>
</tr>
<tr>
<td>4</td>
<td>13</td>
<td>[4, 7, 1, 2, 7, 1, 8, 6, 5, 4, 8, 3, 2]</td>
</tr>
<tr>
<td>5</td>
<td>19</td>
<td>[3, 4, 1, 8, 2, 2, 2, 5, 4, 8, 3, 8, 4, 4, 4, 3]</td>
</tr>
<tr>
<td>6</td>
<td>21</td>
<td>[5, 7, 3, 8, 6, 7, 8, 5, 3, 6, 1, 5, 5, 2, 6, 8, 5, 6, 2, 8]</td>
</tr>
<tr>
<td>7</td>
<td>16</td>
<td>[6, 7, 7, 1, 2, 7, 4, 6, 3, 1, 5, 5, 3, 6, 2, 8, 4]</td>
</tr>
<tr>
<td>8</td>
<td>15</td>
<td>[4, 1, 1, 8, 1, 1, 6, 2, 6, 2, 3, 8, 8, 3]</td>
</tr>
</tbody>
</table>

Table 3: Arriving times of each order.

<table>
<thead>
<tr>
<th>Case number</th>
<th>Arriving time of orders (hour)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>[0, 20, 43, 70, 121, 175, 178, 191]</td>
</tr>
<tr>
<td>Case 2</td>
<td>[0, 9, 67, 86, 134, 146, 153, 227]</td>
</tr>
<tr>
<td>Case 3</td>
<td>[0, 76, 112, 139, 165, 184, 191, 235]</td>
</tr>
<tr>
<td>Case 4</td>
<td>[0, 121, 175, 191, 243, 276, 299, 326]</td>
</tr>
<tr>
<td>Case 5</td>
<td>[0, 67, 112, 134, 153, 227, 265, 342]</td>
</tr>
<tr>
<td>Case 6</td>
<td>[0, 76, 139, 165, 191, 235, 246, 256]</td>
</tr>
<tr>
<td>Case 7</td>
<td>[0, 121, 175, 191, 276, 299, 326, 434]</td>
</tr>
<tr>
<td>Case 8</td>
<td>[0, 67, 134, 153, 227, 243, 265, 402]</td>
</tr>
<tr>
<td>Case 9</td>
<td>[0, 76, 112, 139, 191, 235, 246, 342]</td>
</tr>
</tbody>
</table>

Table 4: Computation result of order 1.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>4, 10, 5, 3, 6, 11, 9, 8, 2, 7, 12, 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Earliness</td>
<td>55</td>
</tr>
<tr>
<td>Tardiness</td>
<td>366.5</td>
</tr>
<tr>
<td>Profit</td>
<td>6387</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mold</th>
<th>Type 1</th>
<th>Type 2</th>
<th>Type 3</th>
<th>Type 4</th>
<th>Type 5</th>
<th>Type 6</th>
<th>Type 7</th>
<th>Type 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note. \(M_i\) \((i = 1, 2, 3, 4, 5, 6, 7, 8, 9)\) represents the nine production processes, including mold manufacture, mold assembling, reinforcement setting, concrete casting, curing, mold removal, finishing and repairing, storing, and transportation respectively.
Table 5 shows the results of the second order. The second order arrives at 76 hours but production begins at 99 hours. Because considering machine availability constraints, M1 is occupied in producing molds for the previous order until 99 hours, and the availability constraint of other machines is satisfied. Figure 7 shows the Gantt chart of the production time of order 2. Although order 2 contains 14 PCs, only 5 new molds were produced due to the sharing of mold. Take job-9 and job-5 as an example, both belong to type 5. The first mold of type 5 that produced during the production of order 1 is applied to job-9. When job-5 begins, it is not available and the waiting time is greater than the production time of the mold; therefore, a new mold of type 5 is produced.

This proposed model is able to process multiple orders production with shared mold, working time, and machine constraints. Experiment 1 shows that model performance achieves its desire. Additionally, the solution quality of the GA is tested by 100 consecutive runs. Figure 8 shows the evolution process of the first run. It can be seen that the value of objective function increases with the increase of generation, and finally the value remains stable. The coefficient of variation of GA is 0.037. Figure 9 shows the results of 100 consecutive runs and indicates that it can provide stable and reliable solution result to proposed PC_OAS model, but tends to converge to a near-optimal solution. Optimal solution total tardiness and earliness of PFSP is much harder to find than makespan. For makespan, the NEH algorithms have been relatively good for small scale problems with machines less than 10 [47]. However, only a few literature on minimizing total tardiness and earliness problem that is able to solve the optimal solution when the machine is less than or equal to 2 [48, 49]. Most of the research has been carried out around approximate solutions [50].

Experiment 2. In order to compare the performance of the PC_OAS model with three difference order acceptance criterion, PA, EDDA, and EDA and determine the value of the corresponding acceptance criteria; this part of experiment carried a serial simulation using data from Table 1, Table 2, and Table 3. The corresponding coefficient value for each simulation is set as follows:

- PA: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9
- EDDA: 1, 2, 3, 4, 5, 6, 7, 8, 9
- EDA: 0.2, 0.3, 0, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.7, 0.8

Table 6 shows the accepted order index and total profit under nine cases and different order acceptance coefficient values. For example, the results of PA (0.1) in case 1 represent that 1st, 2nd, and 5th are accepted, and the PC manufacturer obtained a total of 15950 for producing them. It can be seen that for the three acceptance criteria, a trend is clear that total profit starts to raise with more orders being

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Earliness hour</th>
<th>Tardiness hour</th>
<th>Profit</th>
</tr>
</thead>
<tbody>
<tr>
<td>8, 2, 12, 13, 14, 9, 11, 4, 1, 3, 10, 6, 5, 7</td>
<td>10.5</td>
<td>510.5</td>
<td>6391</td>
</tr>
</tbody>
</table>

Table 5: Computation result of order 2.

<table>
<thead>
<tr>
<th>Mold</th>
<th>Type 1</th>
<th>Type 2</th>
<th>Type 3</th>
<th>Type 4</th>
<th>Type 5</th>
<th>Type 6</th>
<th>Type 7</th>
<th>Type 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
accepted while the coefficient value increases and begins to drop later. The reason is that while the coefficient value of acceptance criteria increases, the possibility of accepting that increases, the number of accepted orders also increased. Then, the production process falls behind orders, which finally leads to the increasing of tardiness hours of order delivery. Therefore, accepting orders exceeding the production capacity will damage the benefit of PC
### Table 6: Accepted orders and total profit of simulations.

<table>
<thead>
<tr>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
<th>Case 5</th>
<th>Case 6</th>
<th>Case 7</th>
<th>Case 8</th>
<th>Case 9</th>
<th>Total Profit</th>
</tr>
</thead>
<tbody>
<tr>
<td>PA (0.1)</td>
<td>1 3 5</td>
<td>1 3 5 7</td>
<td>1 2 3</td>
<td>1 2 3 5 7</td>
<td>1 2 3 4 5 7</td>
<td>1 2 3 4 5 7</td>
<td>1 2 3 4 5 7</td>
<td>1 2 3 4 5 7</td>
<td>38099</td>
</tr>
<tr>
<td>PA (0.2)</td>
<td>1 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>80232</td>
</tr>
<tr>
<td>PA (0.3)</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>38615</td>
</tr>
<tr>
<td>PA (0.4)</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>38615</td>
</tr>
<tr>
<td>PA (0.5)</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>38615</td>
</tr>
<tr>
<td>PA (0.6)</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>38615</td>
</tr>
<tr>
<td>PA (0.7)</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>38615</td>
</tr>
<tr>
<td>PA (0.8)</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>38615</td>
</tr>
<tr>
<td>PA (0.9)</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>38615</td>
</tr>
<tr>
<td>EDDA (0.3)</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>38615</td>
</tr>
<tr>
<td>EDDA (0.35)</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>38615</td>
</tr>
<tr>
<td>EDDA (0.4)</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>38615</td>
</tr>
<tr>
<td>EDDA (0.5)</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>38615</td>
</tr>
<tr>
<td>EDDA (0.6)</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>38615</td>
</tr>
<tr>
<td>EDDA (0.7)</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>38615</td>
</tr>
<tr>
<td>EDDA (0.8)</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>1 2 3 5 7</td>
<td>38615</td>
</tr>
</tbody>
</table>

---

**Accept all**

<table>
<thead>
<tr>
<th>Total Profit</th>
</tr>
</thead>
<tbody>
<tr>
<td>23974</td>
</tr>
</tbody>
</table>
manufacturers. Also, it is obvious that results with order acceptance outperformed that of accepting all orders in most cases.

The PA criterion, it can be seen that maximum total profit is obtained with the coefficient $u_1$ from 0.4 to 0.6 in the nine cases, and the maximal average total profit is greater than that of EDDA, but slightly less than the one of TDA. The results of coefficient near to that gained the best results are found to be rather good, which means that the best coefficient of PA does not require to be precise. Moreover, in each case, the PA does not lead to negative profit, and the gap between minimal and maximal is less than the others. As for the EDDA criterion, the best value of the coefficient $u_2$ varies from 0.35 to 0.6 in different cases, and the maximal average total profit is lower than other criteria. For the TDA criterion, the average total profit is the largest of three when $u_3 = 3$, the best value of the coefficient is more concentrated than others. But its defection is similar to EDDA, which can lead to negative profit if it was set incorrectly. Thus, the overall performance of PC_OASM with PA is superior to the other two.

6. Conclusion

The PC is becoming more attractive in the construction area with its large advantages and policy stimulation. To the best of our knowledge, the authors have not found any paper considering the PC manufacturer facing continuous order flow environment. To deal with this problem, this study proposed a multiple orders PC_OASM based on GA aiming to maximize the profit of the PC manufacturer.

The primary contributions of the paper summarized are as follows:

(1) PCs’ order scheduling model is modified to cope with the multiple orders scheduling, and a new approach of handling the mold sharing constraints is developed.

(2) The multiple orders scheduling model is integrated into the dynamic order acceptance heuristic to formulate PC_OASM. Results of numerical examples indicate that reject orders that are not desirable help to improve overall profits.

(3) This study discovers that the profit-based criterion, PA, is to be the most effective for PC orders acceptance through the comparison of the existed dynamic order acceptance criteria with various acceptance coefficient value tested on total of nine cases.

The proposed PC_OASM alleviates the difficulties of PC manufacturers facing continuous orders under capacity constraints and helps to improve overall profits from a long-term perspective. However, certain defections still remain for future improvement. First, some companies in the construction industry maintain long-term cooperation with others, but this study neglected the cooperative activities within them. Second, the order acceptance coefficient we tested to be the best in our tested cases does not guarantee to be the best if the condition of tardiness penalty or price of PCs changed. To fix that adaptivity could be introduced by reinforcement learning, such as the Markov decision process. The GA used in this study has certain room for improvement, and we plan to implement the local search procedure to further improve the solution quality in future research.

Data Availability

All data used support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interests regarding the publication of this paper.

Acknowledgments

This research was partially supported by the Youth Foundation for Humanities and Social Sciences of Ministry of Education of China (no. 19YJC630063) and Youth Project of Education Department of Sichuan Province (no. 17ZB0335).

References


