Dynamic Planning and Scheduling for One-of-a-kind Assembly Production

Xu Xu¹, Quande Dong¹ and Ming Li²

1. School of Information Science and Engineering, Suzhou University, Anhui, China
2. School of Mechanical and Electronic Engineering, Suzhou University, Anhui, China

Abstract — A one-of-a-kind assembly production system is considered. In this system, every product has its own tree structure process route. The new order for the customized product arrives randomly and follows some known probability distribution. The objective is to assign an early due date to every new ordered product and guarantee that no already-arrived product will be delayed. In order to achieve this goal, we analyze the reasons that lead to tardiness in the considered production system and provide two types of effective dispatching rules that are combined by the three indices that represent the urgency of an operation in two different ways. Then we integrate the proposed rule into the binary searching algorithm for assigning due date. Numerical experiments indicate that the proposed integrated dynamic planning and scheduling algorithm assign comparatively early due dates to newly arrived orders and meanwhile guarantee the delivery reliability of all products.

Keywords - Planning and scheduling; lean manufacturing; OKP; assembly line;

I. INTRODUCTION

With drastic marketing competition and various customer requirements, the mode of multiple products and small batch production is applied more and more widely. In recent years, one-of-a-kind production emerges quickly, in which every kind of product only be produced once. This leads to huge difficulties in the planning and scheduling for the production process. In light of the emergence of the radio frequency identification (RFID) technology, the states of the complex manufacturing system which contain kinds of products can be tracked and control accurately in real time. Then the control in one-of-a-kind assembly production forms two problems: assigning due date for new orders and scheduling the operations of different products dynamically. Because of the personalized requirement for every single product, the objective is usually to deliver the product to the customer in promised due date. Delay of delivery will be penalized greatly. The performance of delivery is assessed in two points: (1) the delivery reliability, that is, satisfied the demand in promised due date; (2) the delivery speed, that is, promising a early due date [1]. It is helpful to win customer orders by assigning early due dates, but early due dates will lower the delivery reliability and lead to vast tardiness penalty. So early due dates are useless without delivery reliability. However, if a manufacturer assigns late due dates which can guarantee deliver on time, it leads to customer’s dissatisfaction and is likely to lose orders. Therefore, how to trade off between due date and delivery reliability is very critical, especially to one-of-a-kind assembly productions. In this sense, assigning an appropriate due date for a new order is very important. Moreover, because of the diversity of the product types, the scheduling of the operations is much more complex than that of general production. In this paper, new orders arrive at the system randomly, so the scheduling has to be determined by the states of the system in real time. The rules which determine the operations priority are called dispatching rules. The dynamic scheduling in this paper is quite different from the traditional static scheduling. Static scheduling is actually kind of combination optimization which aims to obtain the optimal operation sequence [2]. This paper considers two problems and the integration of them. The first one is to assigning a due date to the new ordered product and the second one is to schedule the operations of different products in real time. Finally we integrate these two problems together, aiming to dispatching operations efficiently and ensure that there is no tardiness in the production process. So in this paper, the main contributions are line in the following three aspects: (1) for a one-of-a-kind system, the reasons of the tardiness are revealed; (2) the new dispatching rules that can minimize the mean lead time are proposed. Different from the common rules, the proposed rules consider the tardiness forbidden constraints. (3) a due dates searching algorithm is proposed to minimize mean quoted lead time (due date minus arriving time) and meanwhile to ensure that there is no tardiness in the production process.

Many literature of due date assignment is about the single-machine situation. The objective is to minimize the cost by optimizing statically the due date and the operation sequence of all the jobs [3]. In recent years, to provide optimal due dates and operation sequence, [4] provided a polynomial-time algorithm. [5] considered the integrated due date assignment and job sequencing problems in single machine case under random job process times. [6] also
researched the due date assignment problem on the single-machine case. The topic on multi-machines case is also considered widely. For example, [7] studied this problem for the system that contains parallel machines. And for job shops, [8] proposed kinds of methods to assign due dates for products. All these studies consider the system which produce string-type jobs. However, in this paper we consider the manufacturing system that produces the products with tree structure operation-routing.

Most studies about dispatching rules focus on the jobs with the string-type processing routes. Many rules for this kind of manufacturing have been proposed and summarized in [9]. For the scheduling of a job shop containing m machines, [10] developed a dispatching rule that minimizes the mean weighted tardiness of the jobs. [11] studied a flexible flow shop system in which jobs arrive randomly and the setup times are determined by the sequence. [12] also proposed dispatching rules for job shop. Only a few studies [13-14] considered multilevel jobs case and one-of-a-kind assembly production. In these research, the priorities of operations are assigned offline. That is to say, the priorities of operations are assigned when the according jobs arrive in the system, and do not change in the whole process of production. However, in complex manufacturing process, critical path is not fixed but change along the production process. So the priorities of operations should also be dynamic as the critical path. Besides, the dispatching rules proposed in this paper ensure that each product is finished before its promised due date with the prerequisite that appropriate due dates have been assigned to each product (See details of this part in Section 3).

To summarize, in our research, every product is unique and arrives at the assembly production system randomly. The due dates of products should be assigned both to minimize the mean quoted lead time and to ensure the delivery reliability of every product. The scheduling is dynamic and the scheduling decision is made in real time by the current state of the system.

II. PROBLEM DESCRIPTION AND THE MATHEMATICAL MODEL

We consider dynamic planning and scheduling problem for a OKP production system. New products orders arrive in the manufacturing system randomly. The inter-arrival times between two consecutive new orders follows the exponential distribution. Each product has its unique tree structure processing route. When a new ordered product arrives, an appropriate due date assigned to it according to current state of the manufacturing system. Every product must be finished before the assigned due date, that is to say, tardiness is forbidden. Other assumptions are common as other research on manufacturing: (1) An operation can be loaded only when all of its preceding operations are finished; (2) For every product, each operation need be conducted on the specific machine; (3) one machine can perform only one operation at a time. The following are some notations in this paper:

Notations of operation $i$:
- $pi$: the processing time;
- $si$: the beginning time of the conduction
- $ci$: the finishing time ;
- $Pi$: the set of the immediate predecessors;
- $M_i$: the needed machine;

Notations of product $j$:
- $aj$: the arrival time;
- $fj$: the finishing time;
- $dj$: the promised due date;
- $lj$: the initial length of the critical path ;

Other notations:
- $Ot$: the set of all the operations in the system at time $t$;
- $J$: the set of the products.

The objective of this problem is to optimize the due date for each product and the real time operation dispatching. Therefore, we have the mathematical model of the considered problem as follows:

$$\text{Min } \sum_{j\in J} (dj - aj) \quad (1)$$

s.t.

$$f_j \leq d_j \text{ for any } j \in J \quad (2)$$
$$c_i = s_i + p_i \text{ for any } i \in Ot \quad (3)$$
$$s_i \geq \max\{c_i, k \in P_i\} \text{ for any } i \in Ot \quad (4)$$
$$s_i \geq c_i \text{ or } c_i < s_j \text{, for any } i, k \in Ot, \text{ and } M_i = M_k \quad (5)$$

Constraint (2) describes that every product is finished before its promised due date. Constraint 3 describes that each operation is conducted, it can not be interrupted. Inequality (4) describes one of the necessary conditions, that is, an operation can be performed only when all its preceding operations are finished already. Inequality (5) is the constraint about machines. It assume that each machine can only perform one operation at a time.

III. INTEGRATED DYNAMIC PLANNING AND SCHEDULING METHOD FOR OKP ASSEMBLY SYSTEM

In this section we will propose an integrated due dynamic algorithm for a OKP assembly system. It consists of two parts, i.e., the dispatching rules and the due date searching algorithms. To construct the dispatching rules, we consider the three indexes that reflect the urgency of an operation: the latest finishing time (LFT), the remaining path size (RPS) and the critical path (CP). So this rule is termed as the LFT+RPS+CP rule. The proposed due date searching algorithm optimizes the due date for every new ordered product by the simulation response. The method is termed as due date binary searching algorithm. The details of these proposed methods are described as follows.

A. Reasons That Lead To Tardiness In one-of-a-kind Assembly Production

Because the integrated method should make sure that no product be delayed, it is necessary to investigate the reasons of tardiness. It is assumed that all products in the system already can be completed without tardiness if there is no newly arrived product and at least one product will be...
delayed due to the newly arrived product. The reasons that lead to aforesaid tardiness in one-of-a-kind assembly production and the countermeasures to avoid them are summarized as follows:

(1). The newly arrived product is assigned a too early due date to finish before it. For example, if the due date of a product is earlier than its critical length, this product must be tardy.

(2). An operation cannot be performed before its latest starting time even the newly arrived product in the system is assigned a late enough due date. This situation may occur when all the immediate predecessors of operation \( i \) have finished and this operation need to be performed on a machine, but at this time, the machine is already occupied by operation \( j \) (the operation belongs to the new ordered product). Even the new ordered product has a very late due date and operation \( j \) has a very low priority, it also has the probability to occupy the machine when the machine is free and the immediate predecessors of operation \( i \) have not finished yet. Under this circumstances operation \( i \) must keep waiting until operation \( j \) is finished. If operation \( j \) finish later than the latest starting time of operation \( i \) then the product of operation \( i \) must be tardy. We define this reason that leads to tardiness as direct preemptive occupation. The following example is for illustrating the concept of direct preemptive occupation.

Example 1. In this example there are 4 workstations in which contains one machine. Each product has 4 operations. Each operation should be performed on the specific workstation. Let operation \((i, j)\) denotes operation \( j \) of product \( i \). Operation \( j \) need to be performed on machine \( j \). Candidate operation is the operation whose immediate predecessors have been completed or has no immediate predecessors. The figures of this example are shown in Fig. (1).

When \( t = 0 \), all machines are free, 1st product arrives at this time and operations \((1, 1)\) and \((1, 3)\) are candidate operations. The critical path of 1st product is \( 1 \rightarrow 2 \rightarrow 4 \) and the length of this path is 56. Suppose that the due date of 1st product is 66. In this situation, the Gantt chart of 1st product is shown in Figure 1 (c). When \( t = 15 \), 2nd product arrives and its critical path is \( 2 \rightarrow 3 \rightarrow 4 \), whose length is 69. The due date of 2nd product is assigned to 200, which is very late. Operations \((2, 1)\) and \((2, 2)\) are candidate operations. At this time, machine 1 is performing operation \((1, 1)\), so operation \((2, 1)\) must wait in buffer. Machine 2 is free at this time, so operation \((2, 2)\) begins to be performed on machine 2. When \( t = 20 \), operation \((1, 1)\) is completed and operation \((1, 2)\) becomes a candidate operation, but at this time machine 2 has been occupied by operation \((2, 2)\) of 2nd product. So operation \((1, 2)\) must keep waiting until operation \((2, 2)\) is completed. Because machine 2 is free, although product 2 has a very late due date, before operation \((1, 2)\) becomes a candidate operation, operation \((2, 2)\) occupies Machine 2 preemptively. In this situation, the Gantt chart of 1st product is shown in Figure 1 (d), the finishing time is 70 and the tardiness is 14. The occupation is direct preemptive occupation, which makes the completing time of operation \((2, 2)\) is later than its LFT and finally results in the tardiness of 1st product. The tardiness caused by direct preemptive occupation can not be voided by assigning a later due date to 2nd product.

![Fig. 1. The Direct Preemptive Occupation](image)

The countermeasure to avoid direct preemptive occupation is described as follows:

**Rule 1.** Among all the operations that will be processed on a machine and whose immediate predecessors are completed, the one satisfying the following condition should be scheduled: the completing time of this operation should be earlier than the latest starting times of all the other operations to be performed on this machine; otherwise the operation cannot be scheduled.

By this countermeasure, if the new ordered product is assigned an appropriate due date, direct preemptive occupation will be avoided completely.

(3). Similar to (2), operation \( i \) can be performed on a machine and the machine has been occupied by operation \( j \); however, the only difference is that operation \( j \) will finish...
before the latest starting time of operation \( i \). So operation \( i \) can finish before its latest finishing time, but the completing time of operation \( i \) is delayed when compared with the situation that there is no new ordered product (i.e. operation \( j \) is nonexistent and the machine is not occupied by it). Due to the delay of the operation \( i \), the subsequent operations of operation \( i \) on its path will have less slack time and tardiness possibly occurs at some operation of them. We define this reason that leads to tardiness as indirect preemptive occupation. The countermeasure to avoid indirect preemptive occupation is concluded as the following rule:

**Rule 2.** If a machine is free, consider all the operations that need this machine (whose immediate predecessors are not necessarily completed), and schedule the operation that has the highest priority (see Section 4.2 for the definition of priority) and will not lead to direct preemptive occupation. If the immediate predecessors of the operation have finished, perform the operation, otherwise keep the machine free.

The difference between Rule 1 and Rule 2 is that when a machine is free, Rule 1 just selects the operation from those whose immediate predecessors have been completed while Rule 2 select operation from all the operations that are likely to be performed on this machine.

By this countermeasure, if the newly arrived product is assigned an appropriate due date, indirect preemptive occupation will be avoided completely.

**B. The LFT+RPS+CP Rules**

We propose two operation-based rules to schedule the assembly production. We consider the priority of operations from three aspects which can represent the urgency of operations.

1. The latest finishing time (LFT). Each operation should be finished before its LFT, otherwise the product of the operation surely can not be finished on time. Therefore LFT is the most important and direct index to reflect the urgency. This is the basis factor in our consideration.

2. The number of the remaining operations on the current path of the operation. For two operations with the same LFT, we should compare on its path which one has more remaining operations. Because more remaining operations on its path means that the operation needs to compete more times with other operations for machine. Therefore the operation that has more remaining path size should be more urgent and has higher priority.

3. The critical path. It is defined as the path that currently has the longest total processing time. Different from the traditional concept, in this paper the critical path is dynamic. That is to say, during the production process the critical path may change from one to another. This new consideration evaluates whether an operation is on the longest path in real time. Here are some notations for the following analysis:

- \( \sigma(i) \): the product of operation \( i \);
- \( RPS \): the remaining path size;
- \( MPT \): the mean processing time of all operations;
- \( CP_i \): \( CP_i = 1 \) means the operation is on the current critical path, \( CP_i = 0 \) means the operation is on the normal path currently.

In this subsection two versions of LFT+RPS+CP rules are proposed, i.e. the selective LFT+RPS+CP rule and the combined LFT+RPS+CP rule.

**3.2.1 The selective LFT+RPS+CP rule**

In the selective LFT+RPS+CP rule, we combined the three indexes together. But this three indexes work one by one to adjust the priorities of operations. Every time when the rule is used, only a part of them are selected randomly. Meanwhile, delivering each product as promised should be taken into consideration simultaneously. The operation process of the selective LFT+RPS+CP dispatching rule is as the following steps:

Step 1. If a machine is free, sequence all the operations in the system (not only in this workstation; see Rule 2), that will be performed on this machine by LFT, the operation which has small LFT has high priority.

Step 2. If condition \( LFT_i < LFT_j + MPT \) holds, let the set that contains all the operation meeting the condition be \( S \). Go to the following three sub steps randomly with the same probability. Let \( l = 1 \).

Step 3.1. Go to Step 4.

Step 3.2. Rearrange \( S \). The operations which has great RPS has high priority. Go to Step 4.

Step 3.3. Rearrange \( S \). The operations which are on the critical path has high priority. Go to Step 4.

Step 4.

- If \( \forall j \in \{ j \mid j > i, a_{i, j} < a_{i, j} \} \), \( p_i + t < LST_j \) holds (i.e. Rule 1 is satisfied), go to step 6; otherwise go to Step 5.

Step 5. \( i \leftarrow i + 1 \); go to Step 4.

Step 6. If \( \forall j \in P_i, c_j < t \), perform operation \( i \); otherwise keep the machine free.

**3.2.2 The combined LFT+RPS+CP rule**

In the second LFT+RPS+CP rule, the same three factors are organized in different way. These three indexes are always work together. We define the priority of operation \( j \) as the following equation:

\[
e_j = LFT_j - \omega RPS_j - \tau CP_j
\]

where \( \omega \) and \( \tau \) are the coefficient that represents the proportion of PRS and CP in considering the priority of an operation. Meanwhile, delivering each job as promised should be taken into account simultaneously. The operation process of the combined LFT+RPS+CP rule is as the following steps:

Step 1. If a machine is free, Sequence all the operations need to be performed on this machine according to defined priorities (i.e., \( e_j \)). Let \( i = 1 \).

Step 2.

- If \( \forall j \in \{ j \mid j > i, a_{i, j} < a_{i, j} \} \), \( p_i + t < LST_j \) holds (i.e. Rule 1 is satisfied), go to step 4; otherwise go to Step 3.

Step 3. \( i \leftarrow i + 1 \); go to step 2.
Step 4. If \( \forall j \in P, c_j < t \) holds, perform operation \( i \); otherwise keep the machine free.

The major difference between these two dispatching rules is that the combined rule will select the same operation in the same situation while the selective rule may select different operations in the same situations due to its randomness.

C. Due Date Searching Algorithm

When a new ordered product arrives in the OKP system, the possible earliest due date should be assigned to the product. Meanwhile the due date should ensure that all the products in the system can be completed as promised. The due date searching algorithm that can solve this problem is as the following steps:

Step 1. Once product \( i \) arrives, let \( d^1_i \) be a lower bound of the due date, and \( d^2_i \) an upper bound due date that is large enough to ensure the delivery reliability of product \( i \).

Step 2. Let \( d_i = \frac{1}{2}(d^1_i + d^2_i) \), test if all the products can be finished before \( d_i \) under the combined LFT+RPS+CP rule. If so, let \( d_i^1 = d_i \); otherwise, let \( d_i^1 = d^1_i \).

Step 3. If \( d_i^2 - d_i^1 \approx 0 \), go to Step 4; otherwise, go to Step 2.

Step 4. Termination.

For each new ordered product, we search the optimal due date for it between \( d^1_i \) and \( d^2_i \). \( d^1_i \) is a due date that is too early to complete the product before it. For example, we can let \( d^1_i = a_{i+1} - 1 \). \( d^2_i \) is a sufficiently late due date that can assure all the products in the system can be completed without tardiness, for example, we can let \( d^2_i = a_i + c \times l_i \) and a sufficiently large \( c \) can ensure reliable delivery. Obviously, smaller \( c \) can make the searching converge faster but has a greater probability that the \( c \) cannot guarantee reliable delivery. In the numerical experiments in Section 4, we let \( c = 10 \). Numerical experiments indicate that the \( c \) is big enough.

To solve the integrated problem, a two-layer simulation is adopted. For each new ordered product, record the system states and start a new simulation based on it to search the optimal due date and examine whether the due date can insure that all the products in the system will be completed without tardiness. After several times of inner simulation, a possibly early due date is determined for the new ordered product. Then the outer simulation continues. Once a new product arrives the system the current state is deterministic. So the inner simulation processes for a new ordered product are repeated based on the same deterministic initial status. In inner simulations we do not consider other newly arriving products because we just need to determine whether the new ordered product with the due date will delay other products which are in produce. If every new ordered product is assigned a due date that can insure the delivery reliability of all the products already in the system, then there are no tardy products in the whole production process.

IV. Numerical Experiments

In this section, numerical experiments are conducted to examine the performance of the two LFT+RPS+CP rules and the due date binary searching algorithm. A one-of-a-kind assembly shop that contains 20 workstations is considered. The processing routes of products are different from each other. Every product is finished after 20 operations on the 20 workstations. The processing time of each operation follows a uniform distribution. The inter-arrival between two consecutive new ordered products follows an exponential distribution.

D. Comparison with other dispatching rules

In this subsection, we compare the proposed rules with other four rules that performs well in the dynamic scheduling of complex assembly production. We select four dispatching rules to compare: (1) LFT (The latest finishing date) rule, (2) LSD (The latest starting date) rule, (3) EDD (Earliest due date) rule, (4) ECT (Earliest completing time) rule. EDD rule and ECT rule are product-based dispatching rule, that is to say, priority of all operations in a product depends on some character of the product, such as due date; The other two and proposed dispatching rules are operation-based. According to the numerical experiments we consider two representative allowances (\( c = 1.8 \), 2). When \( c = 1.8 \), products are assigned comparatively early due date and many products will be tardy; when \( c = 2 \), products are assigned comparatively late due date and few products will be tardy. We consider ten groups of products and inter-arrival times and each group are produced under the selective LFT+RPS+CP, the combined LFT+RPS+CP (\( \omega = 1 \), \( \tau = 10 \)), the LFT, the LSD, the EDD and the ECT. Finally the average performance is shown the following tables.

<table>
<thead>
<tr>
<th>TABLE 1. PERFORMANCE EVALUATION (C = 1.8)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Operation-based rules</strong></td>
</tr>
<tr>
<td>Selective</td>
</tr>
<tr>
<td>Combined</td>
</tr>
<tr>
<td>LFT</td>
</tr>
<tr>
<td>LSD</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Product-based rules</strong></th>
<th><strong>Tardy products</strong></th>
<th><strong>Total tardiness</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>ECT</td>
<td>110.5</td>
<td>2826</td>
</tr>
<tr>
<td>EDD</td>
<td>104.8</td>
<td>3025</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE 2. PERFORMANCE EVALUATION (C = 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Operation-based rules</strong></td>
</tr>
<tr>
<td>Selective</td>
</tr>
<tr>
<td>Combined</td>
</tr>
<tr>
<td>LFT</td>
</tr>
<tr>
<td>LSD</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Product-based rules</strong></th>
<th><strong>Tardy products</strong></th>
<th><strong>Total tardiness</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>ECT</td>
<td>38</td>
<td>806</td>
</tr>
<tr>
<td>EDD</td>
<td>36.9</td>
<td>789</td>
</tr>
</tbody>
</table>

From Tables 1 and 2, the performances of the six dispatching rules in reducing tardiness for one-of-a-kind assembly production can be summarized as follows:
(1) Operation-based rules are obviously better than product-based rules.
(2) Two proposed dispatching rules outperform the other two operation-based rules in reducing tardiness. When \( c = 1.8 \), products are assigned comparatively early due dates. In this instance, number of tardy products when using the two LFT+RPS+CP rules are respectively 6.34% and 3% less than that number when using the LFT rule, 10.3% and 7.1% less than that number when using the LSD rule. Total tardiness when using two LFT+RPS+CP rules are respectively 9.2% and 4.3% less than that number when using the LFT rule, 12.9% and 8.2% less than that number when using the LSD rule. When \( c = 2 \), products are assigned comparatively late due dates. In this instance, when comparing the number of tardy products, two LFT+RPS+CP rules are 9.3% and 3.1% less than the LFT rule respectively, 15.2% and 9.4% less than the LSD rule respectively; when comparing total tardiness, two LFT+RPS+CP rules are 14.7% and 3.4% less than the LFT rule respectively, and 21.6% and 11.2% less than LSD rule respectively.

We record the standard deviations of the tardiness under the compared rules. The result is shown in Table 3.

**Table 3. Standard Deviation of the Tardiness**

<table>
<thead>
<tr>
<th>( \sigma )</th>
<th>Selective</th>
<th>Combined</th>
<th>LFT</th>
<th>LSD</th>
<th>ECT</th>
<th>EDD</th>
</tr>
</thead>
<tbody>
<tr>
<td>( c = 1.8 )</td>
<td>394</td>
<td>529</td>
<td>494</td>
<td>431</td>
<td>423</td>
<td>506</td>
</tr>
<tr>
<td>( c = 2 )</td>
<td>207</td>
<td>218</td>
<td>196</td>
<td>261</td>
<td>256</td>
<td>251</td>
</tr>
</tbody>
</table>

We can see from Table 3 that when \( c = 1.8 \) the selective LFT+RPS+CP and ECT rule has smaller deviation; when \( c = 2 \) the selective LFT+RPS+CP and LSD rule has smaller deviation. No dispatching rule outperforms other rules in all cases. But in general, the selective LFT+RPS+CP rule has an impressive stability.

**E. Experiments for the due date searching algorithm**

In this section we will focus on two problems, the first one is whether the integrated method can make sure that all the products can be finished on the promised due dates; the second one is whether the due dates assigned by this method is unnecessarily late, that is to say, whether the method can ensure that the quoted lead times (due date minus arriving time) are close enough to the lead times (completing time minus arriving time). The same ten groups of products as in subsection 4.1 will be assigned due dates and scheduled based on this method in simulation experiments. The mean due date and mean completing time of the products in each group will be calculated and compared. We will integrate the tardiness- forbidden combined LFT+RPS+CP rule into this method.

Though the selective LFT+RPS+CP rule performs very well in the numerical experiments in subsection 4.1, it can not be integrated in the method. Due to the randomness in this rule, the binary searching maybe stop early and assigns an unnecessary late due date to the new ordered product. For example, when we examine whether assigning a due date \( d_i \) to a new ordered product \( i \) can assure the delivery reliability of every product in the systems, tardiness will appear not only because \( d_i \) is too early but also because of the randomness in the dispatching rule. If unfortunately the second reason works, an unnecessary late due date will be assigned to product \( i \). Moreover, as what are mentioned before, even the new ordered product has been assigned a unnecessary late due date, the randomness of the dispatching rule still can not assure that every product will be completed as promised.

**Table 4. Performance of the Integrated Due Date Assignment and Dynamic Scheduling Method**

| Tardy products | 0 |
| Mean quoted lead time | 257.7 |
| Mean lead time | 255.7 |
| Mean ahead of time | 2 |
| Mean critical path length | 149.7 |
| Mean quoted lead time / Mean critical length | 1.73 |

Table 4 shows that this method can ensure that every product is finished before the promised due date. We inspect the ratio of mean quoted lead time to mean critical path length and conclude that the average value is 1.73. However, as shown in the numerical experiments in subsection 4.1, when we assign the due dates by the total work content, there are still many tardy products even the allowance factor equal to 2. This testifies that the integrated method can assign comparatively earlier due dates and ensure the delivery reliability of all products.

**V. Conclusions**

A integrated dynamic planning and scheduling problem in a OKP assembly system is studied in this paper. The new style of combining rules are introduced. We analyze the reasons that lead to tardiness in a OKP assembly system and provide two types of LFT+RPS+CP rules, i.e. the selective LFT+RPS+CP rule and the combined LFT+RPS+CP rule. Numerical experiments show that the proposed LFT+RPS+CP rules performs better in minimizing the number of the tardy products and the total tardiness of them. After that we propose an integrated due date assignment and dynamic scheduling method. This method integrates the combined LFT+RPS+CP rule into the due date binary searching algorithm and the numerical experiments show that the method can assign a comparatively early due date to the new ordered product and make sure that no product will be delayed.

**CONFLICT OF INTEREST**

The authors confirm that this article content has no conflicts of interest.

**ACKNOWLEDGMENT**
This work was financially supported by Anhui Provincial Education Department for Scientific Research of College and Universities (KJ2012Z400, KJ2013B285), Anhui Provincial Natural Science Research Project for college and universities(KJ2014ZD31, KJ2014A24), Suzhou college Natural Science Research Project (SZXYJYXM201233, SZXYSJJD201204).

REFERENCES


