Using simplified drum-buffer-rope for re-entrant flow shop scheduling in a random environment

Yung-Chia Chang* and Wen-Tso Huang

Department of Industrial Engineering and Management, National Chiao Tung University, Hsinchu 300, Taiwan, R.O.C.

Accepted 17 May, 2011

The traditional simplified drum buffer rope (SDBR) does not consider the application of re-entrant flow shop (RFS) in a random environment which might involve variable processing times and machine breakdowns. This paper proposed a weighted layer production buffer and weighted production buffer to monitor the status of the buffer deviation when applying SDBR in a RFS with non-deterministic parameters. The buffer status deviation used the overall urgency and actual urgency to estimate the influence of overall accumulated machine downtime and the influence of actual machine downtime of capacity-constrained-resource (CCR) at different layers. A dispatching rule, called SDBR_DReentry, was applied to decide the priority of all work orders by the buffer status deviation of each CCR machine with the consideration of the re-entry feature. A simulated RFS was designed and four different methods (including ours) were applied in order to demonstrate the effectiveness of our approach. The experimental results show that when compared to six performance indexes related to the due date, our approach has better performance than the other three methods when the product mix with a large proportion of multi-reentrant orders and when the CCR utilization increases from 60 to 90%.

Key words: Drum-buffer-rope, simplified drum-buffer-rope, re-entrant flow shop, scheduling, random environments.

INTRODUCTION

The basic characteristic of a re-entrant flow shop (RFS) is that all jobs have the same production route over the machines of the shop and require the same amount of processing time at each machine. In such a shop, each job may return to the same machine one or more times before completion (Graves et al., 1983). For example, in semiconductor manufacturing, a wafer needs to visit some certain machines several times for processing (Vargas-Villamil and Rivera, 2001). Furthermore, a production system is typically operating under a random environment. Sources of randomness include, for example, machine breakdowns, unexpected releases of high-priority jobs, the processing times that are not precisely known in advance, and the like (Pinedo, 2002). Nevertheless, there was not a complete and clear method for delineating a production system with a RFS in a random environment. For example, many papers have studied the RFS scheduling problems. Hwang and Sun (1998) applied a dynamic programming to solve a two-machine flow shop problem with re-entrant work flows and sequence dependent setup times. Their objective was to minimize the makespan. Park et al. (2000) proposed an approximate method based on the mean value analysis for estimating the average performance of RFS with single job machines and batch machines. Chen et al. (2007) introduced a hybrid tabu search technique to solve the scheduling problem of minimizing the makespan in a RFS. Chen et al. (2008) proposed a hybrid genetic algorithm to minimize the makespan for the RFS scheduling problems. Another line of work applied simplified drum buffer rope (SDBR), a standard production planning method using the theory of constraints (TOC), in a non-RFS environment. Schragenheim (2006) described the make-to-order aspects of implementing the due date setting method and the release policy of SDBR in a non-RFS environment. Lee et al. (2010) extended the work of Schragenheim...
Schragenheim and Dettmer (2001) stated that the advantage of SDBR is that it can be easily applied in most manufacturing environments when other methods seem to be complicated in practice. There is literature discusses the implementation of SDBR in a RFS environment. Chang and Huang (2011) proposed a set of shop-floor control mechanisms (named as SDBR_Reentry) in a RFS and showed that it performs better when the product portfolio has a large proportion of multi-reentrant orders or when the bottleneck machine of the system is full loaded. However, Chang and Huang (2011) did not implement SDBR in a random environment.

This paper extended the work done by Chang and Huang (2011) and proposed a set of SDBR control mechanisms and dispatch rules, named as SDBR_D_Reentry, for a random RFS. In this random RFS, the job process times are random and the machines are subject to random breakdowns.

To show the performance of SDBR_D_Reentry, we applied SDBR_D_Reentry to dispatch jobs with random process times in the simulated RFS based on a case company described in Chang and Huang (2011) but with random machine breakdowns. In addition to SDBR_D_Reentry, three other types of dispatching rules, including the original SDBR, the approach used by the case company, and the critical ratio approach, were also adopted. The simulated results were compared over the six due-date related performance indexes.

THE CORE OF THE SDBR

SDBR

Schragenheim and Dettmer (2001) stated that SDBR assumes that market demand (or the drum) is always a constraint and that the internal resources often have excess capacity. Therefore, no detailed schedule is created for the CCR under SDBR, but only monitors the CCR with total planned load (PL) to ensure that there is enough capacity to meet all due dates. The PL intends to convert each work order into the number of capacity hours of CCR that is needed in the future. For example, suppose work order 1 and work order 2 both require one hour of work on the CCR, the PL is then simply 2 h. Since the market is always a constraint, the buffer in SDBR is simplified into one single production buffer (PB) that represents the time of a work order, from the release date to the due date, and is equal to the cycle time. SDBR estimates the PB of each product using the touch time multiplied by a constant based on practical experience. The touch time represents the shortest processing time of a work order, including the setup time. The total touch time of a work order can be calculated by the summation of the shortest processing time on each machine where a work order passes through to obtain the estimated PB. Since each work order has an accumulated point in the CCR, the release point (or the rope) is obtained by moving each PL point backward by subtracting half of a PB. On the other hand, safe delivery is obtained by adding half of a PB. Usually, the promised due date can be estimated as additional buffer plus safe delivery. Schragenheim (2006) and Schragenheim and Burkhard (2007) introduced buffer management (BM), which is a mechanism that adopts the buffer status (BS) to decide the urgency of a work order. Figure 1 presents an example of buffer management. PB is divided into three equal areas (green, yellow and red). The penetration of a specific work order is obtained by calculating the time difference between its release date and when it reaches the CCR. In other words, penetration means the consumption of PB on a specific work order in front of the CCR, which can be simplified to the ratio of the accumulated flow time on the shop floor to the PB, that is, the BS, by equation (1). The actual sequence should be determined based on the overall urgency of each work order accumulating in front of the CCR by prioritizing their BS from high to low to obtain a complete real-time dispatching schedule for the CCR.

\[
\text{Buffer status (BS)} = \frac{\text{Flow time}}{\text{Production buffer (PB)}} \tag{1}
\]

An illustrative example of the SDBR dispatching rule: Lee et al. (2010) further defined BS as the dispatching rule of CCR by Equation (2).
The numerator in equation (2) represents the penetration. A high penetration rate implies that the increasing consumption of PB is leading to the possibility of delayed delivery. In other words, an order with a high BS value results in a higher priority to be processed by the CCR. Table 1 provides an illustrative example of four orders, in which the fourth work order has a high BS value and a higher priority of the CCR. Chang and Huang (2011) provided an example to illustrate the dispatching results of using the SDBR and SDBR_Reentry, as shown in Table 2. It was assumed that the PB is twice as much as the touch time for each product, and that all operations to be performed by a machine require the same processing time. The resulting outcomes provided in Table 2 show that under three situations, the performance of traditional SDBR would be limited when it is applied to a re-entrant manufacturing environment. By contrast, SDBR_Reentry, which adopts the deviation rate ∆BS as the dispatching rule in a RFS, will choose work orders that might result in possible delay. In fact, neither the dispatching rule BS introduced by Lee et al. (2010) nor the dispatching rule ∆BS illustrated by Chang and Huang (2011) considers machine failure in a RFS.

**SDBR_Reentry MODEL**

The SDBR_Reentry introduced by Chang and Huang (2011) has better performance in a RFS when the product portfolio has a large proportion of multi-reentrant orders, or when the CCR machine of the system is close to full loaded. However, SDBR_Reentry does not consider variable processing times and random machine failures even though in practice these two production variances are common occurrences (De et al., 1993; Arzi and Herbon, 2000; Mohebbi, 2008). In order to improve the applicability of SDBR for RFS in a random environment, this paper proposed the SDBR_Reentry method by extending the layered control of the re-entrant layers proposed by Chang and Huang (2011). The layered control diagram of the re-entrant layers in SDBR_Reentry is shown in Figure 2.

In SDBR_Reentry, the PB of each product in a random RFS is divided into the layer production buffer (LPB) based on the number of CCRs. The estimated value of each LPB includes the influence of variable processing times and machine breakdowns. The difference between SDBR_Reentry and SDBR_Reentry is that SDBR_Reentry only calculates the deviations of the overall buffer status (BS) and the layered buffer status (LBS), while SDBR_Reentry discusses every work order at different CCR layers as well as the influence of overall accumulated machine downtime on the BS and the influence of actual machine downtime at different CCR layers on the LBS.

The deviation between BS and LBS, denoted as ∆BS, is utilized as the dispatching rule for the CCR. The actual sequence on the shop floor should be determined based on the ∆BS of each work order accumulating in front of the CCR. A higher value of ∆BS implies a higher priority. Ranking ∆BS from high to low can obtain a complete real-time dispatching schedule for the CCR layers.

The notations used in this paper are given as:

\[ i : \text{Index of product}; \text{where } i = 1, 2, \ldots, m. \]
\[ j : \text{Index of order}; \text{where } j = 1, 2, \ldots, n. \]
\[ k : \text{Index of CCR machine}; \text{where } k = 1, 2, \ldots, o. \]
\[ l : \text{Index of re-entrant layer number}; \text{where } l = 1, 2, \ldots, p. \]
\[ M: \text{Number of orders}. \]
\[ n: \text{Number of orders of product } i; \text{where } i = 1, 2, \ldots, m. \]
\[ o: \text{Number of CCR machine}. \]
\[ P: \text{Number of re-entrant layers of product } i; \text{where } i = 1, 2, \ldots, m. \]
\[ SB_i: \text{The buffer status ratio for order } i \text{ of product } i; \text{where } i = 1, 2, \ldots, m, j = 1, 2, \ldots, n. \]
\[ PB: \text{The production buffer time for order } j \text{ of product } i \text{ from the release date to the due date}; \text{where } i = 1, 2, \ldots, m, j = 1, 2, \ldots, n. \]
\[ LBS_{ijl}: \text{Layer Buffer Status, the buffer status ratio for order } j \text{ of product } i \text{ at the } l \text{th layer}; \text{where } i = 1, 2, \ldots, m, j = 1, 2, \ldots, n, l = 1, 2, \ldots, pi. \]
\[ LPT_{ijl}: \text{Layer Processing Time, the actual processing time for order } j \text{ of product } i \text{ at the } l \text{th layer}; \text{where } i = 1, 2, \ldots, m, j = 1, 2, \ldots, n, l = 1, 2, \ldots, pi. \]
\[ TFT_{ij}: \text{Total Flow Time, the total flow time for order } j \text{ of product } i \text{ at the } l \text{th layer}; \text{where } i = 1, 2, \ldots, m, j = 1, 2, \ldots, n, l = 1, 2, \ldots, pi. \]
\[ MTBF: \text{The mean time to repair of CCR machine } k; \text{where } k = 1, 2, \ldots, o. \]
\[ MTTR_k: \text{The mean time to repair of CCR machine } k; \text{where } k = 1, 2, \ldots, o. \]

**Table 1. An illustrative example of buffer status.**

<table>
<thead>
<tr>
<th>Order no.</th>
<th>Touch time</th>
<th>PBa</th>
<th>Releasing date</th>
<th>Due date</th>
<th>Remain days to due date</th>
<th>BS</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9</td>
<td>18</td>
<td>1</td>
<td>19</td>
<td>10</td>
<td></td>
<td>(18-10)/18=44%</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>24</td>
<td>1</td>
<td>25</td>
<td>10</td>
<td></td>
<td>(24-10)/24=58%</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>18</td>
<td>1</td>
<td>19</td>
<td>9</td>
<td></td>
<td>(18-9)/18=50%</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>24</td>
<td>1</td>
<td>25</td>
<td>6</td>
<td></td>
<td>(24-6)/24=75%</td>
</tr>
</tbody>
</table>

...a. PB is two times of touch time.
Table 2. An illustrative example of SDBR and SDBR_reentry dispatching results (Chang and Huang, 2011).

<table>
<thead>
<tr>
<th>Situation</th>
<th>Order no.</th>
<th>PBa</th>
<th>CCR Layer</th>
<th>Total CCR layer</th>
<th>Total flow time</th>
<th>Layer flow time</th>
<th>LPBb</th>
<th>SDBR</th>
<th>SDBR_reentry</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>BSc (%)</td>
<td>Priority</td>
</tr>
<tr>
<td>a</td>
<td>1</td>
<td>18</td>
<td>2</td>
<td>2</td>
<td>16</td>
<td>5</td>
<td>8</td>
<td>89</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>24</td>
<td>2</td>
<td>3</td>
<td>16</td>
<td>3</td>
<td>8</td>
<td>67</td>
<td>*</td>
</tr>
<tr>
<td>b</td>
<td>3</td>
<td>18</td>
<td>1</td>
<td>2</td>
<td>12</td>
<td>1</td>
<td>4</td>
<td>67</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>24</td>
<td>2</td>
<td>3</td>
<td>16</td>
<td>1</td>
<td>8</td>
<td>67</td>
<td>*</td>
</tr>
<tr>
<td>c</td>
<td>5</td>
<td>24</td>
<td>3</td>
<td>3</td>
<td>16</td>
<td>1</td>
<td>6</td>
<td>67</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>24</td>
<td>2</td>
<td>3</td>
<td>16</td>
<td>1</td>
<td>8</td>
<td>67</td>
<td>*</td>
</tr>
</tbody>
</table>

- a. PB is assumed to be twice as much the touch times for each product; b. LPB equals to the weight of different layers multiplied by the production buffer; c. BS = Total Flow Time/ PB; d. LBS = Layer Flow Time/ LPB; e. ΔBS = BS(%) - LBS(%).

Figure 2. Layered control diagram of the re-entrant layers in SDBR_reentry (Chang and Huang, 2011).

DD_{ij}: The due date for order j of product i; where i = 1, 2, ..., m, j = 1, 2, ..., n.

OV_{ij}: The value of the orders for order j of product i; where i = 1, 2, ..., m, j = 1, 2, ..., n.

RD_{ij}: The release date for order j of product i; where i = 1, 2, ..., m, j = 1, 2, ..., n.

WV_{ij}: The value of the WIP for order j of product i; where i = 1, 2, ..., m, j = 1, 2, ..., n.

The core concepts of SDBR_reentry are stated thus: first, the layer touch time used in SDBR_reentry was replaced by the actual layer processing time (LPT) at each CCR layer. Therefore, the LPB of each work order is calculated by multiplying LPT by the weight of the CCR machine downtime occurring at different layers and a constant multiplier (MULT), as shown in Equation (3):

\[ LPB_{ij} = LPT_{ij} \times \left(1 + \frac{MTTR}{MTBF} \right) \times MULT, \]
where $i = 1, 2, ..., m$, $j = 1, 2, ..., n_i$, $k = 1, 2, ..., o$, $l = 1, 2, ..., p_i$.

The PB of each work order is then represented by the sum of LPB, as shown in Equation (3):

$$\text{PB}_{ijl} = \sum_{i=1}^{m} L\text{PB}_{ijl}, \text{ where } i = 1, 2, ..., m, j = 1, 2, ..., n_i.$$  (3)

BS is used to estimate the overall urgency of the due date for a specific work order. It is the ratio of the accumulated LFT at each CCR layer on the shop floor to the PB, calculated by Equation (5):

$$\frac{\text{TFT}_{ijl}}{\text{PB}_{ijl}} \times 100\% = \frac{\sum_{i=1}^{m} \text{LFT}_{ijl}}{\text{PB}_{ijl}} \times 100\%,$$

where $i = 1, 2, ..., m$, $j = 1, 2, ..., n_i$, $l = 1, 2, ..., p_i$.  (4)

Step 1: Calculate the LPB of each work order by equation (3).
Step 2: Calculate the PB of each work order by equation (4).
Step 3: Calculate the overall urgency BS of each work order by equation (5).
Step 4: Calculate the actual urgency LBS among the layers for each work order by equation (6).
Step 5: When the CCR needs to determine the priority of work orders accumulating in front of it, calculate the $\Delta$BS by using equation (7).
Step 6: Ranking the $\Delta$BS from high to low can obtain a complete real-time dispatching schedule for CCR.
Step 7: Non-CCR machines are dispatched following the FIFO rule.

Example: We use an example to illustrate the procedure of SDBR_DReentry. The inputs and the resulting outcomes are listed in Table 3. In Table 3, columns 2 to 10 are given as the settings of this example and columns 11 to 15 are calculated by using equations (3) to (7). To implement the SDBR approach in a random environment, this paper adopts a simulation plant designed by Chang and Huang (2011) based on the manufacturing process in Company K. In 2007, Company K implemented the SDBR approach in job dispatching and scheduling (Chang and Huang, 2011). Company K is interested in knowing if SDBR can be applied to a more realistic situation in which processing times are varied and machines randomly break down. To solve this problem, we propose a set of control mechanisms and dispatching rules in applying the SDBR for RFS in a simulated random environment.

Profile of the simulation environment

In order to compare the performance of SDBR_DReentry in a random RFS environment, this paper adopts a simulation plant designed by Chang and Huang (2011) based on the manufacturing process in Company K. In 2007, Company K implemented the SDBR approach in job dispatching and scheduling (Chang and Huang, 2011). Company K is interested in knowing if SDBR can be applied to a more realistic situation in which processing times are varied and machines randomly break down. To solve this problem, we propose a set of control mechanisms and dispatching rules in applying the SDBR for RFS in a simulated random environment.
Table 3. An illustrative example of SDBR_DReentry dispatching results in a random environment.

<table>
<thead>
<tr>
<th>Situation</th>
<th>Order no.</th>
<th>Total CCR layer</th>
<th>CCR layer no.</th>
<th>TFT</th>
<th>LFT</th>
<th>Accumulated down time (%)</th>
<th>LDT (%)</th>
<th>LPB</th>
<th>PB</th>
<th>SDBR BS (%)</th>
<th>Priority</th>
<th>SDBR DReentry LBS (%)</th>
<th>Priority</th>
<th>ΔBS (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>15</td>
<td>4</td>
<td>20</td>
<td>10</td>
<td>8</td>
<td>18</td>
<td>83</td>
<td>*</td>
<td>50</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>20</td>
<td>4</td>
<td>20</td>
<td>30</td>
<td>8</td>
<td>24</td>
<td>83</td>
<td>*</td>
<td>50</td>
<td>32 *</td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>20</td>
<td>4</td>
<td>20</td>
<td>10</td>
<td>6</td>
<td>24</td>
<td>83</td>
<td>*</td>
<td>67</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>20</td>
<td>4</td>
<td>20</td>
<td>30</td>
<td>8</td>
<td>24</td>
<td>83</td>
<td>*</td>
<td>50</td>
<td>32 *</td>
<td></td>
</tr>
</tbody>
</table>

Table 4. Product routing in the simulated plant (Chang and Huang, 2011).

<table>
<thead>
<tr>
<th>Product</th>
<th>Step</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>A</td>
<td>M1</td>
</tr>
<tr>
<td>B</td>
<td>M1</td>
</tr>
<tr>
<td>C</td>
<td>M1</td>
</tr>
</tbody>
</table>

Table 5. Calculation results of the average load for machines in the simulated plant (Chang and Huang, 2011).

<table>
<thead>
<tr>
<th>Product</th>
<th>Machine</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>M1</td>
</tr>
<tr>
<td>A</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
</tr>
<tr>
<td>Sum</td>
<td>6</td>
</tr>
<tr>
<td>Average load</td>
<td>2</td>
</tr>
</tbody>
</table>

aCCR machine which is identified by summing the processing times of all operations to be performed at a machine.

highest average load, can be the CCR in the simulated plant. Therefore, Product A, B, and C require 0, 1, and 2 re-entrant layers, respectively. We assume that PB is three times of touch time. In addition, we assume that there is only one type of WIP provided for all three types of products. The values of the WIP of three different products are 1, 2, and 3 units, respectively. Also, the values of the orders are 2, 4, and 6 units. We fix that one work order represents one lot, processing lot and transfer lot.
This paper considers that each variable processing time of the non CCR machines (M1, 3, 4, 5, and 6) follow a uniform distribution with a range from 3 to 12 h. Also, the CCR machines (M2) follows a uniform distribution ranging from 10 to 20 h. The data are derived from the actual manufacturing environment of the company K as shown in Table 6. This paper intends to verify the following 4 approaches involving the dispatching and release rules. That is, the SDBR_DReentry, the SDBR, the case-study method (we refer the method as the method K), and the critical ratio (CR). This study adopts six performance indexes related to the due date. The throughput dollar day (TDD) which is the summation of the value of the orders, multiplied by the number of days by which their delivery is late (Ho and Li, 2004), as shown in Equation (8):

\[
TDD = \frac{1}{M} \sum_{i=1}^{n} \sum_{j=1}^{m} \left[ OV_{ij} \times \left( Max(0, CD_{ij} - DD_{ij}) \right) \right] \tag{8}
\]

Equation 8 indicates that the manager can focus on the increase of the throughput by the TDD. The inventory dollar day (IDD) is the summation of the dollar value of the WIP multiplied by the time since the WIP entered the plant (Ho and Li, 2004), as shown in Equation (9):

\[
IDD = \frac{1}{M} \sum_{i=1}^{n} \sum_{j=1}^{m} \left[ WV_{ij} \times \left( CD_{ij} - RD_{ij} \right) \right] \tag{9}
\]

Equation 9 is used to guide managers to focus on reducing the plant’s actual WIP and production cycle time. The due date slack time (DDST) is represented by the number of days that the order is completed ahead of schedule, which depicts the ability to protect the due date, as shown in equation (10):

\[
DDST = \frac{1}{M} \sum_{i=1}^{n} \sum_{j=1}^{m} \left( DD_{ij} - CD_{ij} \right) \tag{10}
\]

The due date performance (DDP) is tracked to guide managers to focus on ensuring there is no delay for each order, as shown in Equation (11):

\[
DDP = \frac{1}{M} \sum_{i=1}^{n} \sum_{j=1}^{m} \left( U_{ij} \right) \tag{11}
\]

The average queue length in front of the CCR (Q_CCR) is the total average of queued work orders in front of the CCR, as shown in Equation (12):

\[
Q_{CCR} = \frac{1}{M} \sum_{i=1}^{n} \sum_{j=1}^{m} \left( Q_{ij} \right) \tag{12}
\]

The flow time is calculated by Equation (13):

\[
FlowTime = \sum_{i=1}^{n} \sum_{j=1}^{m} \left( CD_{ij} - RD_{ij} \right) \tag{13}
\]

Test problem design: This study considered variable processing times and machine breakdown situations in applying the SDBR for RFS in a simulated random environment. Each variable processing time for both non-CCR and CCR machine came from a uniform distribution with intervals ranging from 3 to 12 h and from 10 to 20 h, respectively.

The machine breakdown for every machine in the simulation plant is illustrated in Table 6.

The test problem in the simulation plant was considered using two factors in practice:

i) Different utilization ratios of the CCR: The basic assumptions behind SDBR are that the market is always a constraint and that internal resources often have excess capacity. Hence, this paper considered four non-full load CCR utilization ratios: 60%, 70%, 80%, and 90%, respectively.

ii) Different layer mixes of the CCR: Due to the complexity of the re-entrant environment being exacerbated by re-entrant processing (Yan et al., 2000), the performance of SDBR_DReentry might be limited when it is applied to a random environment. This paper took into account three combinational ratios of three layer mixes, which were simulated as M-M-M (1 : 1 : 1), L-H-L (1 : 6 : 3), and L-L-H (3 : 1 : 6). M represented Medium, L represented Low and H represented High. For example, L-L-H (3 : 1 : 6) represented a ration of A : B : C = 3 : 1 : 6. Table 7 shows the twelve test scenarios of the two factors.

Description of SDBR, critical ratio method and method K: To show SDBR_DReentry for RFS in a random environment, this paper
Table 7. The design of the test problems.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Utilization (%)</th>
<th>Layer mix</th>
<th>Proportion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>A&lt;sup&gt;a&lt;/sup&gt;</td>
<td>B</td>
</tr>
<tr>
<td>1</td>
<td>60</td>
<td>M&lt;sup&gt;b&lt;/sup&gt;</td>
<td>M</td>
</tr>
<tr>
<td>2</td>
<td>70</td>
<td>L</td>
<td>H</td>
</tr>
<tr>
<td>3</td>
<td>80</td>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>4</td>
<td>90</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>L</td>
<td>H</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>L</td>
<td>H</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>L</td>
<td>H</td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td>L</td>
<td>L</td>
</tr>
</tbody>
</table>

<sup>a</sup> The simulation plant produces three types of products: A, B, and C. <sup>b</sup> The triplet can be interpreted as follows: M represented Medium, L represented Low, and H represented High.

selected SDBR, CR, and method K to evaluate its efficiency, which was proposed by Chang and Huang (2011). The detailed steps are described thus:

**Procedure of SDBR**

**Step 1. Calculate the touch time of each product and estimate the PB:** Calculate the touch time of each product at the initial stage, then estimate PB of each product by the touch time multiplying the constant times on practical experience.

**Step 2. Calculate the PL of CCR:** When receiving a new order, the managers will convert each work order into the number of capacity hours of CCR that is needed in the future. Combining the resulting outcome with the original CCR, we can derive the planned load. Ensure that it will not exceed the due date after adding the half of a PB.

**Step 3. Decide the release schedule:** The work order is released into the shop floor according to the flow time it will work on CCR minus the half of a PB to get the release date, and prioritize the release date from now to future to obtain the complete release schedule.

**Step 4. Decide the dispatching schedule:** Calculate the overall urgency BS<sub>i</sub> of each work order waiting before CCR according to equation (14) (Russell and Taylor, 2005), and rank the BS<sub>i</sub> from high to low to obtain the complete real-time dispatching schedule for CCR. Non-CCR machines are dispatched under the FIFO rule.

**Procedure of CR**

**Step 1. Decide the CCR dispatching schedule:** Calculate the respective CR of each work order waiting before CCR according to equation (14) (Russell and Taylor, 2005), and rank the CR ratio from low to high to obtain the real-time dispatching schedule of CCR.

\[
\text{CR} = \frac{\text{Time Remaining}}{\text{Work Remaining}} \times 100%
\]

**Step 2. Decide the non-CCR dispatching schedule:** Non-CCR dispatching schedule based on the principle of FIFO rule.

**Procedure of method K**

**Step 1. Decide the release schedule:** The release point is obtained by moving backward each due date was assigned by the client subtracting one average quoted lead time (QLT) of the market, and rank them from the present time to the future to obtain the integrated release schedule.

**Step 2. Decide the dispatching schedule:** Calculate the Modified Critical Ratio (MCR) of each work order waiting before CCR according to equation (15). The difference between CR and MCR is the remaining processing time. That is, the part of denominator is the major difference, and rank the MCR ratio from low to high to obtain the complete CCR real-time dispatching schedule.

\[
\text{MCR} = \frac{\text{(Due Date - Today Date)}}{3 \times \text{Remaining Touch Time}} \times 100%
\]  

**Step 3. Decide the non-CCR dispatching schedule:** Decide the dispatching schedule of non-CCR under the FIFO principle.

**EXPERIMENTAL RESULTS**

The simulation experiments began with an empty plant. Machines were at either up or in idle states at the beginning. Each machine only processed one work order each time, while operating twenty-four hours per day, seven days per week. If a machine breakdown occurred during processing, the work order would start again after the machine's maintenance was completed. To provide a consistent basis for comparison between the four dispatching rules, the due date and release date were kept constant as three of touch times for each product. The four dispatching rules were respectively executed ten
times under twelve scenarios. The resulting simulation outcomes were as follows.

**Analysis of performance under different CCR utilization ratios**

Figures 3 and 4 show the TDD and IDD of the four methods under different CCR utilization ratios. When the PB was three times the touch time and the layer mix was kept constant, the TDD and IDD obtained from SDBR-D_{Reentry} was the smallest under the four CCR utilization ratios. Moreover, it had the largest gap with the other methods if the CCR utilization ratio was 90%. When the CCR utilization ratios increased from 60% to 90%, the gap between SDBR-D_{Reentry} and SDBR also increased. This study inferred that it was driven by the deviation rate $\Delta BS_{ijl}$ of the urgency of the influence on machine breakdowns.

Figures 5, 6 and 7 show the DDST, the $Q_{CCR}$ and the flow time obtained from the four methods, respectively, when the PB was three times the touch time and the layer mix was kept constant under different CCR utilization ratios. It can be seen from Figure 5 that the DDST obtained from SDBR-D_{Reentry} was the largest under all four CCR utilization ratios, which represented a better protection of the due date under SDBR-D_{Reentry}. Figure 6 shows the $Q_{CCR}$ obtained from SDBR-D_{Reentry}. The $Q_{CCR}$ was the smallest under the different CCR utilization ratios and had the biggest gap in comparison with SDBR, when the CCR utilization ratio increased from 60% to 90%. This resulted from the deviation rate $\Delta BS_{ijl}$ of the urgency of the influence on machine breakdowns for each re-entrant layer in SDBR-D_{Reentry}. Figure 7 shows that the flow time obtained from SDBR-D_{Reentry} was the smallest in comparison with other methods under the four CCR utilization ratios. The simulation results showed that the DDP of the four methods was 0%. Therefore, no further discussion was required.

**Analysis of performance under different CCR layer mixes**

Figures 8 to 12 show the changes of the CCR resulting from the five indexes. A change of the CCR showed that the layer mix had changed, but the rest of the factors were the same as before. In Figures 8, 9, 11 and 12, the TDD, the IDD, the $Q_{CCR}$ and the flow time obtained from SDBR-D_{Reentry} were the smallest under the three layer mixes. SDBR-D_{Reentry} had the biggest gap in comparison with the other methods when work orders with two CCR reentries were in the majority, which indicated a better performance if the product portfolio
Figure 4. IDD under different utilizations.

Figure 5. DDST under different utilizations.

had more multi-reentrant work orders.

Figure 10 demonstrates that SDBR\_D\textsubscript{Reentry} had the largest gap with the DDST in other methods when work orders (L-L-H) with two CCR reentries were in the
SDBR_D_reentry had the largest gap with SDBR when work orders with one CCR reentry were in the majority, indicating a better protection of the due date. The above
results showed that the deviation rate $\Delta BSijl$ of the urgency of the influence on machine breakdowns added
DISCUSSION

As mentioned earlier, this paper attempted to illustrate why SDBR_D_reentry could perform better than SDBR for RFS in a random environment. In the example, SDBR utilized the BS to determine the priority of all work orders, while SDBR_D_reentry utilized the ΔBS for RFS in a random environment (Table 3). Note that the BS depicted the planned value of the urgency for a specific work order with re-entrant flows, and did not consider machine failure.
situations for RFS in a random environment. By contrast, SDBR_D\textsubscript{Reentry} adopted the weighted LBS to judge the actual influence of machine failures on each work order among layers. Finally, after the deviation rate $\Delta B_S$ was measured, SDBR_D\textsubscript{Reentry} could choose work orders that would result in the influence of machine failures causing delay for a random RFS. Most previous SDBR studies implemented the SDBR in a non-random environment, but this paper pioneered the application of SDBR for a random RFS. For example, Schragenheim (2006) and Schragenheim and Burkhard (2007) introduced buffer management, which is a mechanism that adopts the buffer status to decide the priority of a work order in front of the CCR, but does not consider the influence of machine failures. By contrast, SDBR_D\textsubscript{Reentry} determines the priority of a work order with a consideration of the influence of machine failures. In addition, SDBR_D\textsubscript{Reentry} emphasizes the concept of a CCR machine that reduces the complexity of scheduling problems from all the machines to the CCR machine.

**CONCLUSIONS AND FUTURE RESEARCH**

The main contribution of this study was to propose SDBR_D\textsubscript{Reentry}, which considers the features of variable processing times and machine breakdowns of SDBR and adopts them for RFS in a random environment. This paper calculated a machine breakdown ratio through the overall accumulated and current CCR layers for each work order, then made a decision on the priority of all work orders based on the deviation rate $\Delta B_S$ for the buffer status of each re-entrant layer.

This study compared six performance indexes related to the due date by applying four methods, as mentioned earlier. The results showed that the proposed modified method has better performance in a random environment with RFS characters when the product portfolio has a large proportion of multi-reentrant orders, or when the CCR utilization ratio increases from 60% to 90%. The TDD, IDD, due date slack time, $Q_{CCR}$ and flow time obtained from SDBR_D\textsubscript{Reentry} also has better performance than the other methods. As this study did not consider the two machine bottleneck situations, future research should investigate this. In addition, further studies could perform bottleneck shiftiness on a system.

**REFERENCES**


