Optimizing cycle time average and variation for a wafer fabrication factory with a dynamic dispatching rule

Hsin-Chieh Wu¹, Toly Chen²,*, Yu Cheng Wang³


Abstract- A dynamic dispatching rule is proposed in the present work to improve the performance of reducing the average cycle time and cycle time variation in a wafer fabrication factory. The dynamic dispatching rule is an extension from the four-factor bi-objective nonlinear fluctuation smoothing rule (4f-biNFS) by dynamically adjusting the factors in 4f-biNFS in a Pareto optimization manner. In addition, the radical transition among successive rules is smoothed by circulating rules of the same type through their weighted geometric mean as an intermediary rule. We have also investigated some theoretical properties of the dynamic dispatching rule. Through validating the effectiveness of the dynamic dispatching rule with a simulated case, some evidences were found to support its effectiveness, from which we also derived several directions that can be exploited in the future.

Keywords: wafer fabrication, dispatching rule, dynamic, fluctuation smoothing

Nomenclature

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>the total number of jobs.</td>
</tr>
<tr>
<td>Rᵢ</td>
<td>the release time of job i.</td>
</tr>
<tr>
<td>RCTᵢₗ</td>
<td>the estimated remaining cycle time of job i since step j.</td>
</tr>
<tr>
<td>SKᵢₗ</td>
<td>the slack of job i at step j.</td>
</tr>
<tr>
<td>wᵤ</td>
<td>the weight of base rule u.</td>
</tr>
<tr>
<td>α</td>
<td>max(Rᵢ) − min(Rᵢ).</td>
</tr>
<tr>
<td>β</td>
<td>max(RCTᵢ) − min(RCTᵢ).</td>
</tr>
<tr>
<td>γ</td>
<td>N − 1.</td>
</tr>
<tr>
<td>λ</td>
<td>the mean release rate.</td>
</tr>
</tbody>
</table>

I. Introduction

Semiconductor manufacturing is undoubtedly one of the most noticeable high-technology industries nowadays because of the widespread applications of semiconductor products. However, the product life cycles of new semiconductor products are getting shorter. Therefore, semiconductor manufacturers are facing pressure to meet the various needs of customers within a shorter time span. More rapid product development, agile production, quick response, and others are considered to be viable strategies. One common feature of these strategies is the compression of the cycle times of the related processes. Among the various types of cycle time, production cycle time is particularly important because it determines the time of delivery to customers. In other words, if the production cycle time is shorter, the delivery to customer will be faster. To this end, shortening the production cycle time through effective job dispatching is an important task [1]. A lot of research has also been done concerning semiconductor shop floor control, in particular in the domains of deterministic scheduling and job dispatching. However, Chen [2-4] believed that to a semiconductor manufacturing factory, job dispatching is very difficult. Theoretically, this is a NP-hard problem. In practice, many semiconductor manufacturing factories suffered from lengthy cycle times, so are not able to make more favorable promises to their customers.

Semiconductor manufacturing can be divided into four stages: wafer fabrication, wafer probing, packaging, and final testing. The most important stage is wafer fabrication, which is also the most time-consuming one. In this study, we investigate job dispatching for this stage. Among the various categories of methods (including dispatching rules, heuristics, data mining-based approaches [5-6], agent technologies [5, 7-9], and simulation) in this field, dispatching rules have received a lot of attention these years [5-7] and are also the most prevalent in practical applications. For example, Altendorfer et al. [10]
proposed the work in parallel queue (WIPQ) rule targeting at maximizing throughput at a low level of work in process (WIP). Zhang et al. [11] proposed the dynamic bottleneck detection (DBD) approach by classifying workstations into several categories and then apply different dispatching rules to these categories. Three dispatching rules including first in first out (FIFO), the shortest processing time until the next bottleneck (SPNB), and critical ratio (CR) were used. Depending on the current conditions in the wafer fabrication factory, Hsieh et al. [6] chose one approach from the fluctuation smoothing rule for the mean cycle time (FSMCT), the fluctuation smoothing rule for cycle time variation (FSVCT), largest deviation first (LDF), one step ahead (OSA), and FIFO. Chen [12] modified FSMCT and proposed the nonlinear FSMCT (NFSMCT) rule, in which he smoothed the fluctuation in the estimated remaining cycle time and balanced it with that of the release time or the mean release rate. To diversify the slack, the ‘division’ operator was applied instead. Followed by Chen [13], in which the one-factor tailored NFSMCT (1f-TNSMCT) rule and the one-factor tailored nonlinear FSVCT (1f-TNFVSVC) rule were proposed. Both rules contain an adjustable parameter in order to customize them for a target wafer fabrication factory. As a multiple-objective study, Chen and Wang [14] proposed a bi-objective nonlinear fluctuation smoothing rule with an adjustable factor (1f-biNFS) to optimize the average cycle time and cycle time variation at the same time. More degrees of freedom seem to be conducive to the performance of customizable rules. For this reason, Chen et al. [15] extended 1f-biNFS to a bi-objective fluctuation smoothing rule with four adjustable factors (4f-biNFS). On drawback of such a rule is that only static factors are used, and its performance cannot evolve over time. To tackle this problem, Chen [16] established a mechanism that was able to adjust the values of the adjustable factor in 1f-biNFS dynamically (dynamic 1f-biNFS). Even with these progresses, it is still very difficult to find out the near-optimal dispatching rule for a wafer fabrication factory. In addition, the performance of a static dispatching rule does not evolve time. These considerations give rise to the emergence of a dynamic dispatching rule, as in [6], [11], and [16].

The current dynamic dispatching rules have some problems:
(1) The adjustment of factors is according to a rule that must be determined offline.
(2) The selection of such a rule is also subjective, and does not take into account the current conditions in the wafer fabrication factory.
(3) Moreover, the rule has not been optimized and leaves considerable room for improvement.

To tackle these problems and to enhance the performance of job dispatching in a wafer fabrication factory, a dynamic dispatching rule is presented in this study. As its name implies, we attempts to optimize two performance measures, the average cycle time and cycle time standard deviation, simultaneously:
(1) A dynamic adjustment mechanism is first designed for tuning the factors in 4f-biNFS in an online manner.
(2) A random weighting model is then used to adapt the dynamic dispatching rule to the target wafer fabrication factory.
(3) The Pareto optimality of the dynamic dispatching rule is continuously checked. To this end, the cycle time related statistics of the most recent finished jobs are recorded as a reference.
(4) Many existing dynamic dispatching rules (such as Hsieh et al. [6], Zhang et al. [11], and Chen [16]) suffer from the problem that the transition among rules is radical. To deal with this problem, we use multiple rules of the same type in a circulation manner, and establish an intermediary rule to smooth every transition.

The other sections of this paper are arranged in the following way. Section 2 provides the details the dynamic dispatching rule. In section 3, a simulated case is used to validate the effectiveness of the dynamic dispatching rule. The performances of some existing rules in this field are also examined using the simulated data and Section 4 finally, concludes this paper and points out some interesting topics for future work.

II. Methodology

Before applying the dynamic dispatching rule, the remaining cycle time required for each job needs to be estimated in advance. There is not too much research in this field, in this study we employ the fuzzy c-means (FCM) and fuzzy back propagation network (FBPN) approach by Chen et al. [17] because of its effectiveness. In the FCM-FBPN approach, FCM is first used to cluster jobs with similar attributes. Two FBPNs are then constructed for each job cluster so that the cycle time and step cycle time of each job herein can be estimated by the FBPNs, respectively. Eventually, we can derive the remaining cycle time of a job by simply subtracting the step cycle time from the cycle time.

The dynamic dispatching rule is revised from the 4f-biNFS rule:

\[
SK_i = (R_i - RCTE_{\min}(R_i)) \cdot f_i \cdot \alpha^{-f_i} + \left(\frac{R_i - RCTE_{\min}(R_i)}{\bar{R}} \cdot f_i \cdot \beta^{-f_i} \right) - \left(\frac{RCTE_{\min}(R_i)}{\bar{R}} \right)^{1/(f_i + f_i)}
\]

where \(f_i \sim f_k\) are constants that have to be determined in advance. Usually, some possible values of them are enumerated, from which the one giving the best result is chosen. However, (1) has several problems:
(1) It is a static one, and cannot reflect the changes in the factory conditions.
(2) Although (1) targets at the mean cycle time and cycle time standard deviation simultaneously, its static structure is not conducive to the fusion of these two considerations, as highlighted in Chen [16]. In this study, we first change (1) into a dynamic one, through specifying dynamic parameters instead:

\[ SK_i = (R_i - RCTE_i + (RCTE_i - \min(R_i)) \cdot f_i(t)) \cdot \alpha^{-f_i(t)} \]

\[ \left( \frac{i - RCTE_i}{\alpha} + \frac{1}{\alpha} \cdot f_i(t) \right) \cdot \left( \frac{\gamma}{\alpha} \right)^{-f_i(t)} \]

\[ (RCTE_i - \min(RCTE_i))^{f_i(t) + f_i(t)} \]

(2)

**Property 1.** The four adjustable factors in the dynamic dispatching rule have to satisfy the following constraints [15]:

If \( f_1(t) = 1 \) and \( f_2(t) = 1 \), then \( f_3(t) = 0, f_4(t) = 0 \), and vice versa

(3)

If \( f_1(t) = 0 \) and \( f_2(t) = 0 \), then \( f_3(t) = 1, f_4(t) = 1 \), and vice versa

(4)

If \( f_3(t) \geq f_3(t) \) and \( f_2(t) \geq f_2(t) \), then \( f_3(t) \leq f_3(t) \) and \( f_4(t) \leq f_3(t) \)

(5)

If \( f_3(t) \leq f_3(t) \) and \( f_2(t) \leq f_2(t) \), then \( f_3(t) \geq f_3(t) \) and \( f_4(t) \geq f_4(t) \)

(6)

where \((f_3(t), f_2(t), f_4(t), f_4(t))\) and \((f_3(t), f_2(t), f_3(t), f_3(t), f_2(t))\) are two different sets of the four dynamic factors. Various models have also been proposed to form such sets [15]:

**Linear model:**

\[ f_1(t) = f_2(t), f_3(t) = f_4(t), f_3(t) = 1 - f_3(t) \]

(7)

**Nonlinear model:**

\[ f_1(t) = f_2^m(t), f_3(t) = f_4^m(t), f_4(t) = 1 / f_3(t), m \geq 0 \]

(8)

**Logarithmic model:**

\[ f_1(t) = \ln (1 + f_2(t)) / \ln 2, f_2(t) = \ln (1 + f_2(t)) / \ln 2, f_3(t) = 1 / f_3(t) \]

(9)

In order to determine the best values of these factors, the conventional way is to evaluate a variety of combinations from which the one giving the best results is chosen [18]. To this end, a series of simulation experiments need to be performed. This is quite time-consuming, and can only consider a handful of combinations. In addition, usually not a single but several good combinations featuring some noninferior solutions can be obtained. To tackle these problems, Chen [18] attempted to relate the scheduling performance to these factors with a BPN so as to find out the optimal setting, but the accuracy was not good enough for the BPN to be useful. Since finding out a single optimal rule is so difficult, why don’t we use several good rules simultaneously, or at least successively, instead? To this end, the following systematic procedure is established:

Step 1. (Initialization) Initialize the value of \( f_1 \) randomly from [0 2.25], and then derive the values of the other factors according to the nonlinear model:

\[ f_{11} = \sqrt{f_1(t) \cdot f_2(t) \cdot f_3(t) \cdot f_4(t)} = \sqrt{1 / f_3(t)} \]

(10)

Step 2. (Job Dispatching) Dispatch jobs using the dynamic dispatching rule with the new factor set for 4 hours.

Step 3. (Performance Evaluation) Calculate the average cycle time and cycle time standard deviation for each product type and priority.

Step 4. (Optimality Check) Record the current setting as a base rule if it is not dominated by any of the previous rules. If the number of base rules reaches 4, go to step 5. Otherwise, go to step 1.

Step 5. (Rule Updating) Generate \( u \) random weights indicated with \( w_u, u = 1 - u \), and then normalize these weights such that \( \sum_{u=1}^{4} w_u = 1 \). Subsequently, update the dynamic dispatching rule by calculating the weighted geometric mean of the base rules:

\[ f_{11}^{(\tau)}(t) = \prod_{u=1}^{4} f_{1u}^{(\tau)}(t) = w_1 f_{11}^{(\tau)}(t) \]

(11)

where \((f_1(t), f_2(t), f_3(t), f_4(t))\) is the factor set of the \( \tau \)-th base rule. Continue dispatching jobs using the current setting for 4 hours. Evaluate the scheduling performance in the same manner as in step 3. If any base rule is dominated by the current setting, replace it with the current setting.

Step 6. (Circulation) Apply the base rules successively in a circulation manner as shown in Fig. 1. Basically, all base rules will be successively applied. After the last base rule is applied, we return to the first one and the process goes on. However, to avoid a radical transition between two successive base rules, the weighted geometric mean of all base rules serves as an
intermediary rule, and every base rule in function is replaced by the weighted geometric mean before transmitting to another base rule:

\[
\text{mod}(\text{int}(t/\theta), a) \quad \text{if} \quad \text{mod}(t/\theta) = 1
\]

\[
\text{mod}(\text{int}(t/\theta) - 1, a) \rightleftharpoons \text{mod}(\text{int}(t/\theta) + 1, a) \quad \text{otherwise}
\]

(12)

where \( r_{\text{effective}}(t) \) is the rule in effect at time \( t \); \( r_u \) is the \( u \)-th base rule; \( \theta \) is the circulation period; \( w \) is a real value between [0, 1]; \( \text{int}(a, b) \) rounds the value to the nearest integer; \( \text{mod}() \) calculates the remainder of dividing \( a \) by \( b \).

**Fig. 1. The circulation of base rules**

An example is given in Table I to illustrate the calculation of the weighted geometric mean. The convergence in the weighted geometric mean can be observed from the series of phase charts in Fig. 2.

**TABLE I**

<table>
<thead>
<tr>
<th>( T )</th>
<th>( f_1(t) )</th>
<th>( f_2(t) )</th>
<th>( f_3(t) )</th>
<th>( f_4(t) )</th>
<th>rule #</th>
<th>replacement</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.25</td>
<td>0.50</td>
<td>4.00</td>
<td>2.00</td>
<td>1</td>
<td>replace rule #3 with rule #4</td>
</tr>
<tr>
<td>168</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>2</td>
<td>replace rule #2 with rule #5</td>
</tr>
<tr>
<td>336</td>
<td>2.25</td>
<td>1.50</td>
<td>0.44</td>
<td>0.67</td>
<td>3</td>
<td>replace rule #4 with rule #6</td>
</tr>
<tr>
<td>504</td>
<td>0.83</td>
<td>0.91</td>
<td>1.21</td>
<td>1.10</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>672</td>
<td>0.59</td>
<td>0.77</td>
<td>1.69</td>
<td>1.30</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>840</td>
<td>0.50</td>
<td>0.70</td>
<td>2.01</td>
<td>1.42</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>1008</td>
<td>0.42</td>
<td>0.65</td>
<td>2.39</td>
<td>1.55</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

**Fig. 2. The convergence process of the weighted geometric mean**

In the dynamic dispatching rule, multiple rules are applied jointly. Compared with the existing approaches with similar treatments, such as Hsieh et al. [6] and Zhang et al. [11], we use rules of the same type to avoid a radical transition among rules. Such a property is further strengthened by the use of the weighted geometric mean as the intermediate rule, as proved in Theorem 2, which is distinct from the previous approach by Chen [16].

**III. Experimental Results and Discussions**

The effectiveness of the proposed dynamic dispatching rule was assessed with simulated data. To this end, a memory fabrication factory was simulated with a monthly capacity of up to 32,000 wafers. In the wafer fabrication factory, more than 500 workstations are devoted to single-wafer or batch production using 58nm~110nm technologies. Such a large scale accompanied with reentrant process flows make production control in the wafer fabrication factory a very tough task. Currently, the release policy adopted by the wafer fabrication factory is the uniform release policy, namely jobs are released into it every a fixed interval, which is a common characteristic of memory fabrication factories. FIFO is employed to sequence jobs on most of the workstations. For details of the production simulator, refer to Chen [15]. The wafer fabrication factory is also seeking better dispatching rules to replace FIFO, in order to shorten the average cycle times and quicken the delivery to its customers.

Although there are more than 10 products in the
wafer fabrication factory, only two major products occupying most of the factory capacity were considered, indicated with A and B. They were assigned various priorities. Jobs with higher priorities will be processed first.

Nine existing approaches, FIFO, earliest due date (EDD), shortest remaining processing time (SRPT), CR, FSVCT, FSMCT, 1f-TNFSVCT, 1f-TNFSMCT, and 4f-biNFS, were also evaluated using the simulated data. In EDD and CR, the internal due date of a job was determined by changing the cycle time multiplier [15]. Then, from several possible values, the one giving the best performance was chosen (see Figs 3 and 4). Two performance measures, the average cycle time and cycle time standard deviation of each product and priority by all approaches were compared, as summarized in Table II and Table III:

(1) The performances of these methods with respect to the average cycle time were compared in Table 2. From the tabulated results, it is obvious that the dynamic dispatching rule achieved a very good performance in shortening the average cycle times. Considering product B with normal priority, the advantage over FSVCT exceeded 40%. All the compared approaches were inferior to the dynamic dispatching rule in this respect.

<table>
<thead>
<tr>
<th>Avg. cycle time (hrs)</th>
<th>A (Normal)</th>
<th>A (Hot)</th>
<th>A (Super Hot)</th>
<th>B (Normal)</th>
<th>B (Hot)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIFO</td>
<td>1256</td>
<td>601</td>
<td>629</td>
<td>1278</td>
<td>457</td>
</tr>
<tr>
<td>EDD</td>
<td>1087</td>
<td>346</td>
<td>306</td>
<td>1433</td>
<td>478</td>
</tr>
<tr>
<td>SRPT</td>
<td>966</td>
<td>350</td>
<td>309</td>
<td>1737</td>
<td>483</td>
</tr>
<tr>
<td>CR</td>
<td>1143</td>
<td>356</td>
<td>301</td>
<td>1497</td>
<td>470</td>
</tr>
<tr>
<td>FSMCT</td>
<td>1401</td>
<td>405</td>
<td>320</td>
<td>1408</td>
<td>430</td>
</tr>
<tr>
<td>FSVCT</td>
<td>1046</td>
<td>385</td>
<td>317</td>
<td>1745</td>
<td>519</td>
</tr>
<tr>
<td>1f-TNFSMCT</td>
<td>1353</td>
<td>379</td>
<td>298</td>
<td>1271</td>
<td>409</td>
</tr>
<tr>
<td>1f-TNFSVCT</td>
<td>1368</td>
<td>365</td>
<td>299</td>
<td>1370</td>
<td>413</td>
</tr>
<tr>
<td>4f-biNFS</td>
<td>1228</td>
<td>356</td>
<td>289</td>
<td>1223</td>
<td>422</td>
</tr>
<tr>
<td>The proposed methodology</td>
<td>926</td>
<td>312</td>
<td>252</td>
<td>1012</td>
<td>349</td>
</tr>
</tbody>
</table>

(2) At the same time, it can be seen from Table 3 that the cycle time standard deviation was well controlled by applying the dynamic dispatching rule. Taking product A with super hot priority as an example, even the most time-consuming job the deviation of its cycle time from the average value was only 12 hours, which is not easy for job dispatching in a wafer fabrication factory and is conducive to making a reliable due date promise.

<table>
<thead>
<tr>
<th>Cycle time std. dev. (hrs)</th>
<th>A (Normal)</th>
<th>A (Hot)</th>
<th>A (Super Hot)</th>
<th>B (Normal)</th>
<th>B (Hot)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIFO</td>
<td>56</td>
<td>24</td>
<td>23</td>
<td>87</td>
<td>40</td>
</tr>
<tr>
<td>EDD</td>
<td>130</td>
<td>25</td>
<td>23</td>
<td>50</td>
<td>39</td>
</tr>
<tr>
<td>SRPT</td>
<td>246</td>
<td>32</td>
<td>23</td>
<td>106</td>
<td>30</td>
</tr>
<tr>
<td>CR</td>
<td>68</td>
<td>30</td>
<td>19</td>
<td>58</td>
<td>37</td>
</tr>
<tr>
<td>FSMCT</td>
<td>42</td>
<td>44</td>
<td>23</td>
<td>35</td>
<td>28</td>
</tr>
<tr>
<td>FSVCT</td>
<td>319</td>
<td>35</td>
<td>28</td>
<td>222</td>
<td>55</td>
</tr>
<tr>
<td>1f-TNFSMCT</td>
<td>81</td>
<td>43</td>
<td>22</td>
<td>49</td>
<td>25</td>
</tr>
<tr>
<td>1f-TNFSVCT</td>
<td>44</td>
<td>28</td>
<td>18</td>
<td>31</td>
<td>21</td>
</tr>
</tbody>
</table>

(3) From Figs 3 and 4, it is obvious that the effects of the cycle time multiplier on EDD and CR were quite different, even if they employed the same method of determining the internal due date.

(4) Compared with the original rule, 4f-biNFS, the dynamic dispatching rule also exhibited considerable superiority, in terms of either the average cycle time or cycle time standard deviation. The advantages in these two respects were 16% and 13% on average, respectively, which confirmed the usefulness of dynamic factor adjustment/optimization to a tailored rule like 4f-biNFS.

To make sure whether the differences between the performance of the dynamic dispatching rule and those of the nine existing approaches are significant, the following hypotheses were tested: 

H₀: The scheduling performance in shortening the average cycle time of the dynamic dispatching rule is the same as that of the compared existing approach.
**H$_{a1}$:** The scheduling performance in shortening the average cycle time of the dynamic dispatching rule is better than that of the compared existing approach.

**H$_{a0}$:** The scheduling performance in reducing cycle time standard deviation of the dynamic dispatching rule is the same as that of the compared existing approach.

**H$_{b1}$:** The scheduling performance in reducing cycle time standard deviation of the dynamic dispatching rule is better than that of the compared existing approach.

Several statistical methods have been developed for testing these hypotheses at a specified significance level $\alpha$. One of the most commonly used methods is Wilcoxon sign-rank test. The results were summarized in Table IV. The null hypothesis $H_0$ was rejected at $\alpha = 0.025$, showing that the dynamic dispatching rule was superior to the existing approaches in reducing the average cycle time. On the other hand, the advantage of the dynamic dispatching rule over seven of the nine existing approaches in reducing cycle time standard deviation was also significant at $\alpha = 0.025$ or 0.05.

<table>
<thead>
<tr>
<th>Method</th>
<th>$Z_{\alpha}$</th>
<th>$Z_{0.025}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIFO</td>
<td>2.02**</td>
<td>1.75*</td>
</tr>
<tr>
<td>EDD</td>
<td>2.02**</td>
<td>1.75*</td>
</tr>
<tr>
<td>SRPT</td>
<td>2.02**</td>
<td>2.02**</td>
</tr>
<tr>
<td>CR</td>
<td>2.02**</td>
<td>2.02**</td>
</tr>
<tr>
<td>FSMC</td>
<td>2.02**</td>
<td>2.02**</td>
</tr>
<tr>
<td>FSVCT</td>
<td>2.02**</td>
<td>2.02**</td>
</tr>
<tr>
<td>IF-TNFSMCT</td>
<td>2.02**</td>
<td>2.02**</td>
</tr>
<tr>
<td>IF-TNFSTVCT</td>
<td>2.02**</td>
<td>1.62</td>
</tr>
<tr>
<td>4f-biNFS</td>
<td>2.02**</td>
<td>1.21</td>
</tr>
</tbody>
</table>

* $p < 0.05$
** $p < 0.025$
*** $p < 0.01$

IV. Conclusions and Directions for Future Research

In this paper, we have presented a dynamic dispatching rule to provide superior performance, in terms of the average cycle time and cycle time standard deviation, for job dispatching in a wafer fabrication factory. Modified from 4f-biNFS, the dynamic dispatching rule dynamically adjusts the four factors in 4f-biNFS in an optimization manner. We have also discussed some theoretical properties of the dynamic dispatching rule.

A simulation experiment was set up to validate the effectiveness of the dynamic dispatching rule:

1. The dynamic dispatching rule incorporates the concepts of dynamic adjustment and optimization, so as to avoid the drawbacks of the existing tailored nonlinear fluctuation smoothing rules. Through self-adjustment and continuously responding to the changing conditions in the wafer fabrication factory, the dynamic dispatching rule proved itself as an effective dispatching rule in the simulation experiment.

2. The bi-objective nature of the proposed rule was fully revealed by the simultaneous improvements in both the average cycle time and cycle time standard deviation, which were also examined and confirmed by statistical analyses.

3. To dynamic dispatching rules, the avoidance of radical transitions among rules is an important issue, which is tackled in this study by circulating rules of the same type through their geometric mean as an intermediary rule. However, another way of dynamically adjusting the factors in similar rules can be exploited further. In addition, there are other techniques of optimization, such as scalarization and compromisation [18-27], that can be applied as well in future studies.

Acknowledgment

This work was supported by National Science Council of Taiwan.

References


[10] Altendorfer K., Kabelka B., Stöher W., A new dispatching rule for optimizing machine utilization at a semiconductor test field,


AUTHORS’ INFORMATION

1Hsin-Chieh Wu is with the Department of Industrial Engineering and Management, Chaoyang University of Technology, Taichung County 413, Taiwan, R.O.C.

2Toly Chen is with the Department of Industrial Engineering and Systems Management, Feng Chia University, Taichung City 407, Taiwan, R.O.C.

3Yu-Cheng Wang is with Ph.D. Program in the Department of Industrial Engineering and Systems Management of Feng Chia University, Taichung City 407, Taiwan, R.O.C.

Hsin-Chieh Wu received the Ph.D. degree in Industrial Engineering from National Tsing Hua University. He is currently an associate professor of the Department of Industrial Engineering and Management at Chaoyang University of Technology.

Toly Chen received the B. S. degree, the M S. degree, and the Ph. D. degree in industrial engineering from National Tsing Hua University. He is now a Professor in the Department of Industrial Engineering and Systems Management of Feng Chia University.

Yu-Cheng Wang received the B.S. degree the M. S. degree, in industrial engineering from Feng Chia University. And now he is a Ph. D. student in the Department of Industrial Engineering and Systems Management of Feng Chia University.