
Ying TANG and MengChu ZHOU

Abstract—Due to expensive, highly complex and time-consuming processes, semiconductor-manufacturing systems have been given a special attention. In our previous work (Tang et al., 2003), a heuristic algorithm to design the reconfigurable automated production system was proposed. However, machine breakdown and planned/unplanned maintenance were not considered. This paper extends that work and addresses the related design issues in reconfigurable back-end semiconductor manufacturing systems considering failures and maintenance effects. Queuing network approaches are used to model the variance of production lines. The work enhances the proposed virtual production line (VPL) design methodology in (Tang et al., 2003) where the deterministic machine processing time is used. A priority is also introduced to each idle machine and updated based on its past performance. The algorithms for Adaptive VPL configuration/reconfiguration are then proposed. The benefit of the proposed methodology and algorithms is illustrated through a simplified back-end semiconductor manufacturing example.

Index Terms—Reconfigurable Manufacturing Systems, Queuing Network Model, Heuristic Algorithm, Virtual Production Line

1. INTRODUCTION

A Reconfigurable Manufacturing System (RMS), identified as a priority technology by the National Research Council (National academy press, 1998), has become one of solutions to addressing manufacturing system flexibility, responsiveness, and integration issues. RMS includes reconfigurable machines and controllers, as well as methodologies for systematic design and diagnosis. Its production capacity can be rapidly adjusted to fluctuations in product demands, and functionality can be cost-effectively adapted to new products from a given product family (Koren et al., 1998; Mehrabi et al., 2000). Due to the mammoth needs of the semiconductor market, semiconductor-manufacturing companies have started to realize the opportunities for necessitating the use of RMS, which can be reconfigured and reprogrammed to provide manufacturers with a rapid response capability. From a production control point of view, the major part of flexibility of a manufacturing system is determined by its capability to make small batches of different part-mixes and rapid changes from one part-mix to another, allowing the use of alternative machines, and accommodating dynamic machine selection among the alternatives during production (Tang and Qiu, 2004). The responsiveness of a manufacturing system is typically characterized by its product production cycle. The smaller the product production cycle time of a manufacturing system, the more responsive the manufacturing system is. The concept of a virtual manufacturing cell has been investigated allowing the set of workstations on the shop floor to be dynamically configured (Irani et al., 1993; Simpson et al., 1982), where cell formation according to machine capacity, processing time and product demands as well as its functionality is restrictedly undertaken. To address reconfiguration issues, the Virtual Production Line (VPL) concept was first introduced in (Qiu and Wysk, 1999). The fundamental issues for the discrete-event driven VPL design was presented in (Tang et al., 2003). However, these methodologies were limited when applied to a system subject to machine failure and periodic maintenance. Actually, the variability due to such degraded behavior of manufacturing systems (e.g., unscheduled equipment downtime and maintenance) is a very important factor in effective production control (Tang and Zhou, 2001). In the recent academic literature, researchers use Petri nets (PN) to model such dynamics (Jeng and Xie, 2001; Kuo and Huang, 2000; Proth and Xie, 1994; Ramaswamy and Valavanis, 1994; Vinod and Altok, 1986; Zhou and Jeng, 1998). Queuing theories are also adopted for this analysis and design (Chen et al., 2000; Ciprut et al., 1999; Tham et al., 1999). Considering optimal production rate control in a failure prone manufacturing system, Hu and Xiang (1993) proved that under the hedging point policy the system is equivalent to an M/M/1 queue. Two Erlangian repairing models for a manufacturing system of identical unreliable machines were proposed by (Ching and Zhou, 1996). To solve the dynamic job shop scheduling problem, Tham et al. (1999) developed a minimum queuing delay dispatching algorithm that considers both the fairness of services to different jobs and the total queuing delays suffered by all the jobs. A number of maintenance dispatching policies were examined in (Mosley et al., 1998) and the results indicated that the choice of maintenance scheduling policy significantly affects the system performance under restrictive staffing levels. Despite these relatively modest activities, it is strongly emphasized that the variability of VPLs is a decisive topic to investigate because of the complexity of the semiconductor manufacturing procedures. The primary goal of this paper is to present a method for the adaptive VPL design, which considers the performance of the failure prone machines.
and periodic maintenance. The rest of the paper is organized as follows. Section 2 considers a VPL with periodic maintenance; Section 3 analyses failure prone machines in a VPL; Section 4 focuses on adaptive algorithms for the VPL design; Section 5 gives an example, followed by our conclusion and future research directions in Section 6.

2. VPL with Periodical Maintenance

A VPL is organized as a sequence of workstations, each with one or more machines of the same type to handle a process in a stage (Tang et al., 2003). When an order is placed, a VPL is configured based on the system status and the order information, and it is dedicated to making this product. Once the order is finished, its VPL can be retained for other orders of the same type; revised for a similar one; or dismissed so that the groups of equipment are reassigned to other VPLs. However, semiconductor-manufacturing equipment may be unreliable regardless of its age (Mosley et al., 1998). The variability due to unscheduled equipment downtime is harmful to the system throughput and thus increases the production cost. Therefore, most of machines need to be maintained regularly for a period of time to achieve high availability. In this section, machines’ periodic maintenance is considered first. The basic terminology and notation are introduced as follows:

**Basic terminology:**

<table>
<thead>
<tr>
<th>System status</th>
<th>The System status includes information of work-in-process (WIP) and machines’ availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order</td>
<td>An order is a document providing authorization for a production to be performed. It specifies the type and size of the production. An order that has not been processed by a system is referred to a new order.</td>
</tr>
<tr>
<td>Workstation</td>
<td>A workstation is a segment of a VPL where a process is performed by one or more machines. Machines in a workstation are functionally identical, but may have different speeds.</td>
</tr>
<tr>
<td>Desired speed</td>
<td>The desired speed of a VPL is the minimal speed that guarantees the order processed on this VPL can be finished on time. It is described as the number of magazines per minute.</td>
</tr>
<tr>
<td>Check time</td>
<td>The check time is discrete-event time for the system to check certain VPLs in a system.</td>
</tr>
<tr>
<td>Machine class</td>
<td>Machine class is a set of machine instances, which have the same or similar properties.</td>
</tr>
</tbody>
</table>

**Notation:**

- $t$: The $t^{th}$ time buck (minute) from the start
- $w$: VPL index.
- $i$: Machine index.
- $j$: Workstation index
- $k$: Machine class index.
- $s_w$: Number of workstations in the $w^{th}$ VPL.
- $c_{ijw}$: Capacity of the $i^{th}$ machine within the $j^{th}$ workstation of the $w^{th}$ VPL.
- $t_{ijw}$: The $i^{th}$ machine’s processing time of a magazine in the $j^{th}$ workstation of the $w^{th}$ VPL.
- $\beta_{ijw}$: Percentage of time that the $i^{th}$ machine is dedicated to the $j^{th}$ workstation of the $w^{th}$ VPL.
- $\nu_{ijw}$: Speed of the $i^{th}$ machine in $j^{th}$ workstation of the $w^{th}$ VPL.
- $\varepsilon_{ijw}$: Number of machines in the $j^{th}$ workstation of the $w^{th}$ VPL.
- $p_i$: Mean failure rate of the $i^{th}$ machine of the $k^{th}$ class.
- $P_{ijw}$: Allowed maximum failure rate of machines in the $j^{th}$ workstation of the $w^{th}$ VPL without getting immediate repair.
- $\lambda_{ijw}$: Maintenance/repair time for the $i^{th}$ machine in the $j^{th}$ workstation of the $w^{th}$ VPL.
- $r_{ijw}$: Repair rate of the $i^{th}$ machine in the $j^{th}$ workstation of the $w^{th}$ VPL.
- $\omega_{ijw}$: Maintenance periodicity of machines in the $j^{th}$ workstation of the $w^{th}$ VPL.
- $k_{ijw}$: Maintenance rate of the $i^{th}$ machine in the $j^{th}$ workstation of the $w^{th}$ VPL.
- $\gamma_{ijw}$: Priority of the $i^{th}$ machine in the $j^{th}$ workstation of the $w^{th}$ VPL.
- $N_{ijw}$: Size of an order (the number of magazines) processed in the $w^{th}$ VPL.
- $n_{aw}(t)$: Number of magazines finished by the $w^{th}$ VPL at time $t$.
- $M_k$: Machine pool of the $k^{th}$ class.
- $R_k$: Repair machine pool of the $k^{th}$ class.
- $I_k$: Idle machine pool of the $k^{th}$ class.
- $\phi_k$: Number of idle machines of the $k^{th}$ class.
- $a_{ijw}$: Average speed of the $j^{th}$ workstation in the $w^{th}$ VPL, which is measured as the number of magazine per minute.
- $\alpha_{ijw}$: Desired speed of the $w^{th}$ VPL.
- $\alpha_{ijw}$: The minimal real-time speed of workstations in the $w^{th}$ VPL.
- $\zeta_{ijw}(t)$: Slack time of the $w^{th}$ VPL at time $t$.
- $\phi_{ijw}(t)$: Tardiness of the $w^{th}$ VPL at time $t$. 
Failure solving the following linear equations (Zhou and idle, busy, and failure conditions of the machines may break down only when they are busy. Section, each machine with failure is modeled as a state imbalance and decrease the speed of workstations. In this exceptions are still in evitable, which in turn cause line prevent machines from failure, some malfunctions and

VPL. Certain workstations with higher frequencies will workload unbalance may occur among workstations in a in that VPL are fully utilized. During the operation, machines in other

changes with the machine’s utilization. The machine utilization is defined as follows:

In this paper, it is assumed that machines may have different maintenance time and the maintenance periodicity of workstations is independent. The speed and the mean maintenance rate of the \( i \)-th machine in the \( j \)-th workstation of the \( w \)-th VPL are calculated as:

Thus, to select a machine with higher speed and lower breakdown has adverse effects on system performance. In other words, if the system can perform with machine \( 2 \), then frequently spend resources to perform the operation associated with machine \( 1 \). Meanwhile, this paper assumes that machines \( 1 \) and \( 2 \) both can serve jobs in a certain stage \( j \), \( v_{ijw} \) is larger than \( v_{2jw} \), but the failure rate of machine \( 1 \) is higher than that of machine \( 2 \). Then, the system may frequently spend resources to perform the operation associated with machine \( 1 \) without success, and then perform with machine \( 2 \) successfully. This will certainly increase the system’s operational cost and decrease the throughput. In other words, if the system can select machine \( 2 \) prior to machine \( 1 \), the system throughput may be improved. Introducing priority for machines and integrating them into VPL design should enable the system to adapt for the best reconfiguration of VPLs.

Based on the past performance of machines, the values of \( \delta_{hk} \) assigned to the \( h \)-th machine in the \( M_k \) is decided as follows: \( \delta_{hk} = e \) if \( v_{ih}(1 - t_{ijw}) \) is the \( e \)-th greatest value in \( \{v_{ijw}(1 - t_{ijw}), \forall \epsilon \in I_k \} \) (assuming the \( h \)-th machine in \( M_k \) is idle and may serve as the \( i \)-th machine in the \( j \)-th workstation of

\* \( \psi_e(t) \): Earliness of the \( w \)-th VPL at time \( t \)
\* \( E \): Set of VPLs with earliness (\( \psi_e(t) > 0 \))
\* \( D \): Set of VPLs with tardiness (\( \phi_t(t) > 0 \))
\* \( ST(w) \): Start time of the \( w \)-th VPL
\* \( Due(w) \): Due time of an order processed in the \( w \)-th VPL
\* \( Date(w) \): Check time of the \( w \)-th VPL
\* \( Com(w) \): Completion time of an order processed in the \( w \)-th VPL
\* \( T_w \): Production time of an order processed in the \( w \)-th VPL (minutes)

Fig. 1. State transition model of the \( i \)-th machine in the \( j \)-th workstation of the \( w \)-th VPL.

Let \( \pi^{\text{idle}}_{ijw} \), \( \pi^{\text{busy}}_{ijw} \) and \( \pi^{\text{failure}}_{ijw} \) denote the probabilities of the idle, busy, and failure conditions of the \( j \)-th workstation of the \( w \)-th VPL, respectively.

Then, the steady-state probabilities are obtained by solving the following linear equations (Zhou and Venkatesh, 1998):

\[
\pi^{\text{idle}}_{ijw} = \left( \pi^{\text{idle}}_{ijw} + \pi^{\text{busy}}_{ijw} + \pi^{\text{failure}}_{ijw} \right) - \pi^{\text{failure}}_{ijw}
\]

4. Adaptive Reconfiguration for VPL

From the above analysis, it is found that machine breakdown has adverse effects on system performance. Thus, to select a machine with higher speed and lower failure rate is very important in VPL design. For example, assume that machines \( 1 \) and \( 2 \) both can serve jobs in a certain stage \( j \), \( v_{ijw} \) is larger than \( v_{2jw} \), but the failure rate of machine \( 1 \) is higher than that of machine \( 2 \). Then, the system may frequently spend resources to perform the operation associated with machine \( 1 \) without success, and then perform with machine \( 2 \) successfully. This will certainly increase the system’s operational cost and decrease the throughput. In other words, if the system can select machine \( 2 \) prior to machine \( 1 \), the system throughput may be improved. Introducing priority for machines and integrating them into VPL design should enable the system to adapt for the best reconfiguration of VPLs.

4. ADAPTIVE RECONFIGURATION FOR VPL

From the above analysis, it is found that machine breakdown has adverse effects on system performance. Thus, to select a machine with higher speed and lower failure rate is very important in VPL design. For example, assume that machines \( 1 \) and \( 2 \) both can serve jobs in a certain stage \( j \), \( v_{ijw} \) is larger than \( v_{2jw} \), but the failure rate of machine \( 1 \) is higher than that of machine \( 2 \). Then, the system may frequently spend resources to perform the operation associated with machine \( 1 \) without success, and then perform with machine \( 2 \) successfully. This will certainly increase the system’s operational cost and decrease the throughput. In other words, if the system can select machine \( 2 \) prior to machine \( 1 \), the system throughput may be improved. Introducing priority for machines and integrating them into VPL design should enable the system to adapt for the best reconfiguration of VPLs.

Based on the past performance of machines, the values of \( \delta_{hk} \) assigned to the \( h \)-th machine in the \( M_k \) is decided as follows: \( \delta_{hk} = e \) if \( v_{ih}(1 - t_{ijw}) \) is the \( e \)-th greatest value in \( \{v_{ijw}(1 - t_{ijw}), \forall \epsilon \in I_k \} \) (assuming the \( h \)-th machine in \( M_k \) is idle and may serve as the \( i \)-th machine in the \( j \)-th workstation of
the \(w^{th}\) VPL. It is clear that the priority of a machine may change depending on the job it serves.

When a new job comes or other events (e.g., breakdown) happen, the system will update the idle machine pools. If the mean failure rate of a machine is higher than a pre-set number \(P_{w}\) (assuming this machine will serve jobs in the \(j^{th}\) workstation of the \(w^{th}\) VPL), remove this machine from its corresponding idle machine pool and add it into \(R_{k}\), where machines receive immediate repairs.

Initially, all machines work in a good condition. During the operations, the mean failure rate of a machine is dynamically updated and assumed known in this paper. Thus, to configure a VPL for a new order, the system will sort machines in \(I_{k}\) with a decreasing order of their priority values. Machines will then be selected from \(I_{k}\) according to their priority values from the highest to the lowest. The algorithm that considers the variability due to machine breakdown and periodic maintenance is presented as follows.

**Algorithm 1 (Adaptive reconfiguration):**

(1) Repeat the following steps for all VPLs:

a. Select the \(w^{th}\) VPL in the set \(X\), remove it from set \(X\), and adjust its \(a_{w}\) as follows:

\[
Min \psi_{w} \quad j = 1 \text{ to } s_{w} \quad (4.4)
\]

Subject to:

\[
a_{w} = \frac{N_{w} - n_{w}(t) \text{ Due}(w) - \text{Date}(w)}{a_{w} \geq a_{j} \quad j = 1 \text{ to } s_{w} \quad (4.7)}
\]

b. If Step a) succeeds, update \(\psi_{w}\); Exclude the redundant machines from this VPL and add them into idle machine pool \(I_{k}\);

c. If \(\psi_{w} = 0\), remove it from the set \(E\);

d. Return to Step (2)

(2) If there is at least one machine available at each stage of the \((u+1)^{th}\) order and \( \sum_{w=1}^{u} (\psi_{w} - \phi_{w}) - \phi_{w+1} > 0 \)

configure this new VPL with a slow rate \(a_{w+1}\), put this new line into set \(D\) and go to Step (10)

(3) Reject this new order, consider it later and exit.

(10) Calculate the production time for the \((u+1)^{th}\) order \(T_{w+1}\) and exit:

\[
T_{w+1} = (N_{w+1} + s - 1) / a_{w+1} \quad (4.8)
\]

Degraded behaviors (e.g., tardiness, malfunctions and exceptions) of semiconductor manufacturing systems are not negligible in practice. Thus, how to reallocate resources to minimize the negative impact of these behaviors is extremely important. Taking this into consideration, an adjustment algorithm is presented based on our previous work (Tang et al., 2003):

**Algorithm 2 (Dynamical adjustment)**

The system adjusts VPLs when tardiness, malfunctions or exceptions happen:

(1) Repeat the following steps for all VPLs:

a) Calculate \(\zeta(t)\), \(\phi_{w}(t)\) and \(\psi_{w}(t)\);

b) If \(\psi_{w}(t) > 0\), update \(E = E \cup \{w\}\); otherwise if \(\phi_{w}(t) > 0\), update \(D = D \cup \{w\}\);

(2) If \(D \neq \emptyset\), do:

\[
(\sum_{j=1}^{s_{w+1}} \nu_{j}^{(w+1)})(1 - t_{j}^{(w+1)}) \beta_{j}^{(w+1)} a_{w+1}^{R} \quad (4.3)
\]

(5) Re-find the minimal speed of the \((u+1)^{th}\) VPL according to \(a_{w+1} = Min \{a_{j}^{(w+1)} \quad j = 1, 2, ..., s_{w+1}\} \quad (4.1)

(6) If \(a_{w+1} \geq a_{w+1}\), keep this new VPL running with speed \(a_{w+1}\); Then, choose the minimum number of machines \(\phi_{j}^{(w+1)}\) using Equation 3.3 such that \(a_{j}^{(w+1)} \geq a_{w+1}\) and they are the first \(\epsilon_{j}^{(w+1)}\) machines in \(I_{k}\), then update \(\phi_{h} = \phi_{h} - \epsilon_{j}^{(w+1)}\), and calculate \(\psi_{j}^{(w+1)}(t)\). If \(\psi_{j}^{(w+1)}(t) > 0\), add this new VPL into set \(E\) and go to Step (10)

(7) If \(X \neq \emptyset\), do:

\[
(\sum_{j=1}^{s_{w+1}} \nu_{j}^{(w+1)})(1 - t_{j}^{(w+1)}) \beta_{j}^{(w+1)} a_{w+1}^{R} \quad (4.3)
\]

(1). Repeat the following steps for all VPLs:

i. Calculate the maximum speed of the \((u+1)^{th}\) VPL:

\[
a_{w+1} = \frac{N_{w+1} + s_{w+1} - 1}{\text{Due}(w+1) - \text{ST}(w+1)} \quad (4.1)
\]

ii. Update idle machine pools: if the failure rate of an idle machine in \(I_{k}\) is larger than \(P_{w+1}\), move it into its corresponding repair machine pool \(R_{k}\); assign priorities to the remaining machines in \(I_{k}\) and order them with a decreasing order of their priority values

iii. Use the system status and calculate the maximum speed of each workstation in the \((u+1)^{th}\) VPL (assuming the arrival rate of jobs is larger than the speed of any machine in a VPL):

\[
a_{j}^{(w+1)} = \sum_{j=1}^{s_{w+1}} \nu_{j}^{(w+1)}(1 - t_{j}^{(w+1)}) \beta_{j}^{(w+1)} a_{w+1}^{R} \quad (4.2)
\]

(3) i. Calculate the maximum speed of workstations with failures and maintenance:

\[
a_{j}^{(w+1)} = \sum_{j=1}^{s_{w+1}} \nu_{j}^{(w+1)}(1 - t_{j}^{(w+1)}) \beta_{j}^{(w+1)} a_{w+1}^{R} \quad (4.3)
\]

(4) i. Calculate the maximum speed of workstations with failures and maintenance: Note that \(a_{w+1}\) is arrival rates of machines in such workstations.
a) If $E \neq \emptyset$, for each VPL in set $E$, Reallocation resources through Equations 4.4-4.7; and update $\phi_k$ and $\psi_n(t)$. If $\psi_n(t) = 0$, remove it from set $E$.

b) Arrange VPLs in $D$ with the ascending order of earliest due dates. For each VPL in $D$, do:

I) For $j = 1$ to $s_w$ (assuming the $j^{th}$ workstation in the $w^{th}$ VPL requires the $k^{th}$ class of machines)

i) If $\phi_k \neq 0$, update $I_k$ using the same way as Step 3 in Algorithm 1, then, assign the first $\theta_j$ ($\theta_j \leq \phi_k$) machines in $I_k$ to the $j^{th}$ workstation to increase its speed.

for $(i = e_{ijw}$ to $e_{ijw} + \theta_j)$ do:

if ($a_{ijw} = 0$) $a_{ijw} = a_{ijw} + \nu_{ijw}(1-t_{ijw})\beta_{ijw}$
else $a_{ijw} = a_{ijw} + \nu_{ijw}(1-t_{ijw})\beta_{ijw}\tau_{ijw}$

ii) Otherwise, check other VPLs in the system (i.e., $x^{th}$ VPL and the $v^{th}$ workstation in the $x^{th}$ VPL requires the $k^{th}$ class of machines). If there is machine $i$ such that its utilization is less than 0.5, this machine can be shared by $w^{th}$ VPL.

for $(x = 1$ to $w (x \neq w))$

for $(i = 1$ to $u (x \neq w)))$

if ($a_{ivx} = 0$)

$\alpha_{ivx} = \alpha_{ivx} + \nu_{ivx}(1-t_{ivx})\beta_{ivx}$
else

$\alpha_{ivx} = \alpha_{ivx} + \nu_{ivx}(1-t_{ivx})\beta_{ivx}\tau_{ivx}$

II) Calculate $q_{iv} = \min \{a_{ivx}, j = 1, 2, ..., s_w\}$

III) If $|q_{ivx} - \alpha_{ivx}| \leq 10^{-2}$, remove this VPL from $D$

IV) Otherwise, update its tardiness $\phi_i$. If $\phi_i > 0$, this order will be delayed $\phi_i$ time to finish based on the present forecast.

5. AN ILLUSTRATIVE EXAMPLE

To better understand the above methods and algorithms, a practical implementation of the concept of VPLs is studied using a simplified back-end semiconductor line. In the exemplified back-end semiconductor manufacturing system, an order is sequentially made through some or all the following processes: Saw, 2/OP, Tape attach, Die attach, Plasma cleaning, Wire bond, 3/OP, Pre mold bake, Plasma cleaning, Glob topping, Mark, Plasma cleaning, Solder ball, Flux clean, Signulation, Final visual and Pack. A simplified process flow is given in Figure 2.

In the simulation run, three cases are analyzed and compared to illustrate the advantage of the proposed methodology.

1) In the baseline case, the system successively processes orders according to their order dates. In other words, the line is predefined to manufacture certain type of products at each time.

2) In the second case, whenever an order comes, the methodology presented in (Tang et al., 2003) is applied to strategically configure a VPL on the fly and monitor all VPLs in the system.

3) In the third one, the adaptive algorithms are used to configure and control VPLs.

For these three cases, this paper considers the following scenario: while the system is already running, two new orders are placed sequentially, both following the process flow in Figure 2. For the simplicity, it is assumed that except in “Saw” and “Wire bond” workstations, machines are reliable. Workstations are equipped with identical machines with different failure rates and machines are fully dedicated to its corresponding workstation. Before configuring a VPL for the first new order, there are ten machines unreliable in $I_5$, which can process jobs in “Wire bond” workstation. Their failure rates are $\{0.004, 0.008, 0.01, 0.01, 0.008, 0.008, 0.008, 0.004, 0.04, 0.04\}$. Based on field engineers’ experience, $P_{ijw}$ depends on $a_{ijw}$. In this example, the threshold for machines in $I_5$ is set as 0.004 (1/min), meaning that the machine with the failure rate higher than 0.004 gets immediate repair. On the fifth day of the first new order being processed, three machines in the “Saw” workstation break down. The next day after that, eighteen idle machines are added into $I_k$ since an old order is finished. The input data for orders and workstations is shown in Tables 1 and 2, respectively.

<table>
<thead>
<tr>
<th>Order</th>
<th>EST</th>
<th>Due</th>
<th>$N_{w_{mag.}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st order</td>
<td>1/1/00</td>
<td>1/16/00</td>
<td>10</td>
</tr>
<tr>
<td>2nd order</td>
<td>1/3/00</td>
<td>1/13/00</td>
<td>10</td>
</tr>
</tbody>
</table>

Fig.2. A simplified process flow.

Table 1. The input data for the two new orders
To show the difference among these cases, the system throughput $g$ is introduced, which is the total number of all completed magazines divided by the difference between maximum completion time and minimum start time of all the orders completed.

\[
g = \frac{\sum_{w=1}^{n} N_w}{\text{Max}\{\text{Com}(w)\} - \text{Min}\{\text{ST}(w)\}} \tag{5.1}
\]

Fig. 3. VPL speeds for different cases when the system processes the two orders.

Table 3. The computational results

<table>
<thead>
<tr>
<th># of days</th>
<th>Baseline case</th>
<th>2nd case (algorithms in (Tang et al., 2003))</th>
<th>3rd case (adaptive algorithms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1st order</td>
<td>2nd order</td>
<td>1st order</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>4.742</td>
<td>4.742</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>4.742</td>
<td>4.659</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>4.742</td>
<td>4.659</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>4.639</td>
<td>0.123</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>4.639</td>
<td>0.123</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>4.429</td>
<td>0.719</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>4.429</td>
<td>0.719</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>4.429</td>
<td>4.639</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>4.429</td>
<td>4.639</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>4.429</td>
<td>4.639</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>4.429</td>
<td>4.639</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>4.429</td>
<td>4.639</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>4.429</td>
<td>4.639</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>4.429</td>
<td>4.639</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>4.429</td>
<td>4.639</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>4.429</td>
<td>4.639</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>4.429</td>
<td>4.639</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>5.358</td>
<td>4.639</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>5.358</td>
<td>4.639</td>
</tr>
<tr>
<td>$g$</td>
<td>5780</td>
<td>7333</td>
<td>7333</td>
</tr>
<tr>
<td>$f$</td>
<td>90.2% (the 5th workstation)</td>
<td>95.4% (the 5th workstation)</td>
<td>99.4% (the 5th workstation)</td>
</tr>
</tbody>
</table>

Our computation results are then presented in Figure 3 and Table 3. The changes of VPL speeds, the system throughput and the utilization of Wire Bond workstation are compared as the two orders go through the three cases. It is clear to see that in the last two cases, the system can concurrently handle multiple orders and adjust the speed of these VPLs according to production line condition and system status. Thus, the system’s throughput increases by 26.7%. In the third case, due to the priority introduced, the system can monitor machines’ performance and immediately repair machines with a high failure rate (i.e., $p_k > 0.004$), instead of keeping such machines running until periodic maintenance. Thus, the second order was finished in advance by one day, and workstation utilization increases by 10.2% and 4.2% compared to those in the first and the second cases, respectively.
6. CONCLUSION

In the competitive and risky environment of semiconductor manufacturing, production planning and control is quite complicated and difficult. In order to meet customers’ needs and adapt to fluctuating market demands, the deployment of RMS has become critical in enhancing the system’s flexibility and responsiveness. Moreover, degraded behaviors, such as failures and maintenance, of a semiconductor manufacturing system are not negligible in practice. This paper extends our previous work in reconfigurable back-end semiconductor manufacturing system design by considering the variability due to such degraded behaviors. A queuing network model is used to analyze a workstation’s throughput due to machine downtime, unplanned and planned maintenance. In light of the adverse effects of unexpected machine breakdowns on system performance, a priority value is introduced to each idle machine, allowing the system to dynamically select the most reliable machines. The algorithms are then developed for adaptive VPL configuration and reconfiguration. From the example, the approach is found to be effective in increasing system throughput and machine utilization.

The research can be extended in several directions. For instance, more factory data is needed to further test the proposed methodology in the future. Different modeling methods for failure prone machines and maintenance scheduling in VPL operations to meet different productivity requirements are worthwhile investigating.

REFERENCES


design, Networking and communication, and System-on-Chip design. She is a member of the IEEE Systems, Man and Cybernetics and Sigma Xi.

MengChu Zhou received his B.S. degree from Nanjing University of Science and Technology, Nanjing, China in 1983, M.S. degree from Beijing Institute of Technology, Beijing, China in 1986, and Ph. D. degree in Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, NY in 1990. He joined New Jersey Institute of Technology (NJIT), Newark, NJ in 1990, and is currently a Professor of Electrical and Computer Engineering, the Director of Discrete-Event Systems Laboratory, and the Director of the M.S. in Computer Engineering Program.


He was invited to lecture in Australia, Canada, China, France, Germany, Hong Kong, Italy, Japan, Korea, Mexico, Taiwan, and US. He served as Associate Editor of IEEE Transactions on Robotics and Automation from 1997 to 2000 and currently Managing Editor of IEEE Transactions on Systems, Man and Cybernetics: Part C, Associate Editor of IEEE Transactions on Automation Science and Engineering, and Editor-in-Chief of International Journal of Intelligent Control and Systems. He was General Co-Chair of 2003 and 2006 IEEE International Conference on System, Man and Cybernetics, 2004 and 2006 IEEE Int. Conf. on Networking, Sensors and Control. He was Program Chair of 1998 IEEE International Conference on System, Man and Cybernetics (SMC) and 1997 IEEE International Conference on Emerging Technologies and Factory Automation, and Guest Editors for IEEE Transactions on Industrial Electronics, and IEEE Transactions on Semiconductor Manufacturing. He organized and chaired over 80 technical sessions and served on program committees for many conferences.

Dr. Zhou has led or participated in twenty-six research and education projects with total budget over $10M, funded by National Science Foundation, Department of Defense, Engineering Foundation, New Jersey Science and Technology Commission, and industry. He was the recipient of NSF’s Research Initiation Award, CIM University-LEAD Award by Society of Manufacturing Engineers, Perlis Research Award by NJIT, Humboldt Research Award for US Senior Scientists, Leadership Award and Academic Achievement Award by Chinese Association for Science and Technology-USA, Asian American Achievement Award by Asian American Heritage Council of New Jersey, and Outstanding Contribution Award from IEEE System, Man and Cybernetics (SMC) Society. He is named Distinguished Lecturer of IEEE SMC Society for 2005-2006. He was the founding chair of Discrete Event Systems Technical Committee of IEEE SMC Society, and Co-Chair (founding) of Semiconductor Factory Automation Technical Committee of IEEE Robotics and Automation Society. He is Fellow of IEEE and a life member of Chinese Association for Science and Technology-USA.