Computers and Chemical Engineering 52 (2013) 112–121

Contents lists available at SciVerse ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

Generation and verification of optimal dispatching policies for multi-product multi-tool semiconductor manufacturing processes

Chin-Feng Chen[a], Kun-Jia Wu[a], Chuei-Tin Chang[a,*], David Shan-Hill Wong[b], Shi-Shang Jang[b]

[a] Department of Chemical Engineering, National Cheng Kung University, Tainan 70101, Taiwan, ROC
[b] Department of Chemical Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan, ROC

ARTICLE INFO

Article history:
Received 9 September 2011
Received in revised form 23 August 2012
Accepted 17 December 2012
Available online xxx

Keywords:
EWMA
Dispatching policy
Queuing system
Semiconductor manufacturing
Numerical simulation

ABSTRACT

Semiconductor manufacturing is one of the fastest-growing industries today. As the recent requirements for feature sizes and wafer sizes change rapidly, it becomes imperative to configure increasingly intricate control schemes to maintain product quality and tool utilization rate. For this purpose, it is assumed in this study that a semiconductor production environment can be viewed as multiple queues operated in parallel and, also, the EWMA controllers can be implemented independently to adjust the process recipes of different products in each queuing system. Based on these assumptions, a MINLP model is formulated to determine the optimal dispatching policies. Systematic numerical simulation procedure is also devised to confirm the validity of the dispatching model. Since accurate estimates of the model parameters may not always be available, the effects of model mismatch have been analyzed and the proper range of controller tuning parameter is recommended to achieve an acceptable level of process capability.

1. Introduction

The typical production environment in a semiconductor plant resembles an automated assembly line in which many similar products with different specifications are manufactured by a step-by-step overall process. Each step is a complicated physiochemical batch process that carried out by a number of similar tools working in parallel. In this study, this environment is viewed as a system with multiple queues (Cassandras & Lafortune, 1999) operated in parallel and more than one product may be processed in each queue.

The run-by-run (RbR) control strategy has already been widely implemented in practice in the semiconductor manufacturing industry for the purpose of ensuring product quality (Moyne, 2001), while the exponentially weighed-moving-average (EWMA) algorithm is by far the most popular RbR control scheme (Patel & Jenkins, 2000; Sachs, Hu, & Ingolfsson, 1995; Smith & Boning, 1997; Tseng, Yeh, Tsung, & Chan, 2003). It is important to note that, although the single-product RbR strategy can be readily applied to a “thread,” i.e., a specific combination of product and tool (Firth, Campbell, Toprac, & Edgar, 2006), the control performance may be substantially degraded in cases of seldom-utilized threads. Zheng, Lin, Wong, Jang, and Hui (2006) showed that, even when the actual root cause of external disturbances is originated from the tool, a single tool-based EWMA controller may still be unstable if different product-related uncertainties are present. Various approaches has been proposed to estimate contributions to the biases due to tool and product individually using historical measurements and then recombine them to determine recipe adjustments for the future runs (Firth et al., 2006; Pasadyn & Edgar, 2005). Based on the aforementioned studies, it is thus assumed in the present work that (1) the statistical parameters of biases due to tools and unprocessed products can be estimated in advance and (2) a distinct EWMA controller can be implemented in each queuing system to adjust the process recipe of every one of the products independently from run to run.

“Dispatch” is a task often required in assigning either workers or vehicles to satisfy customer needs. In particular, this task is performed in response to the calls for taxicabs, couriers, emergency services in everyday life, as well as the tools in semiconductor manufacturing processes. The lot-priority index based dispatch rules are often used in the latter case to rank various queues dynamically. To apply these rules, it is necessary to first obtain various lot and system attributes, such as: arrival time, due date, processing time(s), queue length(s), work content, and setup time (Bhaskaran & Pinedo, 1992; Blackstone, Phillips, & Hogg, 1982; Kutanoglu & Sabuncuoglu, 1999; Raghu & Rajendran, 1993; Rajendran & Holthaus, 1999). The commonly used FCFS (first come first serve) principle is in fact a simple dispatch rule based on arrival time. On the other hand, more sophisticated production planning and scheduling strategies have also been developed. Dynamic dispatch rules have been investigated by Pierce and Yurtsever (1999). Dabbas and his coworkers

* Corresponding author. Tel.: +886 6 2757575x62663; fax: +886 6 2344496.
E-mail address: ctchang@mail.ncku.edu.tw (C.-T. Chang).

0098-1354/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compchemeng.2012.12.009

proposed a modified dispatching approach to incorporate multiple dispatching criteria into a single rule so that multiple performance measures can be optimized (Dabbas, Chen, & Fowler, 2001; Dabbas & Fowler, 2003). Finally, Choi and Reveliotis (2005) studied the allocation problem concerning workstation processing and stepping capacity in a capacitated reentrant line.

Traditionally only the throughput related measures, e.g., the customer demands and the machine capacities, are considered to stipulate the dispatching policy (in order to fully utilize the processing capacity of the plant). To satisfy these criteria, it is often necessary to evenly distribute the production loads to all available machines. On the other hand, as we have also pointed out, the quality of a specific type of products being produced in a particular tool can be significantly affected by their processing frequency if the corresponding thread is under RbR control. Alternative dispatching policies should therefore be considered for optimizing the overall quality of each type of products in addition to operation efficiency. However, the constrained optimization problems of how to maximize product quality without creating backlog in production and how to maximize yield but maintain quality standards have never been studied in details in the past. A mixed integer nonlinear programming model is thus constructed in this work to solve these problems and to address the related issues. In this model the statistical parameters of the biases attributable to different tools and products are assumed to be available. For computation simplicity, the quality statistics of finished products are predicted according to a simplifying assumption, namely, wafers of the same type are processed intermittently and regularly at a *constant* rate. The validity of resulting optimal dispatching policy is then confirmed with a systematic numerical simulation procedure, which is also developed in this study.

The remainder of this paper is organized as follows. The detailed formulation of the proposed mathematical programming model is first given in Section 2. Solutions of an example problem are analyzed in Section 3. Section 4 describes the development of a numerical procedure devised to simulate the multi-product multi-tool plant under threaded RbR control and optimal dispatch. Validations of the optimization results obtained in various case studies in Section 3 are then presented in Section 5. Since accurate estimates of the model parameters may not always be available, the effects of model mismatch are discussed in Section 6 and the proper range of EWMA controller tuning parameter is also recommended for achieving an acceptable process capability. Finally, conclusions and also some comments on future works will be given at the end of it in Section 7.

2. Mathematical programming model

2.1. Optimal dispatch problem

Fig. 1 illustrates a production facility with P products and U tools working in parallel. The operation of each tool is viewed here as a M/M/1 queue and more than one product may be processed in each queue. The dispatching problem at each stage in this environment is concerned with the task of determining the percentages of each type of unprocessed products to be delivered to the available tools. Let us use f_u^p to represent the probability of assigning product p to tool u. Thus, the following constraints should be imposed

$$\sum_{u=1}^{U} f_u^p = 1, \quad 0 \le f_u^p \le 1 \tag{1}$$

where $p = 1, 2, \ldots, P$.

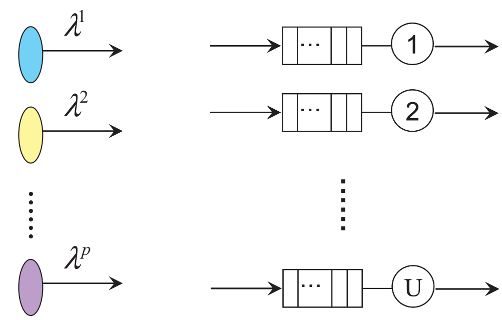

Fig. 1. Parallel M/M/1 queuing systems with P products and U tools.

If λ^p represents the mean arrival rate of product p, the utilization rate of tool u can be expressed as:

$$\rho_u = \sum_{p=1}^{p} f_u^p \lambda^p \tau_u^p, \quad 0 \le \rho_u < 1 \tag{2}$$

where $u = 1, 2, \ldots, U$, and τ_u^p denotes the mean processing time required for product p in tool u. Let us further assume that the mean and variance of characteristic quality variable of the *finished* product p produced in tool u (denoted as y_u^p) can be *estimated* in advance to be $\tilde{\mu}[y_u^p]$ and $\tilde{\sigma}^2[y_u^p]$ respectively. The overall mean and variance for product p collected from all tools in the multi-product multi-tool environment, i.e., $\tilde{\mu}[y^p]$ and $\tilde{\sigma}^2[y^p]$, can then be determined according to the following formulas:

$$\tilde{\mu}[y^p] = \sum_{u=1}^{U} f_u^p \tilde{\mu}[y_u^p] \tag{3}$$

$$\tilde{\sigma}^2[y^p] = \sum_{u=1}^{U} f_u^p \{(\tilde{\mu}[y_u^p])^2 + \tilde{\sigma}^2[y_u^p]\} - \left\{ \sum_{u=1}^{U} f_u^p \tilde{\mu}[y_u^p] \right\}^2 \tag{4}$$

The process capability index of product p can thus be expressed accordingly as (Del Castillo, 2002)

$$C_{pk}[y^p] = \left(\frac{USL^p - \tilde{\mu}[y^p]}{3\tilde{\sigma}[y^p]}, \frac{\tilde{\mu}[y^p] - LSL^p}{3\tilde{\sigma}[y^p]} \right)_{\min} \tag{5}$$

As mentioned previously, one of the primary objectives of dispatching operation is to maximize yield or efficiency. The latter objective can be achieved by minimizing the overall utilization rate of a manufacturing system under steady state condition. However, we must also ensure that (1) the process capability index of each product is kept above a lower bound $(C_{pk})_{\min}^p$ and (2) the utilization rate of every tool is maintained below an upper limit $\rho_{u,\max}$. This constrained optimization problem can be written as:

$$\min_{f_u^p} \sum_{u=1}^{U} \rho_u \tag{6}$$

which is subject to the constraints in (1) and, also, the following inequality constraints

$$C_{pk}[y^p] \ge (C_{pk})_{\min}^p \quad 0 \le \rho_u \le \rho_{u,\max} \tag{7}$$

Alternatively, one can also try to maximize the overall quality performance under the constraints given in (1) and (7). The objective of this alternative optimization problem can be written as

$$\max_{f_u^p} \sum_{p=1}^{P} C_{pk}[y^p] \tag{8}$$

Notice that the above two alternative formulations of the optimal dispatch problem are obviously incomplete. In the mathematical

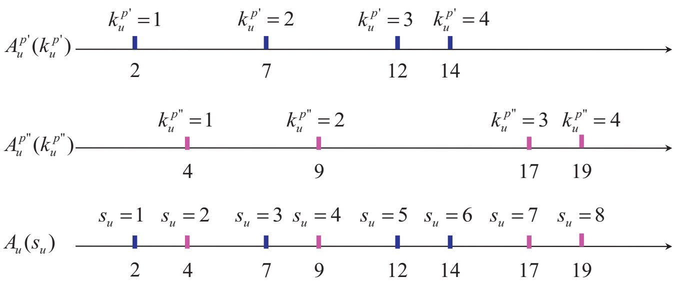

Fig. 2. Relation between thread count k_u^p and queue count s_u.

programming models, it is still necessary to describe the effects of dispatching parameters f_u^p on the quality performance $\tilde{\mu}[y_u^p]$ and $\tilde{\sigma}^2[y_u^p]$ of each thread and also the overall quality performance $\tilde{\mu}[y^p]$ and $\tilde{\sigma}^2[y^p]$.

2.2. Multi-product multi-tool production model

The specific combination of product p on tool u, i.e., thread (p, u), is the basis of RbR control in the present applications. Fig. 2 illustrates the relation between the time sequence, thread sequence and the queue sequence in a multi-product parallel tool production environment. Consider a tool u, product p' arrives at time instants 2, 7, 12, 14; corresponding to thread counts $k_u^{p'} = 1, 2, 3, 4$; and product p'' arrives at time instants 4, 9, 19, 19; corresponding to thread counts $k_u^{p''} = 1, 2, 3, 4$. If there are only these two products in this period, the production queue counts are: $s_u = 1(k_u^{p'} = 1)$, $s_u = 2(k_u^{p''} = 1), \ldots, s_u = 8(k_u^{p''} = 4)$.

Let us assume that the input–output relation of the production thread (p, u) can be written as follows:

$$y_u^p(k_u^p) = \alpha_u^p + \beta_u^p x_u^p(k_u^p) + \eta_u^p(k_u^p) \tag{9}$$

where $p = 1, 2, \ldots, P$ is the product index; $u = 1, 2, \ldots, U$ is the tool index; $k_u^p = 1, 2, \ldots, N_u^p$ is the run number (or thread count) of product p on tool u; $y_u^p(k_u^p)$ and $x_u^p(k_u^p)$ denote the input and output at run k_u^p; α_u^p and β_u^p respectively represent the slope and intercept of linear input–output model; $\eta_u^p(k_u^p)$ is the corresponding process noise. Let us further assume that the process noise can be classified into a product related noise and a tool related noise, i.e.

$$\eta_u^p(k_u^p) = \eta^p(k_u^p) + \eta_u(\Phi(k_u^p)) \tag{10}$$

The product related noise $\eta^p(k_u^p)$ can be expressed as

$$\eta^p(k_u^p) = \mu^p + \varepsilon^p(k_u^p) \tag{11}$$

where μ^p is a constant bias associated with product p and $\varepsilon^p(k_u^p) \in \mathcal{N}(0, \sigma^2[\varepsilon^p])$ is a random noise. Since several different types of products may be produced with the same tool, an assignment function $\Phi(k_u^p)$ is introduced in this work to characterize the one-to-one mapping relation from thread count k_u^p to queue count s_u:

$$s_u = \Phi(k_u^p) \tag{12}$$

where $s_u = 1, 2, 3, \ldots, NR_u$ and $NR_u = \sum_{p=1}^{P} N_u^p$. This assignment function should satisfy the following constraint:

$$\Phi(k_u^{p'}) < \Phi(k_u^{p''}) \quad \text{iff} \quad A_u^{p'}(k_u^{p'}) < A_u^{p''}(k_u^{p''}) \quad \text{and} \quad p' \neq p'' \tag{13}$$

where $A_u^{p'}(k_u^{p'})$ denotes the arrival time of run $k_u^{p'}$ of product p' on tool u, while $A_u^{p''}(k_u^{p''})$ denotes the arrival time of run $k_u^{p''}$ of product p'' on tool u.

In this work, it is assumed that the tool noise is an IMA(1,1) process (Box, Jenkins, & Reinsel, 1994) in terms of the queue count s_u, which can be expressed as

$$\eta_u(s_u) = \eta_u(s_u - 1) - \theta_u \varepsilon_u(s_u - 1) + \mu_u + \varepsilon_u(s_u) \tag{14}$$

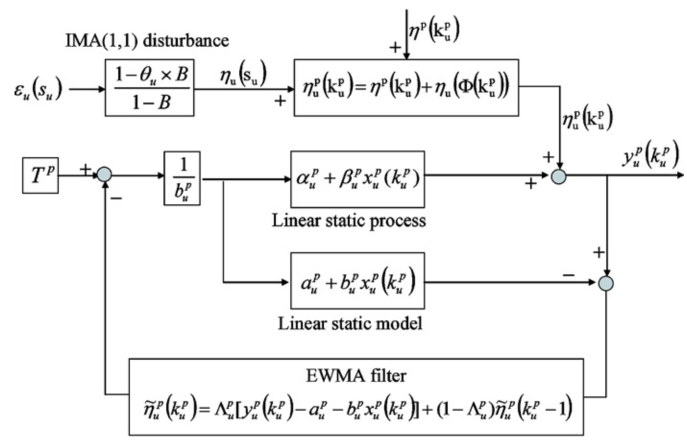

Fig. 3. Threaded EWMA RbR control of product p with random product noise on a static linear tool u with IMA(1,1) noise.

where θ_u is the time correlation of noise due to tool u; and $\varepsilon_u(s_u) \in \mathcal{N}(0, \sigma^2[\varepsilon_u])$ is a zero mean random noise and μ_u is a constant offset due to tool u

2.3. Threaded RbR control strategy

On the basis of the above formulation, the output prediction model can be expressed as (Ma, Chang, Jang, & Wong, 2009):

$$\tilde{y}_u^p(k_u^p) = a_u^p + b_u^p x_u^p(k_u^p) \tag{15}$$

where $\tilde{y}_u^p(k_u^p)$ is the predicted output of run k_u^p for product p on tool u; a_u^p and b_u^p are model estimates of the parameters α_u^p and β_u^p respectively. The offset term for the thread can be estimated with the EWMA algorithm, i.e.

$$\tilde{\eta}_u^p(k_u^p) = \Lambda_u^p[y_u^p(k_u^p) - a_u^p - b_u^p x_u^p(k_u^p)] + (1 - \Lambda_u^p)\tilde{\eta}_u^p(k_u^p - 1) \tag{16}$$

in which Λ_u^p is the adjustable filter parameter of the EWMA controller for thread (p, u). The process input can then be computed according to the following equation:

$$x_u^p(k_u^p + 1) = \frac{T^p - \tilde{\eta}_u^p(k_u^p) - a_u^p}{b_u^p} \tag{17}$$

where T^p is the target response of product p. To facilitate clear illustration of the control scheme, the closed-loop block diagram of thread (p, u) is given Fig. 3.

2.4. Approximate thread disturbance and thread performance under RbR control

The input–output model of thread (p, u) can be written in a different form as:

$$\begin{aligned} y_u^p(k_u^p) &= \alpha_u^p + \beta_u^p x_u^p(k_u^p) + \mu^p + \varepsilon^p(k_u^p) + \mu^u + \eta_u(s_u = \Phi(k_u^p)) \\ &\equiv \breve{\alpha}_u^p + \beta_u^p x_u^p(k_u^p) + \varepsilon^p(k_u^p) + \breve{\eta}_u^p(k_u^p) \end{aligned} \tag{18}$$

where $\breve{\alpha}_u^p = \alpha_u^p + \mu^p + \mu^u$ and $\breve{\eta}_u^p(k_u^p) = \eta_u(s_u = \Phi(k_u^p))$. It should be noted that even though $\eta_u(s_u)$ is an IMA(1,1) series in terms of the tool index s_u, the noise sequence $\breve{\eta}_u^p(k_u^p)$ is in general not an IMA(1,1) series in terms of thread index k_u^p. However, according to Box, Jenkins, and Reinsel (1994), if an IMA(1,1) process with time correlation parameter θ_1 and variance σ_1^2 is sampled at a *constant* sampling interval h, the resulting observations should form an IMA(1,1) process with time correlation parameter θ_h and variance σ_h^2:

$$\frac{h(1-\theta_1)^2}{\theta_1} = \frac{(1-\theta_h)^2}{\theta_h} \tag{19}$$

$$\sigma_h^2 = \frac{\theta_1}{\theta_h}\sigma_1^2 \quad (20)$$

Since the mean arrival rate of product p at tool u can be expressed as $f_u^p\lambda^p$, the following formula is used in this study to compute the average number of arrivals between two consecutive arrivals of product p on tool u:

$$\frac{1}{h_u^p} = \frac{f_u^p\lambda^p}{\sum_{p=1}^{P} f_u^p\lambda^p} \quad (21)$$

Thus the noise series $\breve{\eta}_u^p(k_u^p)$ can be approximated by:

$$\breve{\eta}_u^p(k_u^p) \approx \breve{\eta}_u^p(k_u^p-1) - \theta_u^p\varepsilon_u^p(k_u^p-1) + \varepsilon_u^p(k_u^p) \quad (22)$$

Notice that, in this equation, the time correlation parameter of thread (p, u), i.e., θ_u^p, can be determined according to Eq. (19):

$$\frac{h_u^p(1-\theta_u)^2}{\theta_u} = \frac{(1-\theta_u^p)^2}{\theta_u^p} \quad (23)$$

Also, $\varepsilon_u^p(k_u^p) \in \mathcal{N}(0, \sigma^2[\varepsilon_u^p])$ is a zero-mean normally distributed random noise. The variance of this noise can be computed on the basis of Eq. (20), i.e.

$$\sigma^2[\varepsilon_u^p] = \frac{\theta_u}{\theta_u^p}\sigma^2[\varepsilon_u] \quad (24)$$

According to Del Castillo (2002), if a linear static process with an IMA(1,1) disturbance with time correlation parameter θ and variance σ^2 is under the EWMA control, the error output of the finished product will have an expected mean of zero and an expected variance of

$$E[(\sigma_y)^2] = \frac{1+(\theta)^2 - 2\theta(1-\Lambda\xi)}{1-(1-\Lambda\xi)^2}\sigma^2 \quad (25)$$

In the above equation, the ratio $\xi = \beta/b$ is used to represent the mismatch between model gain and process gain. For the thread (p, u), it will experience an IMA(1,1) disturbance with time correlation parameter θ_u^p, and a random noise with zero mean and variance $\sigma^2[\varepsilon^p]$. Hence the products of thread (p, u) under EWMA-RbR control should have an expected mean of zero, i.e., $\tilde{\mu}[y_u^p] = 0$, and an expected variance of:

$$\tilde{\sigma}^2[y_u^p] = \frac{1+(\theta_u^p)^2 - 2\theta_u^p(1-\Lambda_u^p\xi_u^p)}{1-(1-\Lambda_u^p\xi_u^p)^2}\sigma^2[\varepsilon_u^p] + \frac{2}{(2-\Lambda_u^p\xi_u^p)}\sigma^2[\varepsilon^p] \quad (26)$$

Note that this formula can be found in Del Castillo (2002) and, for the sake of completeness, a derivation is provided in the *Supplementary Material*. If $\xi_u^p = 1$, this formula can be written as

$$\tilde{\sigma}^2[y_u^p] = \frac{1+(\theta_u^p)^2 - 2\theta_u^p(1-\Lambda_u^p) + 2\Lambda_u^p r}{1-(1-\Lambda_u^p)^2}\sigma^2[\varepsilon_u^p] \quad (27)$$

where $r = (\sigma^2[\varepsilon^p])/(\sigma^2[\varepsilon^p])$. The expected mean and variance, as well as the estimated process capability of the product p can therefore be written as:

$$\tilde{\mu}[y^p] = \sum_{u=1}^{U} f_u^p\tilde{\mu}[y_u^p] = 0 \quad (28)$$

$$\tilde{\sigma}^2[y^p] = \sum_{u=1}^{U} f_u^p\tilde{\sigma}^2[y_u^p] \quad (29)$$

$$\tilde{C}_{pk}[y^p] = \frac{USL^p}{3\tilde{\sigma}[y^p]} = -\frac{LSL^p}{3\tilde{\sigma}[y^p]} = \frac{USL^p - LSL^p}{6\tilde{\sigma}[y^p]} \quad (30)$$

Notice that Eq. (30) is valid only under the condition that $USL^p = -LSL^p$.

2.5. Mathematical formulation of MINLP model

In principle, Eqs. (1), (7), (21), (23), (24), (27), (29) and (30) can be used as the constraints of a nonlinear programming (NLP) problem to determine the optimal dispatch strategy according to Eq. (6) or (8). However, these two NLP models are unsolvable in practice since $h_u^p \to \infty$ as $f_u^p \to 0$. To facilitate solution, let us introduce a set of binary variables ($I_u^p \in \{0, 1\}$) along with the following logic constraints:

$$f_u^p \le I_u^p \le f_u^p L \quad (31)$$

$$(1 - I_u^p)L \le h_u^p \le L \quad (32)$$

$$\left(\sum_{p=1}^{P} f_u^p\lambda^p\right) I_u^p = f_u^p\lambda^p h_u^p \quad (33)$$

where L is a large enough positive constant. The implications of these constraints can be summarized as follows

$$I_u^p = 0 \Rightarrow f_u^p = 0, \quad h_u^p = L \quad (34)$$

$$I_u^p = 1 \Rightarrow 0 < f_u^p < 1, \quad 0 < h_u^p < L \quad (35)$$

Thus, two corresponding MINLP models can be constructed by incorporating Eqs. (31)–(33) into the aforementioned NLP models. The input parameters required in solving these MINLPs are: the characteristic parameters of tool disturbances, i.e., θ_u and $\sigma^2[\varepsilon_u]$, the characteristic parameters of disturbance attributed to the products $\sigma^2[\varepsilon^p]$, the arrival rates of unprocessed products λ^p, the upper and lower quality specifications of the finished products, i.e., USL^p and LSL^p, the process capability requirements $(C_{pk})_{\min}^p$, the process times τ_u^p, and the upper limits of utilization rates $\rho_{u,\max}$. The primary decision variables in these model are the dispatch fractions f_u^p, while the auxiliary binary parameters I_u^p of each thread should be selected as well. The needed optimization computation can be readily implemented with the commercial software GAMS.

3. Optimization results

To demonstrate the feasibility of the proposed mathematical programming approach for identifying the optimal dispatching policy, let us consider a system with four products (denoted as a, b, c and d) and seven parallel tools. Model parameters for different tools and products can be found in Table 1. It is also assumed that $\rho_{u,\max} = 1$, $(C_{pk})_{\min}^p = 1$, $USL = 3.2$, $LSL = -3.2$, $\theta_u = 0.8$, $\Lambda_u^p = 0.5$, $\mu[\varepsilon_u] = 0$, and $\xi_u^p = 1$. The dispatch policies can be generated by solving the proposed MINLP model. Three simple cases are considered in the sequel.

3.1. Case 1: uniform distribution of production loads

If there is no preference for any thread, the intuitive dispatch policy of distributing the production loads uniformly among all available tools can be adopted. Thus, the dispatching fractions of all threads should be identical, i.e., $f_u^p = 1/7$. The resulting process capacities should be:

$$\tilde{C}_{pk}[y^a] = 1.775, \quad \tilde{C}_{pk}[y^b] = 1.497, \quad \tilde{C}_{pk}[y^c] = 1.326,$$

$$\tilde{C}_{pk}[y^c] = 1.767 \quad \therefore \sum_p \tilde{C}_{pk}[y^b] = 6.365$$

The corresponding utilization rates are:

$$\rho_1 = 0.765, \quad \rho_2 = 0.766, \quad \rho_3 = 0.764, \quad \rho_4 = 0.767,$$

$$\rho_5 = 0.763, \quad \rho_6 = 0.763, \quad \rho_7 = 0.768 \quad \therefore \sum_u \rho_u = 5.357$$

Table 1
Tool, product and thread parameters of case studies.

τ_u^p	Tool 1	Tool 2	Tool 3	Tool 4	Tool 5	Tool 6	Tool 7	λ^p	μ^p	$\sigma^2[\varepsilon^p]$
Product *a*	50.0	50.5	49.9	49.6	50.2	49.5	50.3	0.012	−0.32	0.108
Product *b*	40.0	40.7	40.2	39.5	39.8	39.7	40.1	0.024	−0.60	0.324
Product *c*	40.0	39.2	39.4	40.9	40.5	40.4	39.6	0.048	−0.32	0.288
Product *d*	40.0	40.5	40.3	39.7	39.2	39.6	40.7	0.048	0.16	0.108
$\sigma^2[\varepsilon_u]$	0.324	0.288	0.288	0.108	0.288	0.108	0.108			
$\alpha_u^p = \alpha_u$	1.0	2.0	1.5	0.5	2.5	1.8	3.0			
$\beta_u^p = \beta_u$	0.8	1.0	2.0	1.5	0.6	2.5	1.3			
$a_u^p = a_u$	1.1	1.9	1.6	0.6	2.4	1.6	3.1			

Notice that the process capabilities of product *b* and *c* are relatively lower than those of *a* and *d*. This is because the variations in their disturbances, i.e., $\sigma^2[\varepsilon^p]$, are higher. Notice also that the utilization rates of all tools are roughly the same and close to 76.5%. This corresponds to a situation in which the production facility is neither flooded with customer demand nor does it have a lot of idle capacity.

3.2. Case 2: minimization of total utilization rate

In this case, the proposed MINLP model is solved to minimize the total utilization rate, i.e., Eq. (6). This and all other optimization problems were solved in this study with solver DICOPT in GAMS on a PC, which is equipped with an Intel® Core™ i5 Quad CPU 760 @ 2.8 GHz and 4.00 GB RAM (3.49 GB usable) 32-bit WIN7 operating system platform. To produce a credible optimum, each problem was solved repeatedly for 2000 runs with randomly generated initial guesses over the given intervals. Only the best solution was kept and reported. The average solution time of a single run is approximately 1 s.

Minimization of total utilization rate will determine how much extra capacity the plant has, and whether additional orders can be delivered. The corresponding optimization results are presented in Table 2. Instead of assigning each product to every tool, each product is mainly dispatched to one to three tools in the present case. Since the utilization rate of tool *u* is given by $\rho_u = \sum_{p=1}^{P} f_u^p \lambda^p \tau_u^p$, let us examine the values of $\lambda^p \tau_u^p$ in Table 3.

Notice that the lowest $\lambda^p \tau_u^p$ associated with each product are: $\lambda^a \tau_6^a = 0.5891$ for product *a*, $\lambda^b \tau_4^b = 0.9405$ for product *b*, $\lambda^c \tau_2^c = 1.8667$ for product *c*, and $\lambda^d \tau_5^d = 1.8667$ for product *d*. It can thus be expected that tools 6, 4, 2 and 5 will be selected first to process products *a*, *b*, *c* and *d* respectively. The most efficient tool for product *b* is tool 4. Moreover, since $\lambda^b \tau_4^b = 0.9405 \leq 1$, it is possible for us to dedicate product *b* to tool 4. The fastest tool for producing product *c* is tool 2. However, tool 2 is not sufficient to take up all productions of product *c*, because $\lambda^c \tau_2^c = 1.8667$. Hence about half of product *c* is allocated to the second most efficient tool, i.e. tool 3, for which $\lambda^c \tau_3^c = 1.8672$. The most efficient tool for product *d* is tool 5. Since $\lambda^d \tau_5^d = 1.8667 \geq 1$, another tool for product *d* must be found. The second most efficient tool for product *d* is tool 6, which is also the most efficient tool for product *a*. Thus products *a* and *d* must share tool 6. Moreover, since tool 6 does not have enough capacity for product *a* and part of product *d*, alternate tools must be found. Tool 1, the third best tool, is thus used for processing product *d*, while tools 3 and 4 are for product *a*. It can be seen in the solution that general dispatch philosophy is to rank the tools in terms of efficiency for each product and try to pack the product to the best ranked tools for this product. Compared with the results obtained in Case 1, the total utilization rate in the present case is slightly improved to 75.1%. However, it can also be observed that tools 2–6 are almost fully operational while the utilization of tool 1 is only 28.5% and tool 7 is completely idle. Finally, notice that the overall process capability does not change much from that achieved with uniform load distribution.

3.3. Case 3: maximization of overall process capability

Another alternative dispatch policy can be stipulated by maximizing the overall process capabilities of all products. The optimization results are shown in Table 4. It was found that the overall process capability can be substantially increased to 6.779 (compared to 6.365 in Case 1 and 6.352 in Case 2) without sacrificing the total utilization, i.e., 76.3% (compared to 76.5% in Case 1 and 75.1% in Case 2). Table 5 lists the sum of variances of the tool disturbances and the product disturbances. It is obvious that tools 4, 6, 7 are better tools for quality control and product *a* and *d* are the products with least product variations. Consequently, product *d* was assigned to tool 6 and 7, while product *a* was assigned to tool 4. The remaining capacity of tool 4 was taken up by part of product *b*. The other portion of product *b* was assigned to tool 3, which is one of the next best available tool. Product *c* was assigned to tools 2 and 5, which are the best tools available other than the occupied tools mentioned above, i.e., tool 4, 6, and 7. So it can be expected that, with the present dispatching model, the tool with the lowest value of $\sigma^2[\varepsilon_u] + \sigma^2[\varepsilon^p]$ should always be selected first to process product *p* unless $\rho_u \to 1$. As a result, almost all tools are dedicated (except tool 4) to a single product. Notice that, in a dedicated tool, i.e., $h_u^p = 1$, the RbR control will be more effective because of the higher action frequencies.

4. Numerical simulation procedure

Let us assume in the aforementioned production environment that the random inter-arrival time of product *p* at tool *u* (denoted as Y_u^p) and the corresponding random service time (denoted as Z_u^p) are both exponentially distributed, that is

$$Pr\{Y_u^p \leq t\} = 1 - e^{-f_u^p \lambda^p t} \tag{36}$$

$$Pr\{Z_u^p \leq t\} = 1 - e^{-t/\tau_u^p} \tag{37}$$

To simulate the aforementioned queuing system, it is convenient to first generate random variates from a uniform distribution between 0 and 1, and then make proper transformations to produce the exponentially distributed inter-arrival times and service times. Notice that $Pr\{X(k_u^p) \leq x\} = x$ if $X(k_u^p) \in U[0, 1]$ is a uniformly distributed random number between 0 and 1. Thus, we could obtain Y_u^p and Z_u^p according to the uniformly distributed random numbers $X'(k_u^p)$ and $X''(k_u^p)$ and let

$$X'(k_u^p) = 1 - e^{-f_u^p \lambda^p Y_u^p(k_u^p)} \Rightarrow Y_u^p(k_u^p) = -\frac{\ln[1 - X'(k_u^p)]}{f_u^p \lambda^p} \tag{38}$$

$$X''(k_u^p) = 1 - e^{-Z_u^p(k_u^p)/\tau_u^p} \Rightarrow Z_u^p(k_u^p) = -\tau_u^p \ln[1 - X''(k_u^p)] \tag{39}$$

Table 2
Optimization results for Case 2. $\sum_u \rho_u = 5.285, \quad \sum_p \tilde{C}_{pk}[y^p] = 6.352.$

f_u^p	Tool 1	Tool 2	Tool 3	Tool 4	Tool 5	Tool 6	Tool 7	$\tilde{C}_{pk}[y^p]$
Product *a*			0.210	0.098		0.692		1.944
Product *b*				0.999		0.001		1.439
Product *c*		0.535	0.465					1.280
Product *d*	0.150		0.001	0.001	0.535	0.313		1.690
ρ_u	0.285	0.999	0.999	0.999	0.999	0.999	0	

Table 3
Products of mean arrival rate and mean processing time in Case 2.

$\lambda^p \tau_u^p$	Tool 1	Tool 2	Tool 3	Tool 4	Tool 5	Tool 6	Tool 7
Product *a*	0.5950	0.6010	0.5938	0.5902	0.5974	0.5891	0.5986
Product *b*	0.9524	0.9691	0.9572	0.9405	0.9476	0.9453	0.9548
Product *c*	1.9048	1.8667	1.8762	1.9477	1.9286	1.9238	1.8858
Product *d*	1.9048	1.9286	1.9191	1.8905	1.8667	1.8858	1.9381

Table 4
Optimization results for Case 3. $\sum_u \rho_u = 5.342, \quad \sum_p \tilde{C}_{pk}[y^p] = 6.779.$

f_u^p	Tool 1	Tool 2	Tool 3	Tool 4	Tool 5	Tool 6	Tool 7	$\tilde{C}_{pk}[y^p]$
Product *a*				1				2.086
Product *b*	0.001		0.564	0.435				1.315
Product *c*	0.001	0.535			0.417		0.046	1.289
Product *d*						0.530	0.470	2.090
ρ_u	0.003	0.999	0.540	0.999	0.805	0.999	0.999	

The k_u^pth arrival time of product p at tool u, i.e., $A_u^p(k_u^p)$, can thus be determined with the following equation:

$$A_u^p(k_u^p) = \sum_{k=1}^{k_u^p} Y_u^p(k) \tag{40}$$

As mentioned before, if P different types of products are produced in tool u, then the run numbers for all products in this tool must be reassigned according to the *precedence order* of their arrival times. These overall run numbers s_u can be determined with the assignment function defined previously in Eqs. (12) and (13). Thus, the arrival time $A_u(s_u)$ and the processing time $Z_u(s_u)$ of any product at tool u can be expressed alternatively with the run number of all products, i.e.

$$A_u(s_u) = A_u^p(k_u^p) \tag{41}$$

$$Z_u(s_u) = Z_u^p(k_u^p) \tag{42}$$

where $s_u = \Phi(k_u^p)$. Let us consider the simple example in Fig. 2 again for further clarification. Since it can be observed from the first two axes $k_u^{p'}$ and $k_u^{p''}$ that:

$$A_u^{p'}(1) < A_u^{p''}(1) < A_u^{p'}(2) < A_u^{p''}(2) < A_u^{p'}(3) < A_u^{p'}(4) < A_u^{p''}(3) < A_u^{p''}(4) \tag{43}$$

the corresponding run number s_u and the arrival time $A_u(s_u)$ can be determined according to in Eqs. (12) and (13) respectively. These results are plotted on the third axis s_u.

After fixing the arrival times $A_u(s_u)$ and the service times $Z_u(s_u)$, the starting times $C_u(s_u)$, the departure times $D_u(s_u)$ and the waiting times $W_u(s_u)$ can be computed with the following formulas:

$$C_u(s_u) = A_u(s_u) + W_u(s_u) \tag{44}$$

$$D_u(s_u) = C_u(s_u) + Z_u(s_u) \tag{45}$$

where the waiting time at run s_u can be determined according to the current arrival time and the departure time at the previous run, i.e.

$$W_u(s_u) = \begin{cases} 0 & \text{if } A(s_u) > D(s_u - 1) \\ D(s_u - 1) - A(s_u) & \text{otherwise} \end{cases} \tag{46}$$

Furthermore, it is assumed in all our simulation runs that $W_u(1) = 0$ and therefore $A_u(1) = C_u(1)$. As an example, the time sequence of a simulated multi-products process is shown in Fig. 4.

Since the utilization rate is defined as the fraction of time horizon that a tool is busy, ρ_u can be calculated with the simulation results according to the following equation:

$$\rho_u = \frac{\sum_{s_u=1}^{NR_u} Z_u(s_u)}{D_u(NR_u)} \tag{47}$$

Table 5
Sum of variances of tool and product noises in Case 3.

$\sigma^2[\varepsilon_u] + \sigma^2[\varepsilon^p]$	Tool 1	Tool 2	Tool 3	Tool 4	Tool 5	Tool 6	Tool 7
Product *a*	0.432	0.396	0.396	0.216	0.396	0.216	0.216
Product *b*	0.648	0.612	0.612	0.432	0.612	0.432	0.432
Product *c*	0.612	0.576	0.576	0.396	0.576	0.396	0.396
Product *d*	0.432	0.396	0.396	0.216	0.396	0.216	0.216

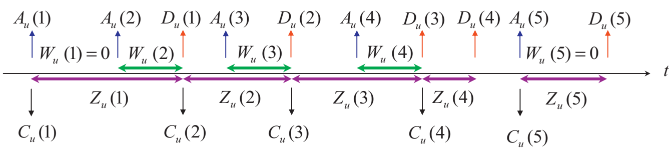

Fig. 4. Time sequence of a fictitious multi-products process.

Having determined the arrival and departure times of all products to be processed in tool u and their individual and overall run numbers (i.e., k_u^p and s_u), the embedded RbR control systems can be simulated according to the model formulation given in Sections 2.2 and 2.3. Consequently, the quality of the finished product $y_u^p(k_u^p)$, the estimated mean and variance of the finished product p on tool u, i.e., $\tilde{\mu}(y_u^p)$ and $\tilde{\sigma}^2(y_u^p)$, can also be calculated on the basis of the simulated data.

5. Simulation results

The uniform dispatch policy adopted in Case 1 and the optimal dispatching policies obtained previously in Section 3 for Case 2 and Case 3 were implemented in numerical simulation studies to confirm the validity of the proposed MINLP model. The model parameters used in the simulation runs are the same as those listed in Table 1. For each of the above-mentioned cases, five (5) repeated simulation runs were carried out. The resulting simulation data were then used for calculating the average values of important performance indices, such as the process capability of each product and the utilization rates of every tool.

The average process capabilities of every product and average utilization rates of every tool obtained for all cases are presented in Figs. 5–10. In terms of the process capacities, the errors between the predictions and simulations are around 15%, 3%, and 4% respectively for Cases 1, 2 and 3. On the other hand, the corresponding errors in utilization rates are 3%, 6%, and 4% respectively. It can be observed that the independently generated model predictions and simulated results are in general consistent, although process capability predictions are in most cases slightly higher. Note also that, although the arrivals of different products at each tool are assumed to be regular in the optimal dispatch model, the inter-arrival times of incoming products in the simulated stochastic processes at exponentially distributed. Due this stochastic nature, the simulation results tend to have larger variances and hence smaller process capabilities. Nevertheless the demonstrated ability to predict the general trends of utilization rate of each tool and process capabilities of each product indicates that the proposed model can be used for determining proper dispatch policies in different scenarios.

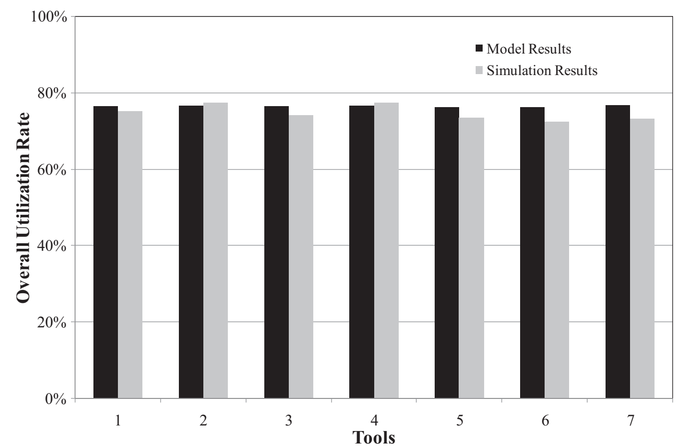

Fig. 6. Comparison of predicted and simulated utilization rates for Case 1.

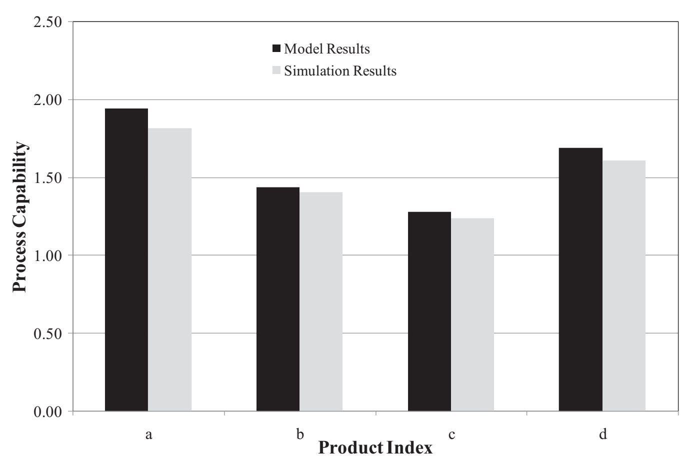

Fig. 7. Comparison of predicted and simulated process capacities for Case 2.

6. Model errors and safe tuning

It was pointed out in Section 3, that optimal dispatch based on minimizing overall utilization rate depends on the manufacturing

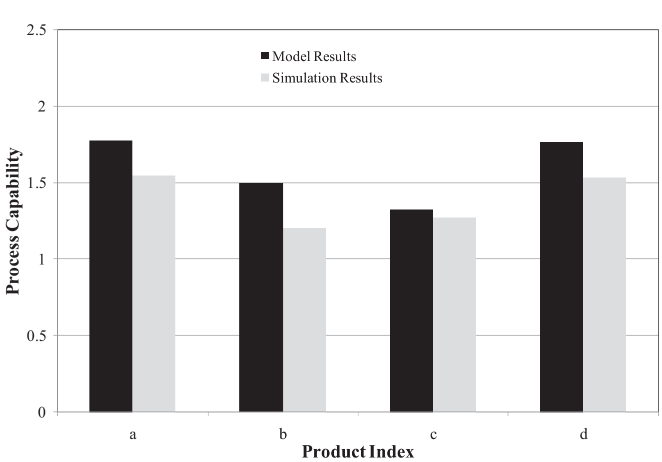

Fig. 5. Comparison of predicted and simulated process capacities for Case 1.

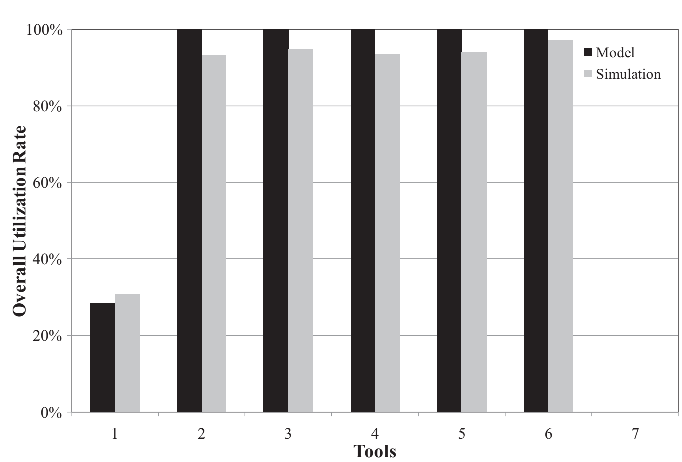

Fig. 8. Comparison of predicted and simulated utilization rates for Case 2.

Table 6
Optimization results for Case 4, maximizing process capability. $\theta_u = 0.6$, $\Lambda_u = 0.4$, $\sum_u \rho_u = 5.319$, $\sum_p \tilde{C}_{pk}[y^p] = 7.338$.

f_u^p	Tool 1	Tool 2	Tool 3	Tool 4	Tool 5	Tool 6	Tool 7	$\tilde{C}_{pk}[y^p]$
Product *a*							1	2.295
Product *b*					1			1.363
Product *c*	0.464	0.535	0.001					1.385
Product *d*			0.001	0.528		0.471		2.294
ρ_u	0.884	0.999	0.004	0.999	0.948	0.887	0.599	

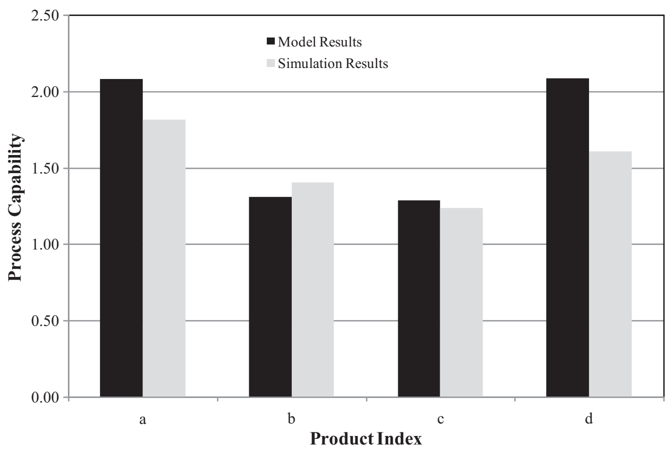

Fig. 9. Comparison of predicted and simulated process capacities for Case 3.

efficiencies of different products on different tools; and optimal dispatch based on maximizing process capabilities depends on the magnitude of different tool-related and product-related noises. These are intrinsic properties of tools and products that can be a priori known to a certain degree of accurateness. However, accurate estimates of the other parameters of the model, such as dynamic parameters θ_u of the tools, may not always be available. It is necessary to investigate the impacts of model mismatch with simulation studies.

It should first be noted that, if a reasonably accurate estimate of the time correlation coefficient θ_u can be obtained and the assumptions that $h_u^p = 1$ and $r = 0$ in Eq. (27) are acceptable, the optimal controller tuning parameter can in fact be determined *exactly* with the following equation

$$(\Lambda_u^p)_{opt} = 1 - \theta_u^p = 1 - \theta_u \quad (48)$$

Since in general $r \neq 0$ and the estimate of θ_u is not reliable, a heuristic approach is developed in this work for practical applications. From extensive simulation results, it can be observed that the model predictions are relatively insensitive to the changes in θ_u and Λ_u. For example, Table 6 presented optimization results including dispatch policy and performance indices obtained using a time correlation parameter of $\theta_u = 0.6$ and $\Lambda_u = 0.4$. The simulation results of average process capabilities of different products obtained with different values of θ_u and Λ_u ($\theta_u = 0.75$, $\Lambda_u = 0.25$) were shown in Fig. 11. Therefore, if accurate estimates of the dynamic parameters cannot be obtained with a high level of confidence, it may be enough to determine a "safe" interval for every EWMA tuning parameter.

Let us define the coefficient of $\sigma^2[\varepsilon_u^p]$ in Eq. (27) as:

$$\Psi(\theta_u^p, \Lambda_u^p, r) = \frac{1 + (\theta_u^p)^2 - 2\theta_u^p(1 - \Lambda_u^p) + 2\Lambda_u^p r}{1 - (1 - \Lambda_u^p)^2} \quad (49)$$

Under the conditions that $h_u^p = 1$ and $\theta_u^p = \theta_u$, Ψ is plotted against Λ_u^p in Fig. 12 for various combinations of θ_u and r (θ_u = 0.01, 0.4, 0.6, 0.9; r = 0, 0.1, 0.2, 0.3). It can be observed that, although the locations of lowest points on these four curves are different, they nearly overlap in the interval [0.4, 0.6] and, also, the corresponding function values are approximately the same as their minima. Thus, it appears that a safe setting of the controller tuning parameter could be chosen around 0.5 if θ_u is unknown. This is because, with such a selection, the variance of each product may reach a

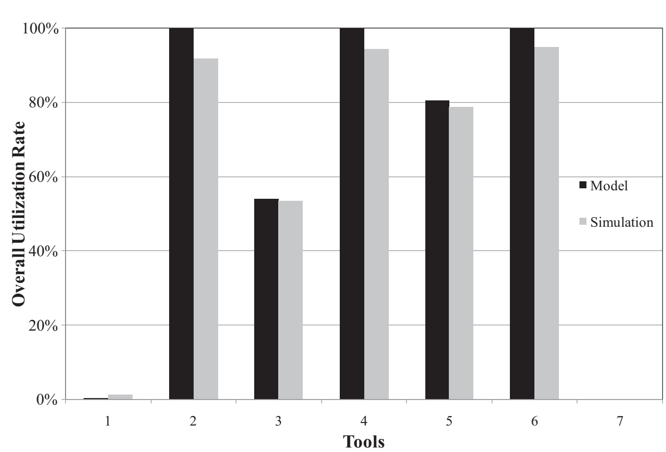

Fig. 10. Comparison of predicted and simulated utilization rates for Case 3.

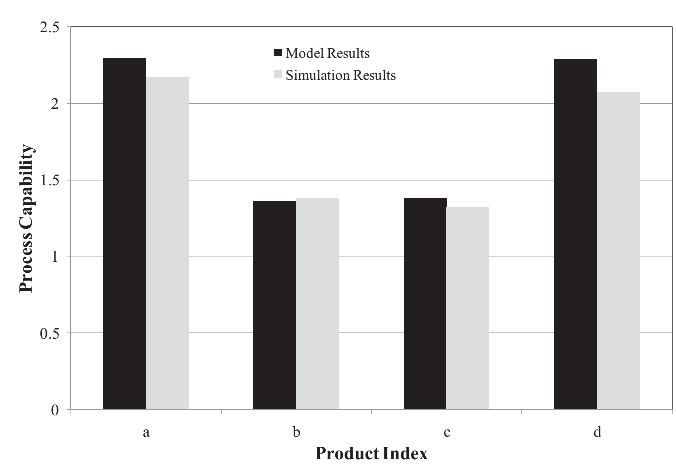

Fig. 11. Comparison of predicted and simulated process capacities for Case 4, when the parameters $\theta_u = 0.6$, $\Lambda_u = 0.4$ used in optimizations are different from those $\theta_u = 0.75$, $\Lambda_u = 0.25$ used in simulation.

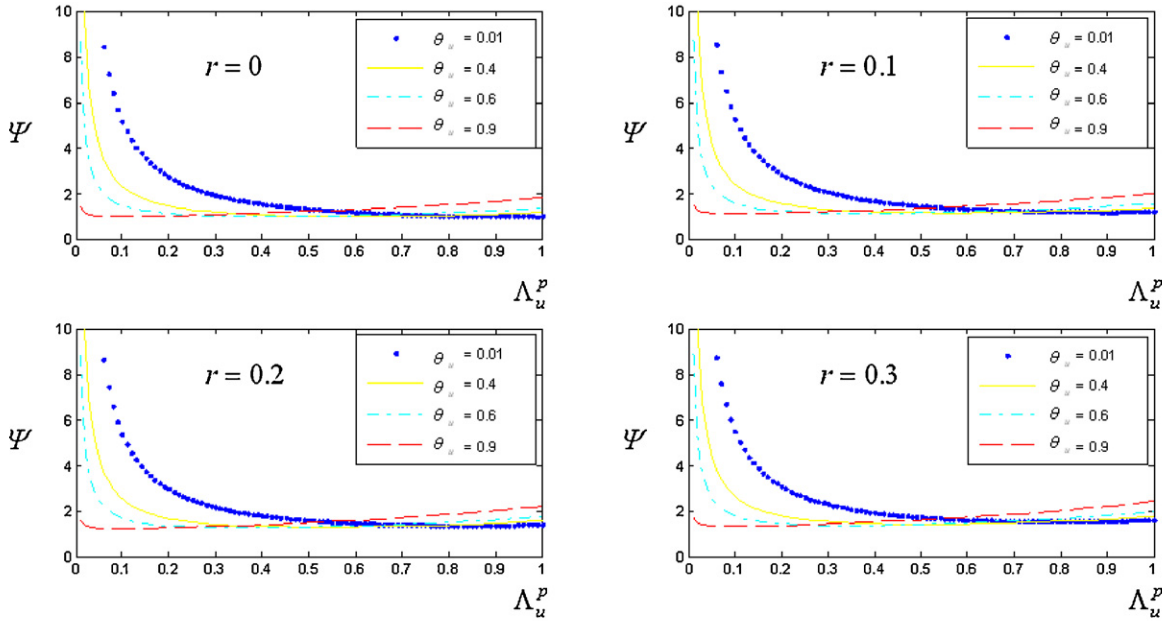

Fig. 12. Plots of $\Psi(\Lambda_u^p)$ under conditions $h_u^p = 1$ and $\theta_u^p = \theta_u$.

near minimum value and, consequently, the corresponding process capability should be at least not too far away from the maximum.

7. Conclusions

By treating the typical semiconductor manufacturing environment as multiple parallel queuing systems, a mixed integer nonlinear programming (MINLP) model has been constructed to determine the optimal dispatching policy. Either process capacity or utilization rate can be used as the objective function of optimization problem. When one of them is selected as the design objective, the limiting value of the other is imposed as the inequality constraint in the MINLP model.

Based on the inherent assumptions of M/M/1 queuing system, the inter-arrival times and processing times of each product can be simulated with random number generators according to the aforementioned optimal dispatching policy. The performance of every individual RbR control system can then be predicted in numerical simulation studies. The simulation results are consistent with those predicted with the aforementioned MINLP model.

Minimizing overall utilization leads to dedication of products to certain tools because of efficiency of these tools in producing the particular products. However, such a policy reduces but does not preclude use of mixed product runs on each tool. Maximizing process capabilities lead to more segregated production in which each tool is dedicated to each product. This is caused by the differences in process abilities of tools to produce finished product of different qualities and the improvement of RbR control performance when a tool is dedicated. Substantial improvement in overall quality can be achieved without much sacrifice in utilization rate in the example studied.

Theoretically the optimal dispatching policy can be implemented in realistic processes if all model parameters are given. Simulation shows that since RbR control is effective, the results only on the intrinsic process capabilities of the tools and products; and do not depend much on the time-correlation of different tool disturbances.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.compchemeng.2012.12.009.

References

Bhaskaran, K., & Pinedo, M. (1992). In G. Salvendy (Ed.), *Handbook of industrial engineering*. New York: John Wiley.
Blackstone, J. H., Phillips, D. T., & Hogg, G. L. (1982). A state-of-the-art survey of dispatching rules for manufacturing job shop operations. *International Journal of Production Research*, *20*, 27–45.
Box, G. E. P., Jenkins, G. M., & Reinsel, G. (1994). *Time series analysis forecasting and control* (3rd ed.). New Jersey: Prentice-Hall.
Cassandras, C. G., & Lafortune, S. (1999). *Introduction to discrete event systems*. Boston: Kluwer Academic Publisher.
Choi, J. Y., & Reveliotis, S. (2005). Relative value function approximation for the capacitated re-entrant line scheduling problem. *IEEE Transactions on Automation Science and Engineering*, *2*, 285–299.
Dabbas, R. M., Chen, H. N., & Fowler, J. W. (2001). A combined dispatching criteria approach to scheduling semiconductor manufacturing systems. *Computers & Industrial Engineering*, *39*, 307–324.
Dabbas, R. M., & Fowler, J. W. (2003). A new scheduling approach using combined dispatching criteria in wafer fabs. *IEEE Transactions on Semiconductor Manufacturing*, *16*, 501–510.
Del Castillo, E. (2002). *Statistical process adjustment for quality control*. New York: John-Wiley & Sons.
Firth, S. K., Campbell, W. J., Toprac, A., & Edgar, T. F. (2006). Just-in-time adaptive disturbance estimation for run-to-run control of semiconductor processes. *IEEE Transactions on Semiconductor Manufacturing*, *19*, 298–315.
Kutanoglu, E., & Sabuncuoglu, I. (1999). An analysis of heuristics in a dynamic job shop with weighted tardiness objectives. *International Journal of Production Research*, *37*, 165–187.
Ma, M. D., Chang, C. C., Jang, S. S., & Wong, D. S. H. (2009). Mixed product run-to-run process control – An ANOVA model with ARIMA disturbance approach. *Journal of Process Control*, *19*, 604–614.
Moyne, J. (2001). *Run-to-run control in semiconductor manufacturing*. Florida: CRC Press.
Pasadyn, A. J., & Edgar, T. F. (2005). Observability and state estimation for multiple product control in semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing*, *18*, 592–604.
Patel, N. S., & Jenkins, S. T. (2000). Adaptive optimization of run-to-run controllers: The EWMA example. *IEEE Transactions on Semiconductor Manufacturing*, *13*, 97–107.
Pierce, N. G., & Yurtsever, T. (1999). Dynamic dispatch and graphical monitoring system. In *Proceedings of the international conference on semiconductor manufacturing operational modeling and simulation* San Francisco, (pp. 57–61).

Raghu, T. S., & Rajendran, C. (1993). An efficient dynamic dispatching rule for scheduling in a job shop. *International Journal of Production Economics*, *32*, 301–313.
Rajendran, C., & Holthaus, O. (1999). A comparative study of dispatching rules in dynamic flowshops and jobshops. *European Journal of Operational Research*, *116*, 156–170.
Sachs, E., Hu, A., & Ingolfsson, A. (1995). Run by run process control: combining SPC and feedback control. *IEEE Transactions on Semiconductor Manufacturing*, *8*, 26–43.
Smith, T. H., & Boning, D. S. (1997). Artificial neural network exponentially weighted moving average controller for semiconductor processes. *Journal of Vacuum Science & Technology*, *15*, 236–239.
Tseng, S. T., Yeh, A. B., Tsung, F., & Chan, Y. Y. (2003). A study of variable EWMA controller. *IEEE Transactions on Semiconductor Manufacturing*, *16*, 633–643.
Zheng, Y., Lin, Q. H., Wong, D. S. H., Jang, S. S., & Hui, K. (2006). Stability and performance analysis of mixed product run-to-run control. *Journal of Process Control*, *16*, 431–443.