Proceedings of the 2011 Winter Simulation Conference
S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, eds.

SCHEDULING POLICIES IN MULTI-PRODUCT MANUFACTURING SYSTEMS WITH SEQUENCE-DEPENDENT SETUP TIMES

Wei Feng
Li Zheng

Department of Industrial Engineering
Tsinghua University
Beijing 100084, CHINA

Jingshan Li

Department of Industrial and Systems Engineering
University of Wisconsin - Madison
1513 University Avenue
Madison, WI 53706, USA

ABSTRACT

Multi-product production systems with sequence-dependent setup times are typical in manufacturing of semiconductor chips and other electronic products. In such systems, the scheduling policies to coordinate the production of multiple product types play an important role. In this paper, we study a multi-product manufacturing system with finite buffers, sequence-dependent setup times and various scheduling policies. Using continuous time Markov chain models, we evaluate the performance of such systems under seven scheduling policies, i.e., cyclic, shortest queue, shortest processing time, shortest overall time (including setup time and processing time), longest queue, longest processing time, and longest overall time. The impact of these policies on system throughput are compared, and the conditions characterizing the superiority of each policy are investigated. The results of this work can provide production engineers and supervisors practical guidance to operate multi-product manufacturing systems with sequence-dependent setups.

1 INTRODUCTION

Flexible manufacturing systems are becoming more and more important in manufacturing industry due to the increasing trends of customization and market changes. For example, microprocessors products in semiconductor factories may vary in sizes, functions and memory capacity; LED lights are produced with different power-specifications; and the final products of LCD monitors may be with various colors and functional packages to satisfy the needs of different customer groups. To manufacture these products, finite buffers are typically used and kept at smaller capacities to reduce storage space, work-in-process (WIP), and production cycle time. In addition, in multiple products environment, setup times usually occur when a machine is switched from one product type to another. These setup times are typically sequence dependent, which implies that the duration of setups time depend on both the immediately preceding product type and the current one. For example, when microprocessors with different package types (e.g., “Ball-package” or “Pin-package”) are processed, the setup time between two “Ball-package” products can be different from the changeover time between a “Pin-package” product and a “Ball-package” product. For the latter case, the temperature in the manufacturing equipment may need to be adapted to a new package type, and such adaption takes time. Moreover, the products are usually processed in batches, to reduce setup times, minimize cost, and improve product quality. Thus, the coordination and scheduling of different product batches become critical. Depending on the specific production environment, many scheduling policies, such as cyclic policy, longest queue policy, etc., have been implemented on the shop floor. These policies can have a great impact on the output of production systems. Therefore, studying multi-product systems with finite buffers, sequence-dependent setup times and different scheduling policies is of significant importance.

During the past decades, substantial research effort has been devoted to the study of multi-product manufacturing systems (such as monographs Viswanadham and Narahari 1992; Buzacott and Shanthikumar 1993; Tempelmeier and Kuhn 1993; Zhou and Venkatesh 1999; and reviews Suri 1985; Buzacott and Yao

1986; Sethi and Sethi 1990; Beach et. al 2000; Takagi 2000). In recent years, there is an increasing research attention on multi-product systems (for instance, Altiok and Shiue 2000; Krieg and Kuhn 2002, 2004; Li and Huang 2005, 2007; Ryan and Vorasayan 2005; Jang 2007; Satyam 2007; Dasci and Karakul 2008; Gurgur and Altiok 2008; Satyam and Krishnamurthy 2008; Tubilla and Gershwin 2009; Wang et. al 2010). However, the issues of either setup times or finite buffers are often ignored. Only a few studies consider the impact of setup times and finite buffers together (e.g., Altiok and Shiue 2000; Krieg and Kuhn 2004; Dasci and Karakul 2008), and only very few studies investigate multi-product systems with sequence-dependent setups (Dasci and Karakul 2008; Tubilla and Gershwin 2009). No analysis and comparison of different scheduling policies have been addressed in these papers. Therefore, an in-depth study on the performance analysis of manufacturing systems with sequence-dependent setups, finite buffers, and various scheduling policies, as well as a comparison among different policies, is necessary and important.

The main contribution of this paper is in studying the scheduling policies in multi-product manufacturing systems with sequence-dependent setups and finite buffers, using a continuous time Markov chain model. A comparison study of the performances under seven different scheduling policies is carried out. The conditions or managerial insights useful for operations of multiple-product systems are obtained.

The remainder of the paper is structured as follows: Section 2 describes the system and formulates the problem. Performance evaluation is introduced in Section 3. The comparison among different scheduling policies and the impacts of system design parameters are discussed in Section 4. Finally, conclusions are given in Section 5.

2 SYSTEM DESCRIPTION AND PROBLEM FORMULATION

In this paper, we study a manufacturing system that processes multiple products (see Figure 1). The following assumptions define the part arrival, processing, setup times, finite buffers, and scheduling policies in the system:

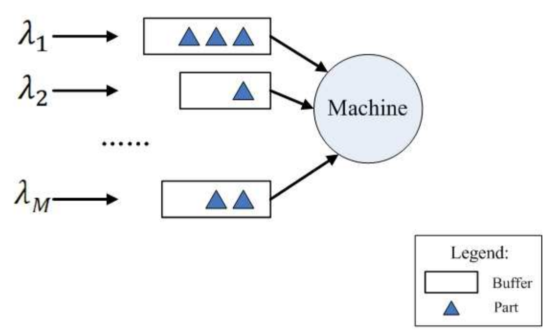

Figure 1: A multi-product manufacturing system producing M products.

(i) The manufacturing system can process M types of products, denoted as products $1,2,\dots M$.

(ii) The products arrive at the manufacturing facility independently. The arrival rate of product type j is defined by a Poisson process with parameter λ_j, $j=1,\dots,M$.

(iii) The processing time for product type j follows exponential distribution with parameter μ_j, $j=1,\dots,M$.

(iv) The system can only accommodate at most N_j parts of type j product at any time period. In other words, there is a buffer b_j for product type j in front of the machine, with capacity N_j (including the one on the machine), $j=1,\dots,M$. Due to its finite capacity, a part may be lost if the corresponding buffer is full at the time of its arrival.

(v) There is an exponential distributed setup time with mean s_{ij} when the facility is switching from product type i to type j, $i,j=1,\dots,M$, $i\neq j$.

(vi) The manufacturing facility will keep processing product i until all parts of type i have been processed and its buffer b_i is empty.

(vii) When a buffer is emptied, the manufacturing facility switches to a non-empty buffer according to a given scheduling policy defined as follows:

(1) *Cyclic policy:* The machine processes products with the order of $1,2,\dots,M$. It returns to process type 1 after type M. If there is no part available in buffer b_j when part type j is to be processed, no setup will be carried out and the facility switches to processing type $j+1$.

(2) *Longest queue (LQ) policy:* The machine always switches to a product with the longest non-empty queue. If there are several longest queues that share the same length, then the nearest product in the cyclic order is processed.

(3) *Shortest queue (SQ) policy:* The machine always switches to a product with the shortest non-empty queue. If there are several shortest queues that share the same length, then the nearest product in the cyclic order is processed.

(4) *Longest processing time (LPT) policy:* The machine always switches to a product with the longest non-zero processing time. The processing time is estimated as the queue length divided by the processing rate of the corresponding product. If there are several queues that share the longest processing time, then the nearest one in the cyclic order is processed.

(5) *Shortest processing time (SPT) policy:* The machine always switches to a product with the shortest non-zero processing time. If there are several queues that share the shortest processing time, then the nearest one in the cyclic order is processed.

(6) *Longest overall time (LOT) policy:* The machine always switches to a non-empty buffer with the longest overall time (processing time plus setup time). If there are several queues that share the longest overall time, then the nearest one in the cyclic order is processed.

(7) *Shortest overall time (SOT) policy:* The machine always switches to a non-empty buffer with the shortest overall time. If there are several queues that share the shortest overall time, then the nearest one in the cyclic order is processed.

(viii) When all buffers are empty, the machine will be starved, and it will start setup whenever a new product arrives (or start processing without setup if the previous type product arrives).

In an appropriately defined state space, system (i)-(viii) is a stationary random process. Let TP^α be the system throughput, i.e., the average number of parts produced per unit of time, under policy α, $\alpha \in \{Cyclic, LPT, LQ, LOT, SPT, SQ, SOT\}$. Then, TP^α is a function of all system parameters:

$$TP^\alpha = f_\alpha(\Lambda, \Gamma, \mathcal{N}, \mathcal{S}), \tag{1}$$

where

$$\begin{aligned}
\Lambda &= [\lambda_1, \dots, \lambda_M], \\
\Gamma &= [\mu_1, \dots, \mu_M], \\
\mathcal{N} &= [N_1, \dots, N_M], \\
\mathcal{S} &= \begin{bmatrix} 0 & s_{12} & \dots & s_{1M} \\ s_{21} & 0 & \dots & s_{2M} \\ \dots & \dots & \dots & \dots \\ s_{M1} & s_{M2} & \dots & 0 \end{bmatrix}.
\end{aligned}$$

The problem addressed in this paper is formulated as follows: *Given the multi-product manufacturing system (i)-(viii), develop a method to estimate system throughput as a function of system parameters, and investigate the impact of different scheduling policies on system throughput.*

Solutions to the problem are presented in Sections 3 and 4 below.

3 PERFORMANCE EVALUATION

To evaluate system throughput, we introduce a continuous time Markov process model and its state space. Let $(\eta,\xi,\alpha,h_1,\ldots,h_M)$ define the system state, where η and ξ take values $1,\ldots,M$, denoting the previous and current product types, respectively; $\alpha=0$ or 1 characterizing the machine status, such that $\alpha=1$ implies the machine is available, and $\alpha=0$ represents the setup state; h_i, $i=1,\ldots,M$, describing the occupancy of part type i in the system, and $0\le h_i\le N_i$. When all $h_i=0$, the machine is starved (idle) for parts. Then the effective states can be obtained by deleting the unreachable ones that could not exist (Feng et al. 2011).

Lemma 1 Under assumptions (i)-(viii), there exist K effective states, where

$$K=(M-1)\sum_{j=1}^{M}\left[(2N_j-1)\prod_{i=1,i\neq j}^{M}N_i\right]. \tag{2}$$

Next, transition equations are introduced to describe the transitions among all effective states. Then, a transition matrix Q with dimensions $K\times K$ will be constructed. The transition rates λ_i, μ_i, and $1/s_{ij}$ are assigned to the matrix elements of Q. Other elements (except $Q(i,i)$) are assigned to be zero. Finally, by letting $Q(i,i)=-\sum_{j=1,j\neq i}^{K}Q(i,j)$, the transition matrix Q is obtained (see the Appendix for details). Let π_S be the stationary probability associated with state $S=(\eta,\xi,\alpha,h_1,h_2,...,h_M)$, i.e.,

$$\pi=[\pi_1,\pi_2,\ldots,\pi_K].$$

we obtain,

Theorem 1 Under assumptions (i)-(viii), the throughput of product j can be calculated as:

$$TP_j^{\alpha}=A_j^{\alpha}\mu_j,\quad j=1,\ldots,M, \tag{3}$$

where α represents the scheduling policy, $\alpha\in\{Cyclic,LPT,LQ,LOT,SPT,SQ,SOT\}$, and A_j denotes the probability that the manufacturing facility is processing product j, i.e.,

$$A_j^{\alpha}=\sum_{i,h_1,\ldots,h_M}\pi_{(i,j,1,h_1,\ldots,h_j>0,\ldots,h_M)}, \tag{4}$$

and π is solved from

$$\pi=\Delta\Phi^{-1}, \tag{5}$$

with

$$\Delta=[0,\ldots,0,1], \tag{6}$$

and

$$\Phi(:,i)=\begin{cases} Q(:,i), \text{ if } i=1,\ldots,K-1,\\ 1, \text{ if } i=K.\end{cases} \tag{7}$$

The the total throughput of the system is

$$TP^{\alpha}=\sum_{j=1}^{M}TP_j^{\alpha}. \tag{8}$$

Proof: See the Appendix. ■

4 COMPARISON OF SCHEDULING POLICIES

In this section, the impact of different scheduling policies on system throughput will be investigated. For simplicity, we assume $\lambda_i=\lambda$ and $N_i=N$, $i=1,\ldots,M$, throughout the section.

4.1 Scenarios with Equal Processing Rates

First, we assume all products have equal processing rate, i.e., $\mu_j = \mu,\ j = 1,\dots,M$. In this case, the LQ and LPT policies are identical, so are the SQ and SPT policies. Therefore, only five policies, SPT, SOT, LPT, LOT and Cyclic, will be studied. We will start with three-product case, then extend to more general scenarios.

4.1.1 Three-product Case

We carry out the investigation numerically by using Theorem 1 under different policies. The numerical cases are constructed as follows:

- First, the system parameters are selected from a given set, where the parameters are chosen in a structured way. Specifically, the utilization $\rho = \frac{3\lambda}{\mu}$ is selected from the set $\{0.6, 0.7, 0.8, 0.9\}$, and the buffer sizes from the set $\{2, 4, 6, 8, 10, 12, 14\}$. Then the arrival rate λ is determined. A total of 35 different setup matrices are selected, where s_{ij} are chosen from $\{0.5, 1, 2, 4, 6\}$ in the unit of $\frac{1}{\mu}$. As a combination of these parameters, 980 different numerical cases are tested.
- Next, randomly selected parameters are used for numerical tests, i.e., N, λ and s_{ij} are randomly chosen from $\frac{3\lambda}{\mu} \in [0.6, 1)$, $s_{ij} \in (0, 6]$, and $N \in [2, 14]$. More than 1000 numerical tests are carried out.

As a result, these numerical experiments lead to the following:

Numerical Fact 1 Under assumptions (i)-(viii), assume $M = 3$, $N_i = N$, $\mu_i = \mu$, $\lambda_i = \lambda$, $s_{ij} = s_{ji}$. Then in terms of total throughput of the system,

(a) $TP^{SOT} \geq TP^{SPT}$ and $TP^{LPT} \geq TP^{LOT}$.

(b) $TP^{SOT} \geq TP^{LPT}$ when N is small, and there exists a possibility that $TP^{SOT} < TP^{LPT}$ when N is large.

The rationale behind Numerical Fact 1(a) is that SOT results in selecting products with shorter setup times than SPT, and LPT leads to shorter setup times than LOT. Since reducing setup times can always lead to larger throughput (Feng et al. 2011), it is observed that SOT and LPT can lead to higher throughput.

Numerical Fact 1(b) indicates two outcomes of the scheduling policy: one is related to setup times, the other is on the queue lengths of different products. For the first one, if a policy always chooses shorter setup times, then it should lead to higher throughput. If a policy always chooses a longer queue, then the system should also have higher throughput, which is due to that choosing longer queues can reduce the frequency of setup times, and this leads to the second outcome. Comparing the SOT and LPT policy, SOT shows advantage in the first one, while LPT could be superior in the second one. Therefore, the policy comparison depends on which outcome plays a more important role. As buffer capacity grows, it is expected that the differences between queue lengths will be larger, which results in choosing a longer queue. Therefore, LPT may lead to higher throughput than SOT.

In addition, we observe that the SOT policy is inferior to LPT if the following cases occur: 1) all the products have equal setup times (i.e., $s_{ij} = s$); or 2) The setup times are all approaching zero. This is because under these two scenarios, the impact of shorter setup times can be neglected, and therefore "longer queue" becomes more critical, which highlights the advantage of LPT policy.

An example of Numerical Fact 1 is illustrated in Figure 2, where

$$M = 3, \quad N \in [2, 14], \qquad \Gamma = [3, 3, 3], \quad \Lambda = [0.9, 0.9, 0.9],$$

$$\mathcal{S} = \begin{bmatrix} 0 & 0.67 & 1.33 \\ 0.67 & 0 & 2 \\ 1.33 & 2 & 0 \end{bmatrix}.$$

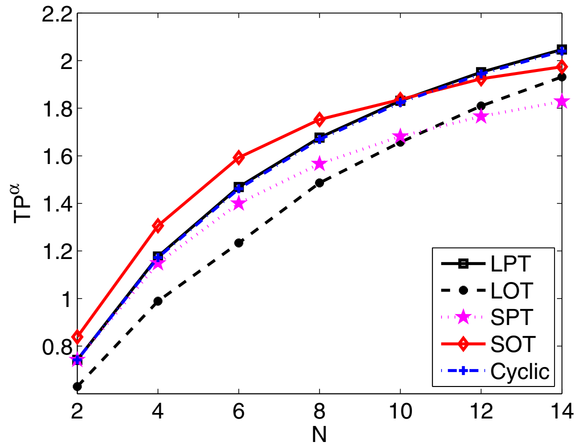

Figure 2: Throughput comparison in equal processing rates case, $M = 3$.

As one can see, SOT policy results in best performance when buffer is small and LPT policy becomes superior with large buffers. Moreover, we observe that the cyclic policy reaches relatively good performance in many scenarios. Therefore, we obtain:

Numerical Fact 2 Under assumptions (i)-(viii), assume $M = 3$, $N_i = N$, $\mu_i = \mu$, $\lambda_i = \lambda$, $s_{ij} = s_{ji}$. Then in most cases, the cyclic policy may not lead to the highest nor the lowest system throughput among all the scheduling policies.

4.1.2 More Than Three-product Case

In the cases of more than three-product, based on extensive numerical studies, we discover that Numerical Facts 1 and 2 still hold. An example of four-product system is illustrated in Figure 3, where

$$M = 4, \quad N \in [2,7], \qquad \Gamma = [1,1,1,1], \quad \Lambda = [0.1731, 0.1731, 0.1731, 0.1731],$$

$$\mathcal{S} = \begin{bmatrix} 0 & 1.4531 & 1.4797 & 1.1272 \\ 1.4531 & 0 & 1.666 & 0.7569 \\ 1.4797 & 1.666 & 0 & 2.9059 \\ 1.1272 & 0.7569 & 2.9059 & 0 \end{bmatrix}.$$

Again, SOT and LPT policies lead to best performance when buffers are small or large, respectively, and cyclic policy always results in relatively good throughput. Therefore, the following hypothesis is proposed:

Hypothesis 1 Under assumptions (i)-(viii), assume $N_i = N$, $\mu_i = \mu$, $\lambda_i = \lambda$, $s_{ij} = s_{ji}$. Then LPT or SOT policies leads to best performance of the system. In addition, LPT policy is favorable when buffer is large and SOT is superior for small buffer system. Moreover, cyclic policy often results in relatively good performance in all scenarios.

4.2 Scenarios with Non-equal Processing Rates

Now we study the case of different processing rates. Since LPT and LQ, SPT and SQ are not identical, all seven scheduling policies defined in Section 2 need to be addressed.

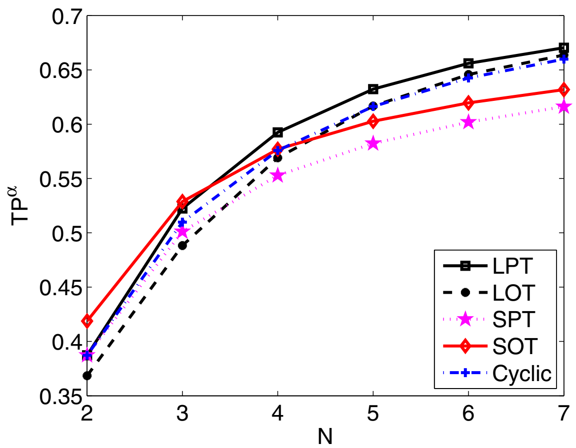

Figure 3: Throughput comparison in equal processing rates case, $M = 4$.

4.2.1 Three-product Case

Again we start with three-product case. Similar approach to Subsection 4.1 is used to construct numerical experiments, but with the processing rates chosen from $[0.3, 1.6]$. Then, through extensive numerical experiments, we discover:

Numerical Fact 3 Under assumptions (i)-(viii), assume $M = 3$, $N_i = N$, $\lambda_i = \lambda$, $s_{ij} = s_{ji}$. Then in terms of system total throughput, most likely we have

(a) $TP^{LQ} \geq TP^{LOT}$ and $TP^{LPT} \geq TP^{LOT}$.

(b) $TP^{LPT} \leq TP^{LQ}$ for large buffers, and there exists a possibility that $TP^{LPT} > TP^{LQ}$ when buffers are small.

It is observed that the conclusion of $TP^{LQ} \geq TP^{LOT}$ is true in more than 98% of the numerical tests. Even if TP^{LQ} is not superior to TP^{LOT}, the difference between them is typically less than 1%. Similar results are observed when LPT and LOT policies are compared.

Remark 1 Numerical Fact 3 not only extends the result of $TP^{LPT} \geq TP^{LOT}$ in Numerical Fact 1 to non-equal processing rates scenario, it also shows that the processing time based policy is more favorable only when buffers are small. The rationale of this is due to that the difference between the queue lengths is not significant for small buffer case, thus, the processing times, influenced by different processing rates, will be critical to system throughput. When the difference in queue length becomes more substantial, the impact of queue-based policy is more dramatic.

An example of Numerical Fact 3 is illustrated in Figure 4, where

$$M = 3, \qquad N \in [2, 14],$$
$$\Gamma = [0.3, 1, 1.6], \qquad \Lambda = [0.4235, 0.4235, 0.4235],$$
$$\mathcal{S} = \begin{bmatrix} 0 & 6 & 6 \\ 6 & 0 & 2 \\ 6 & 2 & 0 \end{bmatrix}.$$

As one can see, both LPT and LQ policies are superior to LOT policy, and LQ is more favorable when buffers are large.

Analogously, when SPT, SQ and SOT policies are compared, we observe:

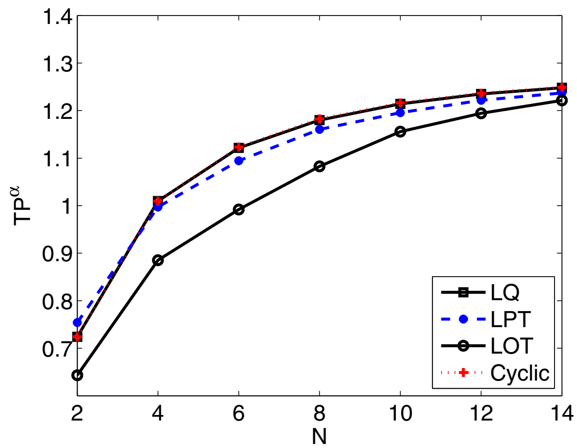

Figure 4: Throughput comparison in non-identical processing rates case under LQ, LPT, LOT, and cyclic policies, $M = 3$.

Numerical Fact 4 Under assumptions (i)-(viii), assume $M = 3, N_i = N, \lambda_i = \lambda$, $s_{ij} = s_{ji}$. Then in terms of total system throughput, practically we have

(a) $TP^{SOT} \geq TP^{SQ}$ and $TP^{SOT} \geq TP^{SPT}$.

(b) $TP^{SQ} \leq TP^{SPT}$ when buffers are large, and there exists a possibility that $TP^{SQ} > TP^{SPT}$ for small buffers.

For Numerical Fact 4, the advantage of SOT policy is due to its capability of reducing setup times. Note that large buffers are not good for SQ policy, in contrast to LQ policy in Numerical Fact 3. This is because SQ policy can be viewed as an inverse of LQ policy.

An example of Numerical Fact 4 is illustrated in Figure 5, where

$$\begin{aligned} M = 3, \qquad & N \in [2, 14], \\ \Gamma = [1.4271, 1.3353, 1.5518], \qquad & \Lambda = [0.4238, 0.4238, 0.4238], \\ \mathcal{S} \;\; = & \begin{bmatrix} 0 & 3.8236 & 0.2853 \\ 3.8236 & 0 & 0.4569 \\ 0.2853 & 0.4569 & 0 \end{bmatrix}. \end{aligned}$$

Clearly, SOT policy results in better throughput.

Similar to equal processing rates case, the cyclic policy often results in relatively good performance, in particular, when buffers are large. Thus, we obtain,

Numerical Fact 5 Under assumptions (i)-(viii), assume $M = 3$, $N_i = N$, $\lambda_i = \lambda$, $s_{ij} = s_{ji}$. Then in most cases, the cyclic policy may not result in the highest nor the lowest system throughput among all the scheduling policies.

Thus, in general, the best scheduling policy is among SOT, LPT or LQ. When comparing the SOT policy with LQ and LPT, we obtain,

Numerical Fact 6 For a multi-product system under assumptions (i)-(viii), assume $M = 3$, $N_i = N$, $\lambda_i = \lambda$, $s_{ij} = s_{ji}$. When buffers ares small, $TP^{SOT} \geq TP^{LQ}$ and $TP^{SOT} \geq TP^{LPT}$; while, there exist possibilities that $TP^{SOT} < TP^{LQ}$ and $TP^{SOT} < TP^{LPT}$ for large buffers.

Therefore, LQ tends to be the best policy among LQ, LPT and SOT when buffer capacities are increasing.

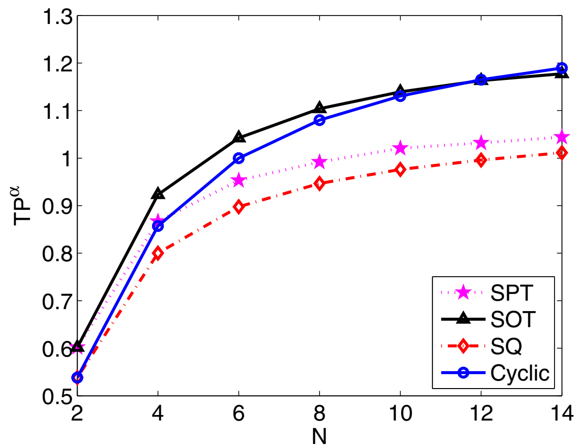

Figure 5: Throughput comparison in non-identical processing rates case under SQ, SPT, SOT, and cyclic policies, $M = 3$.

4.2.2 More Than Three-product Case

Similar conclusions in Numerical Fact 3, 4, 5 and 6 are obtained for the cases of more than three-product. An illustration of four-product case is shown in Figure 6, where

$$M = 4, \quad N \in [2,6], \qquad \Gamma = [0.6790, 0.4445, 0.9273, 0.9028], \quad \Lambda = [0.1317, 0.1317, 0.1317, 0.1317],$$

$$\mathcal{S} = \begin{bmatrix} 0 & 2.8046 & 3.6776 & 3.6066 \\ 2.8046 & 0 & 5.5128 & 3.9612 \\ 3.6776 & 5.5128 & 0 & 3.1241 \\ 3.6066 & 3.9612 & 3.1241 & 0 \end{bmatrix}.$$

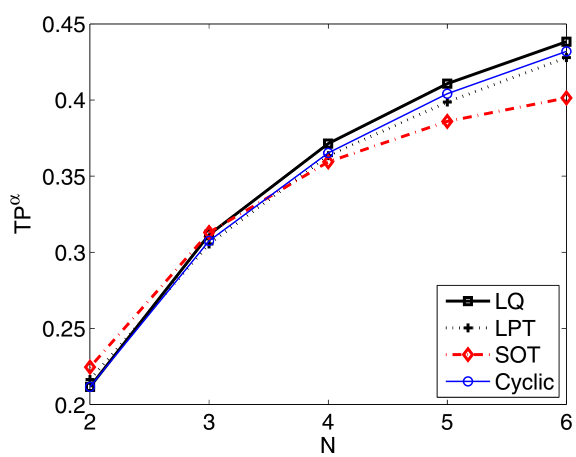

Figure 6: Throughput comparison in non-identical processing rates case, $M = 4$.

To summarize, the following hypothesis is proposed:

Hypothesis 2 Under assumptions (i)-(viii), assume $N_i = N$, $\mu_i = \mu$, $\lambda_i = \lambda$, $s_{ij} = s_{ji}$, Then, in most cases, the SOT policy leads to better throughput than SQ and SPT policies, and LQ and LPT policies exceeds

LOT policy. Moreover, LQ policy is favorable for large buffer case. In addition, the cyclic policy typically leads to relatively good performance in all scenarios.

5 CONCLUSIONS

In this paper, continuous time Markov chain models are developed for multi-product manufacturing systems with sequence-dependent setups, finite buffers, and seven scheduling policies. The impacts of each scheduling policy on system throughput are investigated. It is observed that when all products have equal processing rates, the best policy is either LPT or SOT, while LPT policy is more favorable in large buffer scenario. When the products have different processing rates, the best policy is chosen from LQ, LPT or SOT, and LQ policy is desirable when buffers are large. Moreover, the cyclic policy results in relatively good performance in all scenarios. Such insights are helpful for shop floor management in multi-product manufacturing factories.

ACKNOWLEDGMENTS

This work is supported in part by NSF Grant No. CMMI-1063656 and the National Natural Science Foundation of China (Grant No. 70771058/60834004), the 863 Program of China (2008AA04Z102). In addition, the authors are grateful to Mr. F. Hou of Intel China for valuable discussions.

APPENDIX

Proof of Theorem 1: Let q_{S_1,S_2} denote the transition rate from state S_1 to state S_2. The transition equations under cyclic policy has been presented in (Feng et al. 2011). For other scheduling policies, the transition equations are identical except the following one:

- When the last part in the buffer of type j is being processed, the machine will transit to the setup state of next product with non-empty buffer with rate

$$q_{(i,j,1,h_1,\dots h_j=1,\dots,h_M),(j,l,0,h_1,\dots h_j=0,\dots,h_M)} = \mu_j,$$

In this equation, the relationship between the two states depends on the scheduling policy:

(1) *Cyclic policy:* $i=1,\dots,M,\ j=i+1,\dots,M+i-1,\ l=j+1,\dots,M+j-1,\ h_l=1,\dots,N_l-1,$ $h_{j+1}=\dots=h_{l-1}=0$, and product l has the next non-empty queue after product j in the cyclic order.

(2) *LQ policy*: $i=1,\dots,M,\ j=i+1,\dots,M+i-1,\ l=j+1,\dots,M+j-1,\ h_l=1,\dots,N_l-1,\ h_l > h_m(m=j+1,\dots,l-1), h_l \geq h_k(k=l+1,\dots,j)$.

(3) *LPT policy*: $i=1,\dots,M,\ j=i+1,\dots,M+i-1,\ l=j+1,\dots,M+j-1,\ h_l=1,\dots,N_l-1,$ $\frac{h_l}{\mu_l} > \frac{h_m}{\mu_m}(m=j+1,\dots,l-1),\ \frac{h_l}{\mu_l} \geq \frac{h_k}{\mu_k}(k=l+1,\dots,j)$.

(4) *LOT policy*: $i=1,\dots,M,\ j=i+1,\dots,M+i-1,\ l=j+1,\dots,M+j-1,\ h_l=1,\dots,N_l-1,\ h_m=0$ or $\frac{h_l}{\mu_l}+s_{jl} > \frac{h_m}{\mu_m}+s_{jm}(m=j+1,\dots,l-1)$, $h_k=0$ or $\frac{h_l}{\mu_l}+s_{jl} \geq \frac{h_k}{\mu_k}+s_{kl}(k=l+1,\dots,j)$.

(5) *SQ policy*: $i=1,\dots,M,\ j=i+1,\dots,M+i-1,\ l=j+1,\dots,M+j-1,\ h_l=1,\dots,N_l-1,\ h_m=0$ or $h_l < h_m(m=j+1,\dots,l-1)$, $h_k=0$ or $h_l \leq h_k(k=l+1,\dots,j)$.

(6) *SPT policy*: $i=1,\dots,M,\ j=i+1,\dots,M+i-1,\ l=j+1,\dots,M+j-1,\ h_l=1,\dots,N_l-1,\ h_m=0$ or $\frac{h_l}{\mu_l} < \frac{h_m}{\mu_m}(m=j+1,\dots,l-1)$, $h_k=0$ or $\frac{h_l}{\mu_l} \leq \frac{h_k}{\mu_k}(k=l+1,\dots,j)$.

(7) *SOT policy*: $i=1,\dots,M,\ j=i+1,\dots,M+i-1,\ l=j+1,\dots,M+j-1,\ h_l=1,\dots,N_l-1,\ h_m=0$ or $\frac{h_l}{\mu_l}+s_{jl} < \frac{h_m}{\mu_m}+s_{jm}(m=j+1,\dots,l-1)$, $h_k=0$ or $\frac{h_l}{\mu_l}+s_{jl} \leq \frac{h_k}{\mu_k}+s_{kl}(k=l+1,\dots,j)$.

The remaining proof is similar to that of (Feng et al. 2011). Due to space limitation, the details of the proof are omitted. ■

REFERENCES

Altiok T., and G. A. Shiue. 2000. "Pull-type Manufacturing Systems with Multiple Product Types." *IIE Transactions*, 32: 115-124.

Beach R., A. P. Juhlemann, D. H. R. Price, A. Paterson, and J. A. Sharp. 2000. "A Review of Manufacturing Flexibility." *European Journal of Operational Research*, 122: 41-57.

Buzacott J. A., and J. G. Shantikumar. 1993. *Stochastic Models of Manufacturing Systems*. Englewood Cliffs, NJ: Prentice-Hall.

Buzacott J. A., and D. D. Yao. 1986. "Flexible Manufacturing Systems: A Review of Analytical Models." *Management Science*, 32: 890-905.

Dasci A., and M. Karakul. 2008. "Performance Evaluation of a Single-stage Two-product Manufacturing System Operating under Pull-type Control." *Computers and Operations Research*, 35: 2861-2876.

Feng W., L. Zheng, and J. Li. 2011. "Multi-Product Manufacturing Systems with Sequence-Dependent Setups: Performance Evaluation and System Properties." submitted to *IEEE Transactions on Systems, Man and Cybernetics - Part A*.

Gurgur C. Z., and T. Altiok. 2008. "Decentralized Multi-product Multi-stage Systems with Backorders." *IIE Transactions* 40: 238-251.

Jang Y. J. 2007. *Mathematical Modeling and Analysis of Flexible Production Lines*. Ph.D. thesis, Department of Mechanical Engineering, MIT, Cambridge, MA.

Krieg G. N., and H. Kuhn. 2002. "A Decomposition Method for Multi-product Kanban Systems with Setup Times and Lost Sales." *IIE Transactions*, 34: 613-625.

Krieg G. N., and H. Kuhn. 2004. "Analysis of Multi-product Kanban Systems with State-dependent Setups and Lost Sales." *Annals of Operations Research*, 125: 141-166.

Li J., and N. Huang. 2005. "Modelling and analysis of a multiple product manufacturing system with split and merge." *International Journal of Production Research*, 43: 4049-4066.

Li J., and N. Huang. 2007. "Quality Evaluation in Flexible Manufacturing Systems: A Markovian Approach." *Mathematical Problems in Engineering*, article ID 57128.

Ryan S. M. and J. Vorasayan. 2005. "Allocating Work in Process in a Multiple-product CONWIP System with Lost Sales." *International Journal of Production Research*, 43: 223-246.

Satyam K. 2007. *Analytical Models for Multi-product Systems with Pull Type Material Control Strategies*. Ph.D. thesis, Rensselaer Polytechnic Institute, Troy, NY.

Satyam K., and A. Krishnamurthy. 2008. "Performance Evaluation of a Multi-product System under CONWIP Control." *IIE Transactions*, 40: 252-264.

Sethi A. K., and P. S. Sethi. 1990. "Flexibility in Manufacturing: A Survey." *International Journal of Flexible Manufacturing Systems*, 2: 289-328.

Suri R. 1985. "An Overview of Evaluative Models for Flexible Manufacturing Systems." *Annals of Operations Research*, 3: 13-21.

Takagi H. 2000. "Analysis and Application of Polling Models." In *Performance Evaluation: Origins and Directions*, 423-442. Berlin, Germany: Springer-Verlag.

Tempelmeier H., and H. Kuhn. 1993. *Flexible Manufacturing Systems: Decision Support for Design and Operation*. New York: John Wiley & Sons.

Tubilla F. and S. B. Gershwin. 2009. "Real-Time Scheduling Policy for Manufacturing Systems with Sequence-Dependent Setups." In *Proc. of 7th Conf. on Stochastic Modeling of Manuf. and Service Operations (SMMSO 2009)*, 184-191, Ostuni, Italy.

Viswanadham N., and Y. Narahari. 1992. *Performance Modeling of Automated Manufacturing System*. Englewood Cliffs, NJ: Prentice-Hall.

Wang J., J. Li, J. Arinez and S. Biller. 2010. "Product Sequencing with respect to Quality in Flexible Manufacturing Systems with Batch Operations", *IEEE Transactions on Automation Science and Engineering*, 7: 776-790.

Zhou M., and K. Venkatesh. 1999. *Modeling, Simulation and Control of Flexible Manufacturing Systems: A Petri Net Approach*. Singapore: World Scientific.

AUTHOR BIOGRAPHIES

WEI FENG is a Ph.D. student in Department of Industrial Engineering, Tsinghua University, Beijing, China. He is also a visiting student in Department of Industrial and Systems Engineering, University of Wisconsin - Madison from 2010 to 2011. His research interests are in modeling, analysis and improvement of production systems. His e-mail address is wei.feng1@rwth-aachen.de.

LI ZHENG is a Professor in Department of Industrial Engineering, Tsinghua University, Beijing, China. He was a visiting professor at Georgia Institute of Technology from 1994 to 1996. He received his Ph.D. degree from Tsinghua University in 1991. His research interests include production and operation planning and scheduling, production system analysis, and information driven manufacturing. His e-mail is lzheng@mail.tsinghua.edu.cn.

JINGSHAN LI is an Associate Professor in Department of Industrial and Systems Engineering, University of Wisconsin, Madison, WI. He received his Ph.D. from the University of Michigan, Ann Arbor, in 2000, and was a Staff Researcher in General Motors Research and Development from 2000 to 2006, and an Assistant Professor in University of Kentucky from 2006 to 2010. His primary research interests are in modeling, analysis, control and continuous improvement of manufacturing and health care systems. His email address is jingshan@engr.wisc.edu.