Proceedings of the 2016 Winter Simulation Conference
T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, eds.

MEAN QUEUE TIME APPROXIMATION FOR A WORKSTATION WITH CASCADING

Kan Wu

School of MAE
Nanyang Technological University
102 Nanyang Crescent
Singapore 637820, SINGAPORE

Ning Zhao

Faculty of Science
Kunming University of Science and Technology
68 Wenchang Road
Kunming 650093, CHINA

ABSTRACT

Queueing models can be used to evaluate the performance of manufacturing systems. Due to the emergence of cluster tools in contemporary production systems, proper queueing models have to be derived to evaluate the performance of machines with complex configurations. Job cascading is a common structure among cluster tools. Because of the blocking and starvation effects among servers, queue time analysis for a cluster tool with job cascading is difficult in general. Based on the insight from the reduction method, we proposed the approximate model for the mean queue time of a cascading machine subject to breakdowns. The model is validated by simulation and performs well in the examined cases.

1 INTRODUCTION

The development of queueing theory can be traced back to A.K. Erlang (1909) for the performance evaluation of telecommunication networks. In the mid-twentieth century, researchers (White and Christie 1958; Gaver Jr 1962) applied it to evaluate the performance of manufacturing systems. However, a manufacturing system is much different from a telecommunication network in terms of interruption types, batching, setups and tool configurations, etc. In this paper, we study the performance of machines with complex configurations. Specifically, we propose closed form approximations for the mean queue time of a cluster tool with cascading.

In manufacturing systems, the configuration of a workstation can be more complex than a server (i.e., a customer representative) in telecommunication systems. For example, in a pharmaceutical secondary manufacturing plant, the tablets are made from active pharmaceutical ingredients through dispensing, blending, granulation, compression, and coating. To reduce waiting or transportation time between consecutive steps, all process steps can be done in a single cascading machine rather than by the individual ones. In the food industry, dispensing of a tray meal is a combination of several consecutive steps at a workstation. The workers dispense food items sequentially according to pre-specified weight limits.

In a semiconductor fabrication facility (fab), machines commonly have complex configurations and almost every machine in a fab has multiple load ports. For example, a wet bench is composed of a series of tanks. When a lot, which carries multiple wafers, has to be processed by a wet bench, it starts from setting recipes on a load port and then processed by required tanks sequentially according to the pre-defined recipe (Hyun-Jung et al. 2014).

In contract with the over simplified assumptions in most queueing models, a photo lithographer in a semiconductor fab is the combinations of scanners and tracks, where a track consists of a series of processing chambers (i.e., coaters and developers). A dry etcher has two load ports, a transferring load lock, and four processing chambers. A lot is transferred to one of the load ports dedicated to incoming lot.

978-1-5090-4486-3/16/$31.00 ©2016 IEEE

After pumping down, wafers are transferred inside the dry etcher via a dual-arm robot to one of the processing chambers. The process repeats until all wafers are processed and transferred to the output load port.

A flexible machine in contemporary production systems generally has complex configurations. In above examples, the process chamber (or tank) is commonly called a station of a cluster tool and performs a specific recipe of a process step.

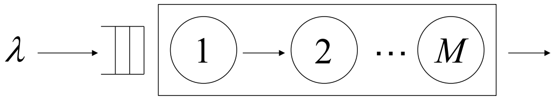

Figure 1: An ideal cascading machine with *M* servers in series.

Due to the limited buffer space, exact queue time analysis for workstations with complex configuration is hard in general (Wu and Zhao 2015a). The analysis may also depend on robot scheduling and processing sequences of each specific job. On the other hand, among the cluster tools with job cascading (or cascading machines), an underlying structure among them is cascading. An ideal cascading machine refers to a workstation, which is composed of multiple single server stations in series (as shown in Figure 1) and no setup before job processing, e.g. a wet bench. Hence, more than one job can be processed by a cascading machine at different process steps simultaneously, where the process step is the service provided by a single server station (or a server) of the ideal cascading machine. Since most tools in the semiconductor manufacturing have load ports, ideal cascading machines are commonly seen in fabs, especially for the tools with simple configurations, such as sorters, which consist of two load ports (one for load and the other for unload) and a sorting station. In this paper, we propose analytical models to approximate system mean queue time of an ideal cascading machine.

Due to the internal blocking and starvation among servers, the exact analysis of cascading machines is difficult except for some special cases. The ideal cascading machine can be modeled as a tandem queue with no buffer in between. When service times of all servers are phase-type distributed or constant, we can analyze system performance exactly. Avi-Itzhak (1965) proved that the departure process from the tandem queue with constant service times and finite buffer capacity is independent of the order of the servers, and Friedman (1965) found that its queue time is determined solely by the bottleneck. Latouche and Neuts (1980) studied exponential tandem queues with finite buffers and showed that the steady-state probability vectors are of matrix-geometric form. Gómez-Corral (2004) derived the sojourn time distribution of two-node tandem queues with finite buffers, Markovian arrival process and phase type service time. Seo and Lee (2011) considered a stationary waiting time in a Poisson driven single-server m-node tandem queue with either constant or nonoverlapping service times. By using (max,+)-algebra, they explicitly expressed the stationary waiting time at each node. Though the matrix-geometric method and (max,+)-algebra are applicable in the above cases, the algorithms are complex and time consuming.

Since even the analysis for two single servers in series is hard (Wu and McGinnis 2013; Wu and Zhao 2015b), the analysis of cascading machines with generally distributed service times is difficult. It is even harder when the servers are subject to interruptions, such as breakdowns or preventive maintenances (PM). When breakdowns exist, most literature approximated the throughput and mean sojourn time by assuming that there is no starvation at the first server of a tandem queue. The prevalent approximation methods are based on the aggregation or decomposition approach (Lim et al. 1990; Dallery and Gershwin 1992; Li et al. 2006; Li et al. 2009).

The analysis of cluster tools has been studied for over a decade. Robot scheduling is essential to this problem and has been studied (Venkatesh et al. 1997; Rostami and Hamidzadeh 2002; Kim et al. 2003). Due to the complexity, simulation studies have been conducted (Mauer and Schelasin 1994; LeBaron and Hendrickson 2000). Perkinson et al. (1994), Wood (1996) and Morrison and Martin (2007) studied system throughput through rough approximate models without differentiating the queue time impacts from different types of interruptions, and the differences between service time and process time. Niedermayer and Rose (Niedermayer and Rose 2003) explained the importance of cluster tools in semiconductor manufacturing and studied their cycle time. They pointed out that the cycle time analysis can only be done by simulation until then. When simulation is used, the simulation optimization technique can be employed (Xu et al. 2015).

To analyze the queue time performance of a cascading machine, we decompose a cascading machine into the bottleneck and non-bottleneck, and then approximate the system queue time based on the reduction method (1965) and its error bounds. Through the identified underlying structure, a closed form model is proposed to approximate the mean queue time of a cascading machine with interruptions.

In the following, we first give definitions and assumptions in Section 2. In Section 3, the mean queue time approximate model is proposed. Simulation validation is given in Section 4. We conclude this paper in Section 5.

2 DEFINITION AND ASSUMPTION

To derive the approximate model, we have to define some terminologies first. The bottleneck server in a cascading machine refers to its throughput bottleneck, which is also the server with the longest service time in an ideal cascading machine (Wu 2005).

In queueing theory, a busy period is the duration of a server between two consecutive idle periods. It is also the duration when a server is continuously running, or in *continuous run mode*. However, when a machine is composed of multiple servers in series (i.e., a cascading machine), continuous run mode cannot guarantee that its bottleneck server is always busy, since the machine is considered as in service, as long as, at least, one of the servers is busy. Hence, there are two levels of continuous run mode: continuous run mode at the machine level and continuous run mode at the bottleneck level, where the previous one only assures that at least one of the servers of the cascading machine is busy, which may not be the bottleneck server.

Clarifying the concept of service time is of fundamental importance in the application of queueing theory. In manufacturing systems, the service time of a cascading machine is closely related to the concept of takt time. Takt time is a notion employed in continuous productive operations. It is defined as 'the rate that a completed product needs to be finished in order to meet customer demand' (iSixSigma 2012), or 'the desired time between units of production output, synchronized to customer demand' (Strategos 2012). Applications of takt time were reported (Labanowski 1997). Rather than using process time, Wu and Hui (2008) found the concept of takt time is closer to the definition of service time in queue theory. However, it fails to tell the impact from non-preemptive interruptions, which should not be counted into service time (Wu 2014). Hence, a more complete and general definition for service time is needed.

In queueing theory, the concept of service time is derived from capacity. And service time is indeed the inverse of capacity, where capacity is the maximum throughput rate in steady state. An important generalization of service time is the *generalized service time* (*G*), which reflects the capacity of a server under the influences of interruptions (Wu et al. 2011):

$$G = \textit{Job departure time} - \textit{The time epoch when the job first claims capacity of the server,} \tag{1}$$

where job departure time is the time a job release the server capacity. A job claims capacity of a server if

1. the job is present at the server,
2. the preceding job has released server capacity,
3. the server is ready to process this job, or is ready to perform a product-induced setup,

where a product-induced setup occurs due to changes in the production process induced by switching products and it cannot be done in parallel with job processing. Although a product-induced setup consumes capacity of a server, it may not be true for a cascading machine. At a cascading machine, if setups can be done in parallel with the pre-bottleneck process steps and the setup time is shorter than the bottleneck service time, the product-induced setups will not consume system capacity in this situation. Hence, at a cascading machine, a job claims machine capacity if

1. the job is present at the machine,
2. the preceding job has released machine capacity,
3. the machine is ready to process this job.

Based on the above definition, generalized service time of a cascading machine is

$$G = S + \sum_{i=1}^{N(S)} D_i, \tag{2}$$

where S is service time, $N(S)$ is the number of preemptive interruptions during S, and D_i is the i-th downtime (Wu 2014).

The following assumptions are made in the derivation of approximate models.

1. The service times and exogenous inter-arrival times are independent and identically distributed (iid) sequences, and both are mutually independent,
2. There is an unique bottleneck server in any cascading machine, and all jobs go through this bottleneck,
3. The cascading machine suffers time-based preemptive-resume breakdowns (Wu 2014). When any of the servers suffers breakdowns, the cascading machine will switch to a non-production mode and all servers have to stop processing immediately (if they are processing jobs). A preempted job (if any) will resume its remaining process upon recovery.
4. Transferring times between consecutive process steps are negligible.
5. The service time at the bottleneck server is strictly longer than the service time at any other server.
6. No breakdown occurs during the recovery of a breakdown.

The first one is a common assumption in queueing models and should not be strict in practice. However, the second assumption may not hold if the cascading machine can process multiple products. In this situation, the presented model has to be modified. In this initial attempt to model a cascading machine with interruptions, we start with analyzing the impact from time-based breakdowns. For other types of interruptions, the queueing models have to be derived separately (Wu 2014).

Since the transferring times between consecutive process steps are relatively shorter than the process time in general, we ignore the transferring times in the analysis. In practical manufacturing systems, to satisfy the requirements of quality and product reliability, service times are often required to be as regular as possible in order to meet the tight specifications. Hence, service time variability is usually small (Inman 1999). Hence, service times at different servers in a cascading machine can be assumed to be nonoverlapping, i.e., if $i \neq j$, $P(S_j \geq S_i) = 1$ (or $P(S_j \leq S_i) = 1$) for all i and j, where S_i and S_j are service time of server i and j respectively. This justifies the 5th assumption.

3 THE APPROXIMATE MODEL

Figure 2 demonstrates an ideal cascading machine subject to time-based preemptive resume breakdowns.

Assume the job mean arrival rate is λ. The service time at server i is S_i ($i = 1, \cdots, M$). D is the duration of the breakdown. η is the machine failure rate (and $1/\eta$ is the mean time to failure after repair). The availability of the system is $A = \frac{1}{1+\eta E(D)}$. The buffer capacity of the first buffer is infinite. The intermediate buffer capacity is finite. The buffers are modeled as servers with zero service times.

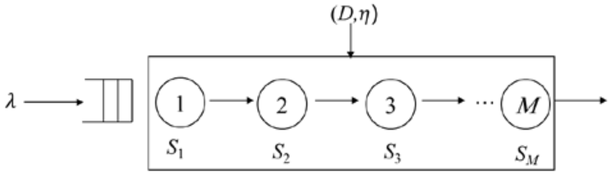

Figure 2: An ideal cascading machine with breakdowns.

When all service times are constant, Avi-Itzhak (1965) and Friedman (1965) found that the system queue time is determined solely by the bottleneck, and is the same as the queue time would be if the bottleneck sees the initial arrival process directly. Since a tandem queue can be reduced to a single server system, it was named reduction method by Friedman (1965). Furthermore, Avi-Itzhak (1965) showed that the reduction method is insensitive to the buffer size. Hence, when all service times are constant, an ideal cascading machine can be analyzed exactly by simply assuming it is equivalent to a single server, which is the bottleneck of the cascading machine.

Tembe and Wolff (1974) extended Friedman's work to tandem queues with nonoverlapping service times. They proved that if the bottleneck is the first server in a tandem queue, its cycle time (i.e., queue time plus service time) is the shortest among all arrangements. Wan and Wolff (1993) further showed that the largest difference among the cycle times of different arrangements of the tandem queues with nonoverlapping service time is the upper bound of the second-longest service time among those of all servers. Wu and Zhao (2015a) gave a tighter bound for tandem queues with nonoverlapping service time.

To serve the needs of practical production lines, prior results have to be extended to the situations with breakdowns. Let T_1 = (1,…, M) and T_2= ([1], …, [M]) be two arbitrary arrangements of servers for an ideal cascading machine with M servers. Assume the longest, the second-longest and the least service time are at server a, b and c, respectively. Denote the cycle time in system T_i by CT^{T_i}, $i = 1, 2$.

Theorem 1 (Bounds for ideal cascading machines with nonoverlapping service times subject to breakdowns)
Let the service times S_i ($i = 1, \cdots, M$) of an ideal cascading machine with time-based preemptive-resume breakdowns be nonoverlapping. If the random duration of the breakdown is D and the machine failure rate is η, then

(1) If $E(CT^{T_1})$ and $E(CT^{T_2})$ exist and are finite, $|E(CT^{T_1}) - E(CT^{T_2})| \leq (\sup(S_b) - \inf(S_c))(1 + \eta E(D))$.
(2) $\lim_{\rho \to 1} \left| E(CT^{T_i}) - E\left(CT^{T_j}\right) \right| / E(CT^{T_i}) \to 0, i, j = 1, 2,$
where ρ is the utilization contributed by jobs at server a and $\rho = \lambda E(S_a)$.

Please see Appendix for the proof. Based on Theorem 1-(1), the bound is tighter if the upper limit of the second-longest service time is smaller or the lower limit of the least service time is greater. Furthermore, the bound will be tighter if machine failure rate and the mean time to repair are smaller. A nice property of the nonoverlapping service time cascading machine is its behavior in heavy traffics. The relative difference between the mean queue times of any two permutations converges to zero in heavy traffic.

An important observation is that in what conditions a cascading machine gives the shortest cycle time among all permutations. If the bottleneck is the first server, the queue time only occurs at the first server and the queue times at all non-bottleneck servers are zero, since the inter-departure times at the bottleneck are greater than the service times at the non-bottleneck servers. This coincides with the reduction method, where the system queue time is determined solely by the bottleneck. Hence, when all service times are constant, the cascading machine also gives the shortest cycle time. The lower bound of an ideal cascading machine occurs when its first server is the bottleneck or all service times are constant. In this situation, the system queue time is simply determined by its bottleneck.

Based on the above analysis, the lower bound of the queue time of an ideal cascading machine is the mean queue time of its bottleneck. Wu (2014) gave the mean cycle time approximation for a single server subject to time-based preemptive resume breakdowns. Since the cycle time of an ideal cascading machine includes its queue time, bottleneck service time and non-bottleneck service times, the lower bound of mean cycle time $E(CT_L)$ can be approximated as follows.

$$E(CT_L) = \frac{\rho_G E(R_G)}{(1-\rho_G)} + (1 - A_{NP})E(R_D) + E(G) + E(G'), \tag{3}$$

where $\rho_G = \lambda E(G)$, $E(R_G) = E(G^2)/2E(G)$, $E(G) = E(S_a)/A$, $E(G^2) = E(S_a^2)[1 + \eta E(D)]^2 + E(S_a)\eta E(D^2)$, $E(R_D) = E(D^2)/2E(D) = \frac{1+c_D^2}{2}E(D)$, $A_{NP} = 1/(1 + \eta E(D))$, and $E(G') = \sum_{i=1,i\neq a}^{M} E(S_i)/A$.

Based on the lower bound and Theorem 1, we can also get the upper bound of the mean cycle time $E(CT_U)$ by Eq. (4).

$$E(CT_U) = E(CT_L) + (\sup(S_b) - \inf(S_c)) * (1 + \eta E(D)). \tag{4}$$

Hence, we can simply approximate the mean cycle time $E(CT)$ of an ideal cascading machine by taking the average of the lower and upper bounds.

$$E(CT) \cong \frac{E(CT_L) + E(CT_U)}{2} = E(CT_L) + (\sup(S_b) - \inf(S_c))(1 + \eta E(D))/2. \tag{5}$$

The above model should give reliable approximations when the bound is tight. This assumption should not be strict due to the requirements of quality and product reliability in practical manufacturing systems.

4 SIMULATION VALIDATION

In this section the mean cycle time approximation of Eq. (5) is validated by simulations. In Section 4.1, we examine the approximate model in an ideal situation where the cascading machine has nonoverlapping service times. In Section 4.2, we examine the model under more practical settings where the cascading machine has overlapping service times and product-induced setups.

4.1 Cascading Machines with Nonoverlapping Service Times

In the following, two cases are examined. The cascading machine in each case has five single servers in tandem. The settings are the same in both except for the service time distributions. Assume that both arrivals follow Poisson distributions. Uptimes between two consecutive breakdowns are exponentially distributed with $1 \,/\, \eta$ =72 hours. Downtimes follow Gamma distributions with mean 6 hours and squared coefficient of variation (SCV) 0.5, i.e., $D\sim$ Gamma(2,3).

Case 1. In the first case, service times at each server of the cascading machine follow triangular distributions, and the probability density function of S_i $(i = 1, \cdots, 5)$ is

$$f(x|a_i, b_i, c_i) = \begin{cases} \dfrac{2(x - a_i)}{(b_i - a_i)(c_i - a_i)}, & a_i < x \le c_i, \\ \dfrac{2(b_i - x)}{(b_i - a_i)(b_i - c_i)}, & c_i < x \le b_i, \\ 0, & others, \end{cases}$$

where $a_1 = 14,\ b_1 = 16,\ c_1 = 15,\ a_2 = 23,\ b_2 = 27,\ c_2 = 25,\ a_3 = 28,\ b_3 = 32,\ c_3 = 30,\ a_4 = 18,\ b_4 = 22,\ c_4 = 20,\ a_5 = 9, b_5 = 11$ and $c_5 = 10$.

The model is validated at 10 different utilizations (ρ_G ranges from 0.1 to 0.95). Thirty replications are conducted at each arrival rate. Each replication consists of 2,000,000 jobs after discarding the first 4,000,000 jobs for warm-up.

Table 1 compares the simulation cycle time (SCT) with approximate cycle time (ACT). The half-width of the 95% confidence intervals of SCT is given after the mean. The sample size is sufficiently large so that the half width of 95% confidence intervals of the mean simulation cycle time (SCT) is less than 1%. The percentage difference between ACT and SCT (i.e., ACT/SCT – 1) is given in “Diff%”. It shows that Eq. (5) overestimates the mean cycle time across all utilizations. Diff% is the largest in light traffic and decreases as ρ_G increases. The approximate error is 3.07% at 95% utilization.

Table 1: Cycle time comparison for the cascading machine in Case 1.

ρ_G	SCT	ACT	Diff%
10%	110.528 ± 0.003	120.275	8.82%
20%	112.833 ± 0.006	122.582	8.64%
30%	115.805 ± 0.008	125.547	8.41%
40%	119.764 ± 0.012	129.502	8.13%
50%	125.297 ± 0.017	135.038	7.77%
60%	133.613 ± 0.033	143.342	7.28%
70%	147.340 ± 0.041	157.182	6.68%
80%	174.710 ± 0.117	184.862	5.81%
90%	256.660 ± 0.445	267.903	4.38%
95%	421.066 ± 2.527	433.985	3.07%

Case 2. In the second case, service times at each server of the cascading machine follow uniform distributions. The probability density function of S_i $(i = 1, \cdots, 5)$ is

$$f(x|\alpha_i, \beta_i) = \begin{cases} \dfrac{1}{\beta_i - \alpha_i}, & \alpha_i < x \le \beta_i, \\ 0, & others, \end{cases}$$

where $\alpha_1 = 14,\ \beta_1 = 16, \alpha_2 = 23,\ \beta_2 = 27,\ \alpha_3 = 28,\ \beta_3 = 32,\ \alpha_4 = 18,\ \beta_4 = 22,\ \alpha_5 = 9, \beta_5 = 11.$

Table 2: Cycle time comparison for the cascading machine in Case 2.

ρ_G	SCT	ACT	Diff%
10%	110.533 ± 0.004	120.276	8.81%
20%	112.845 ± 0.004	122.585	8.63%
30%	115.824 ± 0.006	125.552	8.40%
40%	119.782 ± 0.009	129.510	8.12%
50%	125.327 ± 0.015	135.050	7.76%
60%	133.623 ± 0.028	143.360	7.29%
70%	147.424 ± 0.052	157.210	6.64%
80%	174.702 ± 0.134	184.910	5.84%
90%	256.853 ± 0.590	268.012	4.34%
95%	420.570 ± 2.373	434.214	3.24%

Note that the mean service times of S_i (i=1, ⋯, 5) of Case 1 and 2 are equal. The model is validated at 10 different utilizations as shown in Table 2. The Diff% is similar to that of Case 1. The approximate error is relatively small in heavy traffic. The approximate model performs well in heavy traffic in both Cases 1 and 2.

4.2 Cascading Machines with Overlapping Service Times

It is important to know if the model still performs well in practical manufacturing systems when the cascading machine faces overlapping service times and setups. In the following two cases, we are going to validate the model under those situations. The problem settings are similar to the ones in Section 4.1 except for some minor adjustments to consider the overlapping service times and product-induced setups (i.e., changeovers).

Case 3. In the third case, service times at each server follow triangular distributions. The parameters are as follows: $a_1 = 14,\ b_1 = 16,\ c_1 = 15,\ a_2 = 21,\ b_2 = 29,\ c_2 = 25,\ a_3 = 28,\ b_3 = 32,\ c_3 = 30,$ $a_4 = 18,\ b_4 = 22,\ c_4 = 20,\ a_5 = 9,\ b_5 = 11,$ and $c_5 = 10$. When a new job is loaded, it may face a product-induced setup with constant duration 10 and probability 5%.

The mean service time of S_i is the same as that in Case 1 ($i = 1, \cdots, 5$). However, in Case 3, S_2 and S_3, and S_2 and S_4 are overlapping. The model is validated at 10 utilizations as shown in Table 3. Comparing Table 1 and Table 3, although the Diff% in Table 3 is also decreasing in ρ_G, the approximation error in Case 3 is a little bigger than that in Table 1.

Table 3: Cycle time comparison for the cascading machine in Case 3.

ρ_G	SCT	ACT	Diff%
10%	111.093 ± 0.004	126.775	14.12%
20%	113.425 ± 0.005	129.082	13.80%
30%	116.422 ± 0.008	132.047	13.42%
40%	120.406 ± 0.010	136.002	12.95%
50%	125.965 ± 0.015	141.538	12.36%
60%	134.372 ± 0.025	149.842	11.51%
70%	148.173 ± 0.047	163.682	10.47%
80%	175.660 ± 0.126	191.362	8.94%
90%	258.401 ± 0.606	274.403	6.19%
95%	425.668 ± 2.297	440.485	3.48%

Case 4. In the fourth case, service times at each server follow uniform distributions. Let $\alpha_1 = 14$, $\beta_1 = 16$, $\alpha_2 = 21$, $\beta_2 = 29$, $\alpha_3 = 28$, $\beta_3 = 32$, $\alpha_4 = 18$, $\beta_4 = 22$, $\alpha_5 = 9$, and $\beta_5 = 11$. When a new job is loaded, it may face a product-induced setup with constant duration 10 and probability 5%.

The mean service time of S_i is the same as that in Case 2 ($i = 1, \cdots, 5$). However, in Case 4, S_2 and S_3, and S_2 and S_4 are overlapping. The model is validated at 10 utilizations as shown in Table 4. Although the Diff% in Table 4 is also decreasing in ρ_G, the approximation error in Table 4 is a little bigger than that in Table 2.

Although there are overlapping service times and product-induced setups, the approximate model still performs well in heavy traffic in both Cases 3 and 4.

Table 4: Cycle time comparison for the cascading machine in Case 4.

ρ_G	SCT	ACT	Diff%
10%	111.102 ± 0.004	126.776	14.11%
20%	113.453 ± 0.005	129.085	13.78%
30%	116.461 ± 0.008	132.052	13.39%
40%	120.459 ± 0.010	136.010	12.91%
50%	126.038 ± 0.014	141.550	12.31%
60%	134.381 ± 0.027	149.860	11.52%
70%	148.227 ± 0.045	163.710	10.45%
80%	175.723 ± 0.101	191.410	8.93%
90%	258.154 ± 0.580	274.512	6.34%
95%	424.883 ± 2.235	440.714	3.73%

5 CONCLUSION

Based on the insight from the reduction method, we developed an approximate model for the mean queue time of a cascading machine. The approximate model performs well in heavy traffic. Due to the importance of cluster tools in flexible machine systems, the proposed model plays a critical role in the design and performance evaluation of contemporary production systems.

In practice, each station of a cluster tool may consist of more than one server. While we focus on analyzing the ideal situations in this paper, the complications are left for future research.

Inside a cascading machine, a job is commonly transferred from a server to another server by robots. The transferring time depends on the robot scheduling. We have seen many cases that a cascading machine suffers unnecessary delay due to inferior robot scheduling policies, which not only introduces variability into service times, but also increases the complexity of queue time analysis (Wu et al. 2016). Optimal robot scheduling is essential for both productivity enhancement and performance evaluation, and should be an important topic for future research.

In practice, a machine may be capable to process multiple products with a complex product mix. Under different product mixes, the bottleneck server in a cascading machine may shift. Similar impacts can come from service time variations. If the bottleneck service time is not strictly longer than the non-bottleneck service times, the bottleneck server will shift from time to time. These will inevitably increase the complexity of queue time analysis.

In an ideal cascading machine, we assume there is a single server per station. However, to synchronize the production speed at different stations, the station with lower throughput rate usually has multiple parallel servers inside a cascading machine. This situation is not covered in this paper, and will be left for future research.

ACKNOWLEDGMENTS

This research is supported in part by the GSK-Singapore Partnership for Green and Sustainable Manufacturing under Grant M406884.

APPENDICES

Proof Theorem 1

For customer C_n , n = 1, 2, …, let
A_n = arrival epoch of C_n into the system,
D_n = departure epoch of C_n from the system,
$S_{j,n}$ = service time of C_n at server j, j = 1, …, M,
$CT_n^{T_i}$ = cycle time of C_n in system T_i, $i = 1, 2$.

(1) First, we consider an M-single-server cascading machine with nonoverlapping service times and without time-based breakdowns. From Tembe and Wolff (1974), the departure epochs of C_n from the system in arrangement T_1 and T_2 are

$$D_n^{T_1} = \max_{1\le i_1\le\cdots\le i_M\le n}\left[A_{i_1}+\sum_{k=i_1}^{i_2} S_{1,k}+\cdots+\sum_{k=i_{M-1}}^{i_M} S_{M-1,k}+\sum_{k=i_M}^{n} S_{M,k}\right]$$

$$= \max_{1\le i_1\le n}\left[A_{i_1}+S_{1,i_1}+\cdots+S_{a-1,i_1}+\sum_{k=i_1}^{n} S_{a,k}+S_{a+1,n}+\cdots+S_{M,n}\right].$$

$$D_n^{T_2} = \max_{1\le i_1\le\cdots\le i_M\le n}\left[A_{i_1}+\sum_{k=i_1}^{i_2} S_{[1],k}+\cdots+\sum_{k=i_{M-1}}^{i_M} S_{[M-1],k}+\sum_{k=i_M}^{n} S_{[M],k}\right]$$

$$= \max_{1\le i_1\le n}\left[A_{i_1}+S_{[1],i_1}+\cdots+S_{[a-1],i_1}+\sum_{k=i_1}^{n} S_{[a],k}+S_{[a+1],n}+\cdots+S_{[M],n}\right].$$

Because the service times among servers are not overlapping, for any i_1,

$$A_{i_1}+S_{[1],i_1}+\cdots+S_{[a-1],i_1}+\sum_{k=i_1}^{n} S_{[a],k}+S_{[a+1],n}+\cdots+S_{[M],n}$$

$$\le A_{i_1}+S_{1,i_1}+\cdots+S_{a-1,i_1}+\sum_{k=i_1}^{n} S_{a,k}+S_{a+1,n}+\cdots+S_{M,n}+\sup(S_b)-\inf(S_c).$$

Then $D_n^{T_2} \le D_n^{T_1} + \sup(S_b) - \inf(S_c)$ and $CT_n^{T_2} \le CT_n^{T_1} + \sup(S_b) - \inf(S_c)$.
Since T_1 and T_2 are arbitrary, we have
$\left|CT_n^{T_1} - CT_n^{T_2}\right| \le \sup(S_b) - \inf(S_c)$ for all n.
Taking customer averages,

$$|E(CT^{T_1}) - E(CT^{T_2})| \le E\left(\left|CT_n^{T_1} - CT_n^{T_2}\right|\right) \le \sup(S_b) - \inf(S_c) .$$

If the time-based breakdowns exist, the availability of the system is $\mathrm{A} = \frac{1/\eta}{1/\eta+\mathrm{E(D)}} = \frac{1}{1+\eta\mathrm{E(D)}}$. Then

$$|E(CT^{T_1}) - E(CT^{T_2})| \le (\sup(S_b) - \inf(S_c))(1+\eta E(D)).$$

(2) From Theorem 1-(1), $|\mathrm{E}(\mathrm{CT}^{\mathrm{T}_1}) - \mathrm{E}(\mathrm{CT}^{\mathrm{T}_2})| \le (\sup(\mathrm{S_b}) - \inf(\mathrm{S_c}))(1+\eta\mathrm{E(D)})$.
Since $\lim_{\rho\to 1} E(CT^{T_i}) \to \infty$, then $\lim_{\rho\to 1} |E(CT^{T_1}) - E(CT^{T_2})|/E(CT^{T_i}) \to 0, i = 1, 2$.

REFERENCES

Avi-Itzhak, B. 1965. "A Sequence of Service Stations with Arbitrary Input and Regular Service Times." *Management Science* 11 (5):565-571.

Dallery, Y., and S. B. Gershwin. 1992. "Manufacturing Flow Line Systems: A Review of Models and Analytical Results." *Queueing Systems* 12 (1):3-94.

Erlang, A. 1909. "The Theory of Probabilities and Telephone Conversations." *Nyt Tidsskrift for Matematik B* 20:33.

Friedman, H. D. 1965. "Reduction Methods for Tandem Queuing Systems." *Operations Research* 13 (1):121-131.

Gaver Jr, D. 1962. "A Waiting Line with Interrupted Service, Including Priorities." *Journal of the Royal Statistical Society. Series B (Methodological)*:73-90.

Gómez‐Corral, A. 2004. "Sojourn Times in a Two‐Stage Queueing Network with Blocking." *Naval Research Logistics (NRL)* 51 (8):1068-1089.

Hyun-Jung, K., L. Jun-Ho, and L. Tae-Eog. 2014. "Non-Cyclic Scheduling of a Wet Station." *Automation Science and Engineering, IEEE Transactions on* 11 (4):1262-1274.

Inman, R. R. 1999. "Emperical Evalucation of Exponential and Independence Assumptions in Queueing Models of Manufacturing Systems." *Production and Operations Management* 8 (4):409-432.

iSixSigma. 2012. "Takt Time." http://www.isixsigma.com/dictionary/takt-time/.

Kim, J. H., T. E. Lee, H. Y. Lee, and D. B. Park. 2003. "Scheduling Analysis of Time-Constrained Dual-Armed Cluster Tools." *Semiconductor Manufacturing, IEEE Transactions on* 16 (3):521-534.

Labanowski, L. 1997. Improving Overall Fabricator Performance Using the Continuous Improvement Methodology. Advanced Semiconductor Manufacturing Conference and Workshop, 1997. IEEE/SEMI, 10-12 Sep 1997.

Latouche, G., and M. F. Neuts. 1980. "Efficient Algorithmic Solutions to Exponential Tandem Queues with Blocking." *SIAM Journal on Algebraic Discrete Methods* 1 (1):93-106.

LeBaron, H. T., and R. A. Hendrickson. 2000. Semiconductor Process Equipment Modeling: Using Emulation to Validate a Cluster Tool Simulation Model.

Li, J., D. E. Blumenfeld, and J. M. Alden. 2006. "Comparisons of Two-Machine Line Models in Throughput Analysis." *International Journal of Production Research* 44 (7):1375-1398.

Li, J., D. E. Blumenfeld, N. Huang, and J. M. Alden. 2009. "Throughput Analysis of Production Systems: Recent Advances and Future Topics." *International Journal of Production Research* 47 (14):3823-3851.

Lim, J.-T., S. M. Meerkov, and F. Top. 1990. "Homogeneous, Asymptotically Reliable Serial Production Lines: Theory and a Case Study." *Automatic Control, IEEE Transactions on* 35 (5):524-534.

Mauer, J., and R. Schelasin. 1994. "Using Simulation to Analyze Integrated Tool Performance in Semiconductor Manufacturing." *Microelectronic engineering* 25 (2-4):139-146.

Morrison, J. R., and D. P. Martin. 2007. "Performance Evaluation of Photolithography Cluster Tools." *OR spectrum* 29 (3):375-389.

Niedermayer, H., and O. Rose. 2003. A Simulation-Based Analysis of the Cycle Time of Cluster Tools in Semiconductor Manufacturing. Proceedings of the 15th European Simulation Symposium.

Perkinson, T. L., P. McLarty, R. S. Gyurcsik, and R. K. Cavin III. 1994. "Single-Wafer Cluster Tool Performance: An Analysis of Throughput." *Semiconductor Manufacturing, IEEE Transactions on* 7 (3):369-373.

Rostami, S., and B. Hamidzadeh. 2002. "Optimal Scheduling Techniques for Cluster Tools with Process-Module and Transport-Module Residency Constraints." *Semiconductor Manufacturing, IEEE Transactions on* 15 (3):341-349.

Seo, D. W., and H. Lee. 2011. "Stationary Waiting Times in M-Node Tandem Queues with Production Blocking." *Automatic Control, IEEE Transactions on* 56 (4):958-961.

Strategos. 2012. "Takt Time - Definition, Benefits, Uses and Limits." http://www.strategosinc.com/takt_time.htm.

Tembe, S. V., and R. W. Wolff. 1974. "The Optimal Order of Service in Tandem Queues." *Operations Research* 22 (4):824-832.

Venkatesh, S., R. Davenport, P. Foxhoven, and J. Nulman. 1997. "A Steady-State Throughput Analysis of Cluster Tools: Dual-Blade Versus Single-Blade Robots." *Semiconductor Manufacturing, IEEE Transactions on* 10 (4):418-424.

Wan, Y., and R. W. Wolff. 1993. "Bounds for Different Arrangements of Tandem Queues with Nonoverlapping Service Times." *Management Science* 39 (9):1173-1178.

White, H., and L. S. Christie. 1958. "Queuing with Preemptive Priorities or with Breakdown." *Operations research*:79-95.

Wood, S. C. 1996. "Simple Performance Models for Integrated Processing Tools." *Semiconductor Manufacturing, IEEE Transactions on* 9 (3):320-328.

Wu, K. 2005. "An Examination of Variability and Its Basic Properties for a Factory." *IEEE Transactions on Semiconductor Manufacturing* 18 (1):214-221.

Wu, K. 2014. "Classification of Queueing Models for a Workstation with Interruptions: A Review." *International Journal of Production Research* 52 (3):902-917.

Wu, K., and K. Hui. 2008. "The Determination and Indetermination of Service Times in Manufacturing Systems." *IEEE Transactions on Semiconductor Manufacturing* 21 (1):72-82.

Wu, K., and L. McGinnis. 2013. "Interpolation Approximations for Queues in Series." *IIE Transactions* 45 (3):273-290.

Wu, K., L. McGinnis, and B. Zwart. 2011. "Queueing Models for a Single Machine Subject to Multiple Types of Interruptions." *IIE Transactions* 43 (10):753-759.

Wu, K., and N. Zhao. 2015a. "Analysis of Dual Tandem Queues with a Finite Buffer Capacity and Non-Overlapping Service Times and Subject to Breakdowns." *IIE Transactions* 47 (12):1329-1341.

Wu, K., and N. Zhao. 2015b. "Dependence among Single Stations in Series and Its Applications in Productivity Improvement." *European Journal of Operational Research* 247 (1):245-258.

Wu, K., N. Zhao, and C. K. M. Lee. 2016. "Queue Time Approximations for a Cluster Tool with Job Cascading." *IEEE Transactions on Automation Science and Engineering* 13 (2):1200 - 1206.

Xu, J., E. Huang, C.-H. Chen, and L. H. Lee. 2015. "Simulation Optimization: A Review and Exploration in the New Era of Cloud Computing and Big Data." *Asia-Pacific Journal of Operational Research* 32 (03):1550019.

AUTHOR BIOGRAPHIES

KAN WU is an Assistant Professor at Nanyang Technological University. He received the Ph.D. degree in Industrial and Systems Engineering from Georgia Institute of Technology. He has 10 years' experience in the semiconductor industry, from consultants to managers. His PhD dissertation was awarded the 3rd place for the IIE Pritsker Doctoral Dissertation Award in 2010. His research interests are primarily in the areas of queueing theory, with applications in the performance evaluation of supply chains and manufacturing systems. His email is wukan@ntu.edu.sg.

NING ZHAO is an Assistant Professor in Faculty of Science at Kunming University of Science and Technology. She received the M.S. degree in Applied Mathematics from Shanghai Jiaotong University, and received a Ph.D in Business Information Systems from University of Macau. Her research interest is in queueing theory. Her email is zhaoning1102@gmail.com.