Aggregate modeling of multi-processing workstations *

A.A.A. Kock, L.F.P. Etman, J.E. Rooda,
I.J.B.F. Adan, M. van Vuuren, A. Wierman †

August 12, 2008

Abstract

In this paper an aggregate model for manufacturing systems consisting of flow lines with finite buffers and parallel servers is proposed. The proposed model is a multi-server station with process times depending on the work in process (WIP). An algorithm is developed to measure the WIP-dependent process times directly from industrial data such as arrival times at and departure times from the manufacturing system. Simulation results show that the aggregate model accurately predicts the mean flow time.

Keywords: *Discrete-event simulation, Effective process time, Performance analysis, Queueing approximation*

1 Introduction

In semiconductor manufacturing, there is a trend of proliferation of integrated processing [Wood, 1996]. These integrated processing tools allow multiple wafers of one or more lots to be processed simultaneously. Multiple processes or process steps are contained within a single tool. The logistics inside such integrated tools are often flow line alike (lot cascading). For example, integrated lithography cells allow wafers of up to four lots to be pipelined through a sequence of

*Submitted for publication

†A.A.A. Kock, L.F.P. Etman and J.E. Rooda are with the Department of Mechanical Engineering of the Eindhoven University of Technology
I.J.B.F. Adan is with the Department of Mathematics and Computer Sciences of the Eindhoven University of Technology
M. van Vuuren and A. Wierman were formerly with the Department of Mathematics and Computer Science of the Eindhoven University of Technology. M. van Vuuren is now with CQM BV, Eindhoven. A. Wierman is now with the Computer Science department at Caltech.
A.A.A. Kock, L.F.P. Etman and I.J.B.F. Adan are the corresponding authors, postal address: PO Box 513, 5600 MB, Eindhoven, The Netherlands
{a.a.a.kock,l.f.p.etman,j.e.rooda,i.j.b.f.adan}@tue.nl,vuuren@cqm.nl,adamw@caltech.edu

several processes, including resist coat, expose, and develop. In addition, vacuum processors are integrated around standardized frames that include wafer handlers and loadlocks. Other examples of integrated processing tools are wetbenches (lots traverse through a sequence of chemical baths), metal deposit tools (several surface treatment and metal-alloy deposition processes are combined in a single tool) and ion-implant (ion implant consists of two sequential steps: loading and ion emanation onto the wafers).

Due to the sequence of processes that is carried out in an integrated processing tool, the mean flow time φ and throughput δ in the tool increases as the work in process, WIP, increases. The presence of such tools on the factory floor complicates the performance analysis.

For the performance analysis of semiconductor manufacturing there are two categories of models in common use: (discrete-event) simulation models and analytical models. Simulation models allow the inclusion of various details of the processes. However, every detail requires data to be collected and adds to the computational expense of the simulation model. Arisha and Young [2004]; Nayani and Mollaghasemi [1998]; Pierce and Drevna [1992] develop simulation models of integrated processing tools, with explicit modeling of, e.g., machine downs, repairs, operating rules, setups, maintenance, operator availability and operator skill. The cluster tool model described in Pierce and Drevna uses over 1100 variables and parameters and 500 distributions.

Analytical models, on the other hand, are usually computationally cheap to evaluate and require little input data, such as the mean and variance of process times. However, they adhere to restrictive assumptions, such as, e.g., phase-type distributed process times [Asmussen, 2003]. Shanthikumar et al. [2007] noted that lot cascading in a tool should be well modeled to obtain accurate flow time estimations. An appealing approach to estimate the performance of complex manufacturing systems is to represent (part of) the system by a so-called flow equivalent server (FES) [Norton, 1926]: an exponential single-server station with service rates depending on the WIP. Indeed, under restrictive assumptions, the aggregate system behavior can be described exactly by a FES, i.e., it is possible to replace part of a queueing network (representing the manufacturing system) by a single-server station without affecting the behavior of the rest of the network [Boucherie, 1998; Chandy et al., 1975]. Exact FES models were originally derived for balanced, closed queueing networks with exponential process times. Later, extensions were proposed for special networks with Coxian process times and constant process times [Rhee, 2006; Stewart and Zeiszler, 1980; Thomasian and Nadji, 1981]. However, the assumptions required for an (exact) FES model are too prohibitive to be of practical use in the present context of integrated tools.

In this paper, we propose an aggregate model that, similar to the FES, replaces the integrated tool by a single- or multi-server station with WIP-dependent processing times. However, unlike the FES, we do not make a priori assumptions regarding process time distributions. Key to our approach is that the process time distributions can be obtained directly from arrival and departure events from the factory floor. The advantage is clear: we do not need

to quantify all shop-floor realities individually. To estimate the parameters of the process time distributions we adopt the "Effective Process Time" (EPT) paradigm [Hopp and Spearman, 1996, 2001; Jacobs et al., 2001, 2003].

The system, studied in this paper, is an open network with finite buffers and no feedback; in particular, the configuration is flow-line alike, motivated by the lot cascading tools used in semiconductor manufacturing, which is common for the logistics inside integrated tools. The accuracy of the mean flow time predicted by the aggregate model is investigated for several configurations, ranging from a flow line with twelve sequential servers to a station with twelve parallel servers. Simulation results convincingly demonstrate that the proposed aggregate model yields accurate predictions. Hence, the conclusion is that the modeling framework of multi-server stations with WIP-dependent process times combined with the EPT paradigm provides an effective and powerful tool for the performance evaluation of multi-processing tools.

The outline of this paper is as follows: we first present an overview of the effective process time paradigm in Section 2. In Section 3, we explain the main concept of the aggregate model. We introduce the algorithm to translate arrival and departure data into EPT-realizations in Section 3.3. The algorithm is tested on a set of examples in Section 4. Finally, in Section 5 we present our main findings and the discussion.

2 Previous work using the EPT paradigm

The phrase effective process time was originally introduced by Hopp and Spearman [1996, 2001]. They define the EPT as 'the time seen by a lot at a workstation from a logistical point of view'. The EPT aggregates the raw processing time and all shop-floor realities and disturbances on processing at a workstation into a single process time distribution. The inclusion of multiple phenomena into a single distribution is referred to as aggregation. Hopp and Spearman give explicit expressions to compute the mean EPT and the EPT coefficient of variation from the raw processing time and the various outages, either preemptive (setup-alike) or non-preemptive (breakdown-alike). They use the EPT mean and variance in closed form approximations for $G/G/m$ queues to explain and estimate the mean flow time performance.

In many practical cases, outages may not all be quantifiable. Jacobs et al. [2001, 2003] show that the EPT can be measured without the need to identify and quantify all contributing shop-floor realities. For workstations with ample buffer space and that process a single lot at a time, they present an algorithm to calculate EPT-realizations directly from lot arrival and departure events. The obtained empirical distribution can then be used to fit a parameterized EPT-distribution.

This idea can be generalized into an EPT-based modeling framework, as explained by Kock et al. [2008a]. Event collection, EPT calculation, distribution fitting and aggregate modeling are presented as an integrated framework. The EPT is not only used as a performance metric quantifying capacity (mean) and

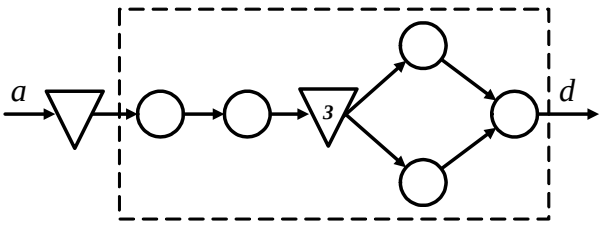


Figure 1: Tandem flow line with finite buffers; circles indicate process steps, triangles buffers, a lot arrivals and d lot departures

variability (variance), but also to build simulation or analytical models fed by parameter values obtained from empirical EPT-distributions.

EPT-algorithms to compute EPT-realizations from arrival and departure events were proposed by Jacobs et al. [2006, 2001, 2003]; Kock et al. [2008a,b]; Vijfvinkel et al. [2007], for infinitely buffered 'single lot' workstations, finitely buffered 'single lot' workstations, assembly workstations and batch workstations. These references focus on discrete-event simulation models. Analytical models may be used as an alternative. Closed form expressions for (mean) performance measures of $G/G/m$ queues can be used for infinitely buffered multi-server workstations. For finitely buffered flow lines and assembly lines, queueing approximations as discussed by Dallery and Gershwin [1992]; Vuuren [2007]; Vuuren et al. [2005] may be used.

3 An aggregate multi-server station

In the present paper, we consider flow lines consisting of multi-server stations with finite buffers. Specifically, we assume that, on arrival, lots are put into an infinite buffer to wait until processing starts, and once in process, lots do not recirculate. An example is visualized in Figure 1.

3.1 Model concept

The idea is to aggregate the entire flow line into a multi-server station with FIFO dispatching and WIP-dependent process times; see Figure 2. The number of servers, denoted by m, is an important user-defined parameter. Initially, one may expect that the choice of m will be related to the structure of the flow line, i.e., the "degree of parallel processing"; this relation will be investigated in Section 4. The process time of a lot depends on the WIP present in the system just before the start of processing. The dependence on the WIP reflects that, in the real system, the mean flow time and throughput depend on the number of lots in the system. Clearly, the real system is *not* a m-server station; hence, the challenge is to subtract the required WIP-dependent process times from the arrival and departure events in the real system. This is explained in the following two sections.

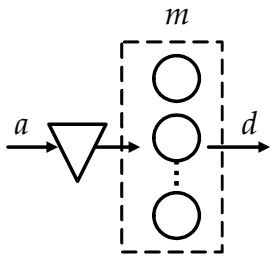


Figure 2: Structure of the proposed aggregate model

3.2 EPT measurement

The input to the calculation of EPT-realizations consists of a chronological list of events obtained from the shop-floor. Each event is defined by the lot id, the event type ev (arrival in the infinite buffer of the flow line, denoted 'A', or departure from the flow line, denoted 'D') and the time of occurrence of the event τ. Then, by acting as if the event list has been produced by an m-server station, we are able to retrieve the EPT-realizations. Since the process times in the multi-server station are WIP-dependent, we introduce bucket b for each WIP-level b, $1 \leqslant b < \infty$. An EPT-realization is assigned to bucket b if b lots are present at the start of the EPT-realization. Thus, each bucket collects EPT-realizations corresponding to a certain WIP, and at the end of the event list, provides an empirical EPT-distribution. Since the EPT-distributions are expected to converge as b tends to infinity, we can limit the number of buckets by N, say, where bucket N contains all process times registered with a WIP $\geq N$.

Most likely, the real system and the m-server station do not perfectly match. Hence, it may happen that, when lot id departs at time τ, it has not yet started processing in the m-server station; this is readily seen to happen when a $G/G/2$ is aggregated into a $G/G/1$, since overtaking takes place in the first, but not in the second system. This inconsistency will be solved as follows. We pick one of the lots in process at time τ, say lot jd that started processing at time t when the WIP was b; the pick rule(s) will be specified in the next section. Then we "interchange" the departure times of lot id and jd; so lot jd leaves at time τ, having received an EPT of $\tau - t$ time units for WIP b, after which lot id immediately enters service and remains so until the "old" departure time of lot jd.

In the next section we describe the algorithm to calculate EPT-realizations in more detail.

3.3 EPT-algorithm

The EPT-algorithm is depicted in Figure 3. It uses the following variables: n represents the current WIP, list rs stores (id, τ, n) containing the start times

of the lots that are in process (according to the m-server station). List ws contains the id of each lot in the system that has not yet started processing (again, according to the m-server station). The algorithm uses the functions `append`, `get`, `remove`, `head`, `tail` and `find` operating on the lists rs and ws. Function `append` adds an element to the end of the list, `get` reads the element with lot id from the list. Function `remove` removes the element with id from the list. Function `head` takes the first element in the list and function `tail` takes all elements except the first. Finally, `find` picks one specific element from the list according to a user-defined rule, to be discussed later.

```
n:= 0; rs:=[]; ws := []
loop
      read id, ev, τ
      if ev = 'A' then
            n := n + 1
            if n ⩽ m then                                   (a1)
                  rs:= append(rs, (id, τ, n))
            elseif n > m then                               (a2)
                  ws:= append(ws, id)
            endif
      elseif ev = 'D' then
            n := n − 1
            if n < m then                                   (d1)
                  (t, b):= get(rs, id)
                  rs:= remove(rs, id)
            elseif n ⩾ m and id ∈ rs then                   (d2)
                  (t, b):= get(rs, id)
                  rs:= remove(rs, id)
                  jd:= head(ws); ws:= tail(ws)
                  rs:= append(rs, (jd, τ, n))
            elseif n ⩾ m and id ∉ rs then                   (d3)
                  (jd, t, b):= find(rs,rule)
                  rs:= remove(rs, jd)
                  rs:= append(rs, (jd, τ, n))
                  ws:= remove(ws, id)
            endif
            write τ − t, b
      endif
endloop
```

Figure 3: EPT-Algorithm

The EPT-algorithm distinguishes five cases:

($a1$) A lot arrives when $n < m$ lots are present. Capacity is available: lot start with id, time τ and WIP-level n is added to rs.

($a2$) A lot arrives when $n \geqslant m$ lots are present. All m servers are busy, thus the lot is stored in the buffer ws.

($d1$) A lot departs, $n < m$ lots remain behind. Bucket b and start time t of the departing lot are retrieved from rs, after which the lot is removed.

($d2$) A lot departs, $n \geqslant m$ lots remain behind and id of the departing lot is known in rs: bucket b and start time t of the lot are retrieved from rs after which id is removed from rs; the first lot waiting in ws is added as new lot start to rs with time τ and WIP-level n.

($d3$) A lot departs, $n \geqslant m$ lots remain behind, and id of the departing lot is not known in rs. So lot id departs, while it has not started processing according to the m-server station. Then, using function `find`, we select an alternative lot that has started already, jd. We compute the EPT-realization using the start time of jd. Then, lot jd is restarted and lot id is removed from buffer ws.

Note that in ($d3$) lot id immediately departs and lot jd (re)starts service, instead of the other way around; the reason is that, although the lot identity is not relevant for the EPT-realization, we should be able to connect the right lot to the departure of lot jd after time τ.

For function `find` in case ($d3$), we propose three rules: 1) random lot, 2) lot with the shortest elapsed process time, 3) lot with the longest elapsed process time. The rationale behind rule 2 is that the lot might be a fast mover, and therefore, we assign the smallest possible process time; the rationale behind rule 3 is opposite. Clearly, for $m = 1$, the pick rules are identical, since then there is only one lot to pick. The impact of the choice of the pick rule on the performance predictions will be investigated in Section 4.

In case ($d1$), ($d2$) and ($d3$), the EPT-realization is printed as $\tau - t$ with bucket b.

3.4 Gantt-chart examples

Figure 4 shows Gantt-charts for two manufacturing systems; Figure 4(a) corresponds to a system without overtaking, and Figure 4(b) to a system with overtaking. The bottom part of the Gantt-charts shows the EPT-realizations computed by the EPT-algorithm, with $m = N = 2$; EPTs are labeled $t(b)$, where t is the duration of the EPT and b the bucket. Note that case ($d3$) is invoked twice in Figure 4(b), but not in Figure 4(a).

4 Model validation

By means of discrete-event simulation we will test the aggregate model in four scenarios depicted in Figure 5; all simulation results are generated using the $\chi - 0.8$ software [Hofkamp and Rooda, 2002].

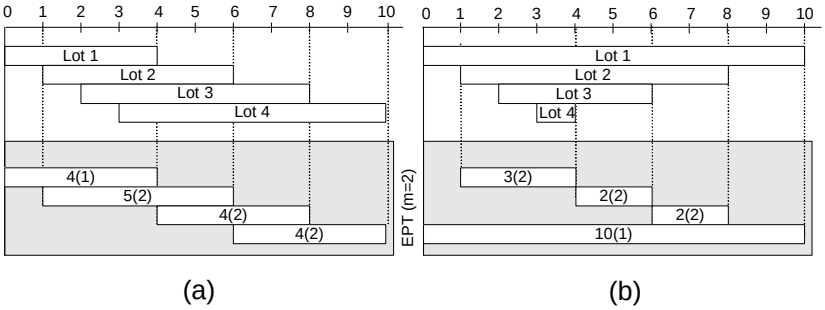


Figure 4: Example Gantt-charts, (a) without overtaking, (b) with overtaking, using rule 2

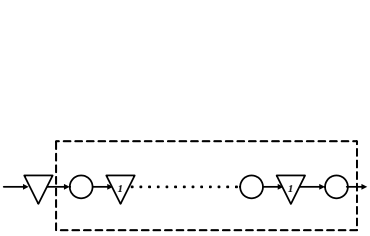
In each example, the arrival process is Poisson with rate δ and the process times are gamma-distributed with mean 1.0 and squared coefficient of variation $c^2 \in \{0.1, 1.0, 2.0\}$. Mean flow time predictions in the real system are based on simulation runs of 2.000.000 lots. The utilization of the system is defined as the ratio of the throughput δ and the maximum attainable throughput $\delta_{\max}$, which is determined in one simulation run of 100.000 lots using unlimited supply of lots. For each scenario EPT-realizations are measured using the EPT-algorithm (Figure 3) in a simulation run of 2.000.000 lots at a given utilization level, the so-called training level. For scenario I, the training level is $\delta/\delta_{\max} \in \{0.6, 0.9\}$ while for scenarios II, III and IV, we take $\delta/\delta_{\max} = 0.8$. On the empirical EPT-distributions, we fit Gamma distributions matching the mean t_e and coefficient of variation c_e^2. Then mean flow times are predicted by the multi-server aggregate station with WIP-dependent Gamma-distributed process times at utilization levels $0.3 \leqslant \delta/\delta_{\max} \leqslant 0.95$; at each utilization level the mean flow time prediction is based on five runs of 10.000.000 lots.

4.1 Scenario I: Twelve sequential single server stations

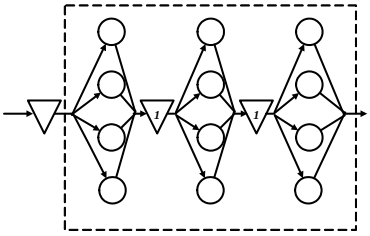
The system consists of a flow line of twelve sequential single-server stations, see Figure 5(a). Each station has one buffer space. For this system, we have $\delta_{\max} = \{0.875, 0.553, 0.440\}$ [lots/hour] for $c^2 = \{0.1, 1.0, 2.0\}$.

In Figures 6 we present EPT-realizations measured for $\delta/\delta_{\max} = 0.9$, $c^2 = 1.0$ and $m = 1$. The x-axis in Figure 6(a) is the WIP (or bucket), whereas the y-z planes represent histograms of the EPT-realizations. Clearly, the bulk of the EPT-realizations is in buckets ranging from 1 to 40, with a peak near 20. The empirical probability distribution function (PDF) is plotted in Figure 6(b). From bucket 30, say, onwards, the distributions do not significantly change; buckets 40 or higher hardly contain any realization explaining the noisy behavior. Hence, it makes sense to aggregate all realizations in buckets ≥ 30 into bucket $N = 30$.

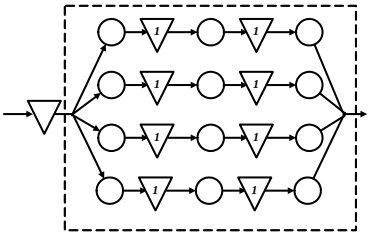
Figure 7 plots the mean EPT t_e and squared coefficient of variation (SCV)



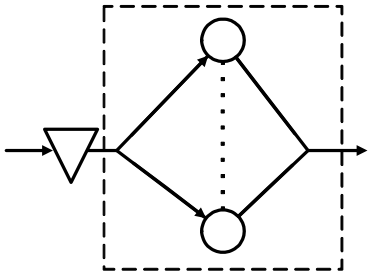
(a) Scenario 1: 12 sequential single-server workstations



(b) Scenario 2: 3 workstations with 4 servers



(c) Scenario 3: 4 parallel lines



(d) Scenario 4: 12 parallel servers

Figure 5: Test scenarios for algorithm of Figure 3

c_e^2 as a function of the WIP-level. Clearly, these plots depend on the squared coefficient of variation c^2 of the processing times in the real system.

The monotonic behavior of t_e as a function of WIP-level is as expected: the higher the WIP in the flow line, the faster lots will leave the line. Also the behavior of c_e^2 may be explained: initially, at low WIP, c_e^2 tends to increase, due to the (random) distribution of the WIP in the flow line, and eventually, c_e^2 will converge to a value close to c^2. We would like to point out that monotonicity properties of t_e and c_e^2, as observed in Figure 7, may be exploited in an analytical model to accomplish, e.g., state space reduction.

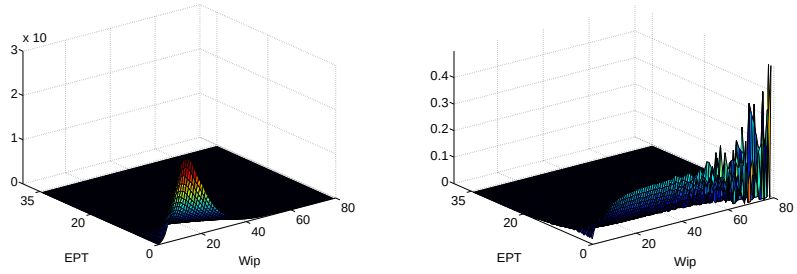
Figures 8(a) and 8(b) show, for various values of m, mean flow time predictions of the aggregate model trained at utilization level $\delta/\delta_{\max} = 0.6$ and 0.9, respectively; the EPT-realizations are obtained by employing pick rule 1. The figure shows that, for $m = 1$, the best prediction is obtained at the training level (as expected). For $m = 1$, mean flow time predictions are also listed in Tables 1 and 2. From the results we can conclude that mean flow times at low utilization levels are more accurately predicted by the aggregate model trained at $\delta/\delta_{\max} = 0.6$ than the one trained at $\delta/\delta_{\max} = 0.9$, whereas the reverse is true for high utilizations. Further, the predictions seem to be more accurate for smaller values of c^2.

A naive approach is to approximate the flow line by an $M/G/1$ queue; in the present context, this means that the flow line is aggregated into a 1-server station with $N = 1$, i.e., all EPT-realizations are assigned to one bucket. This approach would produce poor approximations, since it completely fails to take into account the increased efficiency due to the lot cascading for larger WIP-levels.

Table 1: Scenario I: Mean flow time prediction ($m = 1$, trained at $\delta/\delta_{\max} = 0.6$)

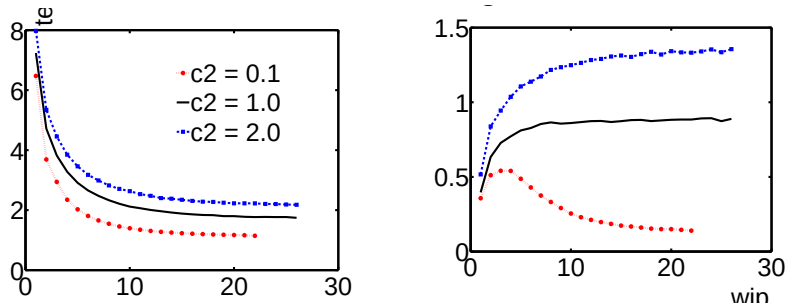
	$c^2 = 0.1$		$c^2 = 1.0$		$c^2 = 2.0$	
$\frac{\delta}{\delta_{\max}}$	Approx.	Real	Approx.	Real	Approx.	Real
0.3	12.02	12.85	14.11	14.67	15.40	15.77
0.5	13.39	13.79	17.16	17.18	19.44	19.29
0.6	14.49	14.49	18.86	18.85	21.62	21.62
0.7	16.23	15.51	20.94	21.06	24.19	24.69
0.85	21.75	18.58	25.22	27.26	29.19	33.54
0.95	69.62	27.07	30.36	48.94	33.38	67.10

The aggregate 1-server station may be slightly refined by exploiting the following observation. There are two possibilities to start processing at WIP-level 1: either a lot arrives in an empty flow line, or the previous departure left behind a single lot. The mean EPT of a lot entering an empty flow line is 12, whereas the mean EPT of a single lot left behind is clearly less (in fact, 6 according to simulation). Thus, splitting bucket 1 in two buckets may improve the predictions. Figure 9 shows mean flow time predictions for $c^2 = 1.0$ with training level $\delta/\delta_{\max} = 0.6$. Since the prediction only slightly improves for low



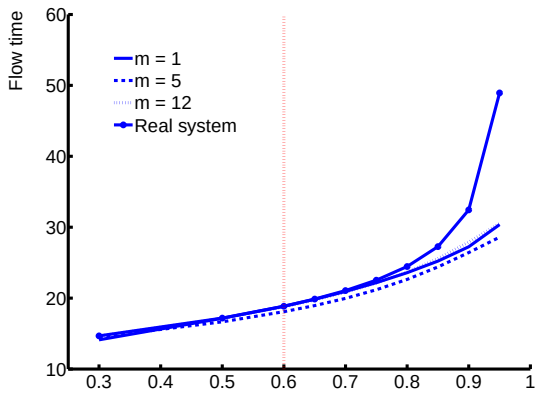
(a) Histogram (b) Empirical PDF

Figure 6: Scenario I: EPT-realizations ($\delta/\delta_{\text{MAX}} = 0.9$, $c^2 = 1.0$ and $m = 1$)

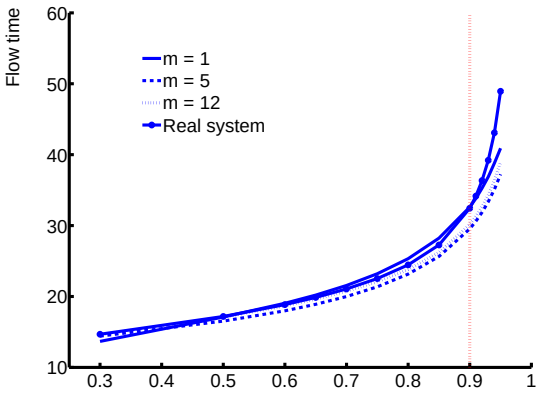


(a) t_{E} per bucket (b) c_{E}^2 per bucket

Figure 7: Scenario I: Mean and SCV of EPT ($\delta/\delta_{\text{MAX}} = 0.9$, $c^2 = 1.0$ and $m = 1$)



(a) TRAINED AT $\delta/\delta_{\text{MAX}} = 0.6$



(b) TRAINED AT $\delta/\delta_{\text{MAX}} = 0.9$

Figure 8: SCENARIO I: FLOW TIME PREDICTION ($c^2 = 1.0$, RULE 1)

Table 2: Scenario I: Mean flow time $\widetilde{\varphi}$ estimation for $m = 1$ if the model is trained at $\delta/\delta_{\text{MAX}} = 0.9$

	$c^2 = 0.1$		$c^2 = 1.0$		$c^2 = 2.0$	
$\frac{\delta}{\delta_{\max}}$	Approx.	Real	Approx.	Real	Approx.	Real
0.3	11.35	12.85	13.66	14.67	15.34	15.77
0.6	13.10	14.49	19.04	18.85	22.82	21.62
0.85	18.13	18.58	28.22	27.26	35.32	33.54
0.9	21.08	20.95	32.60	32.45	41.26	41.18
0.92	22.46	22.55	35.27	36.35	44.84	47.10
0.95	26.14	27.07	40.91	48.94	52.42	67.10

$\delta/\delta_{\max}$, we will not further pursue the option of splitting of buckets.

Next we investigate sensitivity with respect to the number of EPT measurements. Figure 10 shows that, if the number of EPT-realizations is drastically reduced from 2.000.000 to 15.000 lots, the mean flow time predictions are still accurate. This suggests that it is not necessary to collect an "enormous" amount of data, which is convenient from a practical point of view.

Finally we consider an unbalanced flow line: the processing speed of server 6 is is slowed down by a factor 1.5, and thus it becomes the bottleneck station. Mean flow time predictions for utilization levels from 0.3 until 0.95 are depicted in Figure 11. For $m = 1$, the predictions are even slightly more accurate than in the balanced case.

4.2 Scenario II: Three stations, four parallel servers each

The first station in the three station flow line of Figure 5(b) has an infinite buffer, the other two have one buffer place. The maximum obtainable throughput is $\delta_{\max} = \{3.666, 3.174, 2.989\}$ [lots/hour] for $c^2 = \{0.1, 1.0, 2.0\}$. The training level is $\delta/\delta_{\max} = 0.8$.

In Figures 12 we show t_{e} and c_{e}^2 as a function of the WIP-level, for $m = 1$ and $m = 4$. As expected, the shape of the t_{e} and c_{e}^2 curves depend on the choice of m; in particular, the limiting value of t_{e} for $m = 4$ is (roughly) four times the limiting value for $m = 1$.

Figure 13 presents mean flow time predictions in the range of $0.3 \leqslant \delta/\delta_{\max} \leqslant 0.95$. It shows that the predictions for $m = 12$ are accurate at low utilizations, but underestimate the mean flow time at high utilizations; a possible explanation is that the 12-server station allows for more overtaking than in the real system. The predictions for $m = 1$ and $m = 4$ are very accurate in the utilization range $0.6 \leqslant \delta/\delta_{\max} \leqslant 0.9$. In this case, one might initially guess that $m = 4$ would be the best choice, since it properly reflects the "degree of parallel processing"; but, surprisingly, the predictions for $m = 1$ are of the same quality.

Table 3 gives additional results for $m = 4$, demonstrating the effect of the pick rule. The estimates for the three rules are fairly close, but seem to be ordered: rule 3 gives the lowest prediction, rule 2 the highest and rule 1 is in

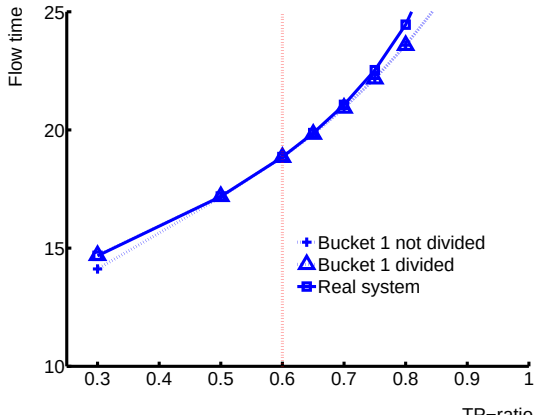


Figure 9: Scenario I: Effect of splitting bucket 1 ($c^2 = 1.0$, $m = 1$, trained at $\delta/\delta_{\text{MAX}} = 0.6$)

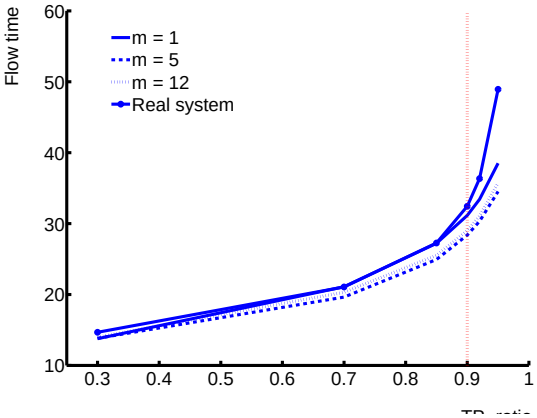


Figure 10: Scenario I: Flow time prediction, 15.000 lots ($c^2 = 1.0$, rule 1, trained at $\delta/\delta_{\text{MAX}} = 0.9$)

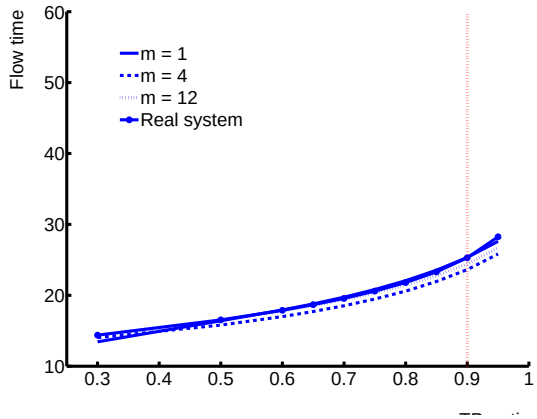


Figure 11: Scenario I: Flow time prediction, unbalance 1.5 ($c^2 = 1.0$, rule 1, trained at $\delta/\delta_{\text{MAX}} = 0.9$)

between. This ordering is also reflected in the c_e^2-curves in Figure 14, which seems to be a direct consequence of the pick rule.

Table 3: SCENARIO II: MEAN FLOW TIME PREDICTION ($m = 4$, TRAINED AT $\delta/\delta_{\text{MAX}} = 0.8$)

	$c^2 = 0.1$				$c^2 = 2.0$			
$\frac{\delta}{\delta_{\max}}$	rule 1	rule 2	rule 3	Real	rule 1	rule 2	rule 3	Real
0.3	2.90	2.90	2.89	3.02	2.50	2.85	2.29	3.03
0.6	3.08	3.08	3.08	3.18	3.12	3.36	3.00	3.33
0.7	3.23	3.23	3.23	3.32	3.51	3.73	3.38	3.62
0.8	3.50	3.50	3.50	3.58	4.12	4.39	3.95	4.18
0.9	4.14	4.15	4.13	4.29	5.46	5.92	5.09	5.68
0.95	4.91	4.94	4.89	5.63	7.28	7.84	6.57	8.04

4.3 Scenario III: Four parallel lines of three sequential, single server stations

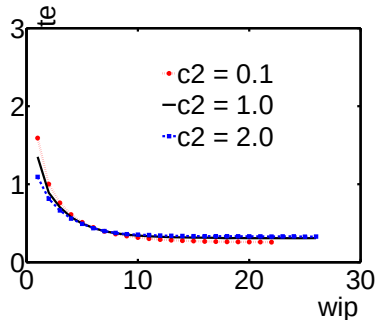
We now consider a system of four parallel single-server flow lines, with three stations per line, see Figure 5(c). Each station has one buffer space, except for the first stations in the lines sharing an infinite buffer. For this system, the maximum obtainable throughput is $\delta_{\max} = \{3.659, 2.691, 2.319\}$ [lots/hour] for $c^2 = \{0.1, 1.0, 2.0\}$. The training level is $\delta/\delta_{\max} = 0.8$.

Figure 15 shows the mean flow time prediction for $0.3 \leqslant \delta/\delta_{\max} \leqslant 0.95$; additional results for $m = 4$ and each of the pick rules are displayed in Table 4. The results for scenario III are comparable to ones for scenario II. Note, however, at high utilizations the prediction errors in scenario III are larger than in scenario II (cf. Figure 15 and Figure 13). Apparently, in scenario II, the aggregate model more accurately captures interaction between lots.

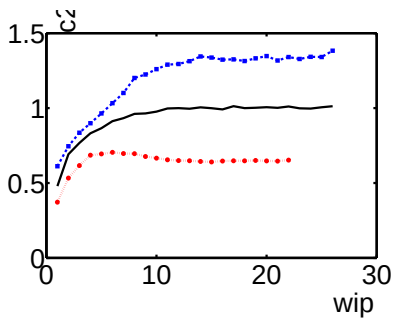
Table 4: SCENARIO III: MEAN FLOW TIME PREDICTION ($m = 4$, TRAINED AT $\delta/\delta_{\text{MAX}} = 0.8$)

	$c^2 = 0.1$				$c^2 = 2.0$			
$\frac{\delta}{\delta_{\max}}$	rule 1	rule 2	rule 3	Real	rule 1	rule 2	rule 3	Real
0.3	3.58	3.60	3.55	3.84	4.20	4.67	3.97	5.23
0.6	4.14	4.15	4.13	4.27	5.57	5.83	5.46	5.96
0.7	4.37	4.38	4.36	4.42	6.11	6.34	6.00	6.28
0.8	4.72	4.74	4.72	4.70	6.83	7.11	6.66	6.92
0.9	5.51	5.54	5.49	5.44	8.16	8.66	7.81	8.86
0.95	6.62	6.66	6.59	6.84	9.60	10.39	9.09	12.75

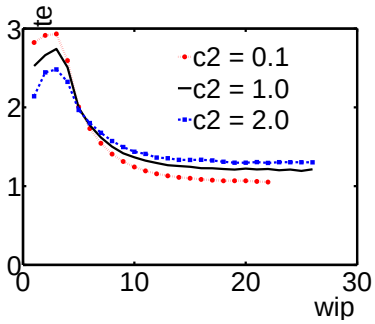
Finally, we note that the picture of mean flow times, obtained by slowing down one of the four lines by a factor 1.5, is similar to Figure 11.



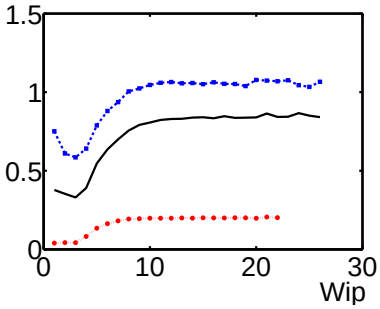
(a) t_{E} PER BUCKET, $m = 1$



(b) c_{E}^2 PER BUCKET, $m = 1$



(c) t_{E} PER BUCKET, $m = 4$



(d) c_{E}^2 PER BUCKET, $m = 4$

Figure 12: SCENARIO II: EFFECTIVE PROCESS TIMES PER BUCKET ($\delta/\delta_{\mathrm{MAX}} = 0.8$, $c^2 = 1.0$, RULE 1)

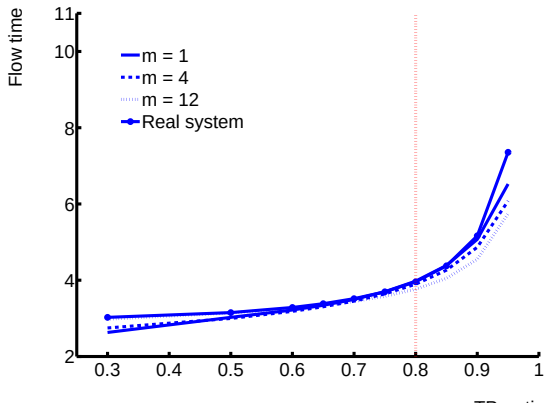
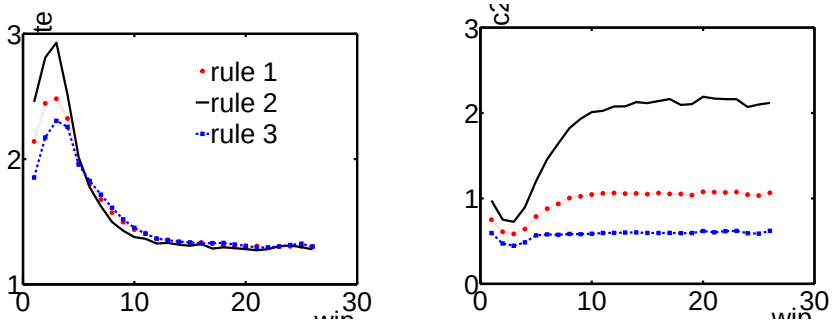


Figure 13: Scenario II: flow time prediction ($c^2 = 1.0$, rule 1, trained at $\delta/\delta_{\text{MAX}} = 0.8$)



(a) t_{E} per bucket, $m = 2$

(b) c^2_{E} per bucket, $m = 2$

Figure 14: Scenario II: Effective process times per pick rule ($\delta/\delta_{\text{MAX}} = 0.8$, $c^2 = 2.0$, $m = 2$)

4.4 Scenario IV: Workstation with twelve parallel servers

To conclude, we consider a workstation with twelve parallel servers, see Figure 5(d). For this system, the maximum obtainable throughput is $\delta_{\max} = \{12, 12, 12\}$ [lots/hour] for $c^2 = \{0.1, 1.0, 2.0\}$. The training level is again set at $\delta/\delta_{\max} = 0.8$.

Figure 16 shows t_e and c_e^2 as a function of the WIP-level for $m = 12$. Clearly, the measurements in buckets smaller than 6 or larger than 15 experience noise (due to few observations): one would expect flat curves here.

Figure 17 shows mean flow time predictions for $0.3 \leqslant \delta/\delta_{\max} \leqslant 0.95$. The figure also depicts the standard $M/G/12$ approximation, i.e., $m = 12$ and $N = 1$. Obviously, now this "naive" approximation is very accurate, and the $M/G/12$ with "WIP-dependent" process times is almost as accurate. Further, the predictions for $m = 1$ are less accurate at low utilization and the ones for $m = 20$ are less accurate at high utilization.

In Scenario II we already touched the issue of selecting the pick rule; see Table 3, demonstrating that the effect of the pick rule on the mean flow time prediction is limited. However, this choice may be relevant in situations where the rule is often invoked. For example, this is expected to happen if the 12-server station is aggregated as a 2-server station; the predicted mean flow time, as a function of $\delta/\delta_{\max}$, is depicted in Figure 18, and indeed, the accuracy now strongly depends on the pick rule. In all examples, however, it appeared that rule 1, i.e., the random rule, performed well and thus, this rule seems to be a safe choice. Moreover, the numerical experiments in this paper convincingly show that the aggregate model with $m = 1$ always produces accurate mean flow time predictions, and in this case, the pick rule is irrelevant.

Finally, we consider an unbalanced case by slowing down the processing speed of six of the twelve servers by a factor 1.5, while keeping $c^2 = 1.0$ for all processing times. Evaluating mean flow time predictions for $m \in \{1, 2, 4, 12\}$ over the range $0.3 \leqslant \delta/\delta_{\max} \leqslant 0.95$ leads to similar results as shown in Figure 17. However, in this case, the standard $M/G/12$ approximation is inaccurate: it sometimes overestimates the mean flow time by more than 10%, while the $M/G/12$ approximation with WIP-dependent process times remains accurate.

5 Conclusions and discussion

In this paper, we propose an aggregate m-server model with WIP-dependent process times. The process times are computed from lot arrivals at and lot departures from the system that is aggregated. An advantage is that these events can be directly measured from the factory floor. An algorithm is presented to calculate the WIP-dependent effective process time realizations.

The accuracy of the mean flow time prediction has been investigated in four scenarios, ranging from a flow line to a single workstation with parallel servers. The results show that predictions are accurate, but the quality depends on the choice of m, and to a lesser degree, on the pick rule; surprisingly, the choice $m =$

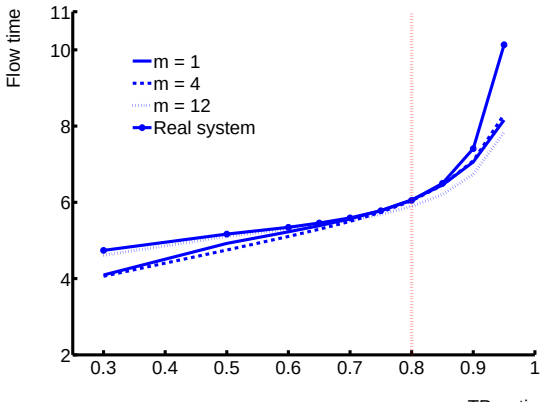
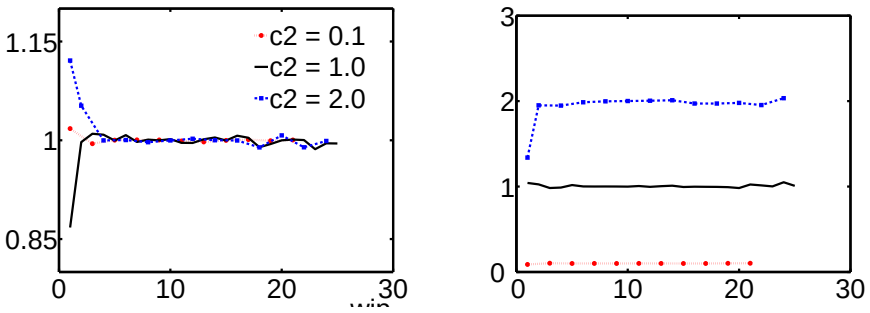


Figure 15: Scenario III: flow time prediction ($c^2 = 1.0$, rule 1, trained at $\delta/\delta_{\text{MAX}} = 0.8$)



(a) t_{E} per bucket, $m = 12$ (b) c^2_{E} per bucket, $m = 12$

Figure 16: Scenario IV: Effective process times per bucket ($\delta/\delta_{\text{MAX}} = 0.8$)

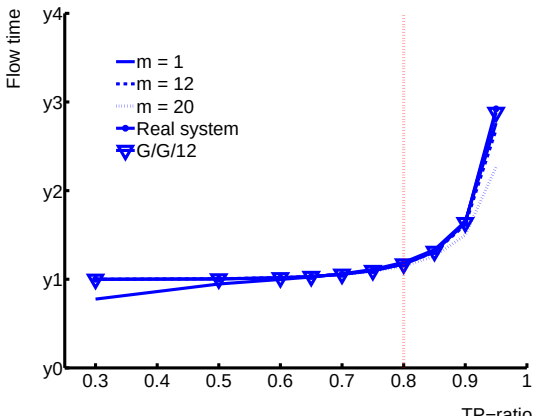


Figure 17: SCENARIO IV: FLOW TIME PREDICTION ($c^2 = 1.0$, RULE 1, TRAINED AT $\delta/\delta_{\text{MAX}} = 0.8$)

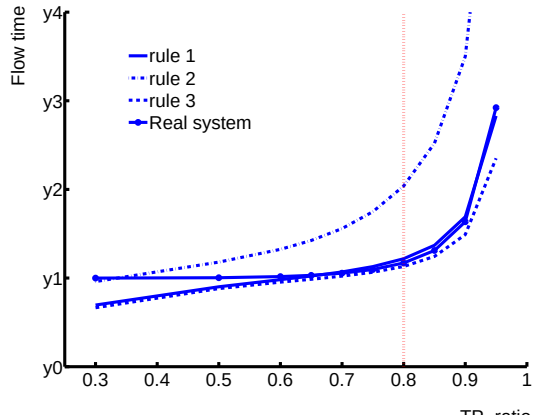


Figure 18: SCENARIO IV: FLOW TIME PREDICTION ($c^2 = 1.0$, $m = 2$, MEASURED AT $\delta/\delta_{\text{MAX}} = 0.8$)

1 appears to be good across all scenarios. The feature of WIP-dependent process times appears to be crucial: the quality of mean flow time predictions by multi-server stations with WIP-*in*dependent process times is usually poor. The overall conclusion is that the aggregate 1-server station always performs well (and, in this case, the choice of the pick rule is not relevant). The simulation study in this paper is restricted to flow lines consisting of multi-server workstations with finite buffers; we expect, however, that the scope of this approach goes (far) beyond this class of manufacturing systems.

The aggregate model has been developed keeping integrated processing equipment in mind. A follow-up paper by Veeger et al. [2008] demonstrates how the present methodology can be applied to workstations with integrated processing tools in a semiconductor manufacturing environment, where commonly used $G/G/m$ approximations perform unsatisfactorily.

Acknowledgments

This research is supported by the Technology Foundation STW, applied science division of NWO and the technology programme of the Dutch Ministry of Economic Affairs. The authors would furthermore like to thank Erjen Lefeber and Casper Veeger of the Eindhoven University of Technology.

References

A. Arisha and P. Young. Intelligent simulation-based lot scheduling of photolithography toolsets in a wafer fabrication facility. In *2004 Winter Simulation Conference*, pages 1935–1942, 2004.

S. Asmussen. *Applied Probability and Queues*. Springer, New York, 2nd edition, 2003.

R.J. Boucherie. Norton's equivalent for queueing networks comprised of quasi-reversible components linked by state-dependent routing. *Performance Evaluation*, 32:83–99, 1998.

K.M. Chandy, U. Herzog, and L. Woo. Parametric analysis of queuing networks. *IBM Journal of Research and Development*, 19:36–42, 1975.

Y. Dallery and S.B. Gershwin. Manufacturing flow line systems: a review of models and analytical results. *Queueing Systems: Theory and Applications*, 12:3–94, 1992.

A.T. Hofkamp and J.E. Rooda. χ *Reference manual*. Systems Engineering Group, Eindhoven University of Technology, 11 2002. URL:http://se.wtb.tue.nl/.

W.J. Hopp and M.L. Spearman. *Factory physics: foundations of manufacturing management*. London: Irwin McGraw-Hill, 1st edition, 1996.

W.J. Hopp and M.L. Spearman. *Factory physics: foundations of manufacturing management*. London: Irwin McGraw-Hill, 2nd edition, 2001. ISBN 0-256-24795-1.

J.H. Jacobs, P.P. van Bakel, L.F.P. Etman, and J.E. Rooda. Quantifying variability of batching equipment using effective process times. *IEEE Transactions on Semiconductor Manufacturing*, 19(2):269–275, 2006.

J.H. Jacobs, L.F.P. Etman, E.J.J. van Campen, and J.E. Rooda. Quantifying operational time variability: the missing parameter for cycle time reduction. In *2001 IEEE/SEMI Advanced semiconductor manufacturing conference*, pages 1–10, 2001.

J.H. Jacobs, L.F.P. Etman, E.J.J. van Campen, and J.E. Rooda. Characterization of operational time variability using effective process time. *IEEE Transactions on Semiconductor Manufacturing*, 16:511–520, 2003.

A.A.A. Kock, L.F.P. Etman, and J.E. Rooda. Effective process time for multi-server flowlines with finite buffers. *IIE Transactions*, 40(3):177–186, 2008a.

A.A.A. Kock, F.J.J. Wullems, L.F.P. Etman, I.J.B.F. Adan, F. Nijsse, and J.E. Rooda. Performance evaluation and lumped parameter modelling of single server flowlines subject to blocking: an effective process time approach. *Computers and Industrial Engineering*, 54(4):866–878, 2008b.

M. Nayani and M. Mollaghasemi. Validation and verification of the simulation model of a photolithography process in semiconductor manufacturing. In *1998 Winter Simulation Conference*, pages 1017–1022, 1998.

E.L. Norton. Design of finite networks for uniform frequency characteristic. taken from http://www.ece.rice.edu/˜dhj/norton/ (last visited 11-12-2007), 1926.

N.G. Pierce and M.J. Drevna. Development of generic simulation models to evaluate wafer fabrication cluster tools. In *Advanced Semiconductor Manufacturing Conference and Workshop, ASMC, 1992*, pages 874–878. IEEE/SEMI, 1992.

Y. Rhee. Some notes on the reduction of network dimensionality in nested open queueing networks. *European Journal of Operational Research*, 174:124–131, 2006.

J.G. Shanthikumar, S. Ding, and M.T. Zhang. Queueing theory for semiconductor manufacturing systems: a survey and open problems. *IEEE Transactions on Automation Science and Engineering*, 4(4):513–522, 2007.

W.J. Stewart and G.A. Zeiszler. On the existence of composite flow equivalent markovian servers. *ACM Sigmetrics Performance Evaluation Review*, 9(2): 105–116, 1980.

A. Thomasian and B. Nadji. Aggregation of stations in queueing network models of multiprogrammed computers. *ACM Sigmetrics Performance Evaluation Review*, 10(3):86–104, 1981.

C.P.L. Veeger, L.F.P. Etman, J. van Herk, and J.E. Rooda. Generating cycle time-throughput curves using ept-based aggregate modeling. In *2008 IEEE/SEMI Advanced Semiconductor Manufacturing Concerence (ASMC)*, Boston, 2008.

M. Vijfvinkel, A.A.A. Kock, L.F.P. Etman, M. van Vuuren, and J.E. Rooda. Performance measurement and prediction of finitely buffered asynchronous assembly lines: an effective process time approach. *submitted*, 2007.

M. van Vuuren. *Performance Analysis of Manufacturing Systems: Queueing Approximations and Algorithms*. PhD thesis, Eindhoven University of Technology, Department of Mathematics and Computer Science, 2007.

M. van Vuuren, I.J.B.F. Adan, and S.A.E. Resing-Sassen. Performance analysis of multi-server tandem queues with finite buffers and blocking. *OR Spektrum*, 27:315–339, 2005.

S.C. Wood. Simple performance models for integrated processing tools. *IEEE Transactions on Semiconductor Manufacturing*, 9:320–328, 1996.