Universität Würzburg
Institut für Informatik
Research Report Series

Discrete–time Analysis of Batch Servers with Bounded Idle Time and Two Job Classes

M. Dümmler and **A. Schömig**

Report No. 162 January 1997

Bayerische Julius–Maximilians Universität Würzburg,
Institut für Informatik, Lehrstuhl für Informatik III,
Am Hubland, D-97074 Würzburg, Federal Republic of Germany
Tel.: +49-931-8885510, Fax: +49-931-8884601,
e-mail: {duemmler|schoemig}@informatik.uni-wuerzburg.de

Abstract: *Production processes where several jobs are loaded into a single machine and processed simultaneously are common in semiconductor manufacturing. These machines are called* batch servers. *In this paper, we extend previous studies of batch servers introducing two job classes. One job class receives privileged service to model prioritized jobs.*

To study the impact of the percentage of prioritized jobs on the system performance we utilize the discrete–time analysis technique, widely used for the performance evaluation of modern communication systems. The intent of this paper is to present first results and to demonstrate the feasibility of discrete–time analysis technique for operational research in semiconductor manufacturing rather than to attempt any rigor mathematical treatment of the numerical algorithms.

Keywords: Stochastic Manufacturing Models, Batch Server, Discrete–time Analysis, Semiconductor Manufacturing.

1 Introduction

Batch servers are machines that can process several jobs at a time. Examples of batch service machines in semiconductor wafer fabrication are diffusion and oxidation ovens. These operations are very time-consuming (Johri 1992). Since it is inefficient to run a long operation without utilizing the full capacity of the machine it may seem desirable always to load full batches, i.e., the maximum number of jobs. Notably, additional jobs cannot be added once the process has been started. A batch server might have to wait a long time for jobs to form a full batch, leaving the machine idle. This waiting time adds to the cycle time of a job and downstream stages face starvation.

For semiconductor manufacturing is among the most complex manufacturing processes, scheduling and sequencing decisions are not necessarily straightforward when there are batch processing machines within a semiconductor wafer production line. To find a fair trade–off between job cycle time and server utilization a nontrivial decision must be made whether to start the partial batch once the machine becomes available or to wait for additional jobs.

(Neuts 1967) presented an analytical study of batch service systems that operate according to a minimum batch size rule. (Gold 1992) and (Tran-Gia et al. 1993) investigate batch service systems in push and pull manufacturing environments using embedded Markov chain techniques. (Glassey and Weng 1991), and (Fowler et al. 1992) approach the problem of controlling batch processing systems in semiconductor wafer fabrication from a operation manager's point of view. These studies utilize online data of the specific facilities under consideration for decision making. (Tran-Gia and Schömig 1996) introduce the discrete–time analysis technique to the performance analysis of a batch service system. They compare the *minimum batch size rule* with a *limited idle time policy* in the single product case. Discrete–time analysis techniques were first used to analyse basic queuing models like single server systems as presented by (Ackroyd 1980) and (Hübner 1993). This technique is used for the analysis of high–performance communication networks.

In this paper we study a batch service system where two clases of jobs are processed: regular jobs and prioritized jobs to model so–called *hot lots*, that receive preferred service. We present numerical approximations using our new algorithm comparing these with simulations of a typical system configuration.

2 Model and Analysis

Batch Service Model

We consider the batch service system depicted in Fig. 1. There exist two types of jobs in our model. Jobs of type I (type II) arrive at queue I (queue II) with rate λ_1 (λ_2). Their interar-

rival time is geometrically distributed according to $a(k) = (1-q_{A_i})\, q_{A_i}^{k-1}, \quad k = 1, 2, \ldots$. Queue I (queue II) has a finite capacity of S_1 (S_2) jobs. If there are already S_1 (S_2) jobs in queue I (queue II), a newly arriving job of type I (type II) will balk. The probability of this occurence is denoted by $p_{B,1}$ ($p_{B,2}$). Jobs of type I will also be referred to as *hot lots*, whereas jobs of type II are *standard lots*. Jobs of different types cannot be served simultaneously. Service times are assumed to be random to account for variability due to setup times, downtimes, material shortage, and/or operator unavailability. Nevertheless, deterministic service times can also be used in our algorithm.

The distribution of service time only depends on the type of jobs in the server, not on

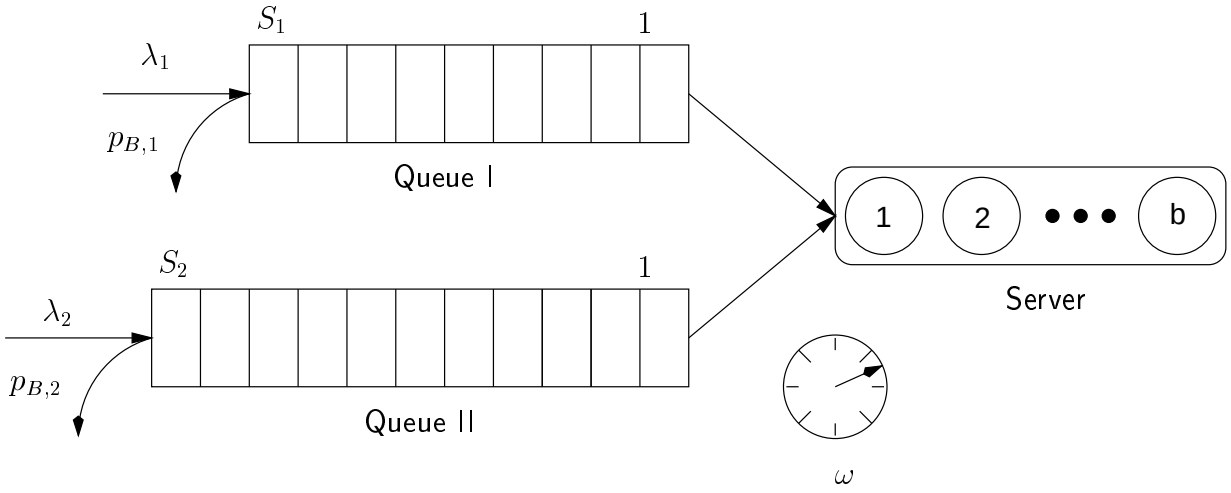

Figure 1: Batch Server with Bounded Idle Time and Two Job Classes

their number. Mean service time for type I jobs is $E[H_1]$, for type II jobs it is $E[H_2]$. Within one queue, jobs are served in FIFO order. Furthermore, no job can join service while the server is busy.

At the end of a service period, a new service is started according to the following strategy:

- ◇ If there is at least one job in queue I, all jobs in queue I, but at most b jobs will move on to the server and will be served.
- ◇ If queue I is empty and there are at least b jobs in queue II, b jobs from queue II will be served.
- ◇ If none of the above applies, a timeout intervall of constant length ω is started.
- ◇ Within the timeout interval, if one job arrives at queue I, the timeout intervall is stopped and this job will be served.
- ◇ If b jobs arrive at queue II within the timeout interval and prior to an arrival at queue I, the timeout intervall is stopped and these b jobs will be served.

- ◇ If none of the latter two holds, the timeout interval ends, and the jobs in queue II (if any) will be served.
- ◇ If both queues are empty when the timeout interval ends, the first job arriving to queue I or queue II will be served.

Note that the batching strategy for queue I can be described as *greedy*, while for queue II a *full batch* strategy is implemented.

Discrete–time Analysis

Random variables considered are of discrete–time nature, i.e. samples are multiples of $\Delta\, t$, for the time axis is divided into intervals of a fixed length $\Delta\, t$, the *unit length*. We assume that the discrete distributions have finite length. Given a random variable X, we denote the distribution (probability mass function) of X by $x(k) \;=\; \Pr\{X \;=\; k\}\,,\; -\infty < k < \infty\,,$ the distribution function by $X(k) \;=\; \sum_{i=-\infty}^{k} x(i)\,,$ the mean by $E[X]$, and the coefficient of variation by c_X.
Figure 2 illustrates a possible evolution of the two–dimensional state space of the batch server model. We observe the state process, $(X^\star(t), Y^\star(t))$, of the system, where $X^\star$ and $Y^\star$ denote the number of jobs in queue I and queue II, respectively. A Markov chain can be embedded at the end of a service period. Henceforth we first investigate the state process of the queue at these embedding points, $(X_n(t), Y_n(t))$.

For the description of the system we introduce the following random variables $(i = 1, 2)$:

A_i	: Interarrival time of jobs of type i
H_i	: Service time for jobs of type i
R_i	: Number of arriving jobs of type i during a service period
$\mathcal{T}_i$	: Number of arriving jobs of type i during the timeout interval of length ω
(X, Y)	: Number of waiting jobs in queue I and II at embedded points
$(X^\star, Y^\star)$	: Number of waiting jobs at arbitrary instants

The distribution $r_i(k)$ of arrivals of job type i during a service period, given $q_{A_i} = (E[A_i] - 1)/E[A_i]$, is

$$r_i(k) = \sum_{m=k}^{\infty} h(m) \binom{m}{k} (1 - q_{A_i})^k q_{A_i}^{m-k}\,,$$

due to the geometric distribution
$a(k) \;=\; (1 - q_{A_i})\; q_{A_i}^{k-1}\,, \quad k \;=\; 1, 2, \ldots\,,$ of the arrival process.

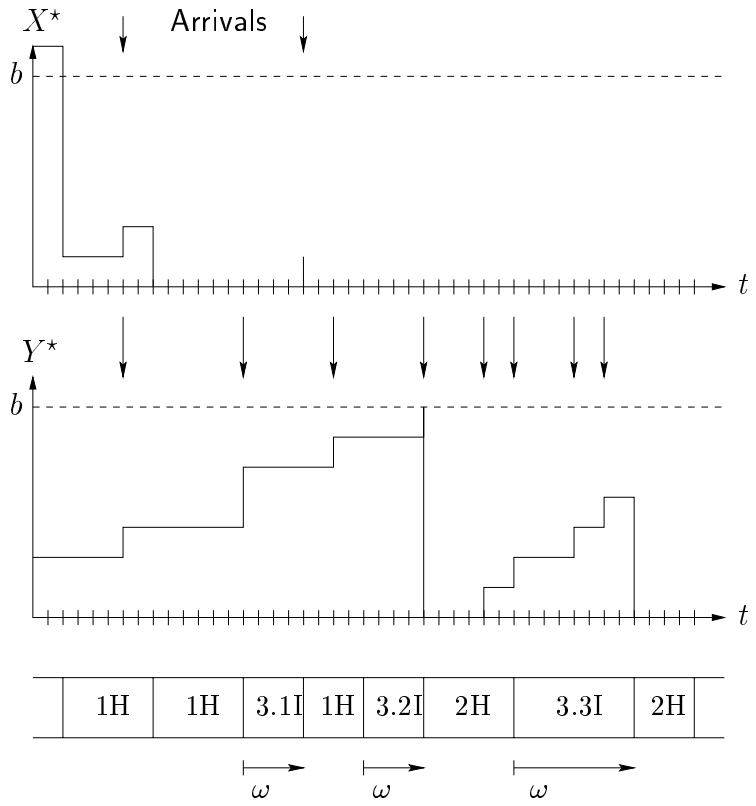

Figure 2: *State space evolution*

(Tran-Gia and Schömig 1996) showed how to use the discrete–time analysis technique to conduct performance analysis of a batch service system having one job class:

- ◇ Derive the state probabilities at the embedded points.
- ◇ Calculate probabilities of certain state description interval types.
- ◇ Combine results to derive state probabilities at arbitrary instants.

To calculate efficiently the steady state probability vector we introduce the following *evolution algorithm*. The discrete–time operators introduced by (Tran-Gia 1986) had to be adjusted to probability distributions on two–dimensional state spaces and new operators had to be added. Here, $z(k,l)$ is the probability that $X = k$ and $Y = l$. The second index of the operators denotes the dimension, to which the operator is applied. Hence the definitions following are given for the first dimension.

$$\sigma_{m,1}(z(k,l)) = \begin{cases} 0 & : k < m \\ z(k,l) & : k \geq m \end{cases} \text{, for } l = 0, \ldots, \infty$$

$$\sigma^{m,1}(z(k,l)) = \begin{cases} z(k,l) & : k \leq m \\ 0 & : k > m \end{cases} \text{, for } l = 0, \ldots, \infty$$

$$\pi_{m,1}(z(k,l)) = \begin{cases} 0 & : k < m \\ \sum_{i=-\infty}^{m} z(i,l) & : k = m \\ z(k,l) & : k > m \end{cases} \text{, for } l = 0, \ldots, \infty$$

$$\pi^{m,1}(z(k,l)) = \begin{cases} z(k,l) & : k < m \\ \sum_{i=m}^{\infty} z(i,l) & : k = m \\ 0 & : k > m \end{cases} \text{, for } l = 0, \ldots, \infty$$

$$\delta_{m,1}(z(k,l)) = z(k-m,l) \text{ , for } l = 0, \ldots, \infty\,.$$

The operators can be defined in a similar way for the second dimension. For example

$$\pi^{m,2}(z(k,l)) = \begin{cases} z(k,l) & : l < m \\ \sum_{i=m}^{\infty} z(k,i) & : l = m \\ 0 & : l > m \end{cases} \text{, for } k = 0, \ldots, \infty$$

Fig. 3 shows a graphical representation of some of those operators. The white areas represent the original distribution, whereas the shaded areas show the distribution after applying the particular operator. The black bar represents the summation realized by the π–operator.

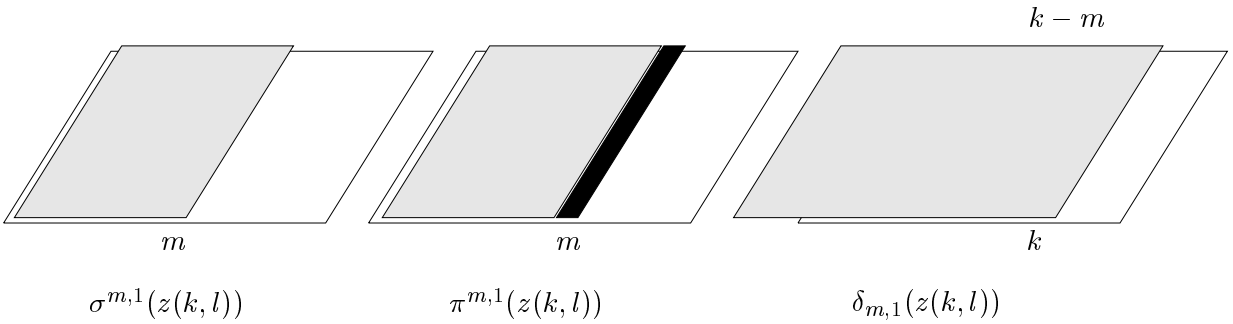

Figure 3: Graphical Representation of Operators

Using these operators, we can derive the evolution algorithm depicted in Fig. 4 which describes the development of states going from embedding point n to embedding point $n+1$. The symbol '$\circledast$' denotes the discrete convolution operation, whereas '$\oplus$' denotes the place where the distributions, which were derived in the individual branches are being added. Each branch is weighted by its factor p_j, which represents the probability of the event corresponding to the individual branches.

To arrive at the distribution at arbitrary points in time, one has to identify the different interval types that occur between embedded points, as described by (Gold 1992). This leads to five different interval types, presented in the lower portion of Fig. 2.

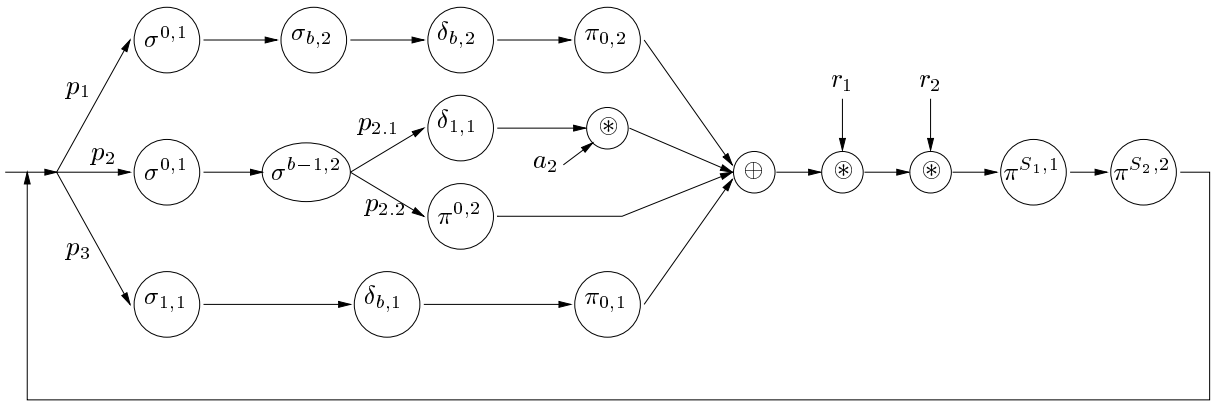

Figure 4: Evolution Algorithm

3 Results

Due to the limited size of this paper we have to confine ourselves to the presentation of few results chosen from the multitude of possible configurations and parameter sets. We observed always a good conformation between analytical and simulative results with relative errors significantly smaller than 5% for most parameter sets. Figure 5 shows the mean waiting time before service (given in terms of $E[H]$) and Fig. 6 the balking probability for both types of jobs (balking probabilities for type I – jobs were zero for coefficient of variation $c_{H_1} = 0.0$), observed for traffic intensities ρ ranging from 0.1 up to 1.1, where

$$\begin{aligned} \rho &:= q \cdot \rho_1 + (1-q) \cdot \rho_2 \\ \rho_i &:= E[H_i]/(b \cdot E[A_i]), i = 1, 2\,. \end{aligned}$$

The variable q is the fraction of jobs of type I in the product mix. The results are given for two different coefficients of variation of the service time distribution, given parameters $b = 8,\ S_1 = 16,\ S_2 = 32$. Mean service time for jobs of type I is 40 time units, for jobs of type II it is 50 time units. Product mix is chosen to be $q = 30\%$ for hot lots and $100\% - q = 70\%$ for regular lots. The length of the timeout interval is $\omega = 0.3 \cdot E[H_2]$ time units.

Firstly, it is obvious that the variability of the service time, denoted by the coefficient of variation, has a major influence on the behaviour of the system in terms of waiting time of jobs before service and balking. Secondly, we notice the a known growth of waiting time with increasing traffic intensity in the system for type II – jobs.

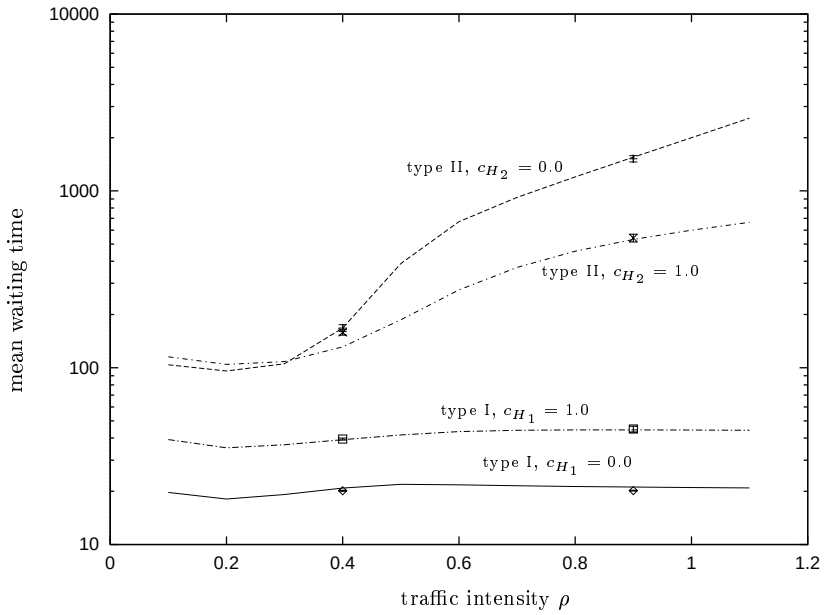

Figure 5: Mean Waiting Time of Jobs Before Service

Furthermore, note that the waiting times for type I – jobs are hardly impacted by a increase in traffic intensity. For traffic intensities higher than 0.3, a peculiar behavior can be observed for waiting times and balking probabilities for jobs of type II; the values observed for coefficient of variation 0.0 are higher than those observed for coefficient of variation 1.0.
Fig. 7 and Fig. 8 show the mean waiting times and balking probabilities for different values of the product mix. Here, the fraction of jobs of type I in the product mix ranges from 0.1 to 0.5. Results are displayed for traffic intensities 0.3 and 0.9, $c_{H_1} = c_{H_2} = 0.0$. All other parameters were the same as in Figures 5 and 6.

Here it can be seen that waiting times for type I – jobs are hardly influenced by changes in the product mix, especially when the load is high. Jobs of type II again show the opposite behavior; their waiting time increases faster for higher loads.

In Fig. 8, only results for type II – jobs are given since balking probabilities for type I – jobs were zero for all parameters.

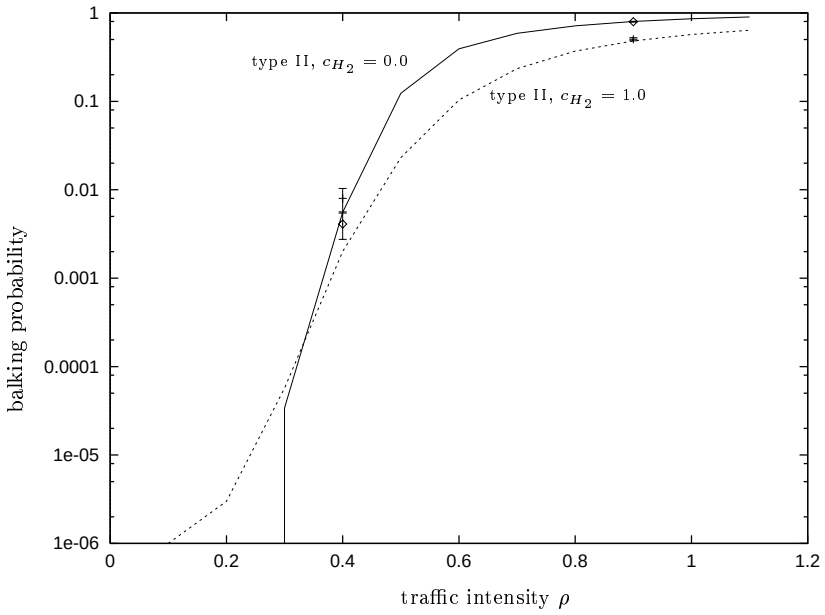

Figure 6: Balking Probability

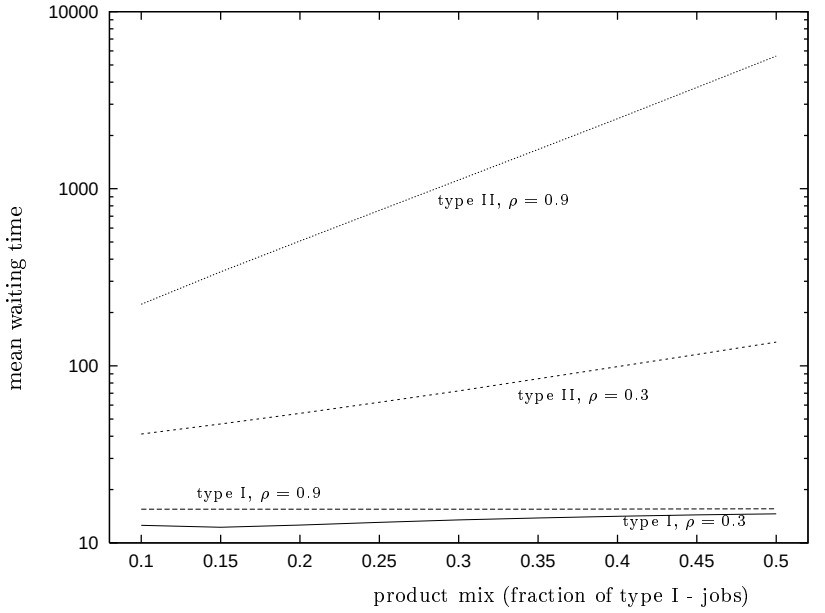

Figure 7: Effect of the Product Mix on the Mean Waiting Time

4 Conclusion and Outlook

In this paper, we demonstrated the feasibility of discrete–time analysis technique for the performance evaluation of a batch service system having two job classes. We employed the bounded idle time rule and discussed how given parameters influence performance

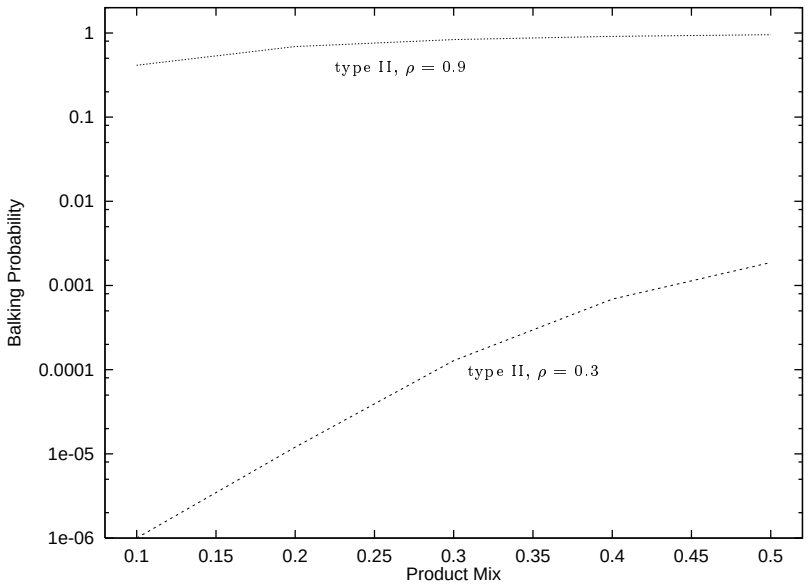

Figure 8: Effect of the Product Mix on the Balking Probability of Arriving Jobs

measures such as balking probability, waiting time, and work in process. Analytical results were validated by discrete event simulation.

We believe it would be advised to introduce *cost functions* to appreciate contradicting economic goals as low mean waiting times of jobs (i.e. small cycle times) and high server utilization (i.e. less manufacturing costs per job).

Further research has to be carried out to consider delays due to setup times as well as so–called *lookahead strategies*.

Acknowledgment

This work was supported by the Deutsche Forschungsgemeinschaft, grant # Tr 257/2–2.

References

Ackroyd, M. H. (1980, January). Computing the waiting time distribution for the G/G/1 queue by signal processing methods. *IEEE Transactions on Communications COM-28*(1), 52–58.

Fowler, J. W., D. T. Phillips, and G. L. Hogg (1992, May). Real-time control of multi-product bulk-service semiconductor manufacturing processes. *IEEE Transactions on Semiconductor Manufacturing 5*(2), 158–163.

Glassey, C. R. and W. W. Weng (1991, May). Dynamic batching heuristics for simultaneous processing. *IEEE Transactions on Semiconductor Manufacturing 4*(2), 77–82.

Gold, H. (1992). *Performance Modelling of Batch Service Systems with Push and Pull manufacturing Management Policies*. Ph. D. thesis, Universität Würzburg, Fakultät für Mathematik und Informatik.

Hübner, F. (1993). *Discrete-time Performance Analysis of Finite-capacity Queueing Models for ATM Multiplexers*. Ph. D. thesis, Universität Würzburg, Fakultät für Mathematik und Informatik.

Johri, P. K. (1992). Practical issues in scheduling and dispatching in semiconductor wafer fabrication. *Journal of Manufacturing Systems 12*(6), 474–485.

Neuts, M. F. (1967). A general class of bulk queues with poisson input. *Ann. Math. Stat. 38*, 759–770.

Tran-Gia, P. (1986). Discrete-time analysis for the interdeparture distribution of GI/G/1 queues. In O. J. Boxma, J. W. Cohen, and H. C. Tijms (Eds.), *Teletraffic Analysis and Computer Performance Evaluation*, pp. 341–357. Elsevier (North-Holland).

Tran-Gia, P., H. Gold, and H. Grob (1993). A batch service system operating in a pull production line. *Archiv für Elektronik und Übertragungstechnik 47*(5/6), 379–385.

Tran-Gia, P. and A. Schömig (1996, June). Discrete–time analysis of batch servers with bounded idle time. In *Modelling and Simulation 1996. (Proceedings of the European Simulation Multiconference)*, Budapest, pp. 999–1003.

Preprint-Reihe
Institut für Informatik
Universität Würzburg

Verantwortlich: Die Vorstände des Institutes für Informatik.

[112] G. Buntrock, und G. Niemann. *Weak Growing Context-Sensitive Grammars.* Mai 1995.

[113] J. García and M. Ritter. *Determination of Traffic Parameters for VPs Carrying Delay-Sensitive Traffic.* Mai 1995.

[114] M. Ritter. *Steady-State Analysis of the Rate-Based Congestion Control Mechanism for ABR Services in ATM Networks.* Mai 1995.

[115] H. Graefe. *Konzepte für ein zuverlässiges Message-Passing-System auf der Basis von UDP.* Mai 1995.

[116] A. Schömig und H. Rau. *A Petri Net Approach for the Performance Analysis of Business Processes.* Mai 1995.

[117] K. Verbarg. *Approximate Center Points in Dense Point Sets.* Mai 1995.

[118] K. Tutschku. *Recurrent Multilayer Perceptrons for Identification and Control: The Road to Applications.* Juni 1995.

[119] U. Rhein-Desel. *Eine „Übersicht“ über medizinische Informationssysteme: Krankenhausinformationssysteme, Patientenaktensysteme und Kritiksysteme.* Juli 1995.

[120] O. Rose. *Simple and Efficient Models for Variable Bit Rate MPEG Video Traffic.* Juli 1995.

[121] A. Schömig. *On Transfer Blocking and Minimal Blocking in Serial Manufacturing Systems — The Impact of Buffer Allocation.* Juli 1995.

[122] Th. Fritsch, K. Tutschku und K. Leibnitz. *Field Strength Prediction by Ray-Tracing for Adaptive Base Station Positioning in Mobile Communication Networks.* August 1995.

[123] R. V. Book, H. Vollmer und K. W. Wagner. *On Type-2 Probabilistic Quantifiers.* August 1995.

[124] M. Mittler, N. Gerlich, A. Schömig. *On Cycle Times and Interdeparture Times in Semiconductor Manufacturing.* September 1995.

[125] J. Wolff von Gudenberg. *Hardware Support for Interval Arithmetic - Extended Version.* Oktober 1995.

[126] M. Mittler, T. Ono-Tesfaye, A. Schömig. *On the Approximation of Higher Moments in Open and Closed Fork/Join Primitives with Limited Buffers.* November 1995.

[127] M. Mittler, C. Kern. *Discrete-Time Approximation of the Machine Repairman Model with Generally Distributed Failure, Repair, and Walking Times.* November 1995.

[128] N. Gerlich. *A Toolkit of Octave Functions for Discrete-Time Analysis of Queuing Systems.* Dezember 1995.

[129] M. Ritter. *Network Buffer Requirements of the Rate-Based Control Mechanism for ABR Services.* Dezember 1995.

[130] M. Wolfrath. *Results on Fat Objects with a Low Intersection Proportion.* Dezember 1995.

[131] S. O. Krumke and J. Valenta. *Finding Tree–2–Spanners.* Dezember 1995.

[132] U. Hafner. *Asymmetric Coding in (m)-WFA Image Compression.* Dezember 1995.

[133] M. Ritter. *Analysis of a Rate-Based Control Policy with Delayed Feedback and Variable Bandwidth Availability.* Januar 1996.

[134] K. Tutschku and K. Leibnitz. *Fast Ray-Tracing for Field Strength Prediction in Cellular Mobile Network Planning.* Januar 1996.

[135] K. Verbarg and A. Hensel. *Hierarchical Motion Planning Using a Spatial Index.* Januar 1996.

[136] Y. Luo. *Distributed Implementation of PROLOG on Workstation Clusters.* Februar 1996.

[137] O. Rose. *Estimation of the Hurst Parameter of Long-Range Dependent Time Series.* Februar 1996.

[138] J. Albert, F. Räther, K. Patzner, J. Schoof, J. Zimmer. *Concepts For Optimizing Sinter Processes Using Evolutionary Algorithms.* Februar 1996.

[139] O. Karch. *A Sharper Complexity Bound for the Robot Localization Problem.* Juni 1996.

[140] H. Vollmer. *A Note on the Power of Quasipolynomial Size Circuits.* Juni 1996.

[141] M. Mittler. *Two-Moment Analysis of Alternative Tool Models with Random Breakdowns.* Juli 1996.

[142] P. Tran-Gia, M. Mandjes. *Modeling of customer retrial phenomenon in cellular mobile networks.* Juli 1996.

[143] P. Tran-Gia, N. Gerlich. *Impact of Customer Clustering on Mobile Network Performance.* Juli 1996.

[144] M. Mandjes, K. Tutschku. *Efficient call handling procedures in cellular mobile networks.* Juli 1996.

[145] N. Gerlich, P. Tran-Gia, K. Elsayed. *Performance Analysis of Link Carrying Capacity in CDMA Systems.* Juli 1996.

[146] K. Leibnitz, K. Tutschku, U. Rothaug. *Künstliche Neuronale Netze für die Wegoptimierung in ATG Leiterplattentestern.* Juli 1996.

[147] M. Ritter. *Congestion Detection Methods and their Impact on the Performance of the ABR Flow Control Mechanism.* August 1996.

[148] H. Baier, K.W. Wagner. *The Analytic Polynomial Time Hierarchy.* September 1996.

[149] H. Vollmer, K.W. Wagner. *Measure One Results in Computational Complexity Theory.* September 1996.

[150] O. Rose. *Discrete-time Analysis of a Finite Buffer with VBR MPEG Video Traffic Input.* September 1996.

[151] N. Vicari, P. Tran-Gia. *A Numerical Analysis of the Geo/D/N Queueing System.* September 1996.

[152] H. Noltemeier, S.O. Krumke. *30. Workshop Komplexitätstheorie, Datenstrukturen und effiziente Algorithmen.* Oktober 1996.

[153] R. Wastl. *A Unified Semantical Framework for Deductive Databases.* Oktober 1996.

[154] R. Wastl. *A Vectorial Well-Founded Semantics for Disjunctive, Deductive Databases.* Oktober 1996.

[155] G. Niemann. *On Weakly Growing Grammars.* Oktober 1996.

[156] W. Nöth, U. Hinsberger, R. Kolla. *TROY — A Tree Oriented Approach to Logic Synthesis and Technology Mapping.* November 1996.

[157] R. Wastl. *Lifting the Well-Founded Semantics to Disjunctive, Normal Databases.* November 1996

[158] H. Vollmer. *Succinct Inputs, Lindström Quantifiers, and a General Complexity Theoretic Operator Concept.* November 1996

[159] H. Baier. *On the Approximability of the Selection Problem.* Dezember 1996

[160] U. Hafner, S.W.M. Frank, M. Unger, J. Albert. *Hybrid Weighted Finite Automata for image and video compression.* Januar 1997