Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

RE-ENACTMENT SIMULATION FOR BUFFER SIZE OPTIMIZATION IN SEMICONDUCTOR BACK-END PRODUCTION

Jelle Adan

Industrial Technology and Engineering Centre
Nexperia
Jonkerbosplein 52
Nijmegen, 6534 AB, THE NETHERLANDS

Stephan Sneijders

Department of Process and Energy
Delft University of Technology
Leeghwaterstraat 39
Delft, 2628 CB, THE NETHERLANDS

Alp Akcay
Ivo J.B.F. Adan

Department of Industrial Engineering
Eindhoven University of Technology
PO Box 513
Eindhoven, 5600 MB, THE NETHERLANDS

ABSTRACT

In this work, we propose a re-enactment simulation-based optimization method to determine the minimal total buffer capacity in an assembly line required to meet a target throughput. A distinguishing feature is the use of real-time event traces, in a fast fluid flow simulation model. Employing real-time event traces avoids the necessity to make restrictive modeling assumptions. The fluid simulation is combined with a multi start search algorithm. To demonstrate its effectiveness, the method is applied to a real-world use case in lead frame based semiconductor back-end manufacturing. This use case considers an assembly line consisting of six machines, for which the proposed method determines optimal buffer size configurations within several minutes of computational time.

1 INTRODUCTION

The back-end semiconductor production includes encapsulating a chip into a plastic package as well as testing and packaging. In this paper, we focus on a serial production line in back-end semiconductor production. The work-in-process inventory is kept in buffer areas between two successive machines to assure a smooth flow of products through the manufacturing line. Motivated by the back-end production of integrated circuits at an assembly and testing facility of Nexperia, we build a data-driven fluid simulation model to optimize the buffer sizes. The combination of data mining and simulation-based business intelligence has proven to be an effective and valuable tool within daily manufacturing operations at Nexperia. A recent work by Wilschut et al. 2014 builds a software tool and perform simulations of the production equipment, based on a fluid flow model, enabling the user to analyze which machine errors contribute most to the production-line downtime.

In Wilschut et al. 2014, the historical data on machine errors are considered and past realizations of machine downtimes and uptimes are recovered, which are subsequently used to obtain a best-fit probability distributions on the uptimes and downtimes of each machine. The resulting distributions are used as input

models to generate the sample paths in the simulation model. In the present paper, contrary to generating simulation input variates from best-fit input distributions, we adopt the data re-enactment simulation (also known as trace-driven simulation) where data traces are collected on the system inputs and then these traces are directly used to drive the simulation model; see Jack et al. 2001 and Sargent 2013. The main motivation behind this approach is the availability of large amount of well-structured historical data on the production line and machine errors as well as the continuous flow of real-time data on the current status of the line. Using the real-world data to drive the simulation allows us to capture many complex relationships between the machine-error types and the production throughput, which can be hard to capture with parametric input models. That is, the main advantage of re-enactment simulation is that there is no need to make assumptions concerning the distributions and independence of the machine uptimes and downtimes.

By following Wilschut et al. 2014, we use fluid flow simulation to model the discrete flow of products through the assembly line similar to a continuous fluid flow. This modeling assumption is viable in this environment because of the high production rates in the machines. It also makes the simulation very efficient without sacrificing the accuracy significantly. In the buffer-size optimization model, we assume that the size of each buffer is a continuous-valued decision variable. We adopt a gradient-based optimization approach where the performance measure throughput is estimated via simulation; we refer to reader Jian and Henderson 2015 for a recent tutorial on simulation-based optimization methods. Since the size of the historical data is finite, the use of historical data directly in the simulation sample-path generation induces an additional layer of uncertainty, making the simulation outputs (i.e., the throughput realization in the production line) themselves random. This makes finding the stochastic gradient estimators itself a challenging problem; we refer the reader to Chau et al. 2014 and Fu 2015 for recent developments in gradient-based simulation optimization methods. In this paper, we develop an algorithm that iteratively increases the length of the historical data until a buffer-size allocation that satisfies a target throughput value is achieved at minimum total buffer size. Our preliminary results shows that the algorithm can effectively optimize the buffer configuration. Since the algorithm uses the same historical data for each feasible buffer configuration, it reduces the variance of the throughput difference between multiple configurations, leading to faster convergence of the algorithm.

There is an extensive literature on the optimization of buffer sizes in manufacturing lines, see Gershwin and Schor 2000. Buffer size optimization requires efficient search algorithms to select the proper configuration (Gershwin and Schor 2000; Shi and Men 2003; Shi and Gershwin 2016; Ng et al. 2017; Dolgui et al. 2017) and also fast methods to evaluate the performance of a given configuration. Performance evaluation is often based on analytical approximations (Dallery and Gershwin 1992; Liberopoulos et al. 2006), which are typically much faster than discrete-event simulation. As explained above, the optimization approach proposed in this paper is a multi start gradient method, based on a fast trace-driven simulation.

The outline of this paper is as follows. First, the general concepts behind the back-end production process is discussed in Section 2. The input data collection procedure is explained in Section 3, and the fluid flow simulation model is explained in Section 4. The buffer-size allocation algorithm is introduced in Section 5, and a real-world case study is presented in Section 6 for the illustration of this algorithm. Section 7 provides some concluding remarks with future research directions.

2 BACK-END PRODUCTION PROCESS

At Nexperia's semiconductor back-end assembly sites, semi-finished ICs are encapsulated in a supporting case to prevent corrosion and physical damage. This case, which is often referred to as "package", supports the electrical contacts that connect it to a circuit board. A large part of this packaging process is done on assembly lines. In Figure 1 a schematic representation of such an assembly line is shown.

From left to right: at the beginning of an assembly line empty lead frame, a (metal) support structure, is fed to the first machine in the sequence. This machine is a so-called die bonder (DB). It picks up semi-finished ICs, so-called dies, from a wafer and bonds them onto the lead frame. Once a wafer is fully consumed, an operator must provide a new wafer to the machine. An assembly line may consist of multiple

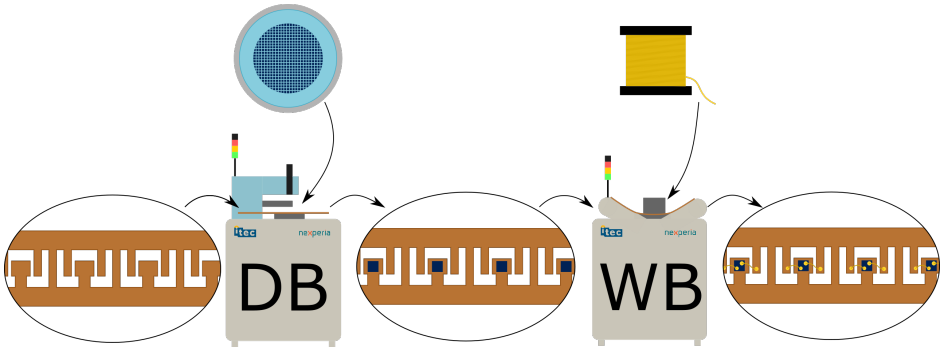

Figure 1: A production line consists of multiple die bonders (DB) and wire bonders (WB) in series with buffers between. Die bonders attach the dies onto a lead frame and wire bonders construct the electrical connection between the die and the lead frame.

die bonders in series. At the next stage, the lead frame, that now contains bonded dies, flows through one or more wire bonders. These machines establish electrical interconnections (wires) between the die and a specific position at the lead frame. Similarly, once the wire supply is finished, an operator must provide a new coil to the machine. In the final step, the device is encapsulated by a molding compound to protect it from the outside environment. In this work, we limit ourselves to the first two steps only; die bonding and wire bonding.

3 DATA COLLECTION

In 2001, Nexperia's Industrial Technology and Engineering Centre (ITEC) introduced its advanced warning and data collection system, abbreviated AWACS, which is used for the analysis of equipment performance. Equipment status monitoring forms the core of this system and is responsible for the collection of a wide range of equipment events. In this work, the focus is on equipment state change events. Equipment can be in one of one of three aggregated states: production, standby or down. In Figure 2 diagram with the three aggregate states and their respective sub-states is shown. The production state indicates that the machine is up and products are being produced. The standby state indicates that the machine is up but no products are being produced, the cause for which is indicated by one of the four standby sub-states. The sub-state wait input means that the upstream buffer is empty and that the upstream machine is down, i.e. the machine is starved. The wait output state means that the downstream buffer is full and the downstream machine is down. The wait material state indicates that the machine is out of some material, e.g. the wafer is consumed, and must wait for an operator to provide new supply. The down state indicates a machine is unable to produce and is also divided into multiple sub-states that indicate the reason. It is noted that the error state is also an aggregate state that contains hundreds of specific errors that can occur.

As mentioned, in this work the influence of buffer capacities on the overall performance of the system is studied. The buffer configuration determines how much time the system spends in the wait input and wait output states. Therefore, both the wait input and output states are removed from the original event log. These events are to be regenerated for each buffer configuration by means of a fluid simulation model, which is described in the next section.

4 FLUID SIMULATION MODEL

In the current work a fluid flow model is used to simulate the behavior of an assembly line. These lines consist of several machines with buffers in between.

We focus here on Nexperia machines, but the technique is applicable to any sequential system that consists of N machines and $N-1$ buffers. The overall process is schematically represented in Figure 3. In this figure squares represent machines while triangles depict buffers. Although the products running

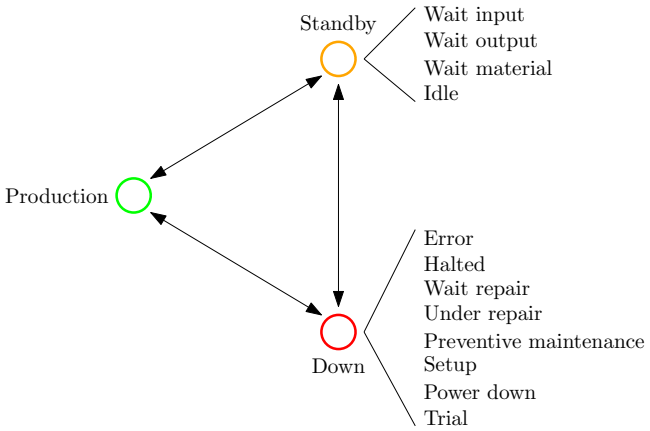

Figure 2: Overview of the equipment states.

through the process are discrete entities, the production rate is so fast that the products literally flow through the process as a continuum. Therefore, it is reasonable to model the process as a fluid flow. Typically, fluid flow models can be simulated much more efficiently when compared to discrete production models. The model used in the proposed framework is similar to the one described in Wilschut et al. 2014. For completeness, the fluid flow model is briefly discussed below.

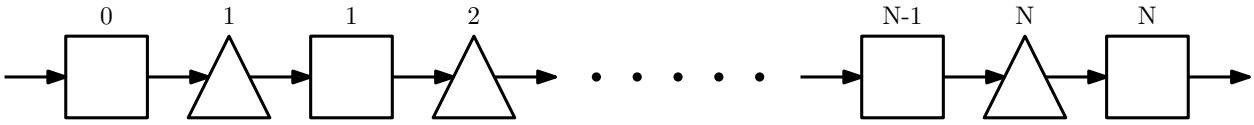

Figure 3: Schematic overview of an assembly line.

The machines and buffers are indexed as shown in Figure 3: machines have indexes ranging from 0 to N and every buffer has the same index as the first machine downstream of the buffer. Since no buffer is situated upstream of the 0^{th} machine, buffer indexes range from 1 to N, i.e. buffer i is between machine $i-1$ and i. Every machine i has its own maximum production rate v_i^{max} in products per time unit and every buffer i has its own product storage capacity c_i in absolute number of products. The actual rate of machine i can vary between 0 and v_i^{max} as it is dependent on the state of all other components in the system. Machines are either in the up or down state, while buffers can be full, empty or neutral. A state change of one component, buffer or machine, may induce a state change on all other constituents of the system. The fluid flow model simulates the production process with discrete events. In the next sections, the behavior of buffers and machines is described in more detail.

4.1 Buffer Behavior

Each buffer in the system has a certain storage capacity. Depending on the relative production rates of the machines directly upstream and downstream of a specific buffer, its product content either increases or decreases. This is true as long as the buffer is in the neutral state. If a buffer reaches the maximum capacity or runs empty, the buffer state is changed to full or empty respectively. Such a state change influences neighboring machines. In case buffer i runs empty, machine i will adjust its production rate to that of machine $i-1$. If machine $i-1$ is down for some reason the rate of machine i simply becomes zero. On the other hand, when a neutral buffer runs full, the production rate of machine i becomes v_i^{max} and the production rate of machine $i-1$ is adjusted to the rate of machine i.

4.2 Machine Behavior

The machines can have two different states, either up or down. If a machine is down, it is not able to produce products due to of problems with the machine itself (e.g. maintenance or errors). When a machine is up, it produces products with a rate that ranges between 0 to v_i^{max} products per time unit, depending on the other machines and buffers in the system. If machine i is up and its downstream buffer $i+1$ is full, its production rate is the same as machine $i+1$ (and if machine $i+1$ is down, its rate is 0). In case the downstream buffer is empty, its rate is dictated by the upstream machines, i.e. if buffer i contains any products, then machine i will produce at maximum rate. However, if buffer i is empty machine i produces at the same rate as machine $i-1$. A trace of up and down times for every machine serves as input for the model. The simulation always starts with all buffers being empty and all machines in the up state. When the production rate of a machine is zero, the up time is frozen until it restarts production, i.e. a machine can only breakdown while in production.

5 MULTI START ALGORITHM

The objective of the multi start algorithm is to minimize the total buffer capacity of an assembly line required to meet a certain target product throughput performance. The fluid flow model requires several input parameters: a set of maximum machine rates $V = \{v_1^{max}, \ldots, v_N^{max}\}$ and a set of buffer capacities $C = \{c_1, \ldots, c_{N-1}\}$. For a given input, the fluid simulation model determines the throughput performance of the system.

The multi start algorithm is initialized by assigning a very large buffer capacity to all the buffers in the system, simply to represent a system with infinite buffer capacities. This set of buffer capacities is inserted in the fluid flow simulation, which determines the throughput. This value is considered the maximum achievable throughput of the system. Hence, the target throughput is set as a fraction of this maximum througput. Next, a new set of buffer capacities is defined through multiplication of each buffer size with a step size α, a number between 0 and 1. This new set is again inserted in the simulation model. If the throughput is still above the required target, the new buffer configuration is feasible and accepted. In this case, another new set of buffer capacities is defined, again by multiplying each buffer size with α. This procedure continues as long as the target remains satisfied. As soon as the performance of the system drops below the specified target, the new set is rejected, α is reduced by a factor two, and the process is repeated. This procedure continues until α drops below a certain threshold, in this case 10^{-7} is chosen. The result of this procedure is a certain C that satisfies the target with minimal total buffer capacity and the special property that all c_i are equal.

After reaching the point where buffer capacities are equally minimized and throughput still meets the target, randomness is incorporated in the search in an attempt to obtain the overall minimum. First, the step size α is reset to 0.9. Secondly, to generate a random start, all buffer capacities are multiplied with a random number between 1 and 10. From this new set of buffer capacities a steepest descent method is used to find a buffer configuration that also meets the target throughput and possibly has a smaller overall capacity. This method starts by independently decreasing each buffer capacity c_i through multiplication with α, yielding $N-1$ different buffer configurations. The buffer configuration that results in the highest throughput is accepted as the new buffer configuration. This process continues until no feasible buffer configuration is found. At this point, the step size is reduced by a factor two. This procedure continues until α drops below a certain threshold. Similar to the previously described procedure, this threshold is set to 10^{-7}. As the name of the algorithm suggests (multi start), this process is repeated numerous times to obtain a complete overview of the solution domain.

Illustrative Example

The algorithm has been tested with an illustrative example. For this purpose a simple system of three machines and two buffers was used, since this allows for easy visualization of the results. Maximum

production rates of the machines were set to: $v_1^{max} = 5$, $v_2^{max} = 4$ and $v_3^{max} = 4$. The target throughput was set to 90% of the maximum throughput. The results of 50 starts are shown in Figure 4. All the data points correspond to the same throughput, i.e. the target throughput. Clearly, in order to reach the target there are several Pareto optimal solutions for the buffer capacity, measured in products (pdts). In our case, we define the optimal solution as the solution where the sum of the buffer capacities is minimized.

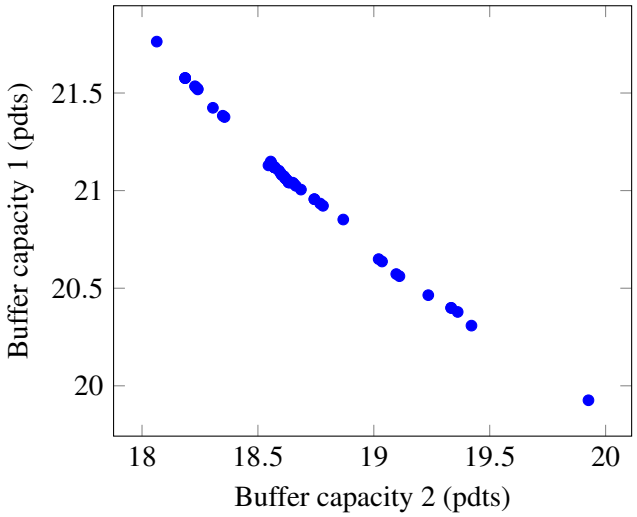

Figure 4: The capacity of buffer 1 versus the capacity of buffer 2 is shown for all the different random starts performed for the illustrative example. The target throughput was set to 90% of the maximum throughput.

6 REAL-WORLD CASE STUDY

In this section the proposed simulation-based optimization method is applied to one of Nexperia's assembly lines. This particular line consists of six machines in total: two die bonders and four wire bonders. Both the fluid simulation model and the optimization method are coded in C# 6.0 and computations are run on a computer with an Intel Core i7 processor running at 2.60 Ghz and 20 GB of RAM. For each of the machines, an event trace (from 2017) with a length of $15 \cdot 10^4$ time units is used. Recall that the wait input and wait output states are removed from the original trace since they are to be regenerated by the simulation model. The other standby states, i.e. the wait material and idle states, are treated as down states. Hence, the resulting event trace is an alternating sequence of up and down times. Note that when the removal of a wait input or wait output state leads to two subsequent up or down states, these are merged into a single up or down state, see Figure 5.

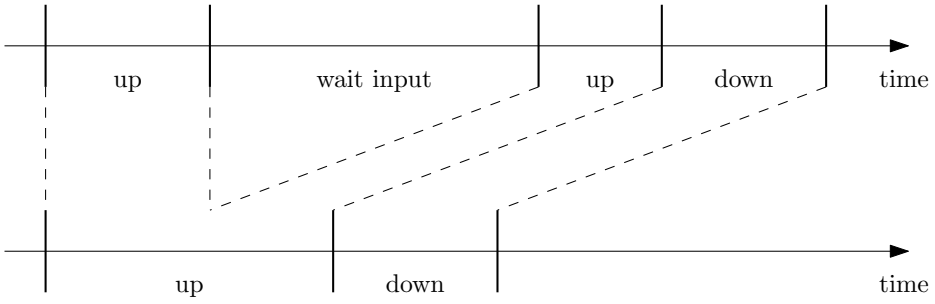

Figure 5: Illustration of filtering wait input and wait output times.

The production rate of each machine is taken from the available data, i.e. the total amount of products produced is divided by the total time spent in the production state during the whole event trace. In Table 1 the production rates are summarized. Additionally, the fraction of the time that the machine is in the up (production) state and the effective rate are shown.

Table 1: The production rates of each machine in products per time unit.

Machine #	Rate (pdts / time unit)	% Uptime (-)	Effective rate (pdts / time unit)
0	272.98	0.84	229.23
1	264.60	0.84	223.06
2	161.45	0.86	142.96
3	163.61	0.91	148.10
4	156.12	0.90	141.20
5	162.00	0.90	145.37

In the following experiment we set the target throughput to 90% of the maximum achievable throughput and determine the minimum total capacity required to achieve this target. Furthermore, the step size s in the optimization procedure is set to 0.1, the minimum step size s_{min} is set to 10^{-5} and a total of 100 starts are generated. Clearly, the length of the simulation must be sufficient in order to derive a reliable solution for buffer size allocation. Therefore, in the following experiment we increase the length of the simulation with increments of 10^4 time units, to investigate when the buffer configuration stabilizes.

In Figure 6 the optimal buffer configurations are shown for increasing simulation lengths. To give an indication of the speed of the fluid simulation, it takes approximately 43 milliseconds to perform the (longest) simulation of length 10^5 time units. For small simulation lengths the sizes of the buffers strongly deviate, possibly due to random effects in the first part of the trace. These effect appear to average out as the simulation length increases. From a simulation of length approximately $7 \cdot 10^4$ and onwards, the buffer allocations seem to stabilize. Clearly, the optimal buffer configuration assigns most of the capacity to buffers two and four. These are the buffers in front of the machines with the lowest effective production rate, see Table 1.

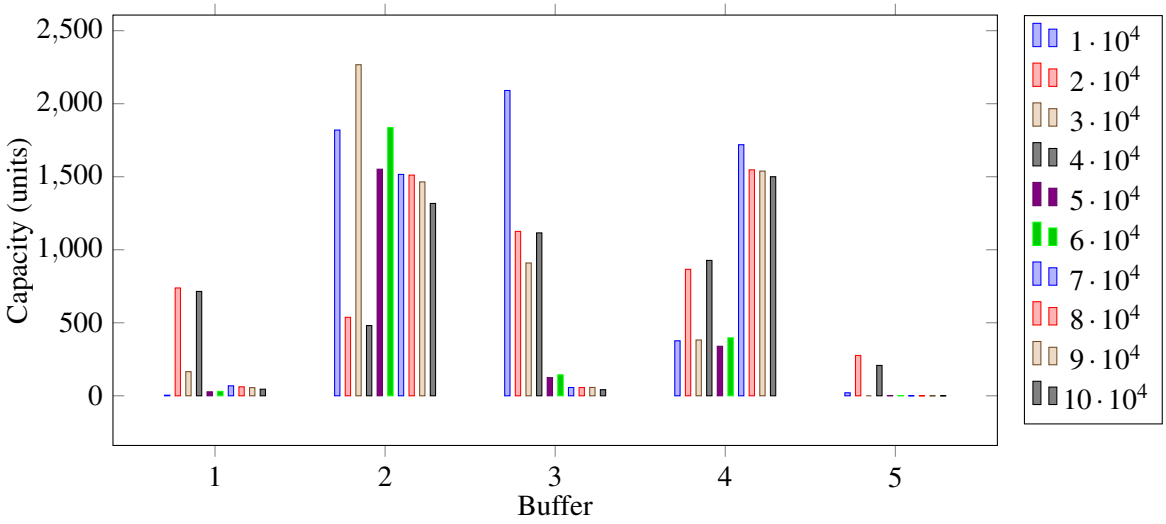

Figure 6: Optimal buffer size allocation for increasing simulation length.

7 CONCLUSIONS AND FUTURE WORK

It has been demonstrated that the combination of a re-enactment simulation model and a multi start optimization algorithm can effectively be applied to determine the optimal buffer size allocation for a discrete product assembly line. For this method, it is essential that the simulation model is very efficient, which has been achieved using a fluid flow simulation model. In a real-world scenario that considers an assembly line consisting of 6 machines the fluid simulation model requires less than 0.1 seconds to evaluate the throughput performance for a given configuration. Combined with the multi start algorithm, the optimal buffer size configuration can be determined within several minutes.

The use of this approach is not limited to buffer size allocation. Another possible application is to use the simulation model for prioritizing errors according to their impact on the throughput performance of the system, in order to steer maintenance engineers and structural improvement programs.

REFERENCES

Chau, M., M. C. Fu, H. Qu, and I. O. Ryzhov. 2014. "Simulation Optimization: A Tutorial Overview and Recent Developments in Gradient-based Methods". In *Proceedings of the 2014 Winter Simulation Conference*, edited by A. Tolk, S. Diallo, I. Ryzhov, L. Yilmaz, S. Buckley, and J. Miller, 21–35. Piscataway, NJ, USA: IEEE Press.

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

Dolgui, A. B., A. V. Eremeev, and V. S. Sigaev. 2017. "Analysis of a Multicriterial Buffer Capacity Optimization Problem for a Production Line". *Automation and Remote Control* 78(7):1276–1289.

Fu, M. C. 2015. "Stochastic Gradient Estimation". In *Handbook of Simulation Optimization*, edited by M. C. Fu, 105–147. New York, NY: Springer New York.

Gershwin, S. B., and J. E. Schor. 2000. "Efficient Algorithms for Buffer Space Allocation". *Annals of Operations Research* 93(1):117–144.

Jack, P., C. Russell, and B. Bert. 2001. "Validation of Trace-Driven Simulation Models: Bootstrap Tests". *Management Science* 47(11):1533–1538.

Jian, N., and S. G. Henderson. 2015. "An Introduction to Simulation Optimization". In *Winter Simulation Conference (WSC), 2015*, 1780–1794. IEEE Press.

Liberopoulos, G., C. Papadopoulos, B. Tan, J. Smith, and S. Gershwin. 2006. *Stochastic Modeling of Manufacturing Systems*. Springer-Verlag Berlin Heidelberg.

Ng, A. H., S. Shaaban, and J. Bernedixen. 2017. "Studying Unbalanced Workload and Buffer Allocation of Production Systems Using Multi-Objective Optimisation". *International Journal of Production Research* 55(24):7435–7451.

Sargent, R. G. 2013. "Verification and Validation of Simulation Models". *Journal of Simulation* 7(1):12–24.

Shi, C., and S. B. Gershwin. 2016. "A Segmentation Approach for Solving Buffer Allocation Problems in Large Production Systems". *International Journal of Production Research* 54(20):6121–6141.

Shi, L., and S. Men. 2003. "Optimal Buffer Allocation in Production Lines". *IIE Transactions* 35(1):1–10.

Wilschut, T., I. J. Adan, and J. Stokkermans. 2014. "Big Data in Daily Manufacturing Operations". In *Proceedings of the 2014 Winter Simulation Conference*, edited by A. Tolk, S. Diallo, I. Ryzhov, L. Yilmaz, S. Buckley, and J. Miller, 2364–2375. Piscataway, New Jersey: IEEE Press.

AUTHOR BIOGRAPHIES

JELLE ADAN is a Sr. Business Process Analyst at Nexperia ITEC and a PhD student at the Department of Industrial Engineering of the Eindhoven University of Technology, where he obtained his M.Sc. degree in Chemical Engineering in 2017. His current research interests are supply chain, manufacturing and chemical process optimization, and data mining. His email address is jelle.adan@protonmail.com.

STEPHAN SNEIJDERS is a PhD student at the Department of Process and Energy of Delft University of Technology. He obtained his MSc degree in Chemical Engineering in 2017. His current research interests are computational physics, computational fluid dynamics and chemical process optimization. His email address is s.sneijders@tudelft.nl.

ALP AKCAY is an Assistant Professor in the School of Industrial Engineering at Eindhoven University of Technology. He holds a Ph.D. in Operations Management from the Tepper School of Business at Carnegie Mellon University. His research interests include statistical decision making under uncertainty, simulation design and analysis, and approximate dynamic programming with applications in manufacturing, maintenance, and supply chain management. His email address is a.e.akcay@tue.nl.

IVO J.B.F. ADAN is a Professor at the Department of Industrial Engineering of the Eindhoven University of Technology. His current research interests are in the modeling and design of manufacturing systems, warehousing systems and transportation systems, and more specifically, in the analysis of multi-dimensional Markov processes and queueing models. His email address is i.adan@tue.nl.