Fortschritt-Berichte VDI

VDI

Reihe 13

Fördertechnik/
Logistik

Nr. 58

Dipl.-Ing. (FH) Jeanette Hamiga,
München

Increasing Output in Transfer Lines through Adaptive Buffer Operation

Increasing Output in Transfer Lines through Adaptive Buffer Operation

Von der Fakultät Maschinenbau der Technischen Universität Dresden

zur

Erlangung des akademischen Grades

Doktoringenieur (Dr.-Ing.)

genehmigte Dissertation.

Bei: Prof. Dr.-Ing. habil. Thorsten Schmidt,

Prof. Dr. T.H.W. (Thomas) Bäck

Dipl.-Ing. (FH) Jeanette Hamiga
geboren am 12.09.1987 in Regensburg

Tag der Einreichung: 04.03.2016

Fortschritt-Berichte VDI

Reihe 13

Fördertechnik/
Logistik

Dipl.-Ing. (FH) Jeanette Hamiga,
München

Nr. 58

Increasing Output in Transfer Lines through Adaptive Buffer Operation

VDI verlag

Hamiga, Jeanette
Increasing Output in Transfer Lines through Adaptive Buffer Operation
Fortschr.-Ber. VDI Reihe 13 Nr. 58. Düsseldorf: VDI Verlag 2016.
144 Seiten, 41 Bilder, 43 Tabellen.
ISBN 978-3-18-305813-6, ISSN 0178-9562,
€ 52,00/VDI-Mitgliederpreis € 46,80.

Keywords: Transfer Lines – Simulation – Multi-Objective-Optimization – Evolution Strategies – Automotive Production – Manufacturing System – Assembly – Buffer – Adaptive Buffer Operation – Decoupling Effect

This thesis is intended for engineers and scientists in the field of production. It deals with the goal of increasing output in (serial) transfer lines and simultaneously decreasing labor costs without need of change to the structure of the production system. For this the method adaptive buffer operation is developed, implemented and validated. Adaptive buffer operation proposes a different way of operating buffers, improving the decoupling effect of buffers. The buffers are filled to certain target fill levels at fixed moments (times of the day). Apart from the target fill levels further parameters, e.g. moments of intervention or the intervention frequency, are identified. To find out how to operate the buffers and which parameter combinations work best, a simulation-based optimization method is proposed. This method is split into the evaluative methodology, here simulation, and the generative technique of evolution strategies, solving the multi-objective optimization problem. Proof of performance is demonstrated while applying the method to a simulation model of an assembly of a German automotive plant.

Bibliographische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliographische Daten sind im Internet unter http://dnb.ddb.de abrufbar.

Bibliographic information published by the Deutsche Bibliothek
(German National Library)
The Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliographie (German National Bibliography); detailed bibliographic data is available via Internet at http://dnb.ddb.de.

Als Manuskript gedruckt. Printed in Germany.

ISSN 0178-9562

ISBN 978-3-18-305813-6

Acknowledgements

Diese Arbeit entstand im Rahmen des Doktorandenprogrammes ProMotion der BMW AG in Zusammenarbeit mit dem Institut für Technische Logistik und Arbeitssysteme der Fakultät Maschinenwesen an der Technischen Universität Dresden.

Ich bedanke mich herzlich bei Prof. Dr.-Ing. habil. Thorsten Schmidt, meinem Doktorvater und Inhaber der Professur für Technische Logistik, für die Übernahme der Betreuung und für das mir entgegengebrachte Vertrauen. Mein besonderer Dank gilt Herrn Prof. Dr. Thomas Bäck von der Natural Computing Research Group am Leiden Institute of Advanced Computer Science (LIACS) für die Übernahme des Koreferats als auch für die anregenden Diskussionen. Die daraus hervorgehenden Impulse haben die Arbeit stets bereichert. Herrn apl. Prof. Dr.-Ing. habil. Michael Völker danke ich für die Unterstützung und Begleitung während der Entstehung dieser Arbeit.

Darüber hinaus bedanke ich mich insbesondere bei Jan Knau, meinem Betreuer, sowie bei allen Kollegen der BMW AG, die mich während meiner Zeit als Doktorandin unterstützt haben. Besonders danken möchte ich hier Markus Ganser für die zahlreichen fachlichen Diskussionen sowie für seine Übernahme der Rolle des Advocatus Diaboli, ohne die diese Dissertation in dieser Form nicht möglich gewesen wäre.

Auch den Kollegen der divis intelligent solutions GmbH, insbesondere Herrn Christophe Foussette und Dr.-Ing. Peter Krause danke ich sehr für die hervorragende Zusammenarbeit, die konstruktive Kritik und Diskussion.

Mein tiefster Dank gilt meiner Familie und meinen Freunden, die mich auf meinem Weg unterstützt haben.

München, März 2016 Jeanette Hamiga

Index

Nomenclature and acronyms

ADDX	Accelerated DDX
AG	Aktiengesellschaft
BMW	Bayerische Motoren Werke
CVA	ClearVu Analytics, software used for optimization
DDX	Dallery-David-Xie
DLL	Dynamic Link Library
DoE	Design of Experiments
DR	Derandomized
EHPP	environmental HPP
FIFO	First In First Out
HPP	hedging point policy
min	minimum
min	minute
NSGA	Non-dominated Sorting Genetic Algorithm
PHPP	prioritized HPP
R	Range (of the target buffer level)
RSM	Response Surface Methodology
s	second
SDHPP	state depended HPP
T	Target buffer level

tol	tolerated
VW	Volkswagen
V&V	Verification and Validation
WIP	work-in-progress
xml	Extensible Markup Language

Notations

$\boldsymbol{x}$	vector $\boldsymbol{x}$
$\boldsymbol{u} \otimes \boldsymbol{v}$	element-wise multiplication of vectors $\boldsymbol{u}, \boldsymbol{v}, \in \mathbb{R}^n$ $\boldsymbol{u} \otimes \boldsymbol{v} = \boldsymbol{w}$ where $\boldsymbol{w} \in \mathbb{R}^n$ and $w_i = u_i \cdot v_i$ for $i \in \{1, \dots, n\}$
$\boldsymbol{x}^T$	transposition of vector $\boldsymbol{x}$
x_i	indexed component of a vector $\boldsymbol{x} = (x_1, \dots, x_n)^T \in \mathbb{R}^n$

Symbols

η	availability of system in %
α	significance level
Optimization	
k	index for constraints $k = 1, 2, \dots, r$
l	index for constraints $l = 1, 2, \dots, m$
K	total number of inequality constraint functions
L	total number of equality constraint functions
V	feasible set of the original problem
M	total number of objective functions
m	index for objective functions
n	number of parents to be optimized
$\mathbf{x}$	offspring vector
i	index number of individual / solution
j	index number of individual / solution
Evolutionary Algorithms	
t	continuous generation index $t = 0, 1, 2, \dots$
P_t	population at generation t
Q_t	offspring population at generation t
p	individual of population P_t
q	individual of population Q_t

Ψ	finite set of strategy parameters
λ	number of offspring
μ	number of parents
γ	number of generations
ρ	number of parents participating in creating offspring
κ	maximum age of an individual

DR2

$\boldsymbol{\phi}_i$	fitness of individual i
$\boldsymbol{z}_i$	random vector of the multivariate normal distribution
$\boldsymbol{\delta}_{scal}$	local step size
δ	global step size
sel	index of selected offspring
β	exponent for global step size
β_{scal}	exponent for local step size
$\boldsymbol{\zeta}$	vector accumulating selected variation information over generations
c	factor controlling weight of last generation in contrast to current generation

Evolutionary Multi-objective Optimization

r_i	non-domination rank of individual / solution i
d_i	local crowding distance of individual / solution i
$<_c$	crowded comparison operator

NSGA-II

N	size of population P
R_t	population formed by joining P_t and Q_t

$\mathcal{F}_i$	different Pareto fronts, with $i = 1, 2, \dots$ etc.

Developed method

$level_{target}(B_i)$	target level of buffer B_i
$level_{current}(B_i)$	current level of buffer B_i
δ_i	Difference between current and target buffer level of buffer B_i
$units(M_i)$	number of units machine M_i has to produce
$units_{tmp}(M_i)$	number of temporary units machine M_i has to produce
$units(max)$	Number of units the machine, with the maximum number of $units(M_i)$ has to produce
$M_i(cycles)$	Number of cycles machine M_i has to stop before machine with $units(max)$
$cycles(M_i)$	Number of cycle times machine M_i has to stop before the last machine stops or production is ceased in general
$range(B_i)$	Indicates the range within which the buffer fill level can lie for the case of the tolerated buffer fill level as target
$level_{min}(B_i)$	Minimum target fill level (tolerated buffer fill level as target)
$level_{max}(B_i)$	Maximum target fill level (tolerated buffer fill level as target)
δ_{i_min}	Difference between current and minimum target buffer level of buffer B_i (tolerated buffer fill level as target)
δ_{i_max}	Difference between current and maximum target buffer level of buffer B_i (tolerated buffer fill level as target)
$count(\delta_{i\_min} < 0)$	Indicates how many buffers are below minimum fill level
$count(\delta_{i\_max} > 0)$	Indicates how many buffers exceed maximum fill level

1 Introduction

Subject of this dissertation is a method to increase output in serial transfer lines without the need of changing the system structure whilst considering costs. The focus of this method is the operation of buffers and simulation-based optimization. With the goal of improving buffer utilization and thus the decoupling effect, possibilities and parameters of adaptive buffer operation have to be designed. To find combinations and solutions for increasing the output, a simulation-based optimization is envisioned to be developed.

1.1 Background and motivation of research

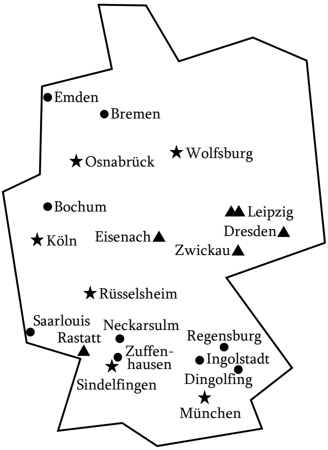
German automotive industry has a far-reaching history, starting with production of the automobile before World War II and resuming production again after reconstruction of the factories. Figure 1.1 shows a map of existing German automotive production sites, classifying these facilities into three periods: production start at the facility before 1945, from 1946 to 1989 and from 1990 until today. Here it can be seen that these factories have a very long life cycle and as they have to be adapted to be able to cope with new products, factories can be seen as complex products, too.[1] Apart from the integration of new products, continuous adaptation of the manufacturing system to environmental conditions[2] with the goal of achieving higher productivity is necessary, so that these production sites in Germany, a high-wage country, remain competitive[3].

Increasing productivity and thus increasing profit can be achieved by lowering costs and increasing output. This is accomplished be improving the processes within or the structure of the system. Yet as the conveyor systems in manufacturing of automobiles are e.g. SKID conveyor systems[4], electric monorail conveyors[5] or suspension chain conveyors to interlink

---

[1] Aldinger *et al.* 2006, 111–112; Westkämper 2008, 93; *ibid.*, 85; Westkämper *et al.* 2006, 143
[2] Westkämper 2008, 85
[3] Brühl 2015, 202
[4] ten Hompel *et al.* 2007, 147–148
[5] *ibid.*, 222

machines[6], structural changes to existing system are unfavorable[7]. Apart from the systems inflexibility, changes cannot be implemented quickly and are very planning- and cost-intensive. Most often modification cannot take place during production. In addition cases of brownfield adaptation and improvement are more frequent than new plannings of greenfield factories, which can be seen in Figure 1.1. Even the youngest factory, BMW plant Leipzig with start of production in 2005[8], has already been adapted to integrate new models into the system.[9] All in all, this shows the necessity to focus on improvement of already existing manufacturing systems. Considering the above, solutions avoiding structural changes are preferred and sought after.



Legend

★ Start of production before 1945
● Start of production from 1946-1989
▲ Start of production after 1989

Company	**Production site**
Adam Opel AG	Rüsselsheim, Bochum (until 2014), Eisenach
Audi AG	Ingolstadt, Neckarsulm
BMW AG	München, Dingolfing, Regensburg, Leipzig
Daimler AG	Sindelfingen, Bremen, Rastatt
Dr. Ing. h.c. Porsche AG	Zuffenhausen, Leipzig
Ford-Werke GmbH	Köln, Saarlouis
Volkswagen AG	Wolfsburg, Emden, Zwickau, Dresden, Osnabrück

Figure 1.1 Map of german automotive production sites[10]

Focus within this dissertation are manufacturing systems, in which personalized make-to-order products are assembled out of many small parts to one finished product of big size, as e.g. automotive vehicle assembly. An important characteristic of the regarded systems is that there is no storage element between production and customer to absorb fluctuation in

[6] *ibid.*, 140
[7] See *ibid.*, 228; here a table gives an overview of flexibility regarding layout change and used conveyor system
[8] Please note, that the Porsche plant in Leipzig first initiated production in 2002. The plant was extended with a new production building for the Panamera with start of production in 2009.
[9] BMW AG 2015
[10] See Adam Opel AG 2015; AUDI AG 2015b; AUDI AG 2015a; BMW AG 2015; Daimler AG 2015a; Daimler AG 2015b; Daimler AG 2015c; Dr. Ing. h.c. F. Porsche AG 2014; Dr. Ing. h.c. F. Porsche AG 2015;Ford Werke GmbH 2015a; Ford Werke GmbH 2015b; Volkswagen AG 2015a; Volkswagen AG 2015b; Volkswagen AG 2015c; Volkswagen AG 2015d; Volkswagen AG 2015e

ordering. Apart from that, the manufacturing systems dealt with already exist and are highly sophisticated, having system availabilities of above 95% and many lean methods already implemented. Here incrementing efficiency and thus output by improving these already applied methods further is rather difficult. This is why the focus of this dissertation is improving the interplay of elements of the system as machines and buffers. In specific the decoupling effect of buffers is studied.

Already developed methods which use buffers and their behavior to increase output can be split into the field of the buffer allocation problem and the optimal production control. Within the buffer allocation problem, focus is put onto the allocation and sizing of buffers.[11] Considering Greenfield projects, where the manufacturing systems are constructed from scratch, or systems with flexible elements, this is easy to realize. Yet for existing e.g. automotive vehicle assembly systems, in which structural changes are very expensive or impossible, this does not work well. In contrast to that, the optimal production control using hedging point policies focuses on the behavior of one single buffer within the system,[12] the buffer between production and customer. This works well for make-to-stock goods, but in manufacturing of personalized make-to-order goods, this buffer is not as relevant. Often these products, as for e.g. automotive manufacturing, are manufactured using transfer lines, consisting of several machines separated by buffers. If the hedging point policy is applied here, not all interactions within the system are regarded and not all elements of the system are viewed in the same manner, as only one buffer is focused on.

1.2 General objective

This dissertation addresses the shortcomings named in section 1.1. The goal and objective is the development of a method to increase output in manufacturing systems, more specifically in transfer lines. On basis of the preceding explanation of the problem, the question for manufacturing companies is how to achieve an increase of output without changing the manufacturing system structurally whilst regarding costs. For this, a method for adaptive buffer operation shall be developed. The goal of adaptive buffer operation is to improve the decoupling effect of buffers.

[11] See Demir *et al.* 2014 for an current literature review of buffer allocation problem literature.
[12] See Kimemia and Gershwin 1983; Bielecki and Kumar 1988; Gershwin 1994 for literature on the optimal production control.

Adaptive buffer operation is a different way of operating buffers. Most studies on system optimization deal with allocation and sizing of buffers, whereas the filling of the buffers is not in the focus, but only observed, e.g. material flow studies. Within this dissertation buffers are filled to certain target fill levels at fixed times of the day (before breaks or end of shift). That way, the decoupling effect of the buffers is changed. The question here is, what is the required filling level for each of the buffers?

At first, the solution to this problem seems trivial, when decomposing the system and only regarding two machines and a buffer in between. If the buffer in between is filled to maximum level, the impact of downtimes of the first machine on the downstream machine is postponed in comparison to an empty buffer. The result, regarding more machines and buffers would be to maximally fill the buffers, so that downtimes do not propagate through the system. There are two reasons, why this trivial solution does not work and why this topic is studied in this dissertation. First of all, filling the buffers maximally results in increased costs. Work-in-process is increased, more operation time is required and thus cost of filling compared to filling the buffers to a lower level is higher. This higher buffer level may not be necessary and interferes with current production principles, e.g. minimizing work-in-process. Second, the elements of the system influence each other and splitting the system into small "isolable units (…) has proved to be insufficient".[13] To understand the behavior of the system, circularity of effects needs to be included,[14] which is not done when decomposing the system.

To find out how to operate the buffers and which parameter combinations work best for adaptive buffer operation, the developed method needs to be tested and optimized. As an adequate testing environment a simulation model shall be build and connected with suitable algorithms to solve the multi-objective optimization problem.

In this context it has to be stated that manufacturing systems can be evaluated by using the analytical approach of queuing models as well. We choose discrete event simulation since the empirical data obtained from most existing manufacturing systems is based on discrete time probability distributions.[15] The studied manufacturing system consists of machines

[13] Bertalanffy 1972, 45

[14] Ashby 1964, 51–54; Bertalanffy 1972, 45

[15] Here and in the following: Matzka 2011, 6–7; Schleyer 2007, 2-3, 17-19; Schleyer and Furmans 2007, 747; Furmans *et al.*, 76

(servers) which have fixed cycle times and gamma-distributed downtimes. Classical, established continuous time queuing models require continuous distributions and cannot use this data directly without approximation. Discrete event simulation in contrast can use this input to the model directly.

The research result aimed at is a simulation-based optimization method using adaptive buffer operation to increase output in transfer lines through improving the decoupling effect of buffers. Proof of performance is demonstrated while applying the method to a real-world problem, an automotive transfer line.

The following two main research issues are addressed and solved in the course of this dissertation:

- How can the manufacturing system be operated to achieve the goal of increasing output?
- How can buffer operation be adapted, without structural interference?
- Which heuristic algorithm is suited to deal with multiple-objectives and to find good, acceptable solutions to the problem within limited calculation time?

1.3 Structure of this dissertation

This dissertation is organized in four chapters. A short overview of the structure of this book is given in Figure 1.2. This chapter introduces the thematic area of the dissertation and explains the principal objective as well as the scope.

The state of the art on buffer management in manufacturing systems is shown in chapter 2. First fundamentals in manufacturing are stated as basis for the existing approaches. Apart from these fundamentals main evaluative and generative solution methodologies are explained. The requirements to a method to increase output in manufacturing systems are defined. Based on the analysis and evaluation of already performed investigations the need for further research is pointed out.

Chapter 3 describes the designed method for increased output in assembly. First the developed method adaptive buffer operation is explained and the main identified parameters are presented. Then the simulation-based optimization method to find out optimized operating points for buffers is introduced. This method is split into the evaluative methodology, here simulation, and the generative technique, in which Evolutionary Algorithms are used for searching the solution space. This chapter is concluded by introducing the developed tool, combining adaptive buffer operation and the simulation-based optimization method.

In chapter 4 the method is applied. The chapter commences by giving an overview on how the existing manufacturing system was transferred into a simulation model. After discussing validity of the model, it is used to perform experiments giving an understanding of the behavior of the model when subjected to adaptive buffer operation. Results of relevant experiments are presented and discussed.

Finally chapter 5 concludes the book with a summary of the contents and gives an outlook on possible further topics to be investigated within the framework of the addressed field of research.

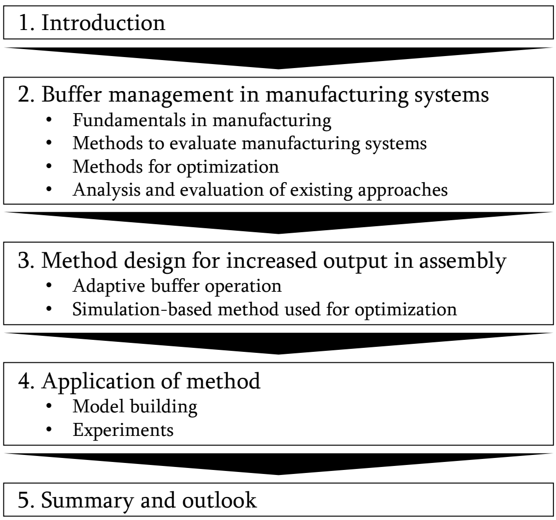


Figure 1.2 Outline of the dissertation

2 Buffer management in manufacturing systems

This chapter gives needed definitions to understand the problem and provides descriptions of the methods applied to solve it. Brief resumes and assessments of these are given as well. To begin with, in section 2.1, Fundamentals in manufacturing, all relevant information is e explained. This is followed by section 2.2, giving an overview of methods to evaluate the performance of manufacturing systems and section 2.3, where an introduction of optimization techniques is explored. Succeeding, in section 2.4 the requirements related to the developed method to increase output are explained. Already existing, relevant approaches to achieve the latter goal are analyzed in section 2.5 and then assessed in section 2.6, which concludes with pointing out the need for further research.

2.1 Fundamentals in manufacturing

To provide a broad and common understanding of the terms used throughout this dissertation these are defined and relevant fundamentals concerning manufacturing are elucidated.

Manufacturing is the transformation of material into goods of higher complexity.[16] This transformation is accomplished by different production processes as processing raw material. In *assembly* it is reached by adding components to each other, as most goods consist of various components.[17] *Manufacturing systems* consist of people, material, production stations as machines, storage areas as buffers, transportation elements and other elements used for manufacturing.[18] In *transfer lines*, a special type of manufacturing system, the material flow is in a fixed sequence through a linear network of machines separated by buffers and each element is only entered once.[19] The *buffer* is a storage element[20] with a

[16] Gershwin 1994, 3; Günther and Tempelmeier 2012, 6; Warnecke *et al.* 1975, 11

[17] Gershwin 1994, 179; *ibid.*, 3; Günther and Tempelmeier 2012, 6; Warnecke *et al.* 1975, 11; Lotter and Wiendahl 2012, 1

[18] Gershwin 1994, 3; VDI 3423 2011, 9; Dallery and Gershwin 1992, 3

[19] Gershwin 1994, 59; Günther and Tempelmeier 2012, 16; Dallery and Gershwin 1992, 3–4

[20] Gershwin 1994, 71; VDI 3633 Entwurf 2013, 15

transportation delay[21] and limited capacity[22]. The buffers in the transfer lines dealt with in this dissertation are conveyor systems with buffering function that have only a short transportation delay and link the material flow from one machine to the next.[23] Units having entered the buffer first, leave the buffer first, too, not changing the sequence, following the First In First Out (FIFO) principle.[24] Figure 2.1 shows a transfer line, where the material gets into the system from the source entering machine M_1, then going to buffer B_1 and proceeding to M_2, repeating this procedure until reaching M_{n+1} and thus leaving the system through the sink.[25]

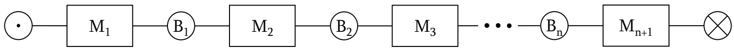


Figure 2.1 General transfer line

In *assembly systems* like the automotive vehicle assembly, the material flow is the same as in a transfer line. Here, small components are added to a large workpiece, which proceeds from one station (or machine) to the next.[26] Another characteristic of the assembly is the proportionally high share of costs regarding the overall manufacturing costs[27], as there is a high variety of manual operations to accomplish the assembly task[28]. In general, manufacturing costs can be broken down to material costs and staff costs, e.g. for personnel needed for inspection and processing.[29]

The two main performance indicators of a manufacturing system are the throughput and the average inventory in-process.[30] An alternative term for the latter mentioned indicator is *work-in-process* or *WIP* and is the material currently found in the system, e.g. in machines or buffers.[31] The *throughput* or *production rate* of a manufacturing system is the rate of products it produces per time unit, e.g. units per shift or hour.[32]

[21] Gershwin 1994, 71; Walenda 1991, 33
[22] VDI 3649 1992, 9; Dallery and Gershwin 1992, 3
[23] ten Hompel *et al.* 2007, 89; VDI 3633 Blatt 1 2014, 30
[24] ten Hompel *et al.* 2007, 107
[25] Gershwin 1994, 59
[26] *ibid.*, 179
[27] Warnecke *et al.* 1975, 13; Lotter 1992, 2
[28] Warnecke *et al.* 1975, 13; Lotter and Wiendahl 2012, 331
[29] Groover 1987, 63; Lotter and Wiendahl 2012, 4
[30] Gershwin 1994, 59; Lotter and Wiendahl 2012, 332
[31] Gershwin 1994, 4; *ibid.*, 5; Groover 1987, 37
[32] Gershwin 1994, 4; Groover 1987, 32

To achieve the desired throughput machines have *cycle times*, which are the time spans each machine requires to complete an operation and for the unit operated on to leave it.[33] Buffers require a time span to transport the units within the buffer and here *lead time* is used for the transportation delay.

Yet the time a part spends in one machine is not predictable and depends on random failure events.[34] The reasons for downtimes can lie within the design or construction of the machine itself, called *technical downtimes*, or within shortcomings in organization, referred to as *organizational downtimes*.[35] These organizational downtimes include downtimes that result as an effect of technical downtimes.[36] If a machine in a transfer line breaks down, the upstream machine continues to operate and the units produced are put in the buffer in front of the broken down machine.[37] This buffer is filled until reaching its maximum capacity and thus the upstream machine is compelled to stop or is *blocked*. The same applies for the reverse case of the downstream buffer. The level of this buffer diminishes until it is empty, forcing the downstream machine to stop, as it is *starved*. To decouple the machines and to mitigate the effects on adjacent machines buffers are used.[38]

To describe how reliable a machine is or how probable it is to find the machine in a functioning state the *availability* η is used.[39] It is the percentage of time the machine is working without being disrupted in operations and is calculated using the mean time between failures (MTBF) and the mean time to repair (MTTR). MTTR is the time a machine is in the state of "not functioning properly".

$$\eta = \frac{MTBF}{MTTR + MTBF} * 100\% \tag{2.1}$$

In this dissertation, the above introduced technical and organizational downtimes are divided into two main classes: *technical downtimes* and *system-induced downtimes*. Here technical downtimes include all downtimes resulting through the machine itself and organizational ones, which happen to the machine itself, e.g. a downtime resulting from an

[33] Gershwin 1994, 5; VDI-Gesellschaft Produktionstechnik 1992, 177; Groover 1987, 107, 145
[34] Dallery and Gershwin 1992, 3; Gershwin 1994, 59
[35] VDI 3423 2011, 6; *ibid.*, 5; Lotter and Wiendahl 2012, 333
[36] Kuhn 2002, 117
[37] Here and in the following: Gershwin 1994, 60; Lotter and Wiendahl 2012, 331; Günther and Tempelmeier 2012, 101–104;
[38] Buzacott 1982, 80; Gershwin 1994, 59–60
[39] See here and in the following: VDI 3581 2004, 2–3; VDI 3423 2011, 8; Groover 1987, 37

operator interrupting a light barrier. System-induced downtimes are those propagated through the system and are subdivided into *blocked* and *starved*. The availability of a machine is calculated by using the technical downtimes and neglecting system-induced downtimes.

Now determination of system availability in transfer lines is rather difficult, as the elements of the system interact with each other,[40] e.g. buffers prevent propagation of downtimes partially. This is why instead of system availability the *utilization* of the system is computed, referring to the actual output regarding the capacity of the system.[41] Here the *output* is the number of units the manufacturing system produces and *capacity* is the maximum production rate that a manufacturing system is able to achieve during a certain time interval.[42]

$$utilization = \frac{output}{capacity} \tag{2.2}$$

2.2 Evaluative solution methodology in manufacturing systems optimization

This section introduces relevant methods for evaluation of manufacturing systems. Popular methods to model the behavior are the queuing theory, simulation models and analytical tools. These are applied to evaluate manufacturing systems and to get a better understanding of these systems.[43] Simulation is introduced in subsection 2.2.1, subsection 2.2.2 presents the analytical tool decomposition and concluding, subsection 2.2.3 explains the Design of Experiments method.

2.2.1 Simulation

Simulation is a method used for designing production processes in manufacturing systems.[44] It is especially used for analyzing systems, when the complexity of the system is high and effects of interventions are not apparent or analytical methods are not available.[45] Simulation uses a model, which represents the elements, the relationships among those and the dynamic processes essential of the real-world system to deduce findings and knowledge of

[40] Hegenscheidt 2003, 24
[41] Groover 1987, 36
[42] Gershwin 1994, 4; Groover 1987, 33
[43] Nyhuis and Wiendahl 2006, 441; Nyhuis *et al.* 2005, 418
[44] Gershwin 1994, 8; Spieckermann and Wortmann 2003, 58; Nyhuis *et al.* 2005, 417
[45] ASIM 1997, 6; Rabe *et al.* 2008, 1; Salt 1993, 1

the system which can be transferred back to reality.[46] Experimental investigations within this model do not interfere with operational processes and thus do not result in further costs or even risks of having negative effects on the running system.[47] In simulation it is important that the results and thus the deduced statements are correct and do not mislead to erroneous decisions.[48] For this verification and validation is used.[49] Verification shows that the model is correct and that more specifically, the transformation from one manner of describing it into another is correct.[50] Validation substantiates that the behavior of the implemented simulation model corresponds to the real system behavior with sufficient accuracy, especially the behavior relevant for reaching the study objectives.[51] Yet it has to be noted that "it is not possible to prove that a model is valid" and that through validation only confidence in the model grows, so that knowledge obtained through experimenting with the model supports decision-making.[52]

2.2.2 Decomposition

For approximate performance evaluation of transfer lines *Gershwin 1987* developed a decomposition method.[53] Indicators as throughput, the average level of each buffer and the probability of blocking and starving of each machine can be computed. In this method transfer lines as depicted earlier in Figure 2.1, referred to as L here, are decomposed into a set of $n-1$ two machine lines named $L(i)$ for $i = 1, \ldots, n-1$.

Figure 2.2 shows the decomposed 5-machine transfer line L. Now line $L(i)$ consists of two machines and a buffer, the upstream machine $M_u(i)$, a downstream machine $M_d(i)$ and the buffer B_i with unchanged maximum capacity as in transfer line L. In the original system each machine has an availability η_i, an $MTTR_i$ and an $MTBF_i$. For the up- and downstream machines these parameters are unknown and are represented by $MTTR(i)_u$, $MTBF(i)_u$ and $MTTR(i)_d$, $MTBF(i)_d$. To determine these parameters and to achieve a similar behavior of material flow through the lines $L(i)$ as through the original line L, *Gershwin* developed a

[46] VDI 3633 Blatt 1 2014, 3
[47] ASIM 1997, 6; VDI 3633 Blatt 1 2014, 9
[48] Rabe *et al.* 2008, 2
[49] *ibid.*, 1
[50] See Balci 1998, 336; Balci 2003, 150; Rabe *et al.* 2008, 14; VDI 3633 Entwurf 2013, 21
[51] See Balci 1998, 336; Balci 2003, 150 ASIM 1997, 17; Rabe *et al.* 2008, 15; VDI 3633 Entwurf 2013, 12; *ibid.*, 20–21
[52] Robinson 2007, 214
[53] Here and in the following: Gershwin 1987

set of equations solved by an iterative procedure. This algorithm is complicated and not robust[54], as it does not always converge[55]. Therefore, *Dallery et al.*[56] improved the equations and introduced the Dallery-David-Xie (DDX) algorithm to solve these initial problems. Further development of the DDX is the Accelerated DDX (ADDX) algorithm[57], which has a higher reliability of convergence and is more accurate and faster.[58]

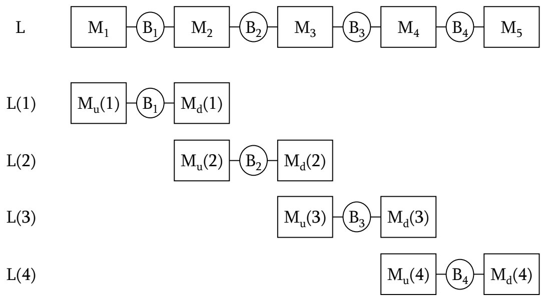


Figure 2.2 Decomposition of 5-machine transfer line L

2.2.3 Design of Experiments

Experiments are used to draw conclusions about cause and effect of processes and systems.[59] In Design of Experiments (DoE) the focus is put on creating well-designed experiments which enable efficient analysis of the collected data. Here the system is seen as a black box with an input, controllable and uncontrollable variables influencing the process or system and the output of the system. These controllable variables are deliberately altered and the response (their effect on the output) is observed. DoE is an essential tool to improve processes and select design parameters.

Within the area of DoE, Response Surface Methodology (RSM) is located.[60] RSM combines DoE with model fitting of responses which are of interest to the user. This response is usually influenced by several input variables and the goal is to find an approximation of the

[54] Dallery *et al.* 1988, 281
[55] Semery 1986
[56] Dallery *et al.* 1988
[57] Burman 1995
[58] *ibid.*, 111
[59] Here and in the following: Montgomery 2013, 1–8; Dean and Voss 1999, 20–22
[60] Myers *et al.* 2009, 1–9; Montgomery 2013, 479–480; Dean and Voss 1999, 559–560; Khuri 2011, 1229–1231

relationship between variables and response. This relationship is plotted graphically as function of the variables and is a surface, which led to the name RSM. It is usually applied to find optimized operating conditions satisfying a certain requirement.

2.3 Generative solution methodology

The general objective of this dissertation is increasing output in transfer lines. Apart from methods, which help to evaluate the manufacturing system as those presented in the previous section 2.2, methods on how to optimize the current systems need to be explained. To begin with, subsection 2.3.1 explains the fundamentals in optimization and the methods applied (subsection 2.3.2). Subsection 2.3.3 is dedicated to a special class of optimization methods, the Evolutionary Algorithms. The methods presented are used for generating possible combinations and to find optimized solutions to the problem in the search space.

2.3.1 Fundamentals in optimization

In optimization problems the goal is to find feasible extreme solutions as a minimum or maximum for a function.[61] The standard form of an optimization problem is:

$$
\begin{aligned}
& f(\boldsymbol{x}) = \text{min! for } \boldsymbol{x} \in \mathbb{R}^n && (2.3)\\
\text{with} \quad & g_k(\boldsymbol{x}) \leq 0, k \in K = \{1, \dots, m\} \\
\text{and} \quad & h_l(\boldsymbol{x}) = 0, l \in L = \{1, \dots, r\} \\
\text{and} \quad & V = \{\boldsymbol{x} \in \mathbb{R}^n : g_k(\boldsymbol{x}) \leq 0, \forall\, k \in K, h_l(\boldsymbol{x}) = 0, \forall\, l \in L\}
\end{aligned}
$$

The optimization task is formed as a minimization task by convention, whereas maximization problems can be solved by minimizing the function $-f(\boldsymbol{x})$. The *objective function* is the function $f: \mathbb{R}^n \rightarrow \mathbb{R}$. A solution $\boldsymbol{x}$ is a vector of *n decision variables*, the terms $g_k(\boldsymbol{x})$ and $h_l(\boldsymbol{x})$ are called constraint functions, where $g_k(\boldsymbol{x})$ represents an inequality constraint and $h_l(\boldsymbol{x})$ an equality constraint. If a solution $\boldsymbol{x}$ satisfies all constraints and variable bounds it is known as a *feasible solution* and gathered in the set V, called the *feasible set* of the original problem.

Depending on the number of objective functions the optimization problems deal with, they are classified into *single-objective optimization problems* for only one objective

[61] Here and in the following: Deb 2004, 13–14; Boyd and Vandenberghe 2004, 127–130

function and *multi-objective optimization problems* for a number of objective functions.[62] The standard form of an optimization problem is stated above in formula (2.3). For multiple objectives it differs, as there are M objective functions $f_m(\boldsymbol{x})$ with $m = 1, 2, \dots, M$, which can be either maximized or minimized.[63]

In case of solving single-objective optimization problems, solutions with better objective function values replace older solutions until the search algorithm reaches its end.[64] Yet practical real-world decision-making problems and optimization tasks (e.g. manufacturing systems design) involve multiple objectives.[65] In the past, due to lack of solution methods multi-objective optimization problems were often simplified to single-objective optimization problems with artificial fix-ups.[66] Solutions were sorted through weighted rating and thus the problem was converted into one composite objective function, an optimization procedure called preference-based multi-objective optimization. Nevertheless it can be observed, that changes in the composite function result in different solutions.[67] Applying such a search strategy, various single-objective runs are needed to give a broad picture of the problem, whereas there are search strategies requiring only one single run (see subsection 2.3.3 Evolutionary Algorithms).[68]

As not all of the large amount of solutions for the multi-objective optimization problem found are relevant and needed, they have to be compared to each other.[69] To do this, the algorithms use the concept of *domination*.[70] As there are M objective functions that can be minimized or maximized, the operators $\triangleleft$ or $\triangleright$ are used between two solutions i and j.[71] If solution i is better than solution j on one particular objective $i \triangleleft j$ is used. In contrast, $i \triangleright j$ stands for solution i being worse than solution j on a particular objective.

[62] Deb 2004, 1; *ibid.*, 13; Collette and Siarry 2003
[63] Deb 2004, 13–14;
[64] *ibid.*, 24
[65] *ibid.*, 13; *ibid.*, 1
[66] Here and in the following: *ibid.*, 5–6; *ibid.*, 25; *ibid.*, 13
[67] *ibid.*, 6
[68] Eckart Zitzler 2012, 885
[69] Collette and Siarry 2003, 19
[70] Deb 2004, 28; Collette and Siarry 2003, 19
[71] Here and in the following: Deb 2004, 28

A solution $\boldsymbol{x}_1$dominates another solution $\boldsymbol{x}_2$ if the following two conditions are true:[72]

1. $\boldsymbol{x}_1$ is as good as $\boldsymbol{x}_2$ or better in all objectives, or $f_j(\boldsymbol{x}_1) \ntriangleright f_j(\boldsymbol{x}_2)$ for all $j = 1, 2, \dots, M$.
2. $\boldsymbol{x}_1$ is strictly better than $\boldsymbol{x}_2$ for at least one objective, or $f_{\bar{j}}(\boldsymbol{x}_1) \triangleleft f_{\bar{j}}(\boldsymbol{x}_2)$ for at least one $\bar{j} \in \{1, 2, \dots, M\}$.

After comparing all solutions, we will find that there is a set of dominated solutions and a set of non-dominated solutions.[73] These *non-dominated solutions* dominate all others, which are outside this set but within the set do not dominate each other. This set is called the *Pareto-optimal set*. In the feasible objective space, the Pareto-optimal solutions can be joined by a curve which is called the *Pareto-optimal front* or *Pareto-front*.[74]

2.3.2 Optimization methods

The solution techniques for combinatorial optimization problems can be classified into exact and heuristic methods.[75] Exact methods give exact solutions to a problem using a finite amount of steps. Examples are complete enumeration, where the best solution is chosen among all possible ones or methods which exclude many configurations before enumerating as branch-and-bound or dynamic programming. Yet, as exact methods can require long computing time heuristic methods are applied for most real life problems.[76] These are able to give a good approximate solution, which is not necessarily the optimum but they solve the problems faster than exact methods.[77] Local search methods or local search heuristics iteratively search the solution space trying to improve the solution for a given problem.[78] Now when the basic principle of the heuristic is applicable on a variety of problem types it is called metaheuristic[79], a term introduced by *Glover 1986*.[80] This type of search strategy is based on a simple, basic search principle, which does not depend on the problem and is an

[72] Here and in the following: Collette and Siarry 2003, 19; Deb 2004, 28
[73] Here and in the following: Collette and Siarry 2003, 19; Deb 2004, 20; *ibid.*, 30; *ibid.*, 31
[74] *ibid.*, 20
[75] Here and in the following: Bangert 2012, 5; Domschke *et al.* 2015, 134–135; Zäpfel *et al.* 2010, 32; Martí and Reinelt 2011, 17; Aarts and Korst 1989, 4
[76] Bangert 2012, 5; Domschke *et al.* 2015, 125; *ibid.*, 135; Martí and Reinelt 2011, 17
[77] Bangert 2012, 5–6; Domschke *et al.* 2015, 135; Martí and Reinelt 2011, 17
[78] Zäpfel *et al.* 2010, 32; Martí and Reinelt 2011, 18–19
[79] Zäpfel *et al.* 2010, 68; Domschke *et al.* 2015, 137
[80] Glover 1986, 541

abstract higher level framework.[81] Examples are Simulated Annealing, Tabu Search, ant colony optimization and Evolutionary Algorithms.[82] In the following these are explained briefly, Evolutionary Algorithms in detail in subsection 2.3.3.

In *dynamic programming*, the problems are broken into subproblems or rather stages, where decisions are made.[83] For each stage, an optimal solution is found and saved. Then these solutions are used to find one optimal solution by working backwards, as it is assumed, that the last decisions taken (which path to take in shortest-path-problems) are the optimal ones.

Tabu Search was introduced by *Glover*[84] and is an iterative local neighborhood search exploring the solution space moving from one neighboring solution to another.[85] It always chooses the best available solution and allows non-improving moves, if these are not forbidden. Solutions can be forbidden or tabu, as the name of the strategy suggests, to avoid getting stuck in a local optimum, which is referred to as cycling. These tabus are memorized on the tabu list, and stay there for a defined number of iterations, the tabu tenure.

Simulated Annealing was introduced by *Kirkpatrick et al.* and by *Černý* independently.[86] This search algorithm is able to escape local optima and can be applied to various problems.[87] It is based on the physical process of cooling material, as in case of the annealing of metal or glass. In this, metal is heated up to a maximum value and then slowly cooled down to obtain a very regular crystalline structure (with minimal energy). Careful annealing can be seen as finding the optimal solution to the problem, where the temperature is the control parameter, the system states are feasible solutions, state changes are modifications to a solution and the energy is the objective function. The Metropolis Algorithm[88]

[81] Zäpfel *et al.* 2010, 72–73; Blum and Roli 2008, 4
[82] Zäpfel *et al.* 2010, 147; Blum and Roli 2008, 4–5; Martí and Reinelt 2011, 42
[83] Here and in the following: Bellman 1957, 3–19; Papadimitriou and Steiglitz 1998, 448–450; Cormen *et al.* 2009, 359
[84] Glover 1986
[85] Here and in the following: Glover and Laguna 1998; Zäpfel *et al.* 2010, 101–104; Domschke *et al.* 2015, 138; Dréo *et al.* 2006, 47–73; Gendreau and Potvin 2005; Martí and Reinelt 2011, 50–56;
[86] Kirkpatrick *et al.* 1983; Černý 1985
[87] Here and in the following: Dowsland and Thompson 2012, 1625; Aarts *et al.* 2005, 187–188; *ibid.*, 191; *ibid.*, 192; Aarts and Korst 1989, 13–17
[88] Metropolis *et al.* 1953

is used to generate new solutions, as "it simulates a thermodynamical [sic!] system by creating a sequence of states or configurations at a given temperature."[89]

The *ant colony optimization* metaheuristic is inspired by the collective behavior of real ants as observed in an ant colony by *Goss*[90].[91] Here Goss investigated an experimental setup with real ants, in which the colony nest and the food source were separated by two paths differing in length. In both directions, to go to the food and back to the nest the ants have to choose one of both paths. It is ascertained that after a while most of the ants chose the shorter path, as the ants leave a pheromone trail on the ground. The path is selected at random first as there is no pheromone trail. After a while, when some ants have already passed the path, they chose the one having a higher concentration of pheromone. This results in choosing the shorter path in the end. In the ant colony optimization algorithms this principle is transferred by introducing artificial ants which collectively try to find the shortest path. They leave an artificial pheromone trail on the path they take. At decision points they use the pheromone trail to compute the probability which path to choose next.

2.3.3 Evolutionary Algorithms

Evolutionary Algorithms are population-based[92] search techniques based on the Darwinian theory of evolution.[93] The three main streams are Evolution Strategies, Genetic Algorithms and Evolutionary Programming.[94] The general metaheuristic of the Evolutionary Algorithms is shared by all of them[95] and is presented in Algorithm 1 as described by *Bäck* and *Rudolph*.[96]

An initial, random population is created during initialization and their *fitness* regarding the objective or fitness function f is evaluated, thereafter the evolution loop is entered, consisting of recombination, mutation, evaluation and selection.[97]

[89] Zäpfel *et al.* 2010, 113
[90] Goss *et al.* 1989
[91] Here and in the following: Dorigo and Di Caro 1999; Zäpfel *et al.* 2010, 82–83; Dréo *et al.* 2006, 127–129
[92] Bäck 1996, 63; Michalewicz 1996, 1; Michalewicz and Fogel 2004, 151; Dréo *et al.* 2006, 77
[93] Bäck 1996, 8; Beyer 2001, 1; Michalewicz 1996, 1; Dréo *et al.* 2006, 11; *ibid.*, 75–76
[94] Rudolph 2012, 674; Bäck 1996, 63; Michalewicz 1996, 1; Michalewicz and Fogel 2004, 151; Dréo *et al.* 2006, 11; *ibid.*, 75–76
[95] Bäck 1996, 131; Michalewicz 1996, 1–2; Michalewicz and Fogel 2004, 151; Dréo *et al.* 2006, 76
[96] Bäck 1996, 66; Rudolph 2012, 675
[97] Here and in the following: Bäck 1996, 61–66; Bäck *et al.* 2013, 8–9; Rudolph 2012, 675; Michalewicz 1996, 1–2; Dréo *et al.* 2006, 11–12; Michalewicz and Fogel 2004, 151

Algorithm 1	General outline of an Evolutionary Algorithm

Initialization
Repeat
  Recombination of individuals
  Mutation of selected individuals to obtain new offspring
  Evaluation of offspring by fitness function
  Selection of individuals according to fitness function
Until Termination criterion fulfilled

During recombination, the parent individuals are selected to be recombined to new individuals (offspring). The offspring is varied during mutation so that new individuals are obtained. Then, each individual's fitness is evaluated by comparing the fitness and chosen to enter the evolution loop from the beginning. This procedure is repeated until the predetermined termination criterion, as reaching a maximum number of evaluations or a target fitness value or the general stagnation of the optimization process, is fulfilled.

Genetic Algorithms

Genetic Algorithms as introduced by *Holland* [98] are the most well-known type of Evolutionary Algorithms.[99] They encode the parameter set of the optimization using a finite-length bit string.[100] The fitness is a scaled objective function value, as solely positive fitness values are required by the selection mechanism of Genetic Algorithms.[101] Within the bit strings, representing individuals of the parameter set, changes of single bits are occasionally introduced.[102] This bit-inversion happens at small mutation rates, so that the individual produced does not totally differ from the ancestor. Recombination is referred to as crossover. Here, segments of different parents are combined to obtain new individuals, as depicted in Figure 2.3.[103]

[98] Holland 1975
[99] Bäck 1996, 106; Dréo *et al.* 2006, 11; *ibid.*, 78
[100] Goldberg 1989, 7; Bäck 1996, 109; *ibid.*, 132
[101] Goldberg 1989, 75–79; Bäck 1996, 111; *ibid.*, 132
[102] Here and in the following: Holland 1975, 109–111; Goldberg 1989, 14; Bäck 1996, 113; *ibid.*, 132
[103] Holland 1975, 97–106; Bäck 1996, 114; *ibid.*, 132; Goldberg 1989, 12

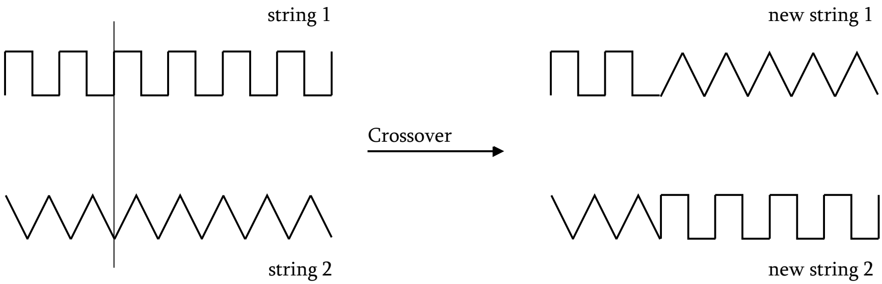


Figure 2.3 Schematic of a simple crossover of two strings[104]

Evolution Strategies

Evolution Strategies were introduced by *Rechenberg* in 1973.[105] They work with real-valued numbers, facilitating application to real-world problems.[106] Recombination of individuals can be done in two ways: *dominant* and *intermediate recombination*.[107] In dominant recombination the properties of the parents are copied at random to the offspring individual, in contrast to that in intermediate recombination the arithmetic mean of the parents' properties is calculated and used for the offspring.Another characteristic is that the individuals that are adapted consist of decision parameters and control parameters as the step sizes.[108] The most general outline of an Evolution Strategy has been specified by *Schwefel and Rudolph* as the $(\mu / \rho, \kappa, \lambda)$-*Evolution Strategy* and is described in Algorithm 2.[109]

A population at generation $t \geq 0$ is denoted by P_t, which is a set of individuals. An individual $p \in P_t$ is a pair $p = (\boldsymbol{x}, \Psi)$, where $\boldsymbol{x}$ is the input of the objective function f, i.e. the design variables, and Ψ is a finite set of strategy parameters of arbitrary kind. The elements of $\boldsymbol{x}$ are mapped to values in $\mathbb{R}$ by the objective function $f: \mathbb{R}^n \to \mathbb{R}$. μ is the number of parents from which ρ parents participate in creating the λ offspring, where $\mu, \rho, \lambda \in \mathbb{N}$ and

[104] Following Bäck 1996, 115; Goldberg 1989, 12; Dréo *et al.* 2006, 12
[105] updated presentation of the subject: Rechenberg 1994
[106] Bäck 1996, 68
[107] Here and in the following: Bäck and Schwefel 1995, 117; Bäck and Schwefel 1996, 24; Bäck 1996, 74; Oyman and Beyer 2000, 268–269
[108] Rudolph 2012, 675
[109] Here and in the following: Schwefel and Rudolph 1995

$\rho \leq \mu$. $\kappa \in \mathbb{N} \cup \{\infty\}$ is the maximum age of an individual and usually either the setting $\kappa = 1$ or $\kappa = \infty$ is used.

Algorithm 2 $(\mu\ /\ \rho, \kappa, \lambda)$-ES

Initialization of P_0 with μ individuals
for all individuals $p \in P_0$ set age $p.\Psi.age = 1$ and evaluate fitness $f = f(\boldsymbol{x})$
set $t = 0$
Repeat
  $Q_t = \emptyset$
  for $i = 1$ to λ **do**
    select ρ parents $p_1, \dots, p_\rho \in P_t$ uniformly at random
    create new offspring q through variation of ρ selected parents
    set age $q.\Psi.age = 0$ and evaluate fitness $f = f(\boldsymbol{x})$
    $Q_t = Q_t \cup \{q\}$ (include individual q in Q_t)
  end for
$P_{t+1} = \emptyset$
include μ best individuals from $Q_t \cup \{p \in P_t$: with $p.\Psi.age < \kappa\}$ in P_{t+1}
for all individuals p $\in P_{t+1}$ increment $p.\Psi.age$
$t = t + 1$
until termination criterion fulfilled

The aspect of random self-adapting step-sizes (control parameters) through evolution has been criticized, as good parameter mutation does not necessarily depend on the step-size but can be due to luck.[110] Thus different, derandomized (DR) Evolution Strategies have been developed, one of those by *Ostermeier et al.*: the *DR2 Evolution Strategy*[111], which is an improved concept of the *DR1 Evolution Strategy*.[112] Algorithm 3 describes the DR2 algorithm in pseudocode.[113]

After initialization (lines 1 and 2) the evolution loop is started. First the time counter is updated (line 4) and then in line 5 the λ new offspring are created by mutation using a global step size δ, local step size $\boldsymbol{\delta}_{scal}$ and the normally distributed random vector $\boldsymbol{z}$:

$$\boldsymbol{x}' = \boldsymbol{x} + \delta \cdot \boldsymbol{\delta}_{scal} \otimes \boldsymbol{z} \tag{2.4}$$

[110] Ostermeier *et al.* 1994a, 371
[111] Ostermeier *et al.* 1994b
[112] Ostermeier *et al.* 1994a, 371; DR1 namely for the first derandomized Evolution Strategy and DR2 namely for the second derandomized Evolution Strategy.
[113] Here and in the following: Ostermeier *et al.* 1994b

Algorithm 3 DR2

1 **Initialization** x, $\boldsymbol{\zeta} = 0, \delta = 1, \boldsymbol{\delta}_{scal} = (1, \dots, 1)^T$
2 $t = 0$
3 **repeat**
4 $\quad t = t + 1$
5 $\quad$ **for** $i = 1$ to λ **do**
6 $\quad\quad$ generate normally distributed random vector: $\boldsymbol{z}_i = N(\mathbf{0}, \boldsymbol{I})$
7 $\quad\quad$ create the offspring $\boldsymbol{x}_i = \boldsymbol{x} + \delta \cdot \boldsymbol{\delta}_{scal} \otimes \boldsymbol{z}_i$
8 $\quad\quad$ calculate fitness $\boldsymbol{\phi}_i = f(\boldsymbol{x}_i)$
9 $\quad$ **end for**
10 select i with best value of $\boldsymbol{\phi}_i$ (offspring with the best fitness)
11 accumulate information of selected mutations over generations:
$$\boldsymbol{\zeta}' = (1 - c) \cdot \boldsymbol{\zeta} + c \cdot \boldsymbol{z}_{sel}$$
12 adapt global step size: $\delta' = \delta \cdot \left(\exp\left(\frac{\|\zeta'\|}{\sqrt{n} \cdot \sqrt{\frac{c}{2-c}}} - 1 + \frac{1}{5n} \right) \right)^{\beta}$
13 adapt local step size: $\boldsymbol{\delta}'_{scal} = \boldsymbol{\delta}_{scal} \otimes \left(\frac{|\zeta'_i|}{\sqrt{\frac{c}{2-c}}} + \frac{7}{20} \right)^{\beta_{scal}}$
14 $\boldsymbol{x} = \boldsymbol{x}_{sel}$
15 $\boldsymbol{\zeta} = \boldsymbol{\zeta}'$
16 $\delta = \delta'$
17 $\boldsymbol{\delta}_{scal} = \boldsymbol{\delta}'_{scal}$
18 **until** termination criterion fulfilled

Here step-size adaptation is based on the most successful $\boldsymbol{z}$, but the information of current successful mutation and of past successful mutations is taken into account and accumulated. The stepsize of each individual is influenced by the local step size $\boldsymbol{\delta}_{scal}$, and additionally parametrized by the global step size δ, having the same value for each individual in the generation.

Then, in line 8, the fitness of all λ offspring is calculated. The offspring with the best fitness is selected (line 10). The vector $\boldsymbol{\zeta} \in \mathbb{R}^n$ accumulates the selected variations so information over generations is taken into account. The factor $c \in (0,1]$ is used to control how much weight the last generation receives in contrast to the current generation (see line 11). In the next step the global step size δ (line 12) and the local step size $\boldsymbol{\delta}_{scal} \in \mathbb{R}^n$ (line 13) are adapted, both based on the vector $\boldsymbol{\zeta}$.

Standard settings for exponents β and β_{scal} and parameter c:

$$\beta = \sqrt{1/n}$$
$$\beta_{scal} = 1/n$$
$$c = \sqrt{1/n}$$

Evolutionary Multi-objective Optimization

Evolutionary Algorithms simulate evolutionary processes with the goal of maximizing or minimizing a certain objective and are able to generate solutions, which do not only yield one point of a Pareto set, but cover the whole.[114] Using populations of solutions in each iteration, more optimal solutions for different objectives are found than in classical approaches with only one run[115]. This makes Evolutionary Algorithms unique in solving multi-objective optimization problems.[116]

As mentioned in section 2.3, the two goals in a multi-objective optimization are on the one hand finding a set of solutions as close as possible to the Pareto-optimal front and on the other hand finding a set as diverse as possible.[117] To identify good solutions and eliminate bad ones a method called *binary tournament selection* is used.[118] Two randomly picked solutions from a population are compared and the better individual is selected and placed in the mating pool until it is full.[119] Then to find a set as diverse as possible the so-called *crowding distance* d_i is determined.[120] This crowding distance is the density of solutions, that surround a particular solution i in the population and it is calculated by taking the average distance of two solutions on either side of solution i along each objective. The *crowded comparison operator* or *crowded tournament selection operator* is based on both: the binary tournament selection and crowding distance measure. According to the author, the crowded comparison operator $<_c$ is used to compare two solutions and

[114] Bäck 1996, 35
[115] Deb 2004, 7–8; Eckart Zitzler 2012, 885; Dréo *et al.* 2006, 207
[116] Deb 2004, 7–8
[117] *ibid.*, 22; *ibid.*, 24
[118] Here and in the following: *ibid.*, 88
[119] Goldberg and Deb 1991, 78–79
[120] Here and in the following: Deb *et al.* 2000, 852

return the winner, so that the selection process is guided towards a uniformly spread-out Pareto-optimal front. It is assumed that every solution i has two attributes:

1. Non-domination rank r_i
2. Local crowding distance d_i

A solution i wins a tournament with another solution j if any of the following conditions are true:

1. $r_i < r_j$
2. $r_i = r_j$ and $d_i > d_j$

The first condition assures, that always the Pareto-optimal solution is chosen. If the solutions lie on the same front, as in the second condition, the solution which is located in a region with a lesser number of points is selected.

Non-dominated Sorting Genetic Algorithm (NSGA-II)

The NSGA-II is a multi-objective Evolutionary Algorithm developed by Deb and his students in 2000.[121] Algorithm 4, following *Deb et al. 2000* represents the pseudocode of the NSGA-II.[122]

Initialization starts with creating a random population P_0 with size N. After assigning fitness to each individual, P_0 is sorted based on non-domination and the individuals are assigned to different fronts $\mathcal{F}_i$ $i = 1,2, \ldots$, etc., where 1 is the best level. Then the offspring population Q_0 with size N is created by using binary tournament selection, recombination and mutation operators. In the following steps, which are repeated until termination, the proceeding differs and is called the NSGA-II procedure (Figure 2.4):

The parent and offspring populations are combined together to form R_t with size $|R_t| = 2N$, the solutions are assigned to different non-dominated fronts using non-dominated sorting ($\mathcal{F} = fast\ non\text{-}dominated\ sort\ (R_t)$). A new population $P_{t+1} = \emptyset$ is set and filled by solutions of different non-dominated fronts. One front is taken at a time, starting with the best non-dominated front and continuing with the second-best non-dominated front and so on. This is done until including the next front would result in the size of P_{t+1} exceeding or being equal N. Now there is still space in P_{t+1} to be filled, so the next front $\mathcal{F}_i$, which

[121] *ibid.*

[122] Here and in the following: *ibid.*, 853–854

has not been included is sorted using the crowded comparison operator $<_c$. Then the first $(N - |P_{t+1}|)$ most widely spread solutions, which still fit into P_{t+1}, so that it reaches size N are included. From this population P_{t+1} the next offspring population Q_{t+1} is created and the NSGA-II procedure started until the termination criterion is fulfilled.

Algorithm 4 NSGA-II

Initialize P_0 a random, $|P_0| = N$
$t = 0$
for $i = 1$ to N **do**
  calculate fitness $\boldsymbol{\phi}_i = f(\boldsymbol{x}_i)$
end for
sort P_0 based on non-domination
create Q_0, $|Q_0| = N$
repeat
  $R_t = P_t \cup Q_t$
  $\mathcal{F} = fast\ non\text{-}dominated\ sort\ (R_t)$
  $P_{t+1} = \emptyset$
  $i = 1$
  do
    $P_{t+1} = P_{t+1} \cup \mathcal{F}_i$
    $i = i + 1.$
  while $|P_{t+1}| + |\mathcal{F}_i| < N$,
  sort($\mathcal{F}_i$, $<_c$), in descending order using $<_c$
  include first $(N - |P_{t+1}|)$, most widely spread solutions
  create Q_{t+1} from P_{t+1}
until termination criterion fulfilled

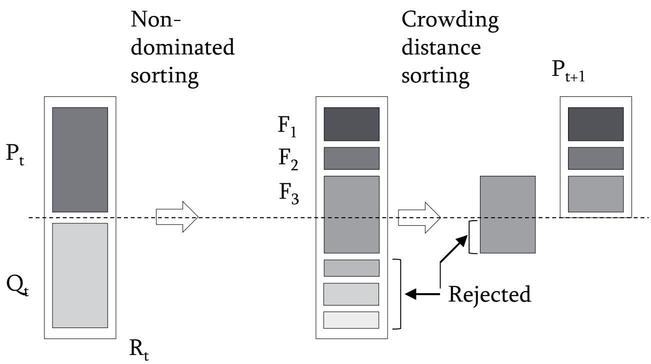


Figure 2.4 Schematic of the NSGA-II procedure following Deb 2004, 246

2.4 Requirements of a solution method to increase output

In the following section, the requirements regarding the method which is to be developed with the goal of increasing output in transfer lines are described. These requirements are used to evaluate the already existing scientific approaches.

2.4.1 System-oriented requirements to the solution method

The system-oriented requirements focus the characteristics of the manufacturing system itself. As the variety of manufacturing systems is very high, the field of application of the method is narrowed down to specific characteristics which result from the motivation and general objective of the dissertation.

- *Applicability to existing systems*: The central system-oriented requirement for the method to be developed is its applicability to already existing systems. Most production sites have a very long life cycle and are continuously improved to increase system efficiency and adapted to new products or new production technology[123] (shown in the introduction, section 1.1). Thus, in most manufacturing system planning cases the systems are not planned from scratch, as they already exist. It has to be assured that the method is applicable to brownfield planning scenarios. This is achieved if the method additionally considers costs for planning or investment (in the case of modification in the existing system).
- *Avoidance of structural changes*: The requirement of avoiding structural changes results from the preceding requirement of applicability to existing systems. When improving already existing systems, the degrees of freedom are confined, as the structure is already given. The difficulty is that most structural changes are accompanied by additional costs and high efforts in prior planning and realization. To facilitate implementation of a method, it is required that no structural changes are needed.
- *Transfer lines*: Additionally, the approach needs to be capable of dealing with transfer lines, as defined in section 2.1, p.7. As explained, this type of manufacturing system is characterized by a fixed sequence of machines. The elements of the system are linked to each other by a conveyor system. The product flow has no possibility to choose its path but always follows this specified sequence. This topology aggravates

[123] Westkämper *et al.* 2006, 143; Westkämper 2008, 85; Westkämper 2008, 93; Aldinger *et al.* 2006, 111–112

changes to the structure of the system even further, as the conveyor system linking the machines is fixed. Moreover, the products worked on cannot be transferred to a parallel line to improve production flow in case of breakdown.

- *Make-to-order production of personalized goods*: It is important that the properties of the products produced in the manufacturing system are regarded, too, as they do influence the system and its operation. The requirement here is for the method to be able to handle make-to-order production of personalized goods. As each unit is personalized and consists of different parts which are directly supplied to the transfer line in a fixed order, the sequence of units in the manufacturing system cannot be changed easily. Each unit needs to be delivered to the customer at a certain time, therefore, there is no storage element between production and customer to absorb the fluctuation in ordering (which additionally would facilitate changes in sequence). This means, that failures resulting from deficits of the unit itself cannot be resolved by simply removing the defective unit from the process, as this would have the consequence of having to rearrange the sequence of supplied parts. Consequently, these deficits have to be dealt with directly in the transfer line.

2.4.2 Application-oriented requirements

In contrast to system-oriented requirements, the application-oriented requirements focus on the characteristics of the method itself.

- *Scalability of system size*: When dealing with transfer lines an important aspect not to be neglected is that each product to be processed in the manufacturing system requires different treatment, sequences, machines and lines. Thus, the arrangement of the elements of the system differs and results in a distinct layout. This is why the approach has to be transferable and flexible enough to be able to tackle a variable number of machines and buffers in a system. The level of detail of each production system can always be simplified, so as to correspond a machine-buffer-machine line of varying amount of elements. Scalability of size is limited to 25 machines in sequence, which is a realistic size of a manufacturing system producing make-to-order personalized goods as e.g. cars.
- *Consideration of all interactions in an interconnected system*: A system (from the Ancient Greek word “σύστεμα” sýstēma, the “whole compounded of several parts or

members") consists of interrelated elements, which are viewed as a whole and are separate to the environment.[124] Thus altering characteristics of elements within a system has effects on other elements of the system. These effects do not need to be obvious immediately, as expressed by the phrase coined by Aristotle: "the whole is greater than the sum of its parts"[125]. This means, that the systems' "constitutive characteristics are not explainable from the characteristics of isolated parts."[126] These properties are referred to as "emergent".[127] Now splitting up the system into small elements and isolating individual casual trains has proved to be insufficient[128] and therefore circularity of effects needs to be included to understand the behavior of the system.[129] Due to this it is required that the solution method considers all interactions in the system and does not focus on the effect one alteration has on only one neighboring element.

- *Equality of treatment of all existing buffers within the system*: An additionally resulting requirement from the above explained characteristics is, that the optimization approach to be developed considers and treats all system elements equally. In serial transfer lines, the number of buffers and machines is high and each element contributes to the system efficiency in a similar manner. It is not differentiated between different types of buffers, as all buffers have the same purpose, with a difference in location. Furthermore, no buffer should be used for the purpose of absorbing passed down failures from other buffers in the system. The optimization approach has to treat all buffers in the same manner and not emphasize one.
- *Practical feasibility*: The solution method to increase output has to be comprehensible to the user in industry. Input parameters need to be available or be easy to access. Moreover, the method used has to be kept as simple as possible and understandable. The methodical approach to the task to be solved needs to be realistic and applicable to real-world problems.

[124] DIN IEC 60050-351 2014, 21; VDI 3633 Entwurf 2013, 19; Bertalanffy 1972, 55–56
[125] the phrase as quoted is attributed to Aristotle, yet in does appear in a different way in the original. For this please see: Aristoteles, 177 (10 1041b);
[126] Bertalanffy 1972, 55
[127] *ibid.*; Ashby 1964, 110–112
[128] Bertalanffy 1972, 45
[129] *ibid.*; Ashby 1964, 51–54

- *Consideration of multiple objectives*: Yet another requirement to the application or method is, that the method is in a position to handle multiple objectives. In most-real world problems, the task is to find a solution that satisfies more than one objective.[130] Apart from improving the applicability of the method to real-world tasks, when considering multiple objectives, this additionally gives the designer the chance to compare and choose between solutions and facilitates decision-making.[131]

2.4.3 Summary of requirements

The development and application of a solution method to increase output in manufacturing systems is subject to many requirements. Figure 2.5 lists all requirements.

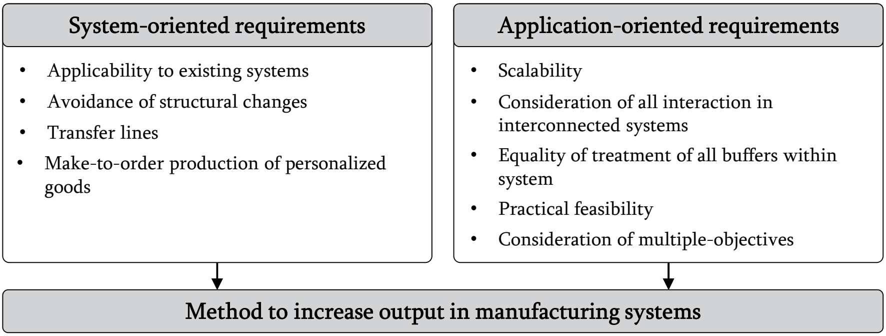


Figure 2.5 Requirements to the solution method

2.5 Analysis of relevant approaches

This section analyses relevant approaches in literature dealing with buffer management and manufacturing system optimization. The research field concerning relevant approaches is split into two areas: the buffer allocation problem and the optimal production control problem solved by the hedging point policy.

[130] Deb 2004, 1; Eckart Zitzler 2012, 872
[131] Deb 2004, 25

2.5.1 Buffer allocation problem

The buffer allocation problem is concerned with determining the size and location of buffers within the system.[132] Many advantages and disadvantages go with buffering. On the one hand increasing buffer sizes lead to additional costs resulting from increased in-process inventory, capital investment and occupied floor space. On the other hand the resulting throughput may be higher, as the decoupling of machines is improved. Downsizing buffers may improve cost but can lead to the production not complying with the demand of the market. This may result in an under-utilization of machines as a consequence of system-induced downtimes. Over the last fifty years extensive research has been devoted on the buffer allocation problem and it still is a current research topic, which is evident from several surveys.[133] The most recently published literature review by *Demir et al. 2014* [134] includes about 100 studies since 1998. In general, existing studies are classified regarding their objective, solution methodology and the type of manufacturing system treated. The three main objectives are maximizing throughput, minimizing total buffer size in line or WIP and minimizing cost. The solution methodology differentiates between the evaluative technique, as e.g. analytical methods or simulation and the search or generative technique as e.g. complete enumeration or Genetic Algorithms. The manufacturing systems are divided into several types, e.g. serial transfer lines, serial-parallel transfer lines or cellular manufacturing systems.

Lee et al.[135] studies unreliable transfer lines with limited buffer space. The buffer allocation problem is solved with applying a two-stage simulation-based optimization method. In the first stage, the author uses Genetic Algorithms for optimization and a simulation model to evaluate the objective function (here: maximize throughput while minimizing total buffer space) and obtain the fitness. This optimization is carried out for many different transfer lines and for each a simulation model is generated and optimized solutions are found. Then in the second stage an artificial intelligence network is built, trained with the solutions from the optimization and validated. This network is able to predict the buffer allocation and to

[132] Here and in the following see: Demir *et al.* 2014
[133] Dallery and Gershwin 1992; Park 1993; Papadopoulos and Heavey 1996; Demir *et al.* 2014; Gershwin and Schor 2000
[134] Here and in the following see: Demir *et al.* 2014
[135] Here and in the following see: Lee *et al.* 2009

increase velocity of finding solutions to the problem. The performance of the developed method was tested on benchmark problems.[136]

Another study employing simulation-based optimization was presented by *Kose and Kilincci.*[137] In this study, a hybrid genetic annealing algorithm, a combination of two meta-heuristics, Genetic Algorithms and Simulated Annealing, is used as search technique to create new buffer sizes. The advantage of Genetic Algorithms, proposing many solutions at the same time, is combined with the good convergence properties of Simulated Annealing, as it has an improved capability of leaving local optima. For evaluation of the objective function, the maximization of the production rate, and to estimate the resulting fitness, the average throughput, simulation is used. To test the performance experimentally, various unreliable benchmark transfer lines[138] were studied. Additionally, the authors vary the number of machines within the transfer lines, including transfer lines with 20, 40 and up to 100 machines, in which the approach showed good results.

Vitanov et al.[139] presented another simulation-based optimization solution of the buffer allocation problem aiming at the objective of throughput maximization. An unreliable closed loop transfer line is studied. The search technique applied is an ant colony optimization algorithm[140] and the methodology is applicable to different line lengths and topologies. Experimental studies validating performance of the method have been conducted using replications comparing it to another simulation-based optimization method.

Sabuncuoglu et al.[141] pursue the objective of maximizing throughput in transfer lines. For this they develop a heuristic which can be applied for reliable and unreliable machines to allocate buffers. This heuristic algorithm works iteratively, identifying bottlenecks and transferring buffers from one place to another. Simulation is employed for evaluation and the performance of the heuristic is tested on benchmark problems[142]. The results suggest

---

[136] Gershwin and Schor 2000; Ho *et al.* 1979
[137] Here and in the following see: Kose and Kilincci 2015
[138] Demir *et al.* 2011; Gershwin and Schor 2000; Shi and Men 2003; Nahas *et al.* 2006; Ho *et al.* 1979; Vergara and Kim 2009
[139] Here and in the following see: Vitanov *et al.* 2009
[140] Gutjahr 2004
[141] Here and in the following see: Sabuncuoglu *et al.* 2006
[142] Schor 1995; Seong *et al.* 1995; Harris and Powell 1999;

that the proposed algorithm achieves qualitatively good results in acceptable time. Additionally, with the obtained results they formulate recommendations on how to design different system configurations.

Demir et al. 2010[143] addresses the issue of throughput maximization under buffer capacity constraints in an unreliable transfer line. An adaptive Tabu Search approach is applied as generative method. The tabu tenure is the number of iterations tabus stay on the tabu list[144] and "is tuned adaptively according to the quality of the current solution and the frequency of the moves"[145]. Initial buffer sizes are not configured randomly but according to the ratio of failure and repair rate. Decomposition, using the ADDX algorithm was applied as evaluative tool and the performance was measured on randomly generated test problems varying the number of machines within the system from 5 to 20 machines. In a following study by *Demir et al. 2011*[146] a Tabu Search approach with constant tabu tenure was presented, solving the buffer allocation problem with the goal of reaching two objectives: maximizing throughput and minimizing total buffer size. Here as evaluative method decomposition (DDX algorithm) is used and the performance is tested on benchmark transfer line problems.[147] They additionally varied the number of machines and obtained good results. A conclusive study, comparing Tabu Search and the adaptive Tabu Search results was performed by *Demir et al. 2012*[148] using the ADDX as evaluation algorithm and showing the superiority of the adaptive Tabu Search method.

In another study addressing the buffer allocation problem for transfer lines *Tempelmeier*[149] uses three evaluative decomposition algorithms in combination with the heuristic optimization approach developed by *Schor*[150]. Depending on the transfer line's characteristics, a different decomposition algorithm is chosen, e.g. for deterministic processing times the ADDX. The objective is minimization of total buffer size. The transfer lines studied are real-life systems and invented systems. With this the author developed an optimization and performance measurement tool to enhance and facilitate planning.

[143] Here and in the following see: Demir *et al.* 2010
[144] Demir *et al.* 2011, 216
[145] Demir *et al.* 2010, 209
[146] Here and in the following see: Demir *et al.* 2011
[147] Ho *et al.* 1979; Park 1993; Gershwin and Schor 2000; Shi and Men 2003; Nahas *et al.* 2006;
[148] Demir *et al.* 2012
[149] Here and in the following see: Tempelmeier 2003
[150] Schor 1995

Nourelfath et al.[151] presents a variation of the buffer allocation problem adding another decision variable, the machine type. Within this problem, the best combination of machines and buffer has to be selected to reach the objective of maximizing efficiency under a cost constraint. The machines differ in production rate, reliability and cost, buffers in capacity and cost. For evaluation decomposition using the DDX algorithm is implemented and an improved ant colony optimization algorithm performs the search. The method was tested on invented problems and it was shown that near optimal solutions were found quickly.

Table 2.1 Overview of relevant buffer allocation literature

Authors	Objective	Evaluation technique	Search technique
Lee et al.	Throughput and total buffer size	Simulation	Genetic Algorithms
Kose and Kilincci	Throughput	Simulation	Hybrid genetic annealing algorithm
Vitanov et al.	Throughput	Simulation	Ant Colony Optimization
Sabuncuoglu	Throughput	Simulation	Heuristic
Demir et al. (2010)	Throughput	Decomposition	Adaptive Tabu Search
Demir et al. (2011)	Throughput and total buffer size	Decomposition	Tabu Search
Tempelmeier	Total buffer size	Decomposition	Heuristic
Nourelfath et al.	Maximize efficiency	Decomposition	Ant Colony Optimization

2.5.2 Optimal production control: the hedging point policy

Much effort has been put in research to minimize total production cost.[152] The hedging point policy (HPP) serves to control or regulate the production rate with the goal of minimizing production costs due to inventory and backlog. It resolves the problem of finding the optimal production policy in order to keep inventory as low as possible yet still meeting customer's demand. Costs for holding inventory (floor space, interest costs, etc.) and costs for lost profit if demand is not met, are included. The difference between production and demand can be positive, called surplus, or negative, named backlog. To minimize costs the production surplus is kept as near to the hedging point as possible. In case the buffer is too empty, production has to be run at the maximum production rate. When reaching the hedging point, the production rate is lowered to the desired average rate and when the buffer is

[151] Here and in the following see: Nourelfath *et al.* 2005

[152] Here and in the following: Kimemia and Gershwin 1983; Bielecki and Kumar 1988; Gershwin 1994, 47–48

too full, above the hedging point, production is ceased. This can be compared to controlling the flow of a fluid through opening or closing the valve.

As the initial research considers single part-type manufacturing systems only, *Perkins and Srikant* [153] extended this to two part-type, single-machine manufacturing systems and solved the problem of optimal scheduling with the objective of minimizing cost. The production control policy used here is a prioritized HPP (PHPP), "where the part-types are prioritized so that among the part-types which are below their respective hedging points, the one with the highest priority is produced at the maximum possible rate, while maintaining all other part-types with a higher priority at their respective hedging points."[154] Additionally they introduce a solution to solve n part-type problems through decomposition of the problem into several two part-type problems. For this analytical solution, the optimal hedging point is numerically computed for various examples.

Giordano and Martinelli [155] present a numerical solution to the problem of finding an optimal production control strategy with the goal of minimizing average backlog or surplus costs. The manufacturing system they investigate is a finite capacity buffer, single part-type, single unreliable machine manufacturing system. The optimal hedging point is computed with an equation developed by the authors. The costs considered material storage and handling expenses, in case of surplus (buffer is too full) and the cost resulting from shortage or unsatisfied demand. The results and performance of this solution are verified by simulation.

Instead of analytically solving the optimal production control problem, *Mok and Porter* [156] present an evolutionary optimization approach, as analytical solutions are available only for simple systems, e.g. single-machine single product-type[157]. Apart from that analytical solutions, they regard constant demand rates on a long run. The approach proposed is able to solve the problem for short-run demands also. Here the HPP introduced by *Kimemia and Gershwin*[158] is applied. Genetic Algorithms, Evolution Strategies and adaptive Evolution Strategies are used. The goal is to minimize the objective function which is the cost function. For single and two product-type cases the optimal hedging levels for the buffers found

[153] Here and in the following see: Perkins and Srikant 1997
[154] *ibid.*, 365
[155] Here and in the following see: Giordano and Martinelli
[156] Here and in the following see: Mok and Porter 2006
[157] Akella and Kumar 1986; Bielecki and Kumar 1988
[158] Here and in the following see: Kimemia and Gershwin 1983

are similar to each other, independently which evolutionary approach is chosen. Compared to analytical solutions, where long-term results are focused, they differ a lot. Yet when increasing the task time and regarding long-run demands, these are approximately the same.

Martinelli [159] investigated a failure prone, single machine, single part-type manufacturing system with the objective of minimizing average costs. For this surplus and not met demand is penalized. The failure rate depends on the production rate, as is realistic for real-world problems. This optimal production control problem is solved by a hedging point type policy referred to as two-threshold policy. These thresholds are the current inventory level and the hedging point. Confirmation of performance was given through numerical examples.

A slight modification of the optimal production control problem is introduced by *Gershwin et al. 2009* [160]. Here backlog is allowed but penalized using a so called defective function. It reproduces customer behavior, when waiting time exceeds their patience they do not complete ordering as a result. The manufacturing system investigated is a reliable single manufacturing facility producing make-to-stock single-items to meet the random demand. The objective pursued is maximizing long-term profit through finding "the optimal production rate as a function of the current surplus and the current demand level."[161] The authors demonstrate that the optimal control policy has a hedging point form. The solution is generated numerically and performance is demonstrated on numerical experiments, too.

Gharbi et al.[162] present another variation of the optimal production control problem. In an unreliable one machine single product-type manufacturing system, another reserve machine is introduced. This reserve machine supports the first machine, here central machine, to meet demand. It runs at an increased cost, only if the level of finished goods inventory drops below a certain level. The problem to be solved here is minimizing cost on a long-run, consisting of production, inventory and backlog cost. For this a state-dependent HPP is developed, as it depends on both the state of the central machine and of the finished product inventory. To determine the parameters and solve the problem, an experimental approach including simulation, DoE and RSM is applied. The results show, that this policy outperforms classical polices.

[159] Here and in the following see: Martinelli 2007
[160] Here and in the following see: Gershwin *et al.* 2009
[161] *ibid.*, 512
[162] Here and in the following see: Gharbi *et al.* 2011

In a further development of the optimal control of unreliable manufacturing systems *Ben-Salem et al.*[163] develop a policy including environmental concerns. They regard a facility producing one product family, where the operation causes harmful emissions per unit produced. Excess of pollution within a period of time is penalized by law, backlog or unmet demand has penalized costs and inventory, too. A policy controlling production rate and emission is developed, named environmental HPP (EHPP). Here a second hedging point, lying below the first one, with the aim of reducing pollution and meeting demand at the same time is introduced. Depending on the current status regarding inventory, demand and emission, the first or second hedging point can be selected and thus the production policy can be adapted. An approach combining simulation, DoE and RSM is chosen to optimize the parameters of the policy and solve the problem. It is ascertained that the EHPP outperforms the HPP.

Wang and Gershwin[164] combine a production and sale policy for a two product-type reliable manufacturing system with downward substitution. In order to meet demand of inferior products and reduce backlog costs, superior products are sold at lower price substituting inferior products if this is worthy. Thus, the costs included consist of inventory costs, backlog costs and profit losses due to downward substitution. The objective is to control the manufacturing system in order to minimize total production costs. To reach this aim, two control policies are developed: one HPP controlling raw material releases into the system and a policy controlling the downward substitution rate. These are solved using dynamic programming. Numerical experiments show performance and state that the HPP introduced here outperforms the prioritized HPP.

[163] Here and in the following see: Ben-Salem *et al.* 2014
[164] Here and in the following see: Wang and Gershwin 2015

Table 2.2 Overview of relevant optimal production control literature

Authors	Reliability	Proof	Production control	part type
Perkins and Srikant	unreliable	numerical	PHPP	multiple
Giordano and Martinelli	unreliable	numerical / simulation	HPP	single
Mok and Porter	unreliable	Evolutionary Algorithm	HPP	single / multiple
Martinelli	unreliable	numerical	HPP (two-threshold feedback)	single
Gershwin et al. 2009	reliable	simulation	HPP	single
Gharbi et al.	unreliable	simulation, DoE, RSM	SDHPP	single
Ben-Salem et al.	unreliable	simulation, DoE, RSM	EHPP	single
Wang and Gershwin	reliable	numerical	HPP and downward substitution policy	two

2.6 Assessment of existing approaches and need for further research

Key concern in manufacturing systems is performance optimization. The foregoing section shows that many approaches to solving this problem exist. In this section these are assessed with regard to the requirements named in in section 2.4. Figure 2.6 shows an overview of all approaches including an assessment of each requirement.

When comparing the different approaches solving the buffer allocation problem, the main differences are within the search and evaluation technique. Approaches as presented by *Lee et al.*, *Kose and Kilincci*, *Vitanov et al.* and *Sabuncuoglu* use simulation as evaluative method. Here the interactions of all elements are always regarded. Now when decomposition is applied, as *Demir et al.*, *Tempelmeier* and *Nourelfath et al.* do, the system is split into smaller pieces and thus some interactions might be neglected. *Lee et al.*, *Demir et al. 2011*, *Kose and Kilincci*, *Demir et al. 2011 Tempelmeier* and *Nourelfath et al.* consider multiple objectives. Only *Nourelfath et al.* include costs of investment and completely meet the requirement of being applicable to existing systems. Yet the principal deficit of these approaches is that the scientific problem implies structural changes, as always buffer sizes are adapted.

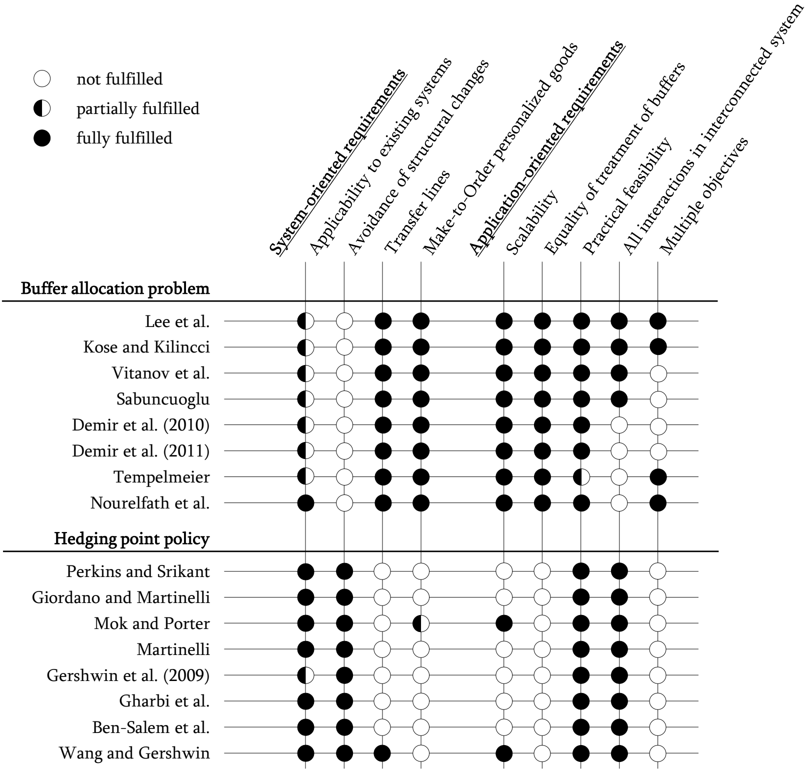


Figure 2.6 Assessment of existing relevant approaches

Now in contrast to these, approaches focusing on the optimal production control problem using a form of HPP do meet this criteria: they do not require adaptation of the system itself but focus on operation control, namely keeping the buffers filled to a certain level, the hedging point. All of them meet the requirement of not involving structural changes and of being capable of dealing with existing systems with one exception: *Gershwin et al. 2009*. The authors focus on reliable systems only, which is not realistic when dealing with existing manufacturing systems. However, as for the buffer allocation problem, the approaches focusing on optimal production control suffer from one core deficit: they concentrate on the inventory level of the finished goods buffer between production and customer. So these policies are only applicable for make-to-stock goods. Furthermore, they only regard one buffer, as in most cases only single-machine single-buffer manufacturing systems are regarded. Even if there is more than one buffer in the system, which is possible in *Wang and*

Gershwin or *Mok and Porter*, these are not treated equally. The latter are the sole ones, which comply with the requirement of being able to solve transfer line problems and scalability, as more machines can be included within the system.

The analysis of the existing approaches reveals, that there is no method of increasing output, that treats all elements equally and at the same time does not require structural changes.[165] On the one hand, when treating all buffers equally and regarding all possible interactions within the system, the methods require structural changes. On the other hand, when no structural changes are required and the implementation of the method within existing systems is possible, the focus is placed on one single buffer. Resulting in the need for further research is the development of a method to increase output, which is capable to deal with existing systems and does not require structural changes. Furthermore this method needs to treat all elements of the system equally.

The method to be developed in this dissertation shall meet the requirements stated in section 2.4. Special focus is put on not introducing structural changes and on equal treatment of all elements within the system. The interactions of the system as a whole shall be regarded. This way it is ensured that the method is applicable to real-world problems. Additionally, the developed method shall be capable of dealing with multiple-objectives.

[165] Please note that the area of lean methods and maintenance policies has not been analyzed, as they are not focus of this thesis. As stated in section 1.1, the here dealt with systems are highly sophisticated, and many lean methods are already implemented, including improved maintenance strategies. This is why these areas are not pursued any further.

3 Method design for increased output in assembly

The method of increasing output in transfer lines is based on improving the decoupling effect of already existing buffers in the system. It is described and explained in this chapter. Beginning with, in section 3.1, a general outline of the principles and work of the developed method is given. Succeeding in section 3.2 the assumptions concerning the studied manufacturing system are listed. Then the specification of the method is started in section 3.3 introducing adaptive buffer operation – ways on how the system can be influenced without structural changes so that an improved decoupling of the machines is achieved. Section 3.4 states buffer filling algorithms, which are necessary to reach the goal of filling all buffers of the system to target fill level. The description to calculate how many units each machine has to produce is provided. Hereinafter, in section 0, the simulation-based method to optimize arbitrary parameters, which are stated in section 3.3, is explained. Concluding the concept, a developed tool on how to implement the simulation-based optimization method is presented in section 3.6.

3.1 General outline of method for increasing output

The method to increase output in manufacturing systems is based on changing the system configuration so that utilization is increased. The structure of the already existing manufacturing system is not changed and is viewed holistically as all elements are tightly connected and influence each other. Increasing utilization can be introduced by improving the decoupling of the machines, which in general is done by adding buffers to the system. Yet this results in the necessity of changing the structure of the system, which is not possible. As the object of study is a highly sophisticated system running at 98% of utilization increasing output through improving the maintenance strategy is possible but not preferable, as the cost is too high. Now a remaining option on improving the decoupling effects of the buffers is an enhanced strategy on steering the system. The buffers cannot be changed physically, but it is possible to influence the filling level of the buffer itself. The idea of the method is simple. The buffers of the manufacturing system are filled to a certain target fill level, resulting in a new system configuration. This is only performed at set moments of the production period, e.g. end of shift, resulting in an improved starting configuration for the next time period as the following shift. Filling or emptying of a buffer naturally results from machine breakdowns. So if machine breakdowns are introduced on purpose, which is done by simply stopping the machine, the filling levels of the buffers can be influenced. Still, when artificially introducing breakdowns the entire system has to be considered as the impact is not only local but global regarding the whole system. The first part of the design deals with how to proceed to fill the buffers. The differences between the possible target buffer levels are explained, moments of intervention and further parameters to vary the method are presented.

After describing the method on how the buffer levels can be adapted, the remaining question to be answered is to which level does each buffer within the system have to be filled? To do this a simulation-based optimization method is introduced. Simulation is used as evaluative methodology and Evolutionary Algorithms are used as search technique, to find optimized solutions. First a simulation model is built, which resembles the real world manufacturing system. Within this model different combinations of fill levels for each buffer and different parameter settings can be tested. As in most cases, there will be a gap between current buffer level and target fill level. Some lines will have to be stopped in advance. To steer when and how to stop the lines the current system status is monitored a defined period in advance of the moment of intervention, e.g. at shift end. Then the time each line has to continue producing is calculated.

To find optimized combinations of buffer fill levels single- and multi-objective optimization is performed using Evolutionary Algorithms. As simulation is used to evaluate the objective function and to obtain the fitness, narrowing down the number of simulation runs is the goal.

A concluding general outline of the method is given in Figure 3.1 and a summary is given explaining the figure: Before starting optimization, as an input, the method of adaptive buffer operation needs to be designed. Here buffer fill levels are adapted following defined rules to reach certain target fill levels. Apart from these, further parameters are presented, which are subsequently used as design variables for the optimization. Parallel to this, a simulation model has to be build, depicting the real-world system which subsequently has to be verified and validated. Only then optimization takes place. Within the Evolutionary Algorithm, which is used as search technique, the evolutionary loop is entered until the termination criterion is fulfilled. The steps of recombination and mutation are performed resulting in new design variables. These in turn are the preset for the simulation run. Simulation is used to evaluate the objective function. The fitness is returned and the evolutionary loop can be started by selecting new individuals for recombination and so on. After finding optimized solutions for the objectives, they are visualized in boxplots or Pareto-fronts and validated by repeating the simulation with the optimized preset.

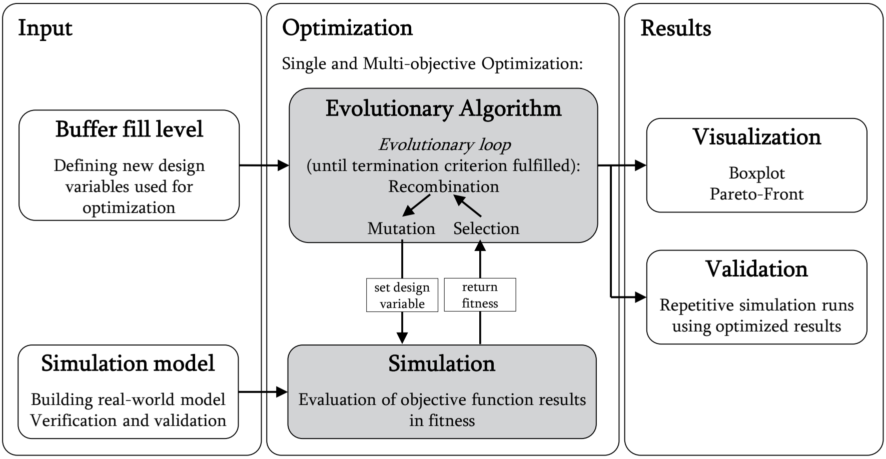


Figure 3.1 Overview of general outline of method for increasing output

3.2 Assumptions and notations

The method of optimizing assembly efficiency is based on a set of assumptions which apply to the regarded manufacturing system:

1. Work is done during shifts; there are two shifts per day, morning and late shift.
2. Shift length is the same in general. Yet it can be adapted – extended or shortened.
3. There are breaks during each shift and they are unavoidable, which means that all workers need to take their assigned breaks.
4. Wage cost for work does not depend on time of the day, it is the same during day or at night.
5. Workers are paid per minute of their presence during a shift.

6. Workers presence is always required during shift, unless there is an artificially introduced break down used for filling or emptying buffers.
7. The number of workers is limited. It is not possible to include additional workers.
8. Downtimes or breakdowns can be introduced artificially at any time.
9. An artificial breakdown at a line equals shortening of the shift for the workers at the affected line, even if work at other lines has not ceased.
10. Artificially introduced downtimes are not operating times for workers and are thus not remunerated in contrast to natural downtimes, resulting from technical downtimes or through blocking and starving.
11. The manufacturing system is a transfer line used for assembly. As described in section 2.1 M_{n+1} machines are separated by B_n buffers. An exemplary transfer line is depicted in Figure 2.1 General transfer line.
12. The model of the manufacturing system is unsaturated, the physically first machine of the system can be starved and the last machine of the system can be blocked.[166]
13. Different models are assembled in the manufacturing system.
14. Operating times related to a set of operations at a machine may be different for different models.
15. The average capacity of an operator during the time of one cycle time is nearly used completely. The average occupancy rate of an operator is about 100%, meaning that the time a worker needs to conclude the assigned set of operations at a station is near cycle time. As these are average values, it is possible, that a worker has an occupancy rate of 95% in one cycle time and of 105% during the next cycle time. This is referred to as "over-cycling".[167]
16. The machines within the system are highly sophisticated, having similar technical availabilities above 98%.

3.3 Adaptive buffer operation: method description and its parameters

In the course of developing a method to increase output in manufacturing systems it has to be explained, how buffer fill levels can be adapted in general. Different parameters on how to intervene the daily routine to do so need to be indicated. These different parameters are

[166] Dallery and Gershwin 1992, 7

[167] "Over-cycling" leads to the drifting away of the worker of the particular station, so that he might have difficulty in reaching his tools or might impede the work of the adjacent station (see Klampfl *et al.* 2006, 278)

a set of design variables used later on in section 0 for optimization. Within this section they are proposed and discussed. To begin with, in subsection 3.3.1 it is described how buffer filling or emptying can be done technically. Ensuing, in subsection 3.3.2 the different target fill levels to be reached are introduced and described in detail, followed by a description of when to intervene at best in subsection 3.3.3. Further intervention parameters are stated in subsection 3.3.4. To finalize, the developed method of adaptive buffer operation is concluded and identified intervention parameters are summarized (subsection 3.3.5).

3.3.1 Buffer filling policy

The filling or emptying of a buffer is a consequence of interaction between elements of the manufacturing system, as already described in section 2.1 Fundamentals in manufacturing. In the daily routine this happens through unpredictable downtimes. Abstractly viewed, changes of buffer levels result from a difference in velocities of consecutives machines.[168] For certain limited time periods of the shift the velocity of machines can be increased or decreased artificially from anywhere in between above original velocity to zero, resulting in stopping the machine.

Stopping machines[169]

Stopping machines results in downtimes which are artificially introduced. Here operators have to cease their work. This is possible at any time during operating hours. If the moment is chosen properly, the operating times of the workers can be shortened and thus wage costs can be saved. Stopping machines for the length of one cycle time results in alteration of the buffer by one unit.

Decreasing velocity[170]

Decreasing velocity of machines excludes the extreme of stopping machines. It results in a raised cycle time, yet the workload for each cycle remains the same. The benefit for the worker is that he can work slower. However, as the operator is already used to the unaltered cycle time he will cease his task before cycle time elapses and will wait for his next operation, which is an idle time. As this means that the worker changes his working process

[168] See Biela 2015, 41
[169] Here and in the following see *ibid.*, 52–57
[170] Here and in the following see *ibid.*, 43–48

sometimes, it finally can become unstable, which is not favorable.[171] Cost savings cannot be achieved, as presence of the worker is required.

Increasing velocity

Increasing velocity of the machine is similar to decreasing velocity. Yet this leads to a reduction of cycle time at an unaltered workload. Now the stations are already "over-cycled", which means the occupancy rate of the worker is already 100% (see 3.2, assumption 15.). Additionally, shortening cycle time results in an even higher utilization, which exceeds manageable "over-cycling" and is thus impossible. Proposed solutions are either involvement of additional manpower, which is not chosen as it leads to increased costs or shifting of the set of operations to different stations.[172] This is difficult to implement, as the assembly order is not flexible. Apart from that, the velocity can only be increased slightly, e.g. cycle times of 60 seconds can be increased to 55 seconds. This leads to a very low effectiveness, as altering the buffer level by one takes very long. For the given example this takes 11 minutes, which can be seen in Table 3.1.

Table 3.1 Example of velocity increasing and resulting amount of units

Cycle time	Time	Units
60 seconds	660 seconds	11
55 seconds	660 seconds	12

Thus the alternative of stopping machines stepwise is chosen for further research, as altering velocity leads to the above explained negative implications.

3.3.2 Target fill level

Considering the method of optimizing assembly efficiency, it is not only important to choose the policy of intervention to fill the buffers, but as well to know which buffer fill level is to be achieved. This level is called the *target fill level* and is the fill level a buffer has to reach in the course of the optimizing strategy explained in the following section 3.4. Here three different filling levels are proposed and the design variables are named and later depicted exemplary in Figure 3.2.

[171] Kletti and Schumacher 2011, 47; *ibid.*, 131
[172] Here and in the following: Schlick *et al.* 2010, 478–479

Exact fill level

The exact fill level is one certain filling level for each buffer in the manufacturing system. If the current level is below target level, the buffer has to be filled. If the buffer level is above target level, the buffer has to be emptied. Other fill levels than the exact fill levels are not permitted. The design variables for the following Evolutionary Algorithm in this case are the buffer levels of each buffer in the system.

Minimum fill level

The minimum fill level of the buffer indicates the level which the buffer has to reach minimally. Buffers with fill levels above minimum fill level do not have to be filled or emptied any further. Yet buffers with a fill level beneath the desired minimum fill level have to be filled, as no buffer is allowed to have a fill level below target fill level. The design variables are the buffer levels of each buffer in the system, equal to the design variables in case of exact filling.

Tolerated fill level

The tolerated fill level is an extension of the exact fill level. Each buffer has one exact buffer filling level. The difference is that around this target buffer level a range is indicated, within which the buffer level can lie. Those ranges differ for each buffer, too. For example, if the tolerated level has the target value 6 and the range is 4 a buffer fill level of 4 to 8 is permitted. The range can equal zero, hence meaning that in this case the exact level has to be achieved. If the buffer fill level lies within range of the tolerated fill level, no further adaption has to be done. In this case the design variables are the buffer levels and in addition the ranges for each buffer.

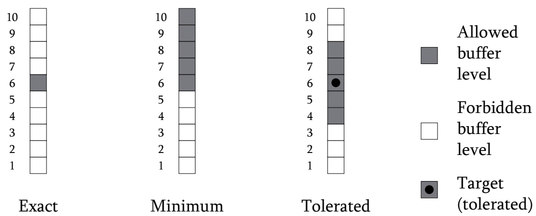


Figure 3.2 Comparison of target fill levels exact, minimum and tolerated

Figure 3.2 depicts allowed and forbidden buffer levels for the different target fill levels exact, minimum and tolerated. For the fill level tolerated, the target is additionally indicted.

Each buffer space of the buffer depicted is shown. Its maximum size is ten. For the exact fill level, the target is 6, for the minimum fill level the target is also 6. The target level of the tolerated fill level is 6, range 4, as in the example described above.

Further considerations

Additionally, it has to be dealt with the problem of technical downtimes occurring while other machines are submitted to artificial downtimes. It is decided that if this happens, the target fill levels for buffers are not adapted, even if this downtime impedes the reach of the target buffer level.

3.3.3 Moments of intervention

Manipulating buffer fill levels can be done at any time of the shift, as it is achieved through stopping machines. Yet stopping machines within the shift arbitrarily leads to additional interruptions of defined time spans, with the disadvantage of having to ensure that all workers return to their station on time. So it is recommended to do this around possible production stops, as the machines are anyway stopped.[173] These planned production stops do not come unexpectedly but are part of the daily routine as the end of shift or a break.

The moments of intervention are described in the following. To facilitate understanding the planned production stop chosen for description is shift end. As mentioned in the general outline of the method in section 3.1, a certain fixed time before shift end the system status is monitored. Using the current buffer levels, it is calculated how many cycle times each machine has to go on operating to reach target fill levels for all buffers.

Figure 3.3 shows the normal routine of stopping machines without intervention (Figure 3.3a) and in comparison to the three here proposed moments of intervention. The manufacturing system in this example is a five machine transfer line.

Before planned production stop[174]

Choosing "before planned production stop" as moment of intervention all machines have to be stopped before the planned production stop (see Figure 3.3b). The machines are stopped stepwise, until the last machine stops with shift end or with the start of a break.

---

[173] See Biela 2015, 55

[174] Here and in the following see *ibid.*, 57–59

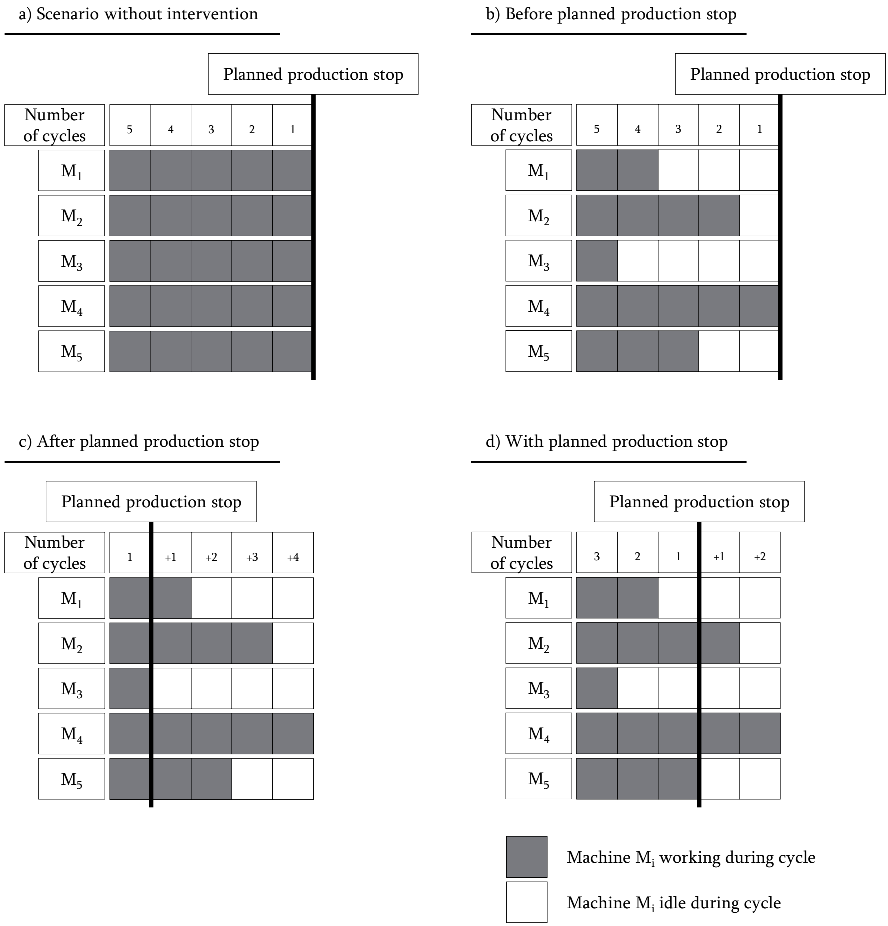


Figure 3.3 Overview of different moments of intervention in comparison

After planned production stop

Another moment to fill the buffers is to stop the first machine at shift end and stop all further machines “after planned production stop” (see Figure 3.3c). All other machines continue producing, concluding a certain number of cycles times to reach the target buffer levels and stop stepwise. This moment of intervention results in an extension of production for some machines, hence it cannot be applied at start of break, due to some workers having a shorter break time. If there are two or more shifts per day, it can only be applied after a shift, when there is an offset of time between end of earlier shift and start of later shift (e.g. it works after late shift and morning shift, if they are not directly consecutive).

With planned production stop

Still, it is not necessary to either stop all machines before shift end or after shift end. Another possibility is to stop some machines before shift end and let other machines operate after shift end, combining the above described moments. The peculiarity of the so called "with planned production stop"-moment of intervention (see Figure 3.3d) is that the machine to stop at end of shift is the physically last machine before the sink. All other machines either stop before or after shift end. This means that this is not only a shift of the moment of intervention, but an additionally introduced constraint. This strategy can be applied in the same manner as the strategy "after planned production stop".

Looking at all moments of intervention depicted in Figure 3.3 one can see that the different moments of intervention result in different operating times for especially the physically last machine M_5. Without applying any strategy and "with planned production stop" machine M_5 operates until planned production stop. In case of "before planned production stop" machine M_5 can cease operation before production stop and thus decreasing operating time of machine M_5. This can result in decreased output. With the moment "after planned production stop" it is possible that machine M_5 goes on producing even longer and more output can be generated. This alteration of output results from the different moments of intervention and is not due to adaptive buffer operation. This is why it has to be taken into account when comparing the moments of intervention and deciding for one alternative.

3.3.4 Variable intervention parameters

Intervening the daily routine with the goal of changing buffer levels is possible at different moments of the day as explained in subsection 3.3.3. Further parameters are the frequency of intervention and the duration of intervention.

Frequency

When choosing "before planned production stop" as moment of intervention the intervention is applicable before any planned production stop, such as before every break or shift end. A normal working day consisting of two shifts, with two breaks during each, gives possibility to intervene six times per day, if done before every planned stop. This number of interventions can be reduced to once per day, e.g. before end of late shift or to once every second or third day. This increasing and decreasing of the frequency results in another control parameter for the method of influencing buffer levels.

Duration of intervention

An additional parameter that can be influenced in regard to the intervention of the daily routine to fill the buffers to a certain level is the allowed time span or duration which this intervention can take. The buffer filling process can be restricted to a limited time span, e.g. five, seven, ten, fifteen or twenty minutes, so that the goal is to reach the best possible buffer target level in a condensed amount of time. This leads to increased planning security as it is known in advance how much time has to be spend on filling the buffers to target fill level at maximum. Of course no limit to duration can be chosen as well, allowing to reach the best possible starting configuration of buffers for the entire system.

3.3.5 Conclusion and identified intervention parameters

This section shows possibilities on how to influence buffer levels. These different possibilities have to be investigated and compared and the optimized solution for each target fill level and moment of intervention, etc. has to be found. To conclude, an overview of the identified levers is given and thus the search space for the following optimization is spread out.

The lever to begin with, is the target fill level of the buffer. Depending on the size of the manufacturing system and thus the number of i enclosed buffers B_i with variable buffer size, the number of combinations is very high. In case of having 10 buffers with the size 5 and possible target buffer fill levels from 0 to 5 this would lead to 6^{10} possible combinations, considering the exact fill level. The number of combinations yet grows, as there is the tolerated and minimum fill level as well.

Then the question on when to intervene has to be answered. It can be done as already explained before, with or after planned production stop. Apart from this three possibilities the frequency of intervention can be alternated and the duration of the intervention can be restricted to different lengths, or not.

Combining these levers leads to a vast amount of combinations which have to be explored and whose characteristics have to be shown to find an optimized solution on influencing buffer levels and thus increasing output.

3.4 Buffer filling algorithms

So far it has been explained how stopping machines influences buffer levels, which different target fill levels of the buffer are possible and at which moments in the daily routine to intervene and adapt the buffer level. Missing until now is how, when and in which sequence machines have to be stopped to reach the required target fill levels for all buffers and how it is initiated at all. This too, is part of the method of adaptive buffer operation. As mentioned in section 3.1 and in subsection 3.3.3 the system status is monitored a certain time before planned production stop (e.g. shift end). For each buffer B_i the current buffer level, $level_{current}(B_i)$, is compared with the target buffer level, $level_{target}(B_i)$ and the difference between those is calculated:

$$\delta_i = level_{target}(B_i) - level_{current}(B_i) \tag{3.1}$$

This is repeated every time an event altering a buffer level occurs. Knowing δ_i, it can be calculated how many units each machine M_i has to produce, so that buffer B_i reaches target fill level, referred to as $units(M_i)$. To begin with, an example on how to calculate $units(M_i)$ for a two machine transfer line, a simplified case of the study, is given and explained. Then algorithms for M_{n+1} machine transfer lines for the different target fill levels (exact, minimum and tolerated) are defined in subsection 3.3.2 are presented.

3.4.1 Simplified example: two machine transfer line

Given a system of two machines M_1 and M_2 with buffer B_1 in between (see Figure 3.4), the current buffer level and the exact target buffer fill level, it is very simple to calculate the amount of units both machines have to produce until stopping.

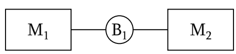


Figure 3.4 Two machine transfer line

First δ_1, the difference between $level_{target}(B_1)$ and $level_{current}(B_1)$, for buffer B_1 is calculated:

$$\delta_1 = level_{target}(B_1) - level_{current}(B_1) \tag{3.2}$$

For $\delta_1 < 0$ the buffer level needs to be reduced and the downstream machine M_2 has to go on producing the amount of $|\delta_1|$ units, while machine M_1 does not work. For $\delta_1 = 0$ both machines can cease work, as the target buffer level is reached. For $\delta_1 > 0$ the buffer in between has to be filled further. The upstream machine M_1 has to go on producing the amount of $|\delta_1|$ units, while the downstream machine M_2 does not work and thus stops $|\delta_1|$ units before machine M_1. An overview of the three cases is given in Table 3.2.

Table 3.2 Results for the different cases of δ_1

$\delta_1 < 0$	$\delta_1 = 0$	$\delta_1 > 0$
$units(M_1) = 0$	$units(M_1) = 0$	$units(M_1) = \|\delta_1\|$
$units(M_2) = \|\delta_1\|$	$units(M_2) = 0$	$units(M_2) = 0$

3.4.2 M_{n+1} machine transfer line

Yet in the manufacturing system dealt with, a serial transfer line consisting of $i = n$ buffers B_i and of $i = n + 1$ machines M_i as shown in Figure 2.1 on p. 8 the difficulty lies within the elements of the system influencing each other. So the filling procedure of a buffer cannot be reduced to the machines around the buffer but the system has to be viewed as a whole. Calculation of $units(M_i)$ is described in the following algorithms for the different fill levels.

Exact fill level as target

Calculating of $units(M_i)$ in case of an exact fill level as target is described in Algorithm 5 in pseudocode.[175]

In line 1 of the algorithm δ_i is calculated for each buffer. Now $units(M_i)$, the number of units each machine has to produce to reach the exact fill level of all buffers, is calculated (line 2). This is done by adding the differences between the target and current buffer levels for all buffers that are located after machine M_i in the sequence, starting with the buffer B_i.

[175] See *ibid.*, 54–55

Algorithm 5 Exact fill level

Calculate δ_i of each buffer ($i = 1$ to $i = n$)

$$\delta_i = level_{target}(B_i) - level_{current}(B_i)$$

Calculate amount of units each machine ($i = 1$ to $i = n + 1$) has to produce to fill the subsequent buffers:

$$units(M_i) = \sum_{i}^{n+1} \delta_i$$

Minimum fill level as target

Calculation of $units(M_i)$ in the case of reaching a minimum target fill level is performed iteratively. The pseudocode, given in Algorithm 6 is explained here in further detail.

Algorithm 6 Minimum fill level

1 Calculate δ_i of each buffer ($i = 1$ to $i = n$)

$$\delta_i = level_{target}(B_i) - level_{current}(B_i)$$

2 **Do**

3 Find leftmost buffer B_j (lowest j) with $\delta_j < 0$

4 Machine M_j has to produce $units_{tmp}(M_j) = |\delta_j|$

5 Machine M_j in total has to produce

$units(M_j) = units(M_j) + units_{tmp}(M_j)$

6 Assign new current buffer level $level_{current}(B_j) = level_{target}(B_j)$

7 Calculate $\delta_{j-1} = \delta_{j-1} - |\delta_j|$

8 Set $\delta_j = 0$

9 **while** any $\delta_i < 0$

To begin with δ_i is calculated for each buffer in the same manner as for the exact fill level. As long as any buffer has a level below the target fill level or $\delta_i < 0$ the following procedure is performed repeatedly (see lines 3 to 8 of the algorithm). First of all, the leftmost[176] buffer B_j which has a current fill level below minimum level is indicated. Machine M_j, which is physically located before buffer B_j, has to virtually[177] produce $units_{tmp}(M_j) = |\delta_j|$ temporary units in this iteration to balance the buffer B_j. The units are temporary units, as during

[176] The leftmost buffer is the first buffer visited by any unit in the manufacturing system. Viewing the system from top material flow is from left to right (source to spring) in the transfer line.

[177] In general while applying the algorithms no units are produced, as the algorithms are used to calculate how many units need to be produced.

balancing of another buffer, the fill level of an already balanced buffer might change, as the elements interfere with each other. Then $units(M_j)$ is calculated by adding the temporary units to the already produced units (line 5). Now buffer B_j is balanced and the buffer level matches the target buffer level (line 6). Yet as we are in an interconnected system, when machine M_j produces a certain number of units, this in turn has to be extracted from the buffer prior to it, which results in a change of its fill level and thus in a new δ_{j-1} (line 7). Concluding the action for buffer B_j its difference in target to current fill level can be set to zero, as it is already balanced for the time being. As mentioned, it might be influenced when balancing another buffer again. Now two δ values, δ_j and δ_{j-1}, have been adapted, and unless all values δ_i lie below zero the do-while loop is entered repeatedly.

Tolerated fill level as target

The code for our last algorithm, Algorithm 7, which is used to reach a tolerated fill level is very similar to Algorithm 6 used for the minimum fill level. As in case of the minimum fill level a minimum level has to be reached, but there is an additional restriction: the buffer fill level may not exceed a maximum level. To begin with, in line 1 $level_{min}(B_i)$ and $level_{max}(B_i)$, the minimum and maximum target fill level for each buffer is calculated using the target fill level and the range, $range(B_i)$. As $range(B_i)$ can adopt an odd value, and the buffer fill levels are restricted to integers, first the odd $range(B_i)$ needs to be adapted by subtracting 1, transforming the $range(B_i)$ to an even number. Hereafter, in line 2, the deviation for both the minimum and maximum buffer level is calculated, represented by δ_{i_min} and δ_{i_max}.

Succeeding, a do-while loop is entered, which is slightly altered from the one explained in Algorithm 6. It differs, as in this case the buffer might either be too full or too empty and two cases have to be regarded, which is done through using an if-else control structure. $count(\delta_{i\_min} < 0)$ indicates how buffers are below minimum fill level, $count(\delta_{i\_max} > 0)$ in contrast indicates the number of buffers exceeding maximum fill level.

If $count(\delta_{i\_min} < 0) \geq count(\delta_{i\_max} > 0)$, in other words there are more empty than overfilled buffers, balancing starts by filling the buffers (lines 5-12). This starts by finding the leftmost buffer below minimum fill level. The procedure of calculating $units(M_j)$ in line 7 is similar to Algorithm 6, with the difference that the temporary units to be produced are $units_{tmp}(M_j) = |\delta_{j\_min}|$ (line 6). As there are minimum and maximum buffer fill level

Algorithm 7 Tolerated fill level

For each buffer ($i = 1$ to $i = n$) calculate δ_{i_min} minimum and δ_{i_max} maximum buffer level

If $range(B_i)$ is odd

$range(B_i) = range(B_i) - 1$

else

End if

$level_{min}(B_i) = level_{target}(B_i) - range(B_i)/2$

$level_{max}(B_i) = level_{target}(B_i) + range(B_i)/2$

Calculate δ_{i_min} and δ_{i_max} of each buffer

$$\delta_{i\_min} = level_{current}(B_i) - level_{min}(B_i)$$
$$\delta_{i\_max} = level_{current}(B_i) - level_{max}(B_i)$$

Do

If $count(\delta_{i\_min} < 0) \geq count(\delta_{i\_max} > 0)$

Find leftmost buffer B_j (lowest j) with $\delta_{j\_min} < 0$

Machine M_j has to produce $units_{tmp}(M_j) = |\delta_{j\_min}|$

Machine M_j in total has to produce

$units(M_j) = units(M_j) + units_{tmp}(M_j)$

Assign new current buffer level $level_{current}(B_j) = level_{min}(B_j)$

Calculate δ_{j-1_min}=$\delta_{j-1\_min} - |\delta_{j\_min}|$

Calculate δ_{j-1_max}=$\delta_{j-1\_max} - |\delta_{j\_min}|$

Calculate δ_{j_max}=$\delta_{j\_max} - |\delta_{j\_min}|$

Assign $\delta_{j\_min} = 0$

else

Find rightmost buffer B_j (lowest j) with $\delta_{j\_max} > 0$

Machine M_{j+1} has to produce $units_{tmp}(M_{j+1}) = |\delta_{j\_max}|$

Machine M_{j+1} in total has to produce

$M_{j+1}(units) = M_{j+1}(units) + M_{j+1}(units\_tmp)$

Assign new current buffer level $level_{current}(B_j) = level_{max}(B_j)$

Calculate $\delta_{j+1\_min} = \delta_{j+1\_min} + |\delta_{j\_max}|$

Calculate $\delta_{j+1\_max} = \delta_{j+1\_max} + |\delta_{j\_max}|$

Calculate $\delta_{j\_min} = \delta_{j\_min} + |\delta_{j\_max}|$

Assign $\delta_{j\_max} = 0$

End if

While any buffer is not within the tolerated range

$\delta_{i\_min} < 0$ or any $\delta_{i\_max} > 0$

deviations, both have to be adapted for the succeeding buffer: δ_{j-1_min} and δ_{j-1_max} (line 9-10). Apart from that for the balanced buffer B_j δ_{j_max} has to be adjusted (line11). Only then δ_{j_min} can be set zero (line 12).

In the other case, if more buffers exceed maximum fill level, the reverse procedure is started (lines 14-21). The rightmost[178] buffer with $\delta_{j-1\_max} > 0$ is found. Machine M_{j+1} consecutive to or on the left side of buffer B_j has to produce $units_{tmp}(M_{j+1}) = |\delta_{j\_max}|$ temporary units, so that the level of buffer B_j drops to maximum fill level. Here producing of machine M_{j+1} results in altering the number of units in the succeeding buffer, by adding $|\delta_{j\_max}|$ to it (line 18-19). Here δ_{j+1_min}, δ_{j+1_max} and δ_{j_min} are adjusted and finally the buffer can be regarded as balanced for the moment and δ_{j_max} is set to zero. This is repeated until all buffers lie within the tolerated range.

3.4.3 Calculation of time to stop before end of production

Until now, for the four presented cases, including the simplified example, only $units(M_i)$, referring to the number of units the machine M_i has to produce to reach target fill levels has been calculated. To indicate how the machines have to be stopped stepwise to reach target fill levels for all buffers, $cycles(M_i)$ is calculated. $cycles(M_i)$ is the number of cycle times machine M_i has to cease production before the last machine to stop ceases production. The sequence of stopping is always the same for one defined target fill level set. The two fixed points are the beginning and the end of intervention. The last machine to stop production is taken as reference for all other machines, as this is the end of the intervention, too.

Algorithm 8 represents the pseudocode of the calculation of $cycles(M_i)$. It can be attached at the end of Algorithm 5, Algorithm 6 and Algorithm 7.

Algorithm 8	Calculation of $cycles(M_i)$
1	Find machine i with maximum amount of units to produce and assign $units\ (max)$: $units\ (max) = units(M_i)$
2	Calculate cycles each machine has to stop before machine i with the maximum amount of units to produce: $cycles(M_i) = units\ (max) - units(M_i)$

[178] The rightmost buffer is the last buffer before exiting the transfer line, when viewing the system from top, material flow is from left to right. The explanation is similar to the explanation of the leftmost buffer given before (minimum fill level).

First of all, the machine M_i, having to produce the maximum number of units so that the target fill levels are reached, needs to be identified among all machines. It is the machine M_i with the highest value of $units(M_i)$. Then $units(max)$ is assigned, which is the number of units the machine found beforehand has to produce. The variable $cycles(M_i)$ is the difference between $units\ (max)$ and $units(M_i)$ for each machine.

3.5 Simulation-based method used for optimization

The challenge in finding optimized solutions to the adaptive buffer operation method to fill the buffers is the large amount of combinations explained in subsection 3.3.5. To test which configurations are best, the buffer fill levels have to be adapted. These changes to the manufacturing system cannot be undertaken with the running system, as it is mainly experimental and the outcome is not sure.[179] This is why a simulation-based method has been developed. A real-world system is depicted in a simplified reproduction of itself in a simulation model. This has to be verified and validated, to show that the basic characteristics relevant for the optimization task are reproduced. Thus the simulation model is used to evaluate the objective function of the optimization.

This section is divided into the following subsections: beginning with subsection 3.5.1 the performance indicators to be monitored are introduced. This is followed by subsection 3.5.2, in which the optimization concept is presented. Concluding the search methodology, the Evolutionary Algorithms used, are described in subsection 3.5.3.

3.5.1 Performance indicators to be monitored

To get a broad understanding of what happens within the system during application of the buffer filling method some performance indicators of the manufacturing system as well as performance indicators for the buffer adaptation procedure are introduced and explained.

System-oriented performance indicators

$Output$: Output is the amount of units the manufacturing system produces within a given time period. In this concept the time period used is the length of one shift. The output then is a number of units per shift.

[179] ASIM 1997, 6; VDI 3633 Blatt 1 2014, 9

Manufacturing cost per unit: The manufacturing cost per unit is calculated using the output and the costs caused at each machine during the period of a shift. The only included costs are wage costs, as the method designed is based on extending or shortening shift duration for the workers. Fixed costs of electricity, maintenance, imputed interest and accounting depreciation are not included, as it is difficult to show them clearly. Machines and illumination is not directly turned off after machine stop because of maintenance. And maintenance is not necessarily being influenced by changing shift duration of some machines. Calculation of imputed interests and accounting depreciation is regularly based on life time estimates of assets, which makes it difficult to indicate those costs and apart from that, some machines are already depreciated. Wage costs in contrast are always influenced in the same manner and can be indicated directly, which is why only those are the focus.

The cost of each machine consists of wage costs for all workers engaged in work at one machine. The cost of all machines during one shift is added and then divided through the number of units produced during one shift, resulting in the manufacturing cost per unit:

$$\frac{cost}{units} = \frac{\sum_{i=1}^{n+1}(cost\ of\ machine\ M_i)}{units} \tag{3.3}$$

System-induced downtimes: Downtimes caused by other downtimes within the system, described as blocking and starving in section 2.1, are referred to as system-induced downtimes.

Method-oriented performance indicators

Operation time reduction: The operation time reduction is the time a machine stops before a planned production stop (as shift end). It is measured for each machine each time adaptive buffer operation is applied. For the moment of intervention “after” the values are always zero, as the first machine stops when the production stop is planned. For each machine the average value is calculated and the maximum value recorded.

Operation time extension: The operation time extension is the time a machine goes on operative after planned production stop (as shift end). It is in the same way as the operation time reduction and the same values are recorded: average and maximum. For the moment of intervention “before” the values are always zero, as the last machine to cease production stops when the production stop is planned.

3.5.2 Optimization concept

The optimization concept is based on finding optimized values for the buffer target fill levels for different objectives. This concept is specified, the objectives presented, the optimization strategies outlined and the Evolution Strategies used for the search are named.

Objectives

The objectives used for optimization are based on the performance indicators described in the previous subsection 3.5.1.To improve the system efficiency, three objectives for optimization are appointed:

1. Maximize system output
2. Minimize manufacturing cost per unit
3. Minimize system-induced downtimes

Maximizing system output is the main target of this optimization problem. Apart from that, another point not to neglect is the costs of maximizing system output, as producing more units at increased costs may not have any benefit. This is why another objective is to minimize manufacturing costs per unit. The concluding objective is minimizing system-induced downtimes. Those downtimes result from insufficient decoupling of the machines in the system and minimizing those may lead to increased output. All together these objectives lead to higher efficiency in manufacturing systems.

Solution methodology

This problem involves multiple objectives, as normal real-world problems do in general.[180] As the objectives might be conflicting (achieve maximize output and minimize cost), it is to be expected, that each objective has a different optimal solution. Thus choosing one solution is difficult, as improving one aspect might result in a compromise in another objective and none of the solutions can be named best solution.[181] Of course it is possible to use preference-based multi-objective optimization, as explained in section 2.3. Yet here the result always depends on the composite function formed before optimization, which in turn depends on the preferred objective. Altering this preferred objective results in a change of the

[180] Deb 2004, 1; Eckart Zitzler 2012, 872
[181] Deb 2004, 1–2; *ibid.*, 3; *ibid.*, 4

trade-off solution, as always only one solution is found.[182] This is why neither single-objective optimization, nor preference-based multi-objective optimization are an adequate methodology for solving the problem. Hence multi-objective optimization, treating all objectives equally, is applied. This solution methodology gives the user freedom in his decision-making process and instead of arbitrarily choosing a solution beforehand, the different solutions can be considered, compared to each other and a comprise solution can be chosen without neglecting negative effects.[183] To facilitate understanding of the solutions and to be able to consider each of the found trade-off solutions, these are joined to the Pareto-optimal front.[184]

The multi-objective optimization is performed with all three objectives at once. Apart from that for the purpose of clearer visualizations it is performed with only two objectives at a time, as the solutions depicted in a two-dimensional Pareto-optimal front are easier to understand.[185] Thus three bi-objective optimization tasks result:

1. Maximize system output and minimize manufacturing costs per unit
2. Maximize system output and minimize system-induced downtimes
3. Minimize manufacturing costs per unit and minimize system-induced downtimes

Nevertheless, before starating with multi-objective optimization, some single-objective-optimization runs are undertaken, to test the potential of the idea in general and to compare the different target fill levels and moments of interventions. With this, a deeper understanding of the functioning of the method can be obtained. The objectives used for this are the first three objectives appointed at the beginning of this subsection. They are neither adapted nor transformed or combined to a composite objective function and are pursued separately.

3.5.3 Evolutionary Algorithms used for search of solutions

As the variables in the proposed optimization task are the buffer fill levels and thus the problem is a combinatorial problem (and not continuous) there are two possible solution

[182] *ibid.*, 6
[183] Eckart Zitzler 2012, 879; Deb 2004, 25
[184] *ibid.*, 20; *ibid.*, 4
[185] Backhaus 2008, 547

methods (see subsection 2.3.2): either exact methods or heuristics.[186] As this real life problem is difficult, because the number of solutions is high and enumeration would require long computing time, heuristics are chosen.[187] This enables obtaining solutions to the problem faster.[188] Within those metaheuristics are chosen, as they are applicable to a variety of abstract problem types.[189] Since metaheuristics e.g. Simulated Annealing and Tabu Search iteratively improve only one solution in the neighborhood, only one optimized solution is found at a time.[190] So to find solutions to multi-objective optimization tasks applying Simulated Annealing or Tabu Search various runs have to be performed. Yet there is another type of metaheuristic, which is unique in solving these kind of problems: The Evolutionary Algorithms.[191] As described in subsection 2.3.3 Evolutionary Algorithms are population-based optimization algorithms having a whole population of solution as result. Evolutionary Algorithms can find multiple optimal solutions covering the Pareto set with only one run and are perfectly suited for and applied to the given optimization task.[192] Within Evolutionary Algorithms Evolution Strategies are selected. Here the variables are real valued numbers[193] and the fitness is the objective function value[194]. This characteristic facilitates application to the optimization problem, as no complicated encoding is required which would be needed when using Genetic Algorithms or Evolutionary Programming.[195]

Evolution Strategy for Single-objective Optimization

The above mentioned reason explains, why Evolution Strategies are chosen for solving the multi-objective problem. Yet, to find out whether the proposed strategy of filling buffers leads to the goal of increased efficiency and to compare the moments of intervention and fill levels to each other the first optimization runs have single-objectives. For single-objective optimization tasks any metaheuristics can be applied. Yet as Evolution Strategies are used for the multi-objective optimization tasks anyways, they are also applied for single-

[186] Bangert 2012, 5; Domschke *et al.* 2015, 134–135; Zäpfel *et al.* 2010, 32; Martí and Reinelt 2011, 17; Aarts and Korst 1989, 4; Collette and Siarry 2003, 5–7
[187] Bangert 2012, 5; Domschke *et al.* 2015, 125; *ibid.*, 135; Martí and Reinelt 2011, 17; Collette and Siarry 2003, 5–7
[188] Bangert 2012, 5–6; Domschke *et al.* 2015, 135; Martí and Reinelt 2011, 17
[189] Zäpfel *et al.* 2010, 68; Domschke *et al.* 2015, 137
[190] Here and in the following: Collette and Siarry 2003, 5–7
[191] Deb 2004, 7–8
[192] Bäck 1996, 35
[193] *ibid.*, 68
[194] *ibid.*, 132
[195] *ibid.*, 108

objective optimization, so that no additional methods have to be adapted or developed for this purpose.

In single-objective optimization the goal is to find an optimum solution, which is done by comparing the objective function values and either accepting the new solution, as it is better than the old one or by rejecting it.[196] This is done using the DR2 Evolution Strategy described in subsection 2.3.3. The DR2 developed in 1994 is not the latest Evolution Strategy. The DR2, comparing to the first Evolution Strategies published, is an improved Evolution Strategy, as it is derandomized which has been explained in subsection 2.3.3.[197] While examining contemporary algorithms Bäck et al. 2013 found the DR2 to be a very robust algorithm, which sometimes even outperforms the (1+1)-Active-CMA-ES[198], which was designated to be the best algorithm.[199] Apart from that the software ClearVu Analytics (CVA) used for optimization, presented in the following section 3.6, uses the DR2 algorithm. Table 3.3 maps the fitness for each of the proposed objectives.

Table 3.3 Objectives and assigned fitness

Objective	Fitness
Maximize	System output
Minimize	Manufacturing cost per unit
Minimize	System-induced downtimes

Evolution Strategy for multi-objective optimization

Now to be able to weigh the different solutions in regard of the different objectives and to be able to choose one adequate solution multi-objective optimization runs are undertaken. Here the goal is not only to find one optimal solution, but to find the Pareto-optimal front and additionally obtain a set of solutions which is as diverse as possible, as all objectives are of equal importance.[200] For this purpose the DR2 Evolution Strategy is combined with the NSGA-II (explained in Algorithm 4) in the software CVA. Here the NSGA-II is responsible

[196] Deb 2004, 24; *ibid.*, 13
[197] See Ostermeier *et al.* 1994a, 371
[198] Arnold and Hansen 2010
[199] Bäck *et al.* 2013, 85
[200] Deb 2004, 24; *ibid.*, 22

for sorting and selecting the individuals and the DR2 is used for mutation, as the NSGA-II does not have a defined rule on how to create offspring (see Algorithm 4, p. 24).

The adapted NSGA-II algorithm only differs slightly from Algorithm 4, which is explained in the following. First, the sizes of the parent population and of the offspring population differ: $|P| = \mu$ and $|Q| = \lambda$. Another difference is, that before creating any offspring population Q the parent population P is recombined using intermediate recombination and from this the offspring population Q is created by using the mutation operators from the DR2 Evolution Strategy.

The DR2 Evolution Strategy helps that individuals mutate towards optimized solutions and the NSGA-II includes sorting of the individuals into the different Pareto-optimal fronts. Apart from that, the NSGA-II supports reaching the goal of a set as diverse as possible, using the crowded comparison operator $<_c$, which was already explained in subsection 2.3.3, in the part of Evolutionary Multi-objective Optimization.[201]

3.6 Implementation of the simulation-based solution method

The solution method consists of many different components, as simulation, optimization and Evolution Strategies which are executed together so that the optimal buffer fill levels are found. To facilitate application and to have an accelerated process, this is automated by a tool described in the following.

Development of an implementation tool

Evaluation of the objective function is done by discrete event simulation, which calculates the fitness. For optimization, the software CVA[202] is used, and a simulation tool is connected to CVA with a Dynamic Link Library (DLL). The optimizer in CVA calls functions from the DLL. The simulation tool was created (in C++) for linear manufacturing systems and is configured with an xml-file, containing information as the succession of machines and buffers and their attributes. The optimization is controlled with an xml-file, containing information as the objectives, design variables and their bounds, which can be adapted manually.[203]

[201] *ibid.*, 252
[202] ClearVu Analytics uses a DR2 Evolution Strategy for optimization.
[203] The interconnection of the optimizer with the simulation tool and the simulation tool itself was created by divis intelligent solutions GmbH.

Initialization of the optimization workflow (see Figure 3.5) begins with choosing design variables randomly at first followed by evaluation. First, the design variables are set in the simulation with the interface function extSetValue, then the simulation is run (extRun). After ending, the fitness value is returned (extGetValue) to the optimization which is then evaluated. After that, it is selected and the evolution loop is entered from the beginning starting with recombination.

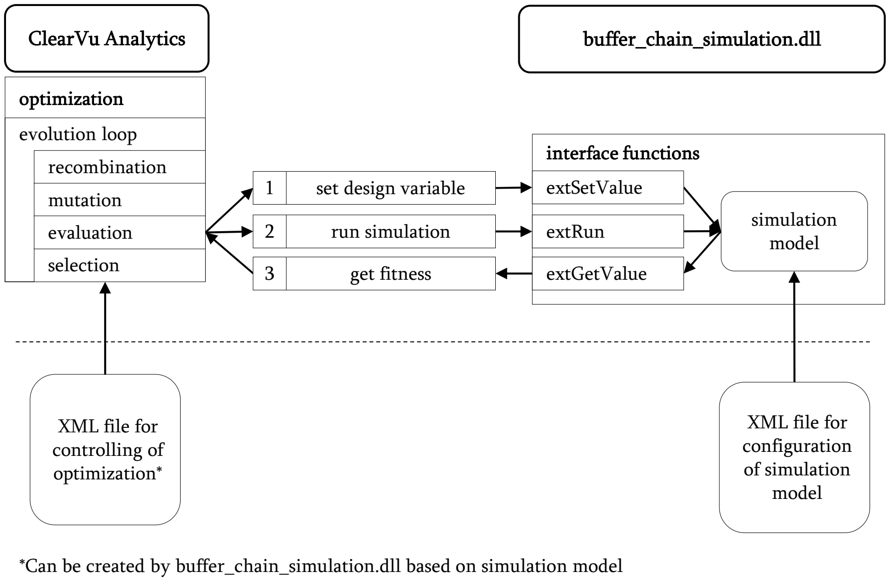


Figure 3.5 Optimization workflow

4 Application of method

In the previous chapter the designed method of advanced buffer operation and simulation-based optimization have been described. In this chapter the developed method is applied to a real-world problem to show how the proposed concept is performing. In section 4.1 model building of the real life manufacturing system is described. The performance of the proposed method is shown in section 4.2, where the results of the undertaken experiments are presented. Concluding, in section 4.3 the developed method is assessed and discussed.

4.1 Model building

The method used for optimization is simulation-based, yet there is no already existing validated simulation model of the regarded manufacturing system. This is why within this section model building, verification and validation is presented. The applied simulation procedure is based on the model developed by *Rabe et al.,* which is outlined in Figure 4.1.[204] The procedure is split into Task Definition, System Analysis, Model Formalization, Implementation and Experiments and Analysis (boxes with rounded off corners in Figure 4.1). The sequence of processing of this task is as proposed. The results of each task are named on the left side (in rectangles). The tasks Data Collection and Data Analysis are not within sequence, as those tasks can be done independently and apart from the rest of the process. Validation and Verification of Data and of the model is done throughout the whole procedure. The needs and the task description has already been given in the previous chapters. A system analysis is performed in subsection 4.1.1 with the resulting conceptual model, which is „*a non-software description of the simulation model (…) describing the objectives, inputs, outputs, content, assumptions and simplifications of the model.*“[205] Data collection and processing is presented in subsection 4.1.2. Model formalization, being a further development of the conceptual model, enabling to transform it into an executable simulation model,[206]

[204] Here and in the following see: Rabe *et al.* 2008, 4–8, which is a development of ASIM 1997, 13–23
[205] Robinson 2007, 65
[206] Rabe *et al.* 2008, 48–50

has been skipped, as the implementation is performed in cooperation with technical experts in the area and the information given in the conceptual model and data is extensive. Apart from that, other authors e.g. *Robinson* do not preview this separate step.[207] Subsection 4.1.3 gives an overview of applied methods of verification and validation used throughout the procedure of developing the model. The executable model is finalized, adapted, verified and validated.

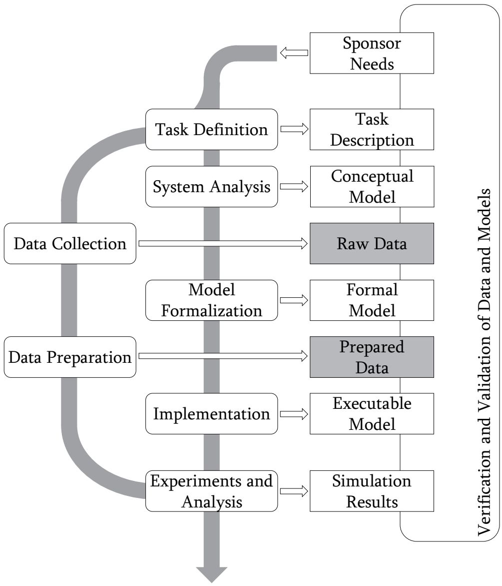


Figure 4.1 Procedure model for simulation including verification and validation[208]

4.1.1 Conceptual model

As described in section 3.1, the general outline of the method, the simulation model is used to evaluate the objective function of the optimization task. The model is simplified, depicting only the main characteristics necessary for the optimization task.[209] Keeping the model

[207] Robinson 2007
[208] Rabe *et al.* 2008, 5
[209] See Robinson 2007, 68; VDI 3633 Blatt 1 2014, 22

simple is a principle that many researchers support, as simple models are easier to understand, undertaking experiments is easier, and changes can be implemented without complications.[210] This principle is followed while setting up the model.

The object investigated in this case study is an already existing transfer line of an automotive manufacturing system, where different models of cars are assembled. The studied transfer line, depicted in Figure 4.2, consists of nine interconnected machines, which are separated by eight FIFO buffers. The model is unsaturated, meaning, that the source and sink can have downtimes as the transfer line is a part of an entire manufacturing system.

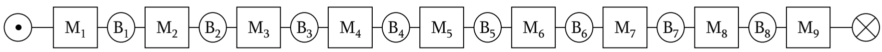


Figure 4.2 Manufacturing system investigated in case study

Each machine has various stations which operate simultaneously. In Figure 4.3 two exemplary machines and buffers are displayed. The stations within machine M_1 are numbered from S_1 to S_6. A stop at one of these stations results in a stop of the entire machine. Machines have cycle times and are unreliable, having unpredictable failures. Buffers have limited capacity and a lead time and function as described in section 2.1. The two shown buffers in Figure 4.3 are not filled completely. The machines and buffers operate only during operation time, meaning during shift. After shift and during breaks production is stopped. The units remain at the position encountered at the moment of the stop.

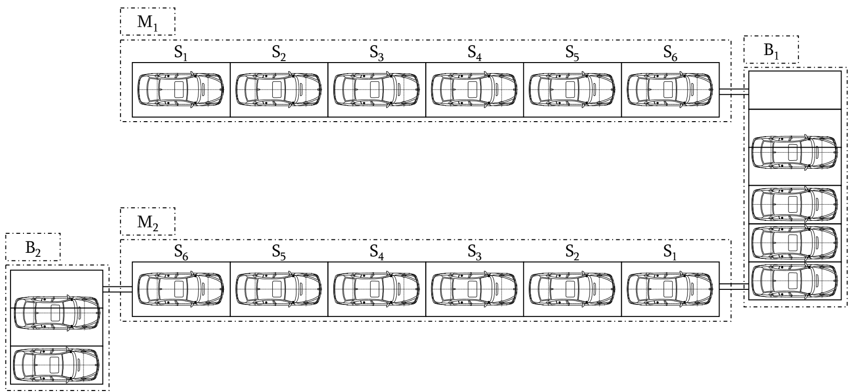


Figure 4.3 Overview of automotive transfer line with stations at machines

[210] See among others Pidd 1996, 721–722; Salt 1993, 1–5; Chwif *et al.* 2000, 452; Robinson 2007, 68; VDI 3633 Blatt 1 2014, 22

The inputs for the components of the system are given in Table 4.1, the outputs of the system are equal to the performance indicators introduced in subsection 3.5.1. The assumptions stated in section 3.2 are valid.

Table 4.1 Inputs needed for each component

Overall	Machines	Buffers
Start / End of shift	Downtime distribution	Maximum fill level
Start / End of break	Availability	Lead time
Number of shifts	Cycle time	Initial buffer fill level
	Cost per minute of operation	

4.1.2 Data collection and processing

The time period of data collection is from March to September 2013, covering 330 shifts. Operations are executed during two shifts per day, early and late shift. Each shift has two breaks.

The data used for simulation modeling is mainly available from the IT system of the manufacturing system. For the machines it consists of the downtime logs and the cycle times. For the buffers it consists of buffer sizes and of buffer fill level logs. The lead times of the buffers have been measured separately. For the whole system, shift and break times are logged in a different file, apart from that existing drawings of layouts are included. Table 4.2 shows the operating hours of the shifts.

Table 4.2 Shift model

	Shift			Break 1		Break 2	
	Duration [min]	Start	End	Start	End	Start	End
Early	525	05:00	14:30	08:15	08:30	11:25	11:55
Late	525	14:30	00:00	17:40	18:10	21:30	21:45

Data processing

The data extracted from the IT system is unprocessed and raw. For the simulation model it is processed as described for each data type in the following subsection. The raw data is prepared, which is done through structuring, condensation and correction.[211] The data used

[211] Wenzel *et al.* 2008, 28; Rabe *et al.* 2008, 52

for building the simulation model concerning machines are listed in Table 4.3 and Table 4.4, the ones concerning buffers in Table 4.5.

Machine downtime logs: The downtimes of the machines are logged with a starting time and an end time for each downtime and can be summarized to the following classifications: technical downtime (including organizational downtimes), starving and blocking. The extracted set of data is corrected, so that all downtimes occur during operating time (e.g. not within a break) and that only one downtime at a time takes place, starting with the downtime having occurred at first. Moreover, plausibility of downtimes is checked. For blocking and starving it is verified, that blocking only occurs, when the downstream machine is down and vice versa for starving. Hereafter the repair time and time between failure for each incident and each machine is calculated. The average downtime per shift is calculated as well, split in technical, blocked, starved and the sum of all downtimes (Table 4.4).

Technical availability: The technical availability results from machine uptime and the technical downtimes and is calculated for each machine. Apart from that, a distribution is fitted to the technical downtimes. The gamma distribution is chosen, as it is commonly used for failures[212] and in this case has the best fit. The parameters α and β are settled.

Machine cycle times: The machine cycle times are extracted from the IT system and are verified by measuring the cycle time during operation. In case of deviation they are reviewed and corrected, as for machine M_2 the cycle time is increased from 50.08 seconds to 55.15 seconds. This is because machine M_2 is a line consisting of various automated stations where the cycle time setting of the most unreliable machine is 50.08 seconds, whereas the other stations work at 55.15 seconds.

Capacity: With the longest machine cycle time of 55.15 seconds the capacity of the system is 571 units per shift.

Output: For each shift the output is counted at each machine. The counter is incremented by one, every time a unit leaves the machine. After extracting the data from the IT system it is corrected, if e.g. before one shift the counter is not set to zero. The output per shift, measured at the last machine of the system, is 558 units on average.

[212] See Robinson 2007, 104; Wiendahl and Hegenscheidt 2003, 74 naming the Erlang distribution, which is a special case of the gamma distribution

Overall system availability: With the above stated capacity and output this results in an availability of approximately 98%, which is highly sophisticated.

Machine cost per minute: The machine cost per minute is needed to be able to calculate the costs per unit during optimization. As explained in subsection 3.5.1 the only included cost are wage costs of all workers at each machine. The machine cost per minute of operation is calculated for each machine separately, as at each machine there is a different number of stations with a different number of workers, having different qualifications. So the cost of a machine per minute is the sum of the cost of each worker to be found at every machine. These costs can vary enormously, as can be seen for machine M_2 in Table 4.3: here automation is higher, which is why less operators are required and thus cost per minute of operation is low in comparison to other machines.

Table 4.3 Machine data used for building simulation model

Machine	Cycle time	Cost per minute	Technical availability	Gamma distribution α	β
source	-	-	99.05%	1.15	8189
M_1	55.15 s	41.96 €	99.64%	1.38	7286
M_2	55.15 s	8.53 €	99.30%	1.99	3155
M_3	55.15 s	46.65 €	99.52%	1.30	6262
M_4	55.13 s	47.61 €	99.39%	1.27	6788
M_5	55.15 s	36.20 €	99.58%	1.29	6430
M_6	55.13 s	25.75 €	99.65%	1.38	3679
M_7	55.13 s	31.62 €	99.47%	1.79	3679
M_8	55.13 s	28.63 €	99.47%	2.03	2315
M_9	55.15 s	37.16 €	99.57%	1.29	4959
sink	-	-	99.36%	1.26	11049

Table 4.4 Downtime distribution per shift in [s] – reference data

	Technical	Blocked	Starved	Sum down
M_1	114	249	300	663
M_2	221	352	133	706
M_3	150	90	305	545
M_4	192	132	358	682
M_5	134	115	375	624
M_6	110	148	443	702
M_7	167	127	616	910
M_8	166	122	362	651
M_9	137	202	409	748

Buffer sizes: The buffer sizes extracted from the IT system are verified by comparing them with the drawing of the layout and during on-site visits. Four buffer sizes are altered comparing to the extracted data. The maximum fill level of buffer B_1 and B_2 is decreased, as within the IT system the stations themselves are falsely counted in. Buffer B_3 and B_4 are decreased, too, according to the layout and after checking on-site. In case of buffer B_6 and B_7 assembly stations in front and after the automated station are neglected in the IT system, so they are downsized, too. Minimum buffer level for each buffer is zero.

Buffer lead times: The buffer lead times are measured during an on-site visit at the shop floor. For the downsized buffers the times are adapted. This was performed measuring the time the units need to enter and exit the buffer and the time to go through the entire buffer. Entering and leaving times remain the same, the time to go through the buffer was adapted accordingly.

Buffer fill level logs: The level of buffer filling is logged each time the event of a unit entering or leaving the buffer happens. After viewing the data it is concluded that it is not to be used, as the deviation is very high and correction is not possible. E.g. some buffers miscount, exceeding maximum buffer level, not following any rule. To ensure that the results obtained are not biased by an inappropriate starting state, the buffer fill levels are set initially, thus avoiding a warm-up period.[213] For this the first shift's buffer fill levels are picked arbitrarily, as the data collected is not suitable.

Table 4.5 Buffer data: collected and adapted for simulation

	Collected data		Adaptation		
	Lead	Buffer level	Lead	Buffer level	
Buffer	time	maximum	time	maximum	start
B_1	115 s	20	40 s	5	1
B_2	172 s	15	172 s	10	5
B_3	191 s	11	139 s	8	5
B_4	211 s	12	106 s	6	3
B_5	190 s	10	190 s	10	5
B_6	454 s	20	227 s	8	5
B_7	475 s	20	238 s	8	5
B_8	121 s	10	121 s	10	5

[213] Wenzel *et al.* 2008, 147

4.1.3 Verification and validation (V&V) of model

This subsection gives an overview of methods applied to verify and validate the simulation model. As already stated, verification and validation is not a single procedure following in the sequence of steps required while building a simulation model, but covers the entire length of the project.[214] This is why within the previous subsections, as describing the conceptual model and data collection and processing, V&V has already been applied, when e.g. comparing automatically logged data to reality. In general, the criteria Rabe et al. proposes are reviewed, and it is checked, if the contents and structures are correct, if the result is accurate enough and if it is applicable to the stated problem.[215] Apart from this, Rabe et al also proposes various techniques to show proof of correctness[216]. The model and data was discussed with experts ("Face Validity" and "Structured Walkthrough"), and all obtained information and data were checked on consistency, correctness and clarity ("Desk Checking"). Relationships between cause and effect in the system have been described in "Cause-Effect Graphs" and are found in the simulation model. To increase confidence in the model "Trace Analysis", checking data logs of single elements of the system to verify their correct performance, and "Submodel Testing", was performed. Simulation runs with different input parameters were executed, obtaining the same effects on downtimes and outputs as expected in reality ("Sensitivity Analysis", "Internal Validity", "Extreme-Condition Test" and "Fixed Value Test"). Finally, the results of the model were compared to historical data.

Objective

The objective of building a simulation model that depicts relevant characteristics of the real manufacturing system is pursued. As the optimization is based on improving the decoupling effects of buffers and as technical downtimes remain unavoidable, downtimes resulting from blocking and starving come into close focus. What matters is to depict these downtimes in a correct ratio to each other for each machine and to reach a system output of about 558 units per shift to depict reality as close as possible.

[214] Rabe *et al.* 2008, 7
[215] *ibid.*, 22–23
[216] Here and in the following: *ibid.*, 95–116

Results

Applying each machines' technical availability from the reference data led to an average output of 565 units per shift, which is too high comparing to the target 558 units per shift. To reach this target, the availability was adapted slightly, deteriorating the availability of each machine by the same percentage until the output was 558 units per shift. The current deviation of availability of 0.41% is acceptable. Yet as the manufacturing system studied is highly sophisticated, having availabilities of above 99%, the effect on technical downtimes per shift is relatively high, when decreasing each machine's availability and comparing to the reference data. The downtimes per shift resulting from this simulation setting are listed in Table 4.6.

Table 4.6 Downtime distribution per shift in [s] – simulation data

	Technical	Blocked	Starved	Sum down
M_1	243	144	323	711
M_2	325	95	279	698
M_3	279	163	278	721
M_4	320	129	256	705
M_5	262	111	348	720
M_6	240	174	288	702
M_7	296	151	269	715
M_8	295	134	293	723
M_9	266	265	204	735

In Figure 4.4 reference data (Table 4.4) is compared to the data obtained by simulation (Table 4.6) for the different types of downtimes in four charts: a) technical, b) starved, c) blocked and d), the sum of downtime. The average downtime per shift in seconds is plotted on the y-axis, the corresponding machine on the x-axis.

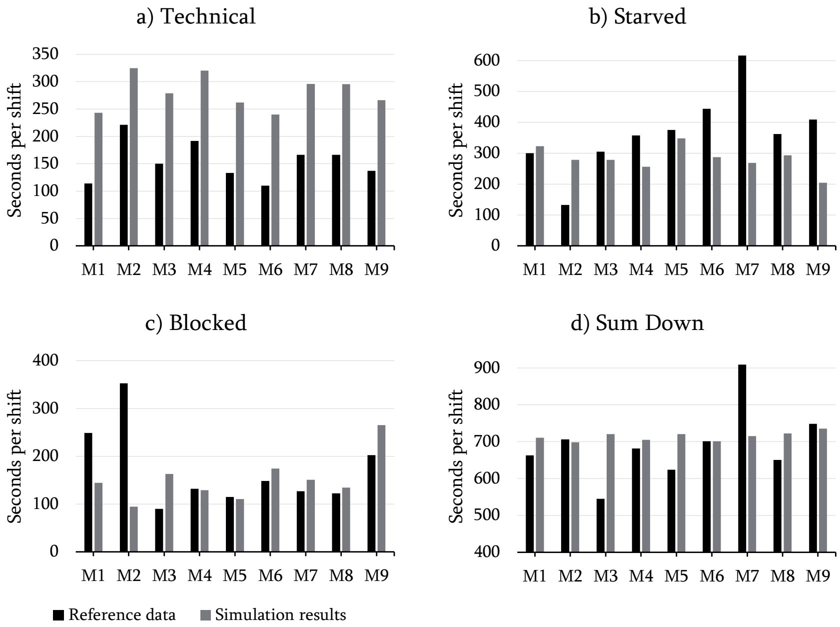


Figure 4.4 Downtimes per shift: comparison of reference and simulation data

When regarding chart a) technical downtimes, it can be seen, that by altering the availability slightly (deterioration of 0.41% as stated before), the impact on the technical downtimes is nearly twice as much. This is due to the fact, that the availability is very high and thus the resulting technical downtimes per shift (110 to 221 seconds or 2 to 4 minutes) in the reference data are small comparing to shift length of 525 minutes.

Yet regarding b) starvation, c) blocking and d) sum all, the downtimes match quite well. The deviations are explained step by step, starting with the chart d).

The most striking deviation is the deviation of machine M_7. Viewing the charts b) starved and c) blocked downtimes it can be seen, that the deviation is a result of the deviation in starvation. This is explained by a measurement error on-site, which was discovered during a visit after obtaining first simulation results. The point for measuring starvation is not directly before machine M_7, but at the entrance of the upstream buffer B_6. Thus in the reference data the technical downtimes of the upstream machine M_6 are directly counted into the starved times of machine M_7.

The discrepancy of all downtimes of machine M_3 can only be explained partly. As the blocked times of the reference data comparing to simulation data are lower, a lower sum of downtimes results automatically. As measurement procedure is correct, the error source is suspected within data processing, as in the raw data blocked downtimes often appeared while another downtime had already started. Although this deviation cannot be explained properly, it was accepted by the experts operating the system.

Deviation of machine M_5 can be explained by the deviation of the technical downtimes (chart a)) which is induced through decreasing each machines availability, as blocked and starved downtimes are nearly equal.

The same applies to M_8. Here the sum of blocked and starved downtimes, comparing reference to simulation data is nearly equal.

Now the most striking discrepancy of reference and simulation data concerning the sum of downtimes have been discussed. When focusing on deviation in starvation machines M_6 and M_9 have to be explained (M_7 has already been discussed). Both have a higher share of blocked times in simulation results. Apart from that the deviation of starvation time in the model is lower than in reality, making this deviation acceptable. An improved decoupling of machines and thus increased output is mainly achieved by preventing starvation, so the improvement potential of the simulation model is not higher than of the real-world system.

The starvation deviation at machine M_2 can be explained, when having a look at blocked times. For machine M_2 blocked reference data is much higher than simulation data. This discrepancy is explained by machine M_2 being a machine consisting of various automated stations, where the signal of blocking is not measured at the last station but in one of the middle stations. This deviation was discussed with specialists and a measurement error was confirmed during an on-site visit. Here downtime logs are compared to the downtimes in real-time and circuit diagrams of the machines were studied. So the amount of blocking in the reference data of machine M_2 leads to a much lower share of starvation downtime at this machine. When the machine is already blocked, no matter if mistakenly, other downtimes, e.g. starvation cannot be logged.

It can be seen, that there are many discrepancies when comparing the simulation results to the reference data, yet the overall result when summing up all downtimes match well and the deviations there are accepted by production experts on-site. Of course, when regarding starved or blocked times there are sole deviations which do not fit well, but they are compensated when summed up, which leads to the assumption, that the reference data measurement has not been performed correctly. Finally, after discussing these results with the experts operating the system their confirmation of confidence in the model is given and the model is used for optimization.

4.1.4 Number of replications and run-length

The run-length is set to 300 shifts and the number of replications to 30. This run-length and number of replications was determined using the methods *Robinson* proposes.[217]

Run-length was determined graphically. In Figure 4.5 the cumulative means from three replications are plotted on a graph, the exact results are attached in appendix D, Table D.2. It can be seen, that the cumulative means converge at around 250 shifts. At 300 shifts, the set run-length, the level of convergence reaches 0.04%. Additionally, for each replication the plots including a confidence interval of ($\alpha = 5\%$) are attached (see Figure D.2 to Figure D.3).

With the above set run-length, the number of replications was determined using the confidence interval method. Figure 4.6 shows the cumulative mean and the confidence intervals graphically. The significance level α of 1% is selected. It can be seen, that the interval is narrow and the cumulative mean line is flat at the point of 30 replications. This significance level is already reached with three replications (see Table D.1). Further reason to apply as many as 30 replications is to possibly depict the whole spectrum of states the simulation model can take. The high variety of combinations resulting from adaptive buffer operation (target fill level, moment and duration of intervention, frequency, etc.) is additionally increased through the high variety of system states the simulation model can assume. Within the model downtimes can occur at different moments, different machines and be of different duration. To depict as many combinations and to ensure, that the optimized results obtained from adaptive buffer operation, the number of replications is increased.

[217] Here and in the following: Robinson 2007, 152–158

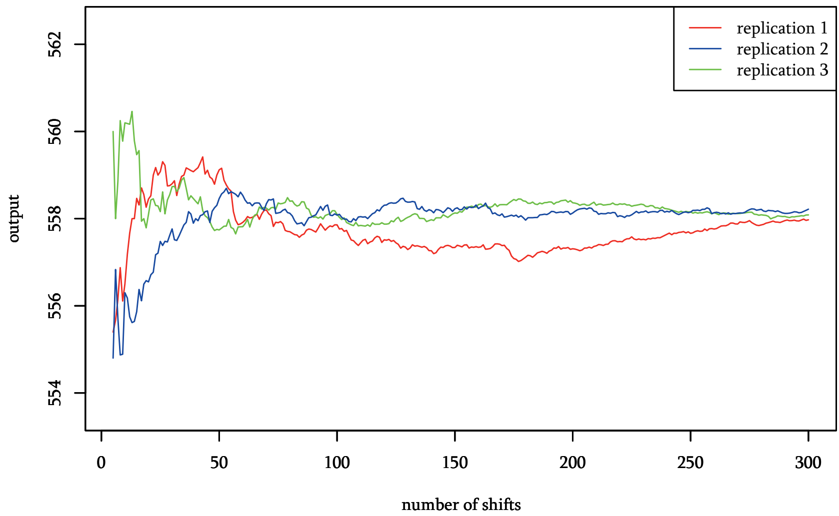


Figure 4.5 Cumulative means from three replications

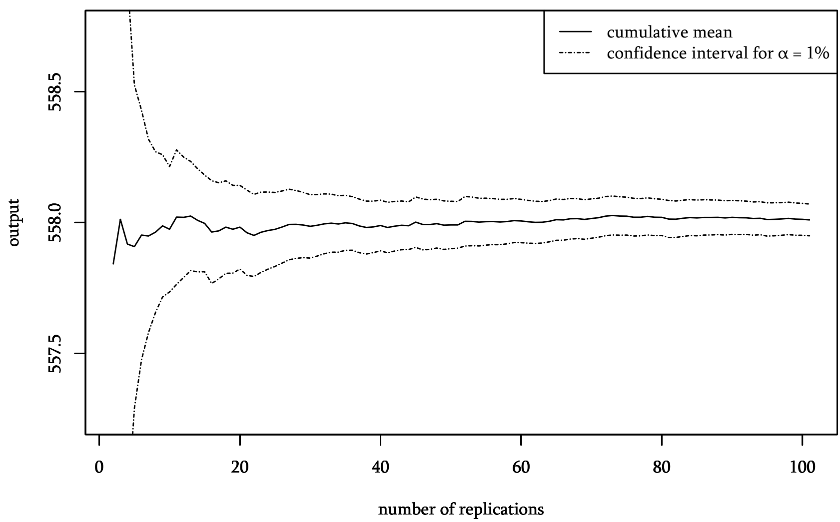


Figure 4.6 Cumulative mean and 99% confidence intervals (output)

4.2 Experiments

In this section experiments to test the proposed method's performance are conducted. Optimized buffer level parameters are determined using optimization. For a first performance evaluation of the different buffer levels and of the proposed moments of intervention, the optimization task is started with single-objectives (subsection 4.2.1) and succeeded by multiple objectives (4.2.3). In subsection 4.2.4, further experiments with different duration of intervention and frequency are performed and the obtained results presented.

Table 4.7 gives recommended parameter settings for the single-objective optimization, Table 4.8 for multi-objective-optimization.

Table 4.7 Parameter settings for the single-objective optimization

Number of offspring	$\lambda = 10$
Number of parents	$\mu = 2$
Maximum number of generations	200
Number of shifts	300
Number of replications	30

Table 4.8 Parameter settings for the multi-objective optimization

Number of offspring	$\lambda = 35$
Number of parents	$\mu = 5$
Maximum number of generations	400
Number of shifts	300
Number of replications	10

4.2.1 Single-objective optimization: minimize manufacturing cost per unit

In this subsection, the results obtained from the single-objective optimization run with the objective of minimizing manufacturing costs are presented and discussed. The interventions resulting from adaptive buffer operation were performed every two shifts at end of late shift. Monitoring of the system state was initiated 30 minutes before end of shift and maximum operation time extension for "after" and "with" was set to 30 minutes. Combining the different moments of intervention and target fill levels, nine optimization runs result. The identified target fill levels are listed in Table 4.9 (exact fill level), Table 4.10 (minimum fill level, referred to as "min") and Table 4.11 (tolerated fill level, named "tol"). Figure 4.7 depicts all buffers with their maximum size and within these all possible buffer levels depending on the moment of intervention and target buffer level are indicated. It can be observed, that the target fill levels are very similar to each other, especially when comparing the moment of intervention for one fixed target fill level, e.g. see "exact" and compare "before", "with", and "after".

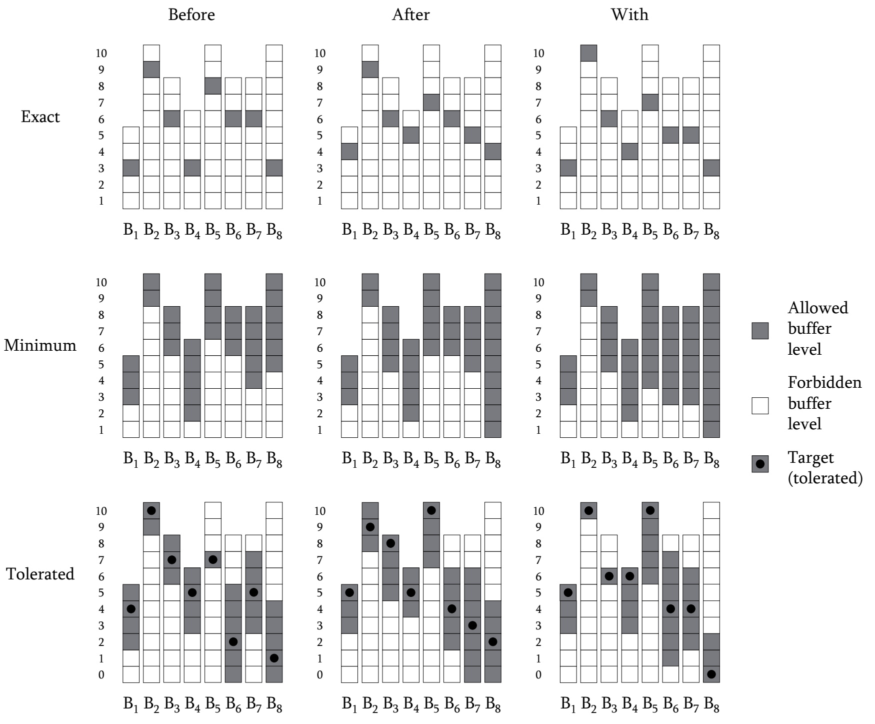


Figure 4.7 Feasible and unfeasible buffer levels depending on moment of intervention and target fill level[218]

Table 4.9 Exact target fill levels for minimizing manufacturing costs

Moment of intervention	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
Before	3	9	6	3	8	6	6	3
With	3	10	6	4	7	5	5	3
After	4	9	6	5	7	6	5	4

[218] Notice: range zero for tolerated fill level is possible as stated in subsection 3.3.2 and results here.

Table 4.10 Minimum fill levels for minimizing manufacturing costs

Moment of intervention	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
Before	3	9	6	2	7	6	4	5
With	3	9	5	2	4	3	3	1
After	3	9	5	4	6	6	5	1

Table 4.11 Tolerated fill levels (T = target and R = range) for minimizing manufacturing costs

Moment of	B_1		B_2		B_3		B_4		B_5		B_6		B_7		B_8	
intervention	T	R	T	R	T	R	T	R	T	R	T	R	T	R	T	R
Before	4	4	10	3	7	2	5	5	7	0	2	6	5	4	1	7
With	5	4	10	1	6	1	6	6	10	9	4	6	4	5	0	5
After	5	4	9	2	8	6	5	3	10	7	4	4	3	7	2	4

With these target buffer levels 3000 replications with run-length of 300 shifts have been performed. The obtained results regarding manufacturing cost per unit (referred to as “cost per unit” or “cost” in the following) and output are given in Table 4.12. The last line of the table, “savings” indicates the savings per unit in comparison to not intervening at all (referred to as “without strategy” or “none - none” in the following). The average manufacturing cost per unit and output for “without strategy” is listed in Table 4.13. Apart from the tables, the results are visualized in Figure 4.8, p. 81, including “without strategy”.

Table 4.12 Results obtained with different intervention strategies (average values; cost per unit in [€])

	Before			After			With		
	Min	Tol	Exact	Min	Tol	Exact	Min	Tol	Exact
Cost per unit	285.76	285.55	285.50	285.82	285.66	285.66	285.65	285.46	285.41
Output	556.1	556.3	555.9	559.5	560.1	561.1	559.0	558.0	558.2
Savings [€]	0.36	0.57	0.62	0.30	0.46	0.46	0.47	0.66	0.71

Table 4.13 Manufacturing cost per unit and output achieved “without strategy”

Cost per unit [€]	286.12
Output	558

First the achieved output is discussed. When choosing the moment of intervention before shift the output is less compared to without strategy, as expected (see subsection 3.3.3), because stopping machines means introducing additional downtimes. The moment of intervention after shift end always reaches more output, especially in combination with the exact target level. Stopping with shift end results in slightly higher output than “without strategy”. Regarding cost, it can be seen that applying any intervention outperforms not intervening at all. The best results are achieved when using the “exact” or “tolerated” fill level, the alternative “minimum” fill level is not recommended. Intervening before shift end results in lower costs than after shift end, when comparing each target fill level to each other. This is due to the fact, that the time all machines within the system operate simultaneously is shorter, resulting in lower cost compared to intervening after shift end.

Yet all in all, the moment of intervention “with” shift end outperforms all other moments when regarding costs. As stated in subsection 3.3.3, where the moments of intervention are explained, the moment of intervention “with” shift end has an additional constraint comparing to “before” and “after”: the physically last machine is obliged to stop at shift end. This leads to a different behavior regarding machine operation, as can be seen when viewing Figure 4.9, and Figure 4.10, both on p. 82. In these the differences of the average operation time extension or reduction (going on operating after shift end or stopping before shift end) for each machine are depicted. For exact results see appendix A, Table A.1 and Table A.2. In Figure 4.9, showing the average operation time extension, behavior of “with” and “after” can be compared. The machines M_1 to M_4 go on operating after shift end, but not as extensively as in case of the moment “after” shift end. At the same time, the latter machines M_5 to M_9 nearly stop at shift end, while “after” shift end goes on producing one or two units (for “tol” and “exact”; with the cycle time of about 55 seconds). In Figure 4.10, depicting average operation time reduction the results for “before” and “with” can be seen. Here the average operation time reduction of each machine is less for the moment “with” compared to “before” and at the same time the output is increased (see Figure 4.8). “With” outperforms before, due to the already stated opposed results: slightly increased costs and higher output for “with” shift end.

Figure 4.11 and Figure 4.12 (both p. 83) show the maximum operation time extension and reduction for each machine. The maximum limit of time extension (30 minutes) has only been reached once. For exact results see appendix A, Table A.3 and Table A.4.

To see how exactly the method of adaptive buffer operation interferes with the operation of systems the system-induced downtimes have to be regarded (Figure 4.13 on p. 84; exact results: appendix A, Table A.5). For each machine the system-induced downtimes per shift are depicted. It can be seen that for the machines M_3 to M_8 the system-induced downtimes are below the results obtained without interfering. Additionally, Figure 4.14 (p. 84) shows resulting starved times and Figure 4.15 (p. 85) shows the resulting blocked times. Exact results are listed in Table A.6 and Table A.7. Here, it can be seen that the reduction of system-induced downtimes results from the reduction of starved times. Please note, that the model depicted in the simulation is unsaturated. Due to this, in Figure 4.15 the blocked times of machine M_9 have the similar values for all strategies, as the downtimes of the sink are not influenced. This also applies to the starved times of machine M1 in Figure 4.14. The downtimes of the source are not influenced and not decoupled by any buffer which underlies adaptive buffer operation.

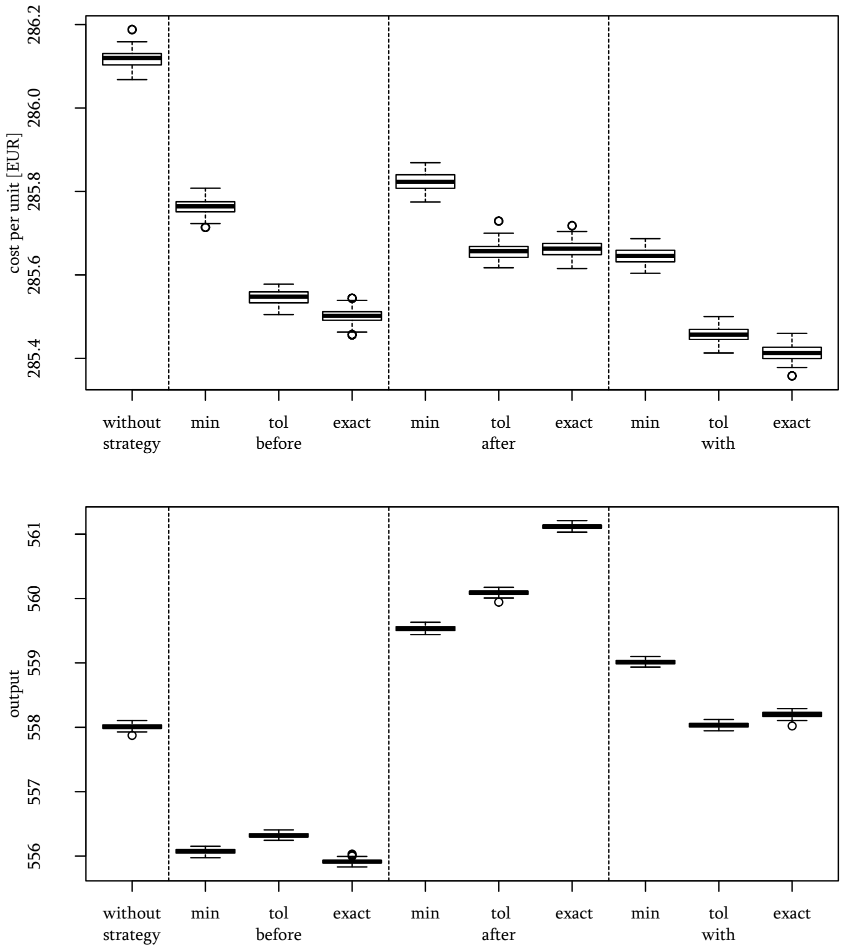


Figure 4.8 Results comparing combinations of moments of intervention and target fill levels

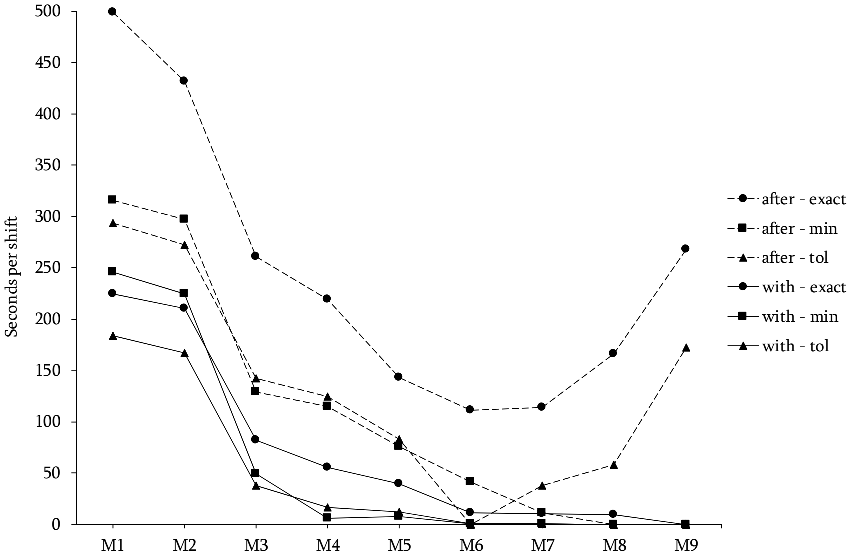


Figure 4.9 Average operation time extension for "with" and "after" (for results see Table A.1)[219]

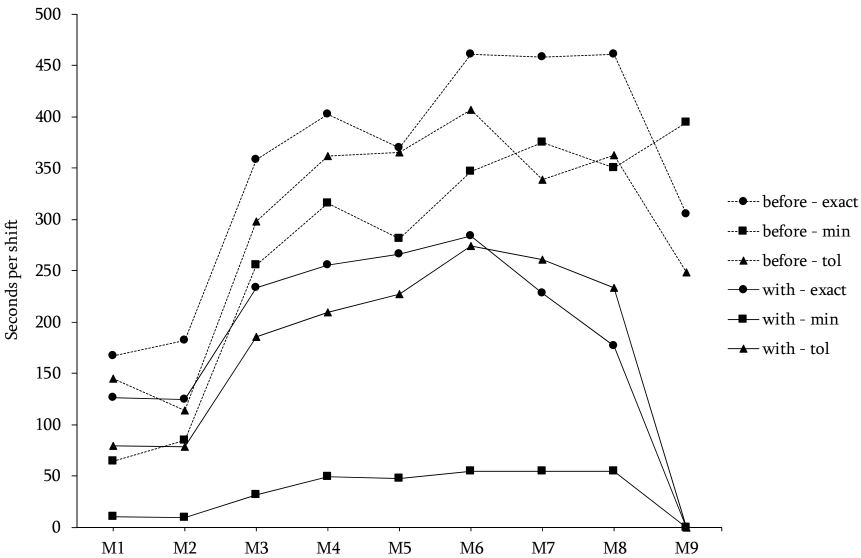


Figure 4.10 Average operation time reduction for "before" and "with" (for results see Table A.2)

[219] Note, here and in the following: For better visualization line diagrams are used instead of bar diagrams.

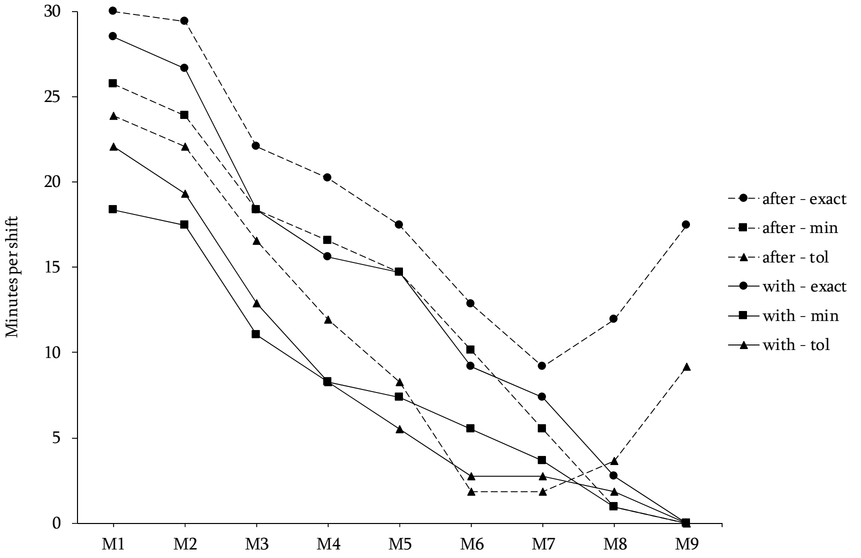


Figure 4.11 Maximum operation time extension (for results see Table A.3)

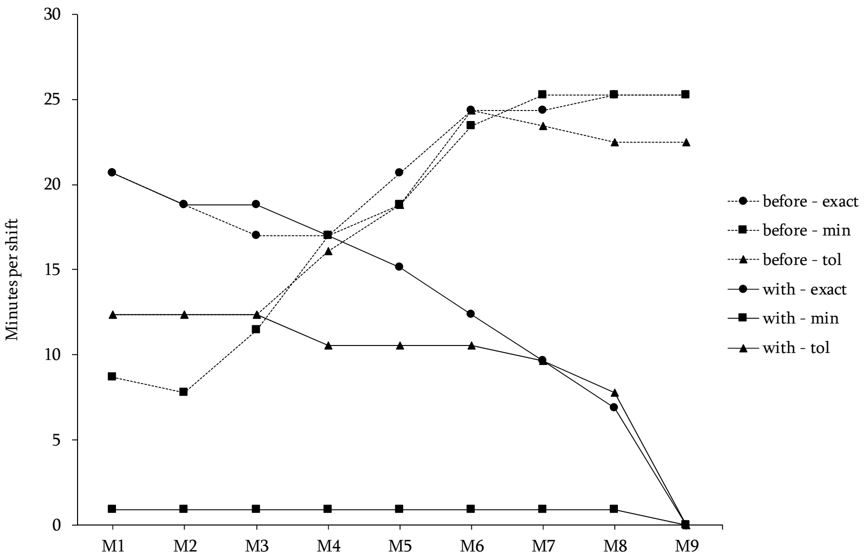


Figure 4.12 Maximum operation time reduction (for results see Table A.4)

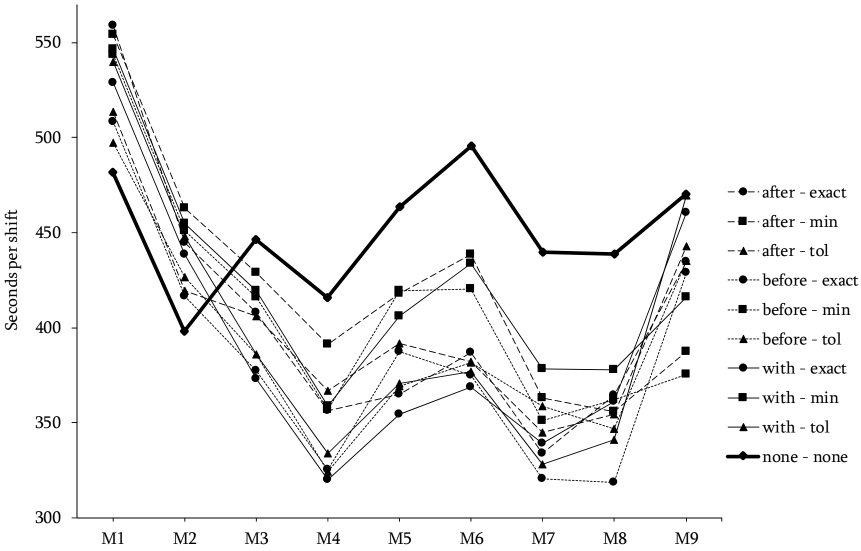


Figure 4.13 Comparison of system-induced downtimes (for results see Table A.5)

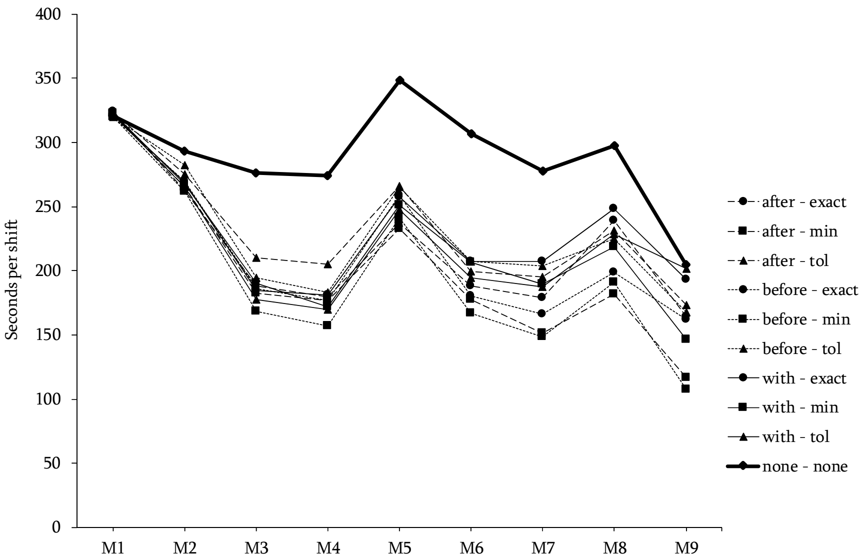


Figure 4.14 Comparison of starved times (for results see Table A.6)

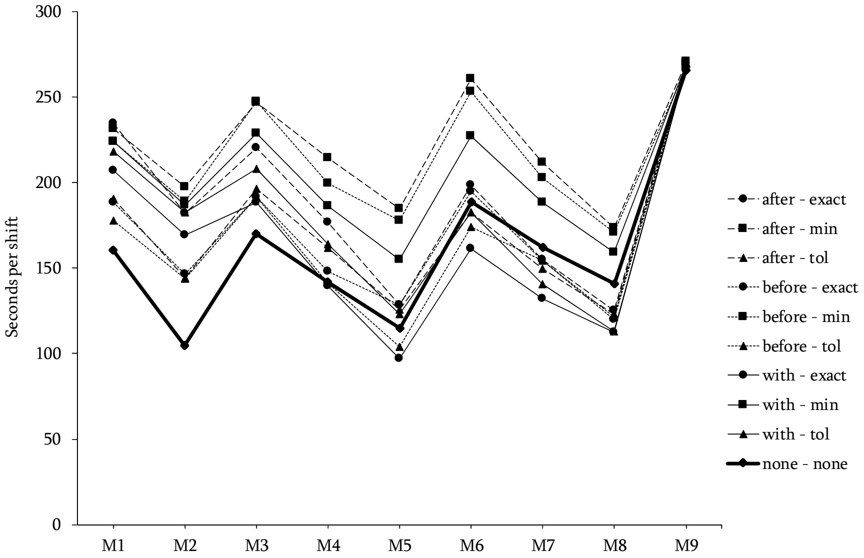


Figure 4.15 Comparison of blocked times (for results see Table A.7)

4.2.2 Single-objective optimization: maximize output

Single-objective optimization with the goal of maximizing output was performed with the moment of intervention "before" and all target fill levels. Adaptive buffer operation was performed every second shift during late shift. Monitoring to initiate adaptive buffer operation started 30 minutes before shift end, as for the objective of minimizing manufacturing cost per unit. Table 4.14 shows the resulting target fill levels for exact, minimum and tolerated and the achieved output and cost, which are compared to the obtained output of 558 units and 286.12 € manufacturing cost per unit when not interfering (Table 4.13).

Table 4.14 Target fill levels for maximizing output (moment of intervention: before)

Target fill	B_1		B_2		B_3		B_4		B_5		B_6		B_7		B_8			
level	T	R	T	R	T	R	T	R	T	R	T	R	T	R	T	R	cost [€]	units
Exact	3	-	6	-	5	-	4	-	8	-	5	-	6	-	6	-	285.72	556.27
Minimum	0	-	3	-	2	-	1	-	3	-	3	-	3	-	2	-	286.09	557.96
Tolerated	3	5	7	10	6	8	4	6	7	8	7	7	8	7	6	9	286.12	557.94

For the exact fill level, the output is below "without strategy" and at lower costs. When intervening in the operation of the system, and additionally introducing artificial downtimes, an increased output cannot be achieved, as expected. This is confirmed when regarding the results of the minimum and tolerated target fill level. The achieved output and cost is approximately the same as "without strategy". The therewith associated target buffer levels nearly allow any target buffer fill level combination, and thus do not require any intervention, as can be seen in Figure 4.16.

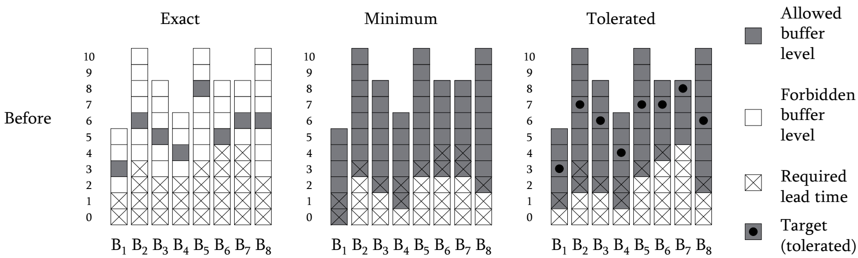


Figure 4.16 Feasible and unfeasible buffer levels for single-objective maximize output for exact, minimum and tolerated

The forbidden buffer levels are very low levels, e.g. for buffer B_5 for both minimum and tolerated at least 3 units need to be in the buffer. For the downstream machine to remain operative, the buffer needs to be filled to a level considering the buffer lead time. If the lead time is 190 seconds, as for buffer B_5, the fill level required for machine M_6 to remain operative without any interruptions is 3 units. This amount corresponds the result of the target fill levels for the tolerated and minimum fill level. This required fill level for each buffer is indicated in Figure 4.16 as "required lead time". Now the slight deviation of output and cost for minimum and tolerated comparing to "without strategy" is due to the fact that with these target buffer levels, sometimes the operation is stopped artificially, as the optimization algorithm did not find the global optimum.

4.2.3 Multi-objective optimization

The first striking result obtained from the three bi-objective optimization runs is, that the objectives minimize cost per unit and minimize system-induced downtimes are not conflicting objectives, as the fitness converges to one point. This shows, that the idea of focusing on system-induced downtimes and trying to minimize those, automatically results in minimized manufacturing costs. It reinforces the application of the proposed method. Thus the number of objectives is reduced to two: maximizing output and minimizing manufacturing cost. Figure 4.17 shows the Pareto-fronts for the nine different combinations of moments of intervention and target buffer fill level. The exact results can be viewed in appendix B from Table B.1 to Table B.9. Additionally, the output and cost achieved without applying any strategy is included.

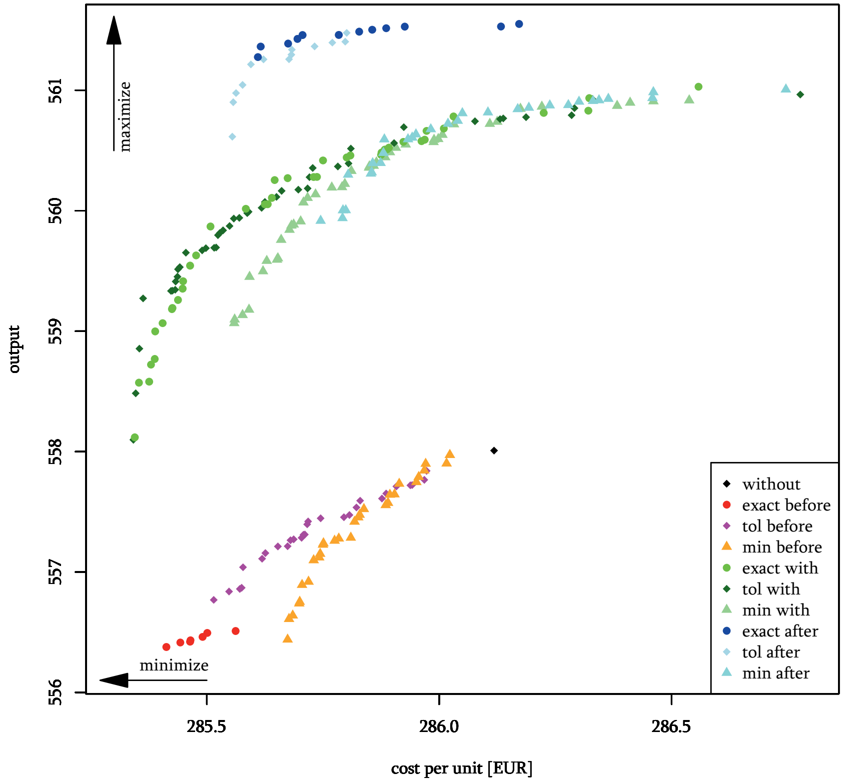


Figure 4.17 Pareto-fronts resulting from multi-objective optimization (maximize output and minimize cost; for results see Table B.1 to Table B.9)

It can be observed, that no additional output can be gained when intervening “before” shift end. This has already been confirmed in subsection 4.2.2 when discussing the results of the single-objective optimization for output maximization. When applying the intervention “with” or “after” shift end this output exceeds the output “without” intervention although at lower manufacturing cost.

Depending on the circumstances and requirements for a manufacturer any of those strategies may be chosen, yet it is recommended to use and adapt the strategy with the lowest manufacturing cost per unit: the moment of intervention “with” in combination with the buffer target fill level “exact” or “tolerated”. As discussed in subsection 4.2.1, the moment of intervention “with” outperforms “before” and “after” regarding manufacturing cost per unit. The target fill levels “exact” and “tolerated” achieve similar cost savings, which is why both are recommended.

Application of the moment of intervention “with” can be adapted to the manufacturer’s requirements. Currently “with” stops some machines before shift end and lets other machines operate after shift end, with the constraint, that the physically last machine stops at end of shift (see p. 48, subsection 3.3.3). The experimentally obtained output is the same as without strategy (see Table 4.12, “with – exact” or “with – tol”). If more output is required, it is proposed to extend shift length or to add additional shifts if possible. Choosing “after” instead of “with”, leads to an increased output, yet, at increased costs. For the contrary case, the need of less output, shortening shift length or cancelation of shifts is a solution. Another obstacle to implement adaptive buffer operation using “with” is the need of operation after shift end. If it is not possible to extend operation time after shift end, stopping the machines can be initiated earlier, so that the last machine to stop, stops at shift end (the physically last machine stops before shift end). Figure 4.18 depicts this adjustment compared to the unaltered process. The moments for initiation of stopping the machines and for initiation of monitoring of the system is pulled forward, yet the underlying calculation logic remains the same. It has to be noted, that this has the effect of reduced output.

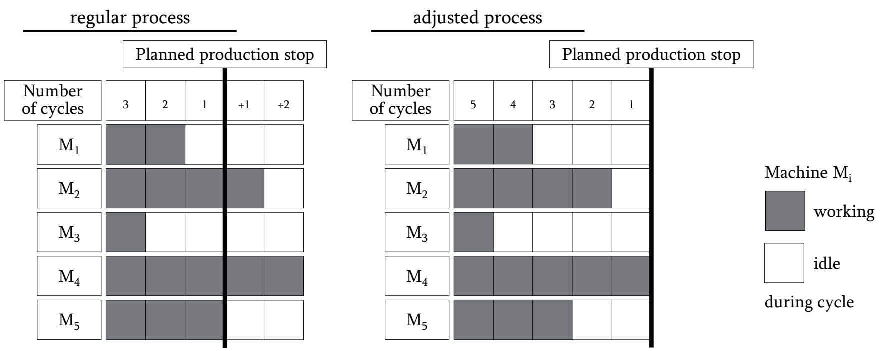


Figure 4.18 Adjustment to use the moment of intervention "with" without extension of operating time after planned production stop

4.2.4 Further experimentation

To obtain broader knowledge of the proposed method's performance further experiments have been undertaken. In this subsection the results from the experiments undertaken with the variable intervention parameters presented in subsection 3.3.4 are presented and discussed. The experiments for varying duration of intervention and frequency are performed for the moment of intervention "before" shift end and the target fill level "exact".

Variation of duration of intervention

As explained in subsection 3.3.4, the duration of intervention, in other words, the time span needed for the process of filling the buffers can be varied. The experiments presented in the preceding subsections have a maximum duration of intervention of thirty minutes. This is the time monitoring of the current system status is initiated before shift end for the moments of intervention "before" and "with". For the moment "after", the duration of intervention was limited to thirty minutes, too.

Here, for the moment of intervention "before", to vary the duration of the intervention, the time system monitoring is initiated before shift end was altered. When monitoring of the system is started, the machines can be stopped to achieve the target fill levels of the buffers. The moment monitoring is initiated can be altered. Changing this moment results in a variation of duration of intervention. The time of monitoring is reduced stepwise by five minutes, starting with the initial setting of 30 minutes. An additional duration of intervention of 3 minutes is included. Extending the duration of intervention above 30 minutes is not necessary, as this time has never been required, as seen in the previous optimization runs in subsection 4.2.1, displayed in Figure 4.12, showing the maximum operation time reduction for each machine. For the different durations of intervention single-objective optimization runs with the goal of minimizing cost and finding optimized target fill levels for each buffer have been performed. The resulting target fill levels can be seen in appendix C in Table C.1, or as graphical representation in Figure C.1. When looking at those, it can be noticed that they are similar to each other. With these target fill levels, the fitness and the achieved average output was calculated by replicating the simulation model 3000 times. The obtained cost per unit, output and savings comparing to "without strategy" (286.12 €) are written down in Table 4.15 and are shown in Figure 4.19. Additionally, the resulting cost and output obtained without intervention are included.

Table 4.15 Achieved results through variation of duration of intervention

Duration of intervention	3	5	10	15	20	25	30[220]
Cost per unit [€]	285.89	285.81	285.58	285.50	285.50	285.50	285.50
Output	557.71	557.34	556.22	555.88	555.88	555.77	555.92
Savings [€]	0.23	0.31	0.54	0.62	0.62	0.62	0.62

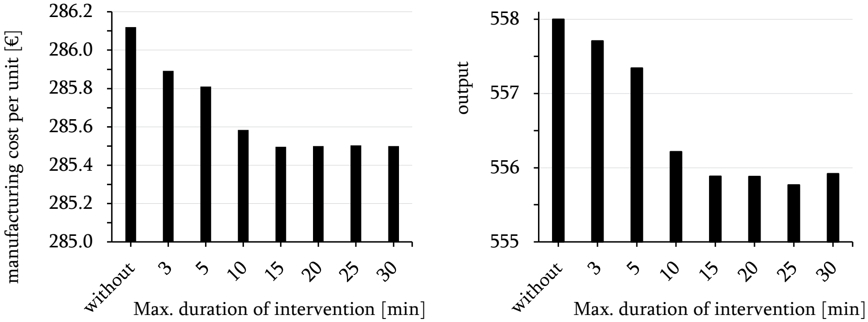


Figure 4.19 Obtained results through variation of duration of intervention

When regarding Figure 4.19 it can be seen that the manufacturing cost per unit is decreasing stepwise while maximum duration of intervention is increased from zero to up to 15 minutes. From 15 to 30 minutes the cost settles around 285.50 €, with slightly varying output. For the presented problem it is recommended to use a setting of 15 minutes as maximum duration, if maximum savings are to be achieved. Further extension of duration of intervention does not imply further improvement, and implementation is facilitated with shorter intervention times.

Figure 4.20 depicts the average operation time reduction per shift for the different maximum duration of intervention. Exact results are provided in Table C.2. It can be observed, that the average operation time reduction for 10 to 30 minutes is similar, with a slightly lower reduction for 10 minutes. This result for 15 to 30 minutes confirms the above findings, that possible extension times above 15 minutes do not imply further improvements, as they have a similar behavior in general. For a maximum duration of three and five minutes, the average operation time reduction is of course lower.

[220] This maximum duration of intervention corresponds the settings of the single-objective optimization in subsection 4.2.1, obtaining the same target fill levels and results.

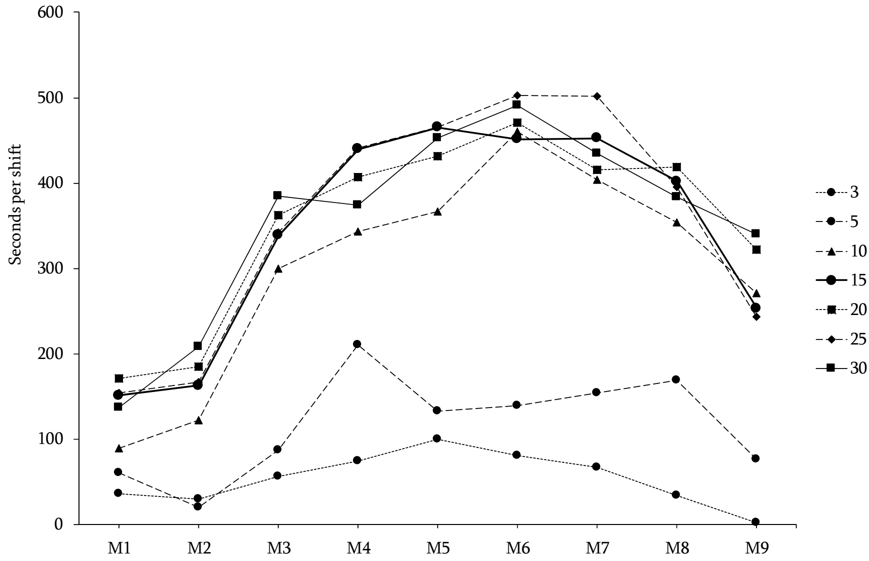


Figure 4.20 Average operation time reduction depending on duration of intervention (for results see Table C.2)

Variation of intervention frequency

An experiment varying intervention frequency was performed with a duration of intervention of 30 minutes. In the optimization run in subsection 4.2.1 the target fill levels are calculated every late shift, in other words every second shift. The frequency was increased to every shift (intervention before early and late shift) and reduced to every forth, sixth, 10th, 12th and 20th shift. These frequencies are chosen, as the period of time of 300 shifts can be divided into parts of equal length and therefore the resulting cost and output are not biased. The target fill levels obtained through the optimization with the goal of minimizing manufacturing costs are listed in Table C.3 and depicted in Figure C.2. These are very similar when compared to each other. 3000 replications of 300 shifts were run and the resulting cost and output for each was included in Table 4.16. Additionally, the savings in comparison to "without strategy" (cost per unit 286.12 €) are included. Frequency 1 corresponds intervening every shift, 2 every second shift, 4 every fourth etc. Figure 4.21 depicts the results in comparison to without intervention.

Table 4.16 Achieved results through variation of frequency of intervention

Frequency	1	2[221]	4	6	10	12	20
Cost per unit [€]	285.08	285.50	285.80	285.91	286.01	286.02	286.07
Output	554.09	555.92	556.91	557.15	557.57	557.52	557.75
Savings [€]	1.04	0.62	0.32	0.21	0.11	0.1	0.05

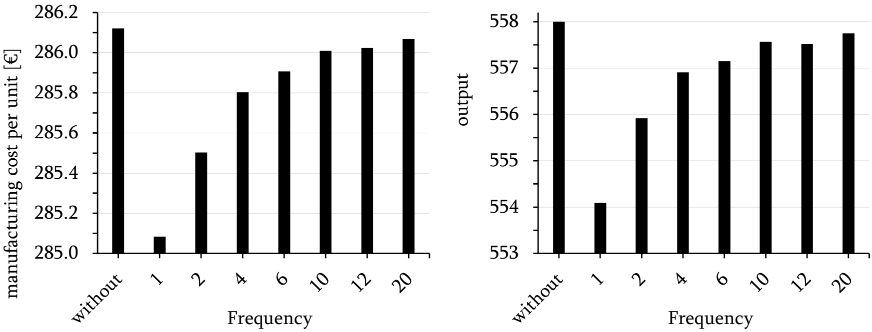


Figure 4.21 Obtained results through variation of frequency of intervention

It can be seen, that reducing intervention frequency leads to increased output and lower savings. Now when comparing every shift to every second and fourth shift, it can be seen, that the costs are further reduced. The reduction, however, is not proportional. When comparing intervention 1 and 2 (every shift to every second shift), savings of cost is not twice as high. This is why it is suggested to intervene every second shift. Here any moment of intervention and target fill level can be chosen without restriction, as there is a time span between end of late shift and beginning of early shift.

Figure 4.22 depicts the average operation time reduction for all possible frequencies of intervention. The underlying data is attached in Table C.4. The average operation time reduction for intervening every shift is the lowest in general. When intervening every second shift, the time reduction is slightly higher, as the system has more time to leave the adapted status, in comparison to intervening once per shift. All other intervention frequencies do not seem to follow any rule, apart from the fact, that they have average operation reduction times which lie higher than when intervening once per shift. This is due to the system

[221] This frequency of intervention corresponds the settings of the single-objective optimization in subsection 4.2.1, obtaining the same target fill levels and results.

returning back to a “chaotic” status quo. Now resetting the buffers within the system to the “optimized” target fill level from this status is similar for each case.

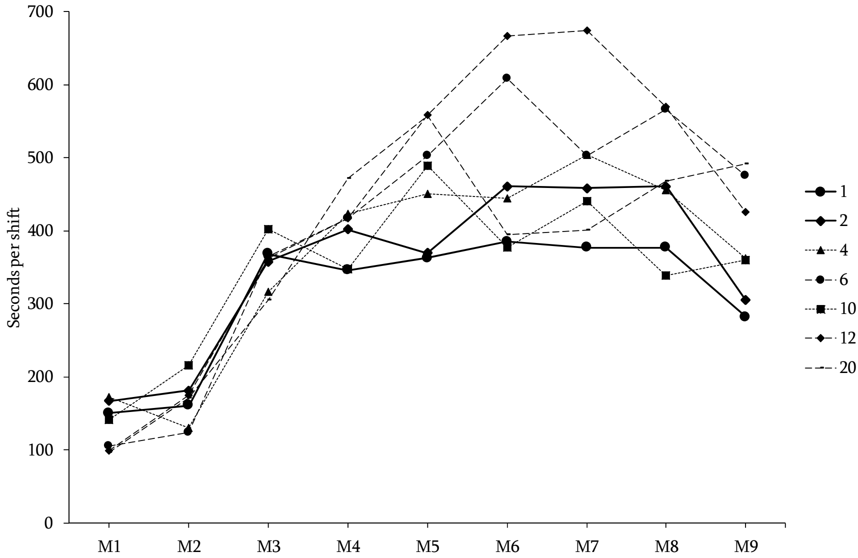


Figure 4.22 Average operation time reduction depending on frequency of intervention (for results see Table C.4)

4.2.5 Conclusion of results and findings

The objective of the performed experiments was to show the performance of adaptive buffer operation and to be able to compare the proposed variations of strategies. All experiments affirm that application of adaptive buffer operation outperforms not doing so, when regarding the indicator manufacturing cost per unit. Additionally, for the moments of intervention “with” and “after” even the output can be increased simultaneously. Furthermore, the experiments confirm that when reducing system-induced downtimes, especially due to starvation, manufacturing costs per unit are reduced also. In the following the findings are concluded and stated explicitly.

Minimizing system-induced downtimes equals minimizing manufacturing costs per unit

Central result of this dissertation is that the objectives minimize cost per unit and minimize system-induced downtimes equal each other as they are not conflicting (stated in subsection

4.2.3). Minimizing system-induced downtimes, which include blocking and starvation automatically results in reduced manufacturing costs per unit. Yet this does not mean that minimizing system-induced downtimes results in increased output.

Starvation has a higher impact on output than blocking

In subsection 4.2.3., the analysis of experimental results of the objective minimize manufacturing costs per unit, it has been stated that system-induced downtimes were reduced and that this was due to a decrease of starvation. Therefore, minimizing starvation equals minimizing manufacturing cost per unit. In contrast to minimizing system-induced downtimes, minimizing starvation, e.g. through adaptive buffer operation, can lead to increased output. This is due to the fact that minimizing starvation results from filling the buffers to a higher level. Blocked times might be increased. Yet minimizing blocked times, achieved through lower fill levels might result in a higher rate of starvation downtimes and less output. This is why, starvation has a higher impact on output than blocking.

Full buffers increase output at increased costs

Filling buffers maximally increases output as starvation downtimes are decreased, yet this does not equal a cost-optimal operating of the manufacturing system, as more idle times through blocking result. In Figure 4.7 (p. 78) depicting cost-optimal buffer target levels, it can be seen that the filling of the buffers is not maximal.

Introduction of additional interventions during operation results in decreased output

As seen in the single-objective optimization run with the goal of maximizing output (subsection 4.2.2, “before – exact”) interventions during operation do not result in increased output, when compared to not intervening. Thus adaptive buffer operation cannot lead to increased output given the same time of operation as without applying it, but to minimized costs of manufacturing per unit. If there is a possibility to extend manufacturing time, e.g. application of “with” is possible, the output can be increased at reduced cost at the same time.

4.3 Concluding assessment and discussion

Within this section a concluding assessment of the solution methodology is given, which is based on the requirements stated in section 2.4. An overview of the assessment of the fulfilment of the system- and application oriented requirements is given in Figure 4.23.

Requirements to the method to increase output in manufacturing systems			
System-oriented requirements		**Application-oriented requirements**	
Applicability to existing systems	●	Scalability	◕
Avoidance of structural changes	●	Consideration of all interactions in the system	●
Transfer lines	●	Equality of treatment of all buffers within the system	●
Make-to-order production of personalized goods	●	Practical feasibility	●
		Consideration of multiple-objectives	●

Requirements …
○ not
◐ partially
◕ primarily
● fully
… fulfilled

Figure 4.23 Assessment of fulfilment of the requirements to the method to increase output in manufacturing systems

This simulation-based optimization method using adaptive buffer operation to increase output in transfer lines has especially been developed for existing systems. Thus *applicability to existing systems* and to *transfer lines* is directly fulfilled. In adaptive buffer operation the fill levels of the buffers are adapted, creating a new and optimized system constellation to improve the decoupling effect buffers have on downtimes. In contrast to the methods solving the buffer allocation problem discussed in subsection 2.5.1 here *avoidance of structural changes* is guaranteed. Furthermore, *make-to-order production of personalized goods* can be dealt with without difficulties. The sequence of units within the system does not have to be modified and the focus is put on all buffers equally in the system, and not just the buffer between production and customer as in case of the hedging point policy (subsection 2.5.1). When regarding application-oriented requirements, *scalability* of the system is primarily and not fully fulfilled, as in this dissertation only one system has been tested and the number of machines within the system has not been varied. As the optimization method is simulation-based *consideration of all interactions in the interconnected system* is ensured. Of course the simulation model depicts a simplified model of reality, but the level of detail is sufficient to state that all necessary resulting interactions are included. In adaptive buffer operation all buffers are regarded equally and none of the buffers is subject to special focus, so the requirement *equality of treatment of all buffers within the system* is fully fulfilled. *Practical feasibility* is given, as the method adaptive buffer operation is simple, understandable and applicable to the dealt with real-world problem. In the proposed simulation-based optimization method algorithms *considering multiple-objectives* are included. This additionally increases practical feasibility, as most real-world problems are not only restricted to a single-objective, but many different possibilities have to be weighed.

5 Summary and outlook

To remain successful in global competition, production sites of automotive industry in high-wage countries as Germany need to increase productivity. Due to their far-reaching history most manufacturing systems already exist, which aggravates adaptation of the system to reach an increase of output and at the same time a decrease of cost. Additionally, these systems are often highly sophisticated, so implementation of further lean methods or of more elaborate maintenance strategies scarcely improves the system behavior. This is why current, already existing approaches to reach both goals focus on enhancing the interplay of elements within the system through optimizing the decoupling effect of buffers. Yet these approaches often require structural changes of the system, considering existing manufacturing systems only insufficiently or do not treat all elements equally.

Thus, in this dissertation a method to increase output in transfer lines was developed, which closes these research gaps. Adaptive buffer operation, a method to improve the decoupling effects of buffers, and a simulation-based optimization method to find optimized parameter settings for adaptive buffer operation was developed. In adaptive buffer operation the structure of the manufacturing system is not changed. The method is based on changing the configuration of the system through altering the fill levels of the buffers within the system at the end of given periods, e.g. at the end of shift. For this the machines within the system are stopped at different moments before ceasing production, which can be viewed as artificially introducing downtimes. Doing this, an improved starting configuration for the next shift is prepared, resulting in diminished propagation of technical failures through the system. To find out to which fill level each buffer within the system should be filled simulation-based optimization was applied. As this optimization method uses a simulation model depicting the manufacturing system, the latter is regarded holistically, and does not focus on one special buffer but on all in the same intensity. Additionally, all interactions resulting from the interventions and the behavior of the system are included. Furthermore, with the named possibilities to modify the method through varying the moment of intervention, duration or frequency, the proposed method can meet with different situation-dependent demands. It has been shown that the proposed method is capable of increasing output and at the same time lowering manufacturing costs without structurally intervening. Moreover, a direct connection between downtimes resulting from blocking or starvation and manufacturing costs has been substantiated. Pursuing reduction of costs automatically results in reduction of blocking and starvation.

The main deficit of the proposed method is that it might be conflicting with the existing organizational structure. Adaptive buffer time operation requires extending or reducing shift length for some employees. This results in working time fluctuation, which usually can be compensated for by the employee's working time account. Yet if specific machines have to extend shift length (or reduce shift length) repeatedly, the employee's working time account cannot be balanced as easily anymore. For this two solutions are proposed: operator rotation and adaption of cycle times of the affected machine. Operator rotation requires the operator to be qualified to work at different machines and stations. This is the solution's main disadvantage, as sometimes the scope of work is too big to be able to qualify the operator. Now adaptation of cycle times of the machines is possible, if the work packages at a station can be modified or split up. As example, within a manufacturing system one machine's shift length was extended by five minutes per shift on average. If these five minutes of operation can be included at other machines through cycle time adaptation, no more extension of shift length should be necessary at the affected machine in future. This has the subsidiary effect of an improved operating of the entire system.

Additionally, the shuttle busses between production site and home provided for the employees by most German automotive manufacturers are in conflict with flexible working times. Those leave at fixed times, resulting in less time to get to the shuttle bus or in more time waiting until the shuttle leaves the production site and would lead to unequal treatment of employees.

The proposed method can be extended in further research work. The manufacturing system investigated here has a limited system size and high availabilities. To increase the scope of manufacturing systems the proposed method can handle, the influence of system size and overall system availability has to be tested in more detail. An additional extension covering further manufacturing systems is to include preassemblies and their buffers when joining the main transfer line. In this case, downtimes resulting from lack of parts can be influenced directly, whereas in the current solution, they are taken into account in the technical downtimes only. Hence, the behavior of the system underlying adaptive buffer operation can be studied in further depth.

Another shortcoming of this thesis is that the developed method has solely been tested on a simulation model. The model used here can be afflicted with errors, resulting from simplification of the model or data processing. Therefore, implementation of the method into a real manufacturing system is necessary. This includes an integration to the IT system used for monitoring the current production status. With real-time simulation, the optimized parameters for adaptive buffer operation can be determined not only statically, as the here investigated method does, but dynamically, when needed and depending on the current system status. Due to the chosen optimization algorithms, the Evolution Strategies, computation time is fast enough for real-time application. Furthermore, an artificial neuronal network system could be attached, so that the parameters decided on are based on experience and the obtained knowledge.

In conclusion, a completely different method to increase output in transfer lines has been developed. Apart from being applicable to existing systems, as no changes to the structure are required, the system is viewed as a whole. Solutions regarding multi-objectives can be generated, making the method applicable to real-world problems and facilitating decision-making, as several options to solve the problem are given.

A Results single-objective optimization

Table A.1 Average operation time extension per shift for each machine in [s]

	None	Before			After			With		
	None	Min	Tol	Exact	Min	Tol	Exact	Min	Tol	Exact
M_1	0	0	0	0	316	294	499	246	184	225
M_2	0	0	0	0	297	272	432	225	167	211
M_3	0	0	0	0	129	142	261	49	38	82
M_4	0	0	0	0	115	125	219	6	17	56
M_5	0	0	0	0	76	83	143	8	12	40
M_6	0	0	0	0	42	0	112	0	0	11
M_7	0	0	0	0	12	37	114	0	1	11
M_8	0	0	0	0	0	58	167	0	0	10
M_9	0	0	0	0	0	173	268	0	0	0

Table A.2 Average operation time reduction per shift for each machine in [s]

	None	Before			After			With		
	None	Min	Tol	Exact	Min	Tol	Exact	Min	Tol	Exact
M_1	0	65	145	168	0	0	0	10	79	126
M_2	0	84	114	182	0	0	0	9	79	125
M_3	0	256	298	358	0	0	0	32	186	234
M_4	0	315	362	402	0	0	0	49	210	256
M_5	0	281	365	370	0	0	0	48	227	266
M_6	0	346	407	461	0	0	0	54	274	284
M_7	0	375	339	459	0	0	0	54	261	229
M_8	0	351	362	461	0	0	0	54	233	177
M_9	0	395	248	306	0	0	0	0	0	0

Table A.3 Maximum operation time extension per shift for each machine in [min]

	None	Before			After			With		
	None	Min	Tol	Exact	Min	Tol	Exact	Min	Tol	Exact
M_1	0	0	0	0	26	24	30	18	22	21
M_2	0	0	0	0	24	22	29	17	19	19
M_3	0	0	0	0	18	17	22	11	13	19
M_4	0	0	0	0	17	12	20	8	8	17
M_5	0	0	0	0	15	8	17	7	6	15
M_6	0	0	0	0	10	2	13	6	3	12
M_7	0	0	0	0	6	2	9	4	3	10
M_8	0	0	0	0	1	4	12	1	2	7
M_9	0	0	0	0	0	9	17	0	0	0

Table A.4 Maximum operation time reduction per shift for each machine in [min]

	None	Before			After			With		
	None	Min	Tol	Exact	Min	Tol	Exact	Min	Tol	Exact
M_1	0	9	12	21	0	0	0	1	12	21
M_2	0	8	12	19	0	0	0	1	12	19
M_3	0	11	12	17	0	0	0	1	12	19
M_4	0	17	16	17	0	0	0	1	11	17
M_5	0	19	19	21	0	0	0	1	11	15
M_6	0	23	24	24	0	0	0	1	11	12
M_7	0	25	23	24	0	0	0	1	10	10
M_8	0	25	22	25	0	0	0	1	8	7
M_9	0	25	22	25	0	0	0	0	0	0

Table A.5 Average system-induced downtimes per shift for each machine in [s]

	None	Before			After			With		
	None	Min	Tol	Exact	Min	Tol	Exact	Min	Tol	Exact
M_1	482	544	497	502	554	514	559	547	540	519
M_2	398	451	426	411	463	419	445	445	448	438
M_3	446	416	386	397	429	406	408	419	386	373
M_4	416	357	324	317	391	367	356	359	334	320
M_5	463	419	369	351	418	392	365	406	371	354
M_6	495	421	382	363	438	382	387	434	377	369
M_7	440	351	359	335	363	345	334	378	328	339
M_8	439	362	347	359	356	354	364	378	341	361
M_9	470	375	435	438	387	443	435	416	470	461

Table A.6 Average starved times per shift for each machine in [s]

	None	Before			After			With		
	None	Min	Tol	Exact	Min	Tol	Exact	Min	Tol	Exact
M_1	321	320	320	320	323	323	324	323	321	322
M_2	293	262	283	272	265	275	263	268	266	269
M_3	277	169	195	194	182	210	188	190	178	185
M_4	274	157	183	174	177	205	180	172	170	181
M_5	349	241	266	251	233	266	237	251	247	257
M_6	307	167	207	198	178	200	188	206	194	207
M_7	278	148	204	198	151	195	179	190	187	207
M_8	298	191	225	241	182	231	239	219	228	249
M_9	205	107	168	172	117	173	164	146	202	193

Table A.7 Average blocked times per shift for each machine in [s]

	None	Before			After			With		
	None	Min	Tol	Exact	Min	Tol	Exact	Min	Tol	Exact
M_1	161	224	178	182	231	190	235	224	218	207
M_2	105	189	144	138	197	144	182	187	182	170
M_3	170	247	191	204	247	196	220	229	208	188
M_4	142	200	141	143	215	162	177	186	164	139
M_5	115	178	104	101	185	126	128	155	123	97
M_6	189	253	174	165	261	182	198	227	183	161
M_7	162	203	155	136	212	149	155	189	141	132
M_8	141	171	122	117	174	123	125	159	113	112
M_9	266	268	268	267	271	269	271	269	268	268

B Results multi-objective optimization

Table B.1 "min-before": target fill levels, cost per unit in [€] and output for each solution of the Pareto-optimal set

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	Cost	Units
1	4	8	6	3	5	5	6	3	285.67	556.44
2	3	8	6	3	5	5	4	5	285.68	556.61
3	4	8	4	3	6	3	5	4	285.69	556.64
4	3	8	5	3	4	5	3	5	285.70	556.74
5	4	7	4	3	7	4	6	3	285.70	556.75
6	2	8	5	3	3	5	5	4	285.71	556.89
7	3	7	5	3	5	5	5	2	285.72	556.92
8	3	7	4	2	5	4	6	4	285.73	557.10
9	3	6	5	3	4	6	5	4	285.74	557.12
10	3	6	5	3	4	5	5	1	285.74	557.15
11	3	6	5	2	4	4	5	2	285.75	557.23
12	2	7	4	3	4	3	4	3	285.75	557.24
13	3	6	4	3	2	4	4	3	285.78	557.26
14	2	6	4	3	6	4	4	2	285.78	557.28
15	3	5	5	3	3	4	1	0	285.81	557.29
16	2	6	4	2	3	5	1	2	285.82	557.42
17	3	5	4	2	4	3	4	2	285.83	557.45
18	3	5	3	3	4	3	4	3	285.83	557.48
19	2	5	4	3	4	4	4	2	285.84	557.52
20	2	4	5	2	3	4	2	1	285.89	557.56
21	2	4	5	2	3	3	4	2	285.89	557.57
22	2	5	3	1	4	4	2	1	285.89	557.64
23	2	5	3	1	3	3	2	1	285.90	557.65
24	2	4	2	3	2	4	3	1	285.91	557.73
25	2	4	3	2	2	4	2	0	285.95	557.75
26	2	4	2	1	1	1	1	2	285.96	557.79
27	2	3	3	2	2	3	3	0	285.97	557.84
28	2	0	2	2	3	2	3	1	285.97	557.90
29	0	3	3	2	3	4	4	1	286.02	557.90
30	1	1	2	1	2	3	3	1	286.02	557.97

Table B.2 "min-after": target fill levels, cost per unit in [€] and output for each solution of the Pareto-optimal set

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	Cost	Units
1	3	9	5	4	6	5	6	5	285.75	559.92
2	4	8	6	4	7	4	4	7	285.79	559.94
3	3	7	7	5	6	5	5	6	285.79	560.00
4	4	10	5	4	6	6	5	6	285.80	560.01
5	3	9	6	5	5	5	6	8	285.80	560.30
6	1	10	6	3	7	7	7	6	285.85	560.31
7	5	7	6	5	5	6	6	8	285.86	560.32
8	4	8	5	5	9	4	6	7	285.86	560.40
9	2	10	4	5	6	6	7	8	285.87	560.40
10	3	9	6	4	7	5	7	8	285.88	560.48
11	3	8	6	4	7	6	6	9	285.88	560.59
12	3	8	7	4	7	6	6	9	285.93	560.59
13	4	10	4	3	7	6	8	8	285.94	560.61
14	4	8	7	5	5	6	7	9	285.95	560.64
15	4	8	4	5	7	5	7	10	285.98	560.68
16	3	8	5	5	7	7	7	9	286.02	560.72
17	5	9	5	5	6	6	7	10	286.04	560.75
18	1	9	6	4	8	7	7	9	286.05	560.81
19	4	10	6	4	7	5	8	10	286.11	560.82
20	5	8	8	4	8	5	7	10	286.17	560.84
21	4	9	5	5	8	6	8	9	286.19	560.86
22	4	8	7	5	8	6	7	10	286.24	560.88
23	3	10	7	5	7	7	7	10	286.28	560.88
24	3	9	4	5	9	6	8	10	286.30	560.91
25	3	9	7	5	8	7	7	10	286.33	560.91
26	3	8	7	4	9	6	8	10	286.34	560.92
27	5	8	8	4	9	5	8	10	286.36	560.93
28	4	10	7	5	9	5	8	10	286.46	560.94
29	5	8	8	5	8	7	8	9	286.46	560.99
30	5	9	5	5	10	7	8	10	286.75	561.01

Table B.3 "min-with": target fill levels, cost per unit in [€] and output for each solution of the Pareto-optimal set

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	Cost	Units
1	3	8	4	2	3	5	5	2	285.56	559.07
2	3	9	4	1	3	4	4	3	285.56	559.10
3	2	9	4	2	2	6	5	3	285.58	559.13
4	3	9	4	3	2	5	5	2	285.59	559.18
5	3	10	6	2	4	6	5	1	285.59	559.45
6	3	9	5	3	5	6	4	4	285.62	559.50
7	4	8	6	2	5	6	6	3	285.63	559.58
8	3	10	4	3	7	5	2	6	285.65	559.59
9	4	9	5	5	4	5	5	4	285.65	559.61
10	3	9	5	4	5	6	6	4	285.66	559.76
11	3	8	6	4	6	6	5	5	285.68	559.84
12	3	9	6	5	7	5	4	4	285.68	559.88
13	3	9	4	4	6	5	5	7	285.69	559.89
14	3	10	6	4	6	6	6	4	285.70	559.91
15	3	10	6	4	7	5	6	5	285.71	560.07
16	4	8	6	4	6	7	6	5	285.72	560.11
17	4	9	4	4	7	6	4	8	285.73	560.14
18	4	10	6	5	5	7	5	6	285.77	560.19
19	5	9	5	5	6	6	4	8	285.79	560.20
20	3	10	4	4	6	7	7	6	285.80	560.22
21	3	9	6	3	7	8	7	5	285.81	560.33
22	4	8	6	4	7	6	6	8	285.85	560.36
23	4	8	5	3	7	6	7	8	285.85	560.37
24	4	9	6	4	7	6	6	8	285.86	560.37
25	4	9	6	5	7	5	6	8	285.87	560.40
26	5	9	5	3	7	6	7	8	285.88	560.45
27	4	9	4	6	7	6	5	9	285.89	560.49
28	4	9	6	4	8	5	6	9	285.91	560.52
29	4	9	6	4	8	4	6	10	285.93	560.55
30	4	8	6	4	8	6	7	8	285.99	560.57
31	4	10	6	5	7	7	6	8	285.99	560.59
32	4	10	4	3	6	8	7	9	286.00	560.60
33	5	9	6	4	7	6	7	9	286.01	560.63
34	4	10	6	5	7	7	7	8	286.03	560.72
35	4	9	6	6	7	6	7	9	286.11	560.72
36	4	8	6	5	8	5	7	10	286.13	560.74
37	4	9	5	3	7	8	7	10	286.18	560.85
38	4	10	6	4	6	7	8	10	286.22	560.87
39	5	8	7	5	8	7	7	10	286.38	560.87
40	4	10	3	6	8	7	8	10	286.41	560.90
41	4	9	5	5	7	8	8	10	286.46	560.91
42	5	9	5	5	7	8	8	10	286.54	560.92

Table B.4 "exact-before": target fill levels, cost per unit in [€] and output for each solution of the Pareto-optimal set

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	Cost	Units
1	3	8	6	4	7	6	5	4	285.41	556.38
2	3	8	6	4	8	6	6	5	285.44	556.41
3	3	8	6	3	8	6	6	5	285.46	556.42
4	2	8	6	4	7	6	5	6	285.47	556.43
5	3	7	6	4	7	6	5	6	285.49	556.46
6	3	7	6	4	6	7	5	6	285.50	556.49
7	2	6	6	4	8	6	6	6	285.56	556.51

Table B.5 "exact-after": target fill levels. cost per unit in [€] and output for each solution of the Pareto-optimal set

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	Cost	Units
1	5	8	5	5	8	6	4	5	285.61	561.28
2	5	10	5	5	8	5	5	4	285.62	561.36
3	5	8	5	6	9	6	5	2	285.68	561.39
4	5	9	6	6	8	5	3	4	285.70	561.43
5	5	9	5	6	9	5	3	4	285.71	561.46
6	4	9	7	5	9	3	3	6	285.78	561.46
7	3	10	8	6	8	5	2	7	285.83	561.49
8	4	9	6	6	10	5	2	5	285.86	561.50
9	4	10	7	6	9	3	3	6	285.89	561.52
10	3	9	8	6	9	3	2	7	285.93	561.53
11	4	10	8	6	9	2	3	6	286.13	561.53
12	4	10	8	6	10	3	2	6	286.17	561.55

Table B.6 "exact-with": target fill levels, cost per unit in [€] and output for each solution of the Pareto-optimal set

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	Cost	Units
1	3	9	7	3	7	6	4	3	285.35	558.12
2	4	9	6	4	7	5	7	2	285.35	558.57
3	4	10	6	5	8	5	5	3	285.38	558.58
4	3	9	6	3	9	6	5	3	285.38	558.72
5	4	8	6	5	8	6	6	2	285.39	558.77
6	3	9	5	4	8	6	5	4	285.39	559.00
7	4	10	6	4	8	6	6	3	285.41	559.07
8	3	9	7	3	8	4	6	5	285.43	559.18
9	3	9	6	5	8	5	6	4	285.43	559.19
10	3	9	7	4	7	6	6	4	285.44	559.26
11	4	9	7	5	7	6	6	4	285.45	559.35
12	3	9	6	5	8	6	6	4	285.45	559.42
13	4	9	6	5	7	7	6	4	285.46	559.54
14	4	9	6	5	7	5	8	4	285.48	559.63
15	3	9	7	4	7	6	6	6	285.51	559.87
16	4	9	6	5	6	6	6	7	285.58	560.02
17	4	10	5	4	9	6	6	6	285.63	560.05
18	4	9	6	4	8	6	7	6	285.63	560.05
19	4	9	6	4	8	6	5	8	285.64	560.11
20	3	8	6	3	8	6	6	8	285.65	560.26
21	4	7	6	5	8	5	6	8	285.67	560.27
22	4	9	7	4	7	6	6	8	285.73	560.28
23	4	9	5	5	8	7	6	7	285.74	560.28
24	4	8	6	5	7	6	7	8	285.75	560.42
25	4	10	7	5	8	4	6	9	285.80	560.44
26	3	8	6	4	8	6	7	8	285.81	560.46
27	4	10	5	5	9	6	6	8	285.88	560.46
28	4	9	7	5	7	7	5	9	285.88	560.48
29	3	9	6	5	8	6	6	9	285.88	560.51
30	2	9	7	4	8	5	7	9	285.89	560.51
31	4	10	7	5	8	5	7	8	285.89	560.52
32	3	10	6	5	7	6	7	9	285.92	560.57
33	4	8	7	5	8	5	7	9	285.96	560.58
34	4	10	6	5	6	7	7	9	285.97	560.59
35	3	10	6	4	9	7	7	8	285.97	560.66
36	4	7	6	5	8	5	7	10	286.01	560.68
37	4	8	7	4	10	5	7	9	286.03	560.78
38	4	7	6	5	8	6	8	10	286.23	560.81
39	2	9	7	5	9	5	8	10	286.32	560.83
40	4	9	6	5	8	6	8	10	286.32	560.94
41	3	9	8	5	10	5	8	10	286.56	561.03

Table B.7 "tol-before": target fill levels, cost per unit in [€] and output for each solution of the Pareto-optimal set

	B_1		B_2		B_3		B_4		B_5		B_6		B_7		B_8			
	T	R	T	R	T	R	T	R	T	R	T	R	T	R	T	R	Cost	Units
1	4	4	9	3	8	6	6	4	6	3	4	4	8	5	2	2	285.52	556.77
2	5	5	9	5	8	7	6	5	8	3	6	6	7	7	2	4	285.55	556.84
3	4	4	10	4	7	5	5	5	9	6	4	5	7	8	2	5	285.57	556.86
4	4	5	10	4	7	5	6	5	8	6	7	6	7	6	2	7	285.58	556.87
5	5	5	10	6	8	8	5	5	8	2	4	5	7	6	3	5	285.58	557.04
6	4	3	8	4	8	5	4	3	9	10	7	5	5	4	3	6	285.62	557.11
7	5	5	9	7	7	5	5	5	9	6	4	6	8	7	3	5	285.63	557.16
8	4	5	10	7	8	6	4	4	8	6	6	6	7	7	3	7	285.65	557.21
9	4	5	7	5	8	6	6	4	7	5	5	5	8	6	3	5	285.67	557.21
10	5	4	9	7	8	8	5	6	8	5	8	7	7	8	4	5	285.68	557.26
11	4	4	9	6	8	7	5	4	6	5	4	6	7	6	4	4	285.69	557.27
12	4	5	9	7	7	8	5	5	7	4	4	5	8	7	2	7	285.70	557.28
13	4	5	9	7	7	7	6	4	6	6	6	6	7	6	4	5	285.71	557.31
14	5	3	9	10	7	5	4	5	4	7	5	5	8	6	4	5	285.71	557.31
15	4	4	9	6	6	5	5	5	8	10	7	8	6	8	3	7	285.72	557.40
16	5	5	8	7	7	6	3	4	4	7	6	6	7	7	3	6	285.72	557.42
17	5	5	9	6	6	7	5	6	7	7	5	7	5	8	4	6	285.75	557.45
18	4	5	9	7	7	8	5	6	7	6	7	6	8	8	4	7	285.80	557.46
19	4	5	8	7	8	8	4	4	5	9	7	6	6	8	3	6	285.81	557.47
20	4	5	8	7	7	7	4	6	6	8	5	7	7	6	4	6	285.82	557.54
21	4	5	8	7	7	8	5	6	6	10	6	5	7	7	5	5	285.83	557.59
22	4	5	8	7	7	8	5	6	6	8	5	6	7	7	5	6	285.88	557.61
23	4	5	8	9	8	8	5	6	7	6	6	8	8	7	5	8	285.89	557.65
24	4	5	8	8	7	8	5	6	7	8	6	7	7	7	4	9	285.91	557.71
25	4	5	8	8	6	8	5	6	7	9	7	6	7	7	5	7	285.94	557.72
26	4	5	6	9	6	8	5	6	7	8	7	8	6	6	4	7	285.94	557.72
27	4	5	8	8	5	7	4	6	7	9	6	7	6	6	6	9	285.97	557.77
28	3	5	7	8	7	8	4	5	7	8	7	6	6	7	4	8	285.97	557.84

Table B.8 "tol-after": target fill levels, cost per unit in [€] and output for each solution of the Pareto-optimal set

	B_1		B_2		B_3		B_4		B_5		B_6		B_7		B_8			
	T	R	T	R	T	R	T	R	T	R	T	R	T	R	T	R	Cost	Units
1	5	4	9	1	8	4	5	2	8	2	1	6	6	3	1	7	285.56	560.62
2	5	4	10	5	7	2	6	3	9	1	4	5	2	6	1	8	285.56	560.90
3	5	4	8	1	8	4	5	1	8	1	1	7	6	2	1	6	285.56	560.98
4	4	3	9	1	7	2	5	0	8	0	2	6	4	3	2	5	285.58	561.05
5	5	4	9	0	6	1	6	2	9	1	1	7	4	4	1	5	285.60	561.22
6	5	2	9	3	6	1	5	1	9	1	1	7	2	4	0	10	285.62	561.26
7	4	3	10	3	7	2	5	1	9	1	1	7	1	5	1	9	285.68	561.26
8	5	3	9	3	7	1	6	2	8	1	0	6	1	6	1	8	285.68	561.30
9	4	3	9	0	7	2	6	0	9	1	1	7	4	0	0	9	285.68	561.34
10	5	5	10	1	8	3	6	1	9	1	2	6	3	3	2	5	285.73	561.37
11	5	5	9	0	8	2	5	0	10	1	1	6	4	2	1	3	285.77	561.40
12	4	4	9	2	7	1	6	1	9	0	1	6	2	3	1	3	285.80	561.40
13	5	2	9	3	7	1	6	1	9	0	0	7	1	4	1	10	285.80	561.48

Table B.9 "tol-with": target fill levels, cost per unit in [€] and output for each solution of the Pareto-optimal set

	B_1		B_2		B_3		B_4		B_5		B_6		B_7		B_8			
	T	R	T	R	T	R	T	R	T	R	T	R	T	R	T	R	Cost	Units
1	4	2	10	1	7	3	6	3	8	1	5	0	4	2	1	2	285.34	558.10
2	4	2	10	0	7	2	5	4	8	0	5	0	6	2	1	3	285.35	558.48
3	4	2	10	0	7	2	5	3	7	0	6	1	5	2	2	2	285.36	558.86
4	4	2	9	0	7	2	6	3	8	0	6	0	5	1	3	2	285.36	559.27
5	4	2	10	2	6	2	6	2	9	0	6	0	5	2	3	2	285.42	559.33
6	3	1	10	1	6	4	6	3	7	3	7	1	6	0	4	1	285.43	559.34
7	4	2	10	2	7	2	5	2	8	1	6	0	5	2	4	2	285.43	559.34
8	3	2	10	2	7	3	6	3	7	0	7	1	5	2	4	2	285.43	559.41
9	4	2	10	1	6	3	5	2	6	0	8	1	5	1	4	2	285.44	559.45
10	4	2	10	2	8	2	6	3	7	0	6	1	6	1	4	1	285.44	559.51
11	4	2	10	2	7	2	5	2	8	1	6	0	6	1	4	2	285.44	559.53
12	3	2	9	2	7	3	6	3	9	1	5	0	5	1	5	2	285.46	559.65
13	4	2	9	0	8	2	6	3	7	0	6	0	7	1	4	2	285.49	559.67
14	5	2	10	1	7	2	6	4	8	0	6	1	5	1	5	2	285.50	559.69
15	4	1	10	2	6	2	6	4	8	0	6	0	6	2	5	1	285.52	559.69
16	4	2	10	1	7	3	6	3	8	1	7	0	6	1	4	2	285.52	559.70
17	4	2	10	1	7	3	6	2	8	2	6	0	7	1	4	2	285.52	559.80
18	4	2	9	1	7	3	6	3	8	1	6	0	5	2	6	2	285.53	559.82
19	4	2	10	2	7	1	5	2	7	0	6	0	7	1	5	3	285.54	559.84
20	4	3	10	1	7	3	6	3	7	0	6	0	7	0	5	1	285.55	559.87
21	5	1	9	0	7	2	6	4	9	0	6	0	6	1	5	1	285.56	559.94
22	4	1	10	2	7	2	6	2	8	1	6	0	5	2	7	0	285.57	559.94
23	5	2	10	1	8	2	6	4	7	0	6	1	6	1	6	2	285.59	559.98
24	5	2	10	0	7	3	6	3	8	0	6	0	7	1	5	1	285.59	559.99
25	4	2	9	0	8	2	6	2	9	0	6	1	6	1	5	1	285.62	560.02
26	5	2	9	2	7	3	6	3	9	0	7	1	6	1	6	0	285.63	560.07
27	3	2	10	2	7	3	6	3	8	0	7	0	6	1	6	1	285.65	560.12
28	4	1	10	2	7	3	6	3	9	1	6	0	6	1	6	2	285.66	560.17
29	4	2	10	0	7	2	6	3	8	0	6	0	6	1	7	2	285.70	560.18
30	4	1	9	2	7	3	5	3	8	0	6	0	6	2	8	1	285.72	560.19
31	4	1	9	2	7	3	5	2	6	0	5	0	7	1	9	1	285.72	560.28
32	4	2	9	2	8	2	6	3	9	0	6	0	5	0	8	1	285.73	560.36
33	5	2	9	1	8	2	6	4	8	1	7	0	7	1	7	2	285.78	560.37
34	5	1	10	2	8	3	6	3	9	2	6	0	6	1	8	1	285.81	560.39
35	5	1	10	2	7	2	6	3	7	0	7	1	6	2	9	0	285.81	560.52
36	5	2	10	1	7	3	6	4	8	0	7	0	7	1	8	1	285.90	560.56
37	4	1	10	1	7	3	6	4	8	2	6	0	7	1	10	1	285.92	560.69
38	4	1	10	1	7	2	6	3	8	1	8	0	7	0	9	0	286.08	560.74
39	5	2	9	1	8	2	6	5	8	0	8	2	8	1	9	1	286.13	560.76
40	5	2	10	1	8	1	6	3	9	0	8	1	7	1	9	1	286.14	560.77
41	3	2	9	1	7	2	6	3	8	0	8	0	7	0	10	1	286.19	560.78
42	5	2	9	2	8	1	6	4	9	1	8	2	7	2	10	0	286.29	560.79
43	5	2	9	2	8	0	6	4	9	1	8	2	8	1	9	0	286.29	560.85
44	5	2	10	1	8	1	6	4	8	1	8	1	8	1	10	1	286.33	560.93
45	5	1	9	2	7	1	6	3	9	1	8	0	8	0	10	0	286.78	560.97

C Results further experiments

Table C.1 Target fill levels for different maximum duration of intervention (before - exact)

Duration of intervention	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
3	3	8	6	4	5	5	4	0
5	1	10	8	2	6	6	6	2
10	4	10	6	4	8	5	5	4
15	3	9	7	4	6	6	5	3
20	3	9	6	4	7	5	6	4
25	3	9	7	4	7	6	4	3
30[222]	3	9	6	3	8	6	6	3

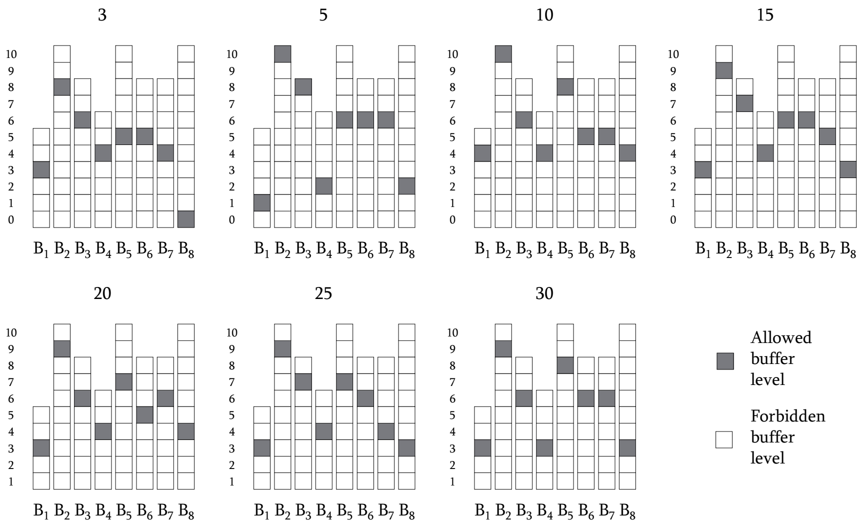


Figure C.1 Feasible and unfeasible buffer levels for variation of duration of intervention (before – exact)

[222] This maximum duration of intervention corresponds the settings of the single-objective optimization in subsection 4.2.1, obtaining the same buffer levels and results.

Table C.2 Average operation time extension per shift for each machine in [s] for varying duration of intervention (before - exact)

Duration of intervention	3	5	10	15	20	25	30
M_1	36	60	89	151	171	154	137
M_2	30	20	122	162	185	167	208
M_3	56	87	300	339	362	342	385
M_4	74	210	343	440	407	441	374
M_5	100	132	366	465	432	465	453
M_6	81	139	460	451	471	502	491
M_7	67	154	404	453	415	502	435
M_8	34	169	354	402	419	395	384
M_9	2	76	271	253	321	243	340

Table C.3 Target fill levels for different frequency of intervention (before - exact)

Frequency	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
1	3	10	5	4	7	6	6	4
2[223]	3	9	6	3	8	6	6	3
4	2	9	7	4	6	7	5	4
6	3	10	6	5	8	4	7	4
10	4	9	4	6	4	7	4	6
12	4	9	6	6	8	6	4	3
20	4	8	8	5	3	6	7	6

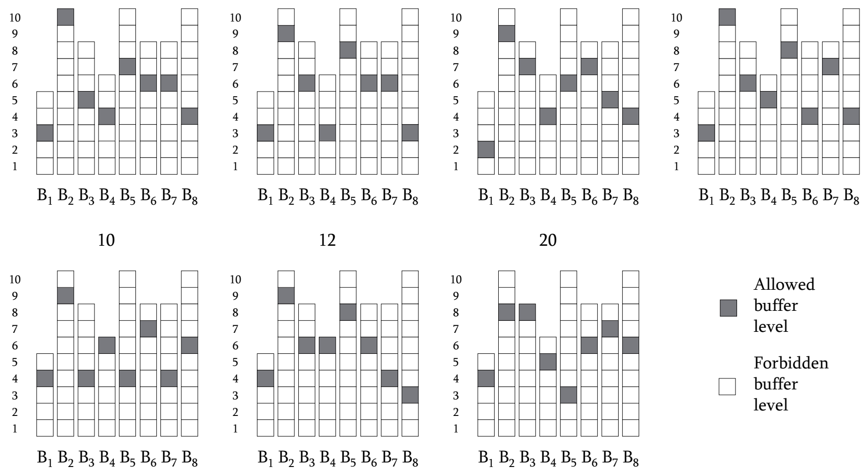


Figure C.2 Feasible and unfeasible buffer levels for variation of frequency of intervention (before – exact)

[223] This frequency of intervention corresponds the settings of the single-objective optimization in subsection 4.2.1, obtaining the same buffer levels and results.

Table C.4 Average operation time extension per shift for each machine in [s] for varying frequency of intervention (before-exact)

Frequency	1	2	4	6	10	12	20
M_1	151	168	172	106	142	99	97
M_2	161	182	130	124	216	175	172
M_3	369	358	317	366	403	363	306
M_4	346	402	423	418	347	417	471
M_5	363	370	451	503	489	558	558
M_6	385	461	444	609	378	667	394
M_7	377	459	504	503	440	674	402
M_8	377	461	456	567	339	570	468
M_9	282	306	363	476	360	426	492

D Validation of simulation

Table D.1 Confidence interval method: results from 30 replications[224]

Replication	Output	Cum. mean average	Standard deviation	Lower Interval	Upper Interval	% deviation
1	557.6467	557.65	n/a	n/a	n/a	n/a
2	558.0367	557.84	0.276	545.43	570.25	2.23%
3	558.3533	558.01	0.354	555.98	560.04	0.36%
4	557.6333	557.92	0.346	556.91	558.93	0.18%
5	557.87	557.91	0.300	557.29	558.53	0.11%
6	558.1733	557.95	0.289	557.48	558.43	0.09%
7	557.9267	557.95	0.264	557.58	558.32	0.07%
8	558.0667	557.96	0.248	557.66	558.27	0.06%
9	558.18	557.99	0.243	557.72	558.26	0.05%
10	557.8567	557.97	0.233	557.73	558.21	0.04%
11	558.4833	558.02	0.269	557.76	558.28	0.05%
12	558.0133	558.02	0.257	557.79	558.25	0.04%
13	558.0833	558.02	0.246	557.82	558.23	0.04%
14	557.79	558.01	0.245	557.81	558.21	0.04%
15	557.8333	558.00	0.240	557.81	558.18	0.03%
16	557.47	557.96	0.267	557.77	558.16	0.04%
17	558.04	557.97	0.259	557.78	558.15	0.03%
18	558.23	557.98	0.259	557.81	558.16	0.03%
19	557.8233	557.97	0.254	557.81	558.14	0.03%
20	558.1333	557.98	0.250	557.82	558.14	0.03%
21	557.5367	557.96	0.262	557.80	558.12	0.03%
22	557.7333	557.95	0.260	557.79	558.11	0.03%
23	558.2267	557.96	0.261	557.81	558.12	0.03%
24	558.1233	557.97	0.257	557.82	558.12	0.03%
25	558.0767	557.97	0.253	557.83	558.11	0.03%
26	558.2133	557.98	0.252	557.85	558.12	0.02%
27	558.24	557.99	0.252	557.86	558.13	0.02%
28	558.01	557.99	0.247	557.86	558.12	0.02%
29	557.9233	557.99	0.243	557.87	558.12	0.02%
30	557.83	557.99	0.241	557.86	558.11	0.02%

[224] following: Robinson 2007, 152-158

Table D.2 Run-Length Selection: Results from three replications

	Replication 1		Replication 2		Replication 3		
Shift	Output	Cum. mean	Output	Cum. mean	Output	Cum. mean	% Convergence
1	559	559.0	550	550.0	548	548.0	2.01%
2	558	558.5	560	555.0	562	555.0	0.63%
3	553	556.7	555	555.0	569	559.7	0.84%
4	555	556.3	561	556.5	558	559.3	0.54%
5	552	555.4	548	554.8	563	560.0	0.94%
6	557	555.7	567	556.8	548	558.0	0.42%
7	559	556.1	549	555.7	564	558.9	0.57%
8	562	556.9	549	554.9	570	560.3	0.97%
9	550	556.1	555	554.9	556	559.8	0.88%
10	560	556.5	569	556.3	564	560.2	0.70%
11	564	557.2	555	556.2	560	560.2	0.72%
12	563	557.7	551	555.8	560	560.2	0.79%
13	562	558.0	554	555.6	564	560.5	0.87%
14	558	558.0	556	555.6	551	559.8	0.75%
15	565	558.5	559	555.9	555	559.5	0.65%
16	556	558.3	564	556.4	561	559.6	0.57%
17	565	558.7	552	556.1	532	557.9	0.47%
18	556	558.6	563	556.5	559	558.0	0.37%
19	553	558.3	558	556.6	554	557.8	0.30%
233	556	557.5	562	558.2	563	558.3	0.14%
234	562	557.6	560	558.2	557	558.3	0.13%
235	556	557.5	558	558.2	548	558.3	0.13%
236	561	557.6	555	558.1	558	558.3	0.12%
237	557	557.6	568	558.2	553	558.2	0.12%
238	564	557.6	559	558.2	568	558.3	0.12%
239	564	557.6	548	558.2	550	558.2	0.11%
240	564	557.6	563	558.2	556	558.2	0.10%
241	564	557.7	554	558.2	556	558.2	0.10%
242	549	557.6	567	558.2	553	558.2	0.10%
243	566	557.7	549	558.2	555	558.2	0.09%
244	557	557.7	549	558.1	554	558.2	0.09%
245	563	557.7	553	558.1	554	558.1	0.08%
280	562	557.9	561	558.2	561	558.1	0.06%
281	561	557.9	547	558.2	556	558.1	0.05%
282	566	557.9	555	558.2	556	558.1	0.05%
283	561	557.9	566	558.2	552	558.1	0.05%
284	563	557.9	556	558.2	542	558.0	0.05%
285	563	557.9	557	558.2	565	558.0	0.04%
286	552	557.9	553	558.2	562	558.0	0.04%
287	557	557.9	564	558.2	567	558.1	0.05%
288	556	557.9	551	558.2	555	558.1	0.04%
289	564	557.9	551	558.1	556	558.0	0.04%
290	564	558.0	555	558.1	559	558.1	0.03%
291	562	558.0	558	558.1	553	558.0	0.03%
292	553	558.0	560	558.1	555	558.0	0.03%
293	559	558.0	562	558.1	564	558.0	0.03%
294	561	558.0	564	558.2	557	558.0	0.03%
295	553	557.9	556	558.2	561	558.1	0.04%
296	557	557.9	553	558.1	560	558.1	0.03%
297	565	558.0	560	558.1	561	558.1	0.03%
298	564	558.0	566	558.2	557	558.1	0.03%
299	549	558.0	565	558.2	564	558.1	0.04%
300	562	558.0	566	558.2	559	558.1	0.04%

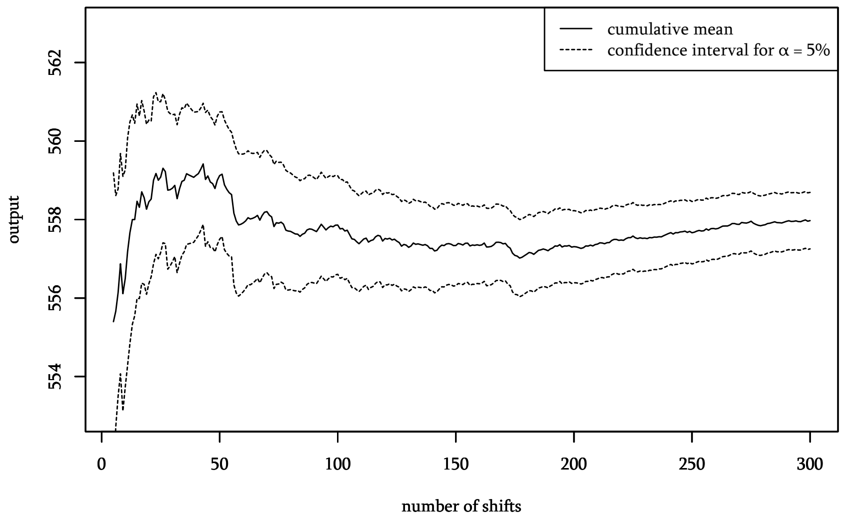


Figure D.1 Cumulative mean for replication 1 and 95% confidence intervals

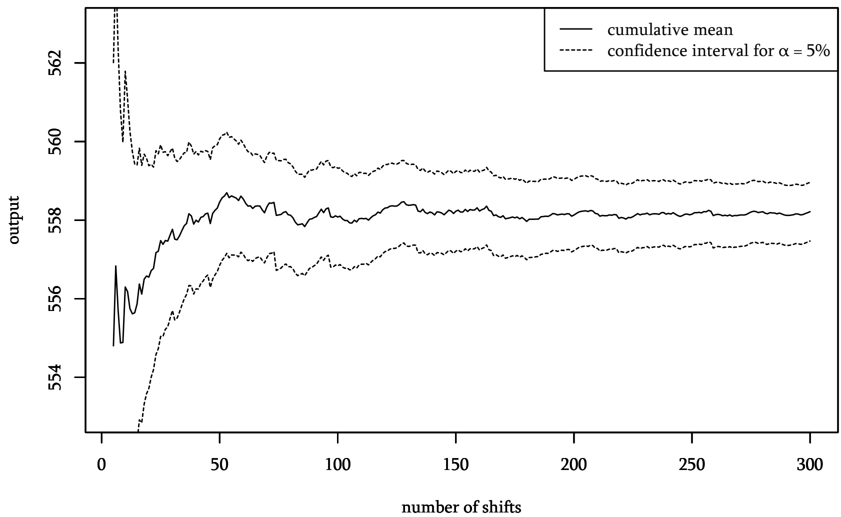


Figure D.2 Cumulative mean for replication 2 and 95% confidence intervals

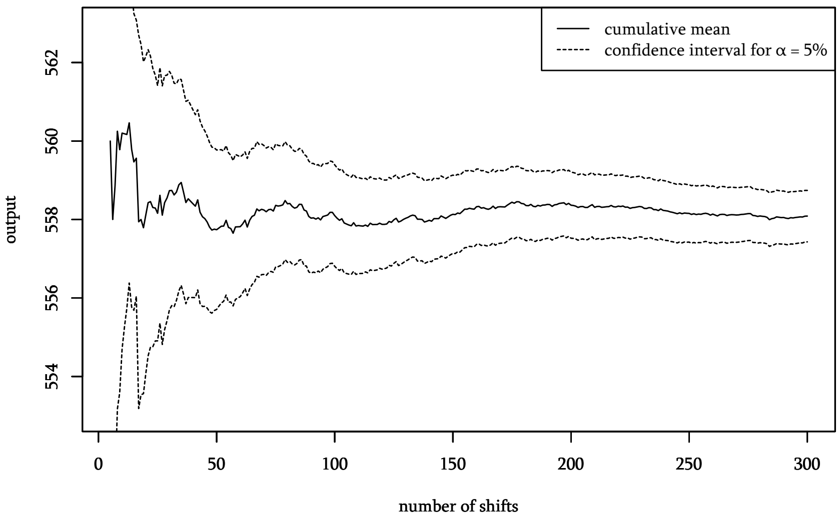


Figure D.3 Cumulative mean for replication 3 and 95% confidence intervals

References

Aarts, Emile and Jan Korst. 1989. *Simulated annealing and Boltzmann machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing.* Chichester: Wiley.

Aarts, Emile, Jan Korst, and Wil Michiels. 2005. Simulated Annealing. In *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, edited by Edmund K. Burke and Graham Kendall, 187–210. New York: Springer US.

Adam Opel AG. 2015. *Year in Review 2014: Facts & Figures*. Available from <http://www.opel.de/content/dam/Opel/Europe/germany/nscwebsite/de/03_ExperienceBrand/01_AboutOpel/Opel-Company-FactsFigures2014_en-de.pdf>. Accessed 29 September 2015.

Akella, Ramakrishna and Prabhat Ranjith Srikanta Kumar. 1986. Optimal Control of Production Rate in a Failure Prone Manufacturing System. *IEEE Transactions on Automatic Control* AC-31 (2):116–126.

Aldinger, Lars, Carmen Constantinescu, Vera Hummel, Rita Kreuzhage, and Engelbert Westkämper. 2006. Neue Ansätze im "advanced Manufacturing Engineering". *wt Werkstattstechnik online (2006)* 96:110–114.

Aristoteles. Siebentes Buch: Von der Wesenheit. In *Metaphysik*, edited by Ernesto Grassi and Hector Carvallo, 138–175. Reinbek: Rowohlt.

Arnold, Dirk. V. and Nikolaus. Hansen. 2010. Active Covariance Matrix Adaptation for the (1+1)-CMA-ES. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, edited by M. Pelikan and J. Branke. New York: ACM.

Ashby, Ross W. 1964. *An Introduction to Cybernetics.* London: Methuen.

ASIM. 1997. *Leitfaden für Simulationsbenutzer in Produktion und Logistik*. ASIM-Mitteilungen.

AUDI AG. 2015a. *Audi MediaCenter: Ingolstadt*. Available from <https://www.audi-mediacenter.com/de/audi-ag-werk-ingolstadt-2862>.

AUDI AG. 2015b. *Audi MediaCenter: Neckarsulm*. Available from <https://www.audi-mediacenter.com/de/neckarsulm-3290>.

Bäck, Thomas. 1996. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms.* New York: Oxford University Press.

Bäck, Thomas, Christophe Foussette, and Peter Krause. 2013. *Contemporary Evolution Strategies.* Berlin: Springer.

Bäck, Thomas and Hans-Paul Schwefel. 1995. Evolution Strategies I: Variants and their computational implementation. In *Genetic algorithms in engineering and computer science*, edited by Gabriel Winter, J. Periaux, M. Galan and P. Cuesta, chapter 6, 111–126. Chichester: Wiley.

Bäck, Thomas and Hans-Paul Schwefel. 1996. Evolutionary computation: an overview. In *IEEE International Conference on Evolutionary Computation*, 20–29.

Backhaus, Klaus. 2008. *Multivariate Analysemethoden: Eine anwendungsorientierte Einführung.* Berlin: Springer.

Balci, Osman. 1998. Verification, Validation and Testing. In *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*, edited by Jerry Banks, 335–393. New York: Wiley.

Balci, Osman. 2003. Verification, Validation, and Certification of Modeling and Simulation Applications. In *Proceedings of the 2003 Winter Simulation Conference: Driving Innovation*, edited by Stephen E. Chick, Paul J. Sanchez, David M. Ferrin and Douglas J. Morrice, 150–158: WSC.

Bangert, Patrick. 2012. *Optimization for Industrial Problems.* Berlin: Springer.

Bellman, Richard. 1957. *Dynamic Programming.* Princeton, NJ: Princeton University Press.

Ben-Salem, A., A. Gharbi, and A. Hajji. 2014. An Environmental Hedging Point Policy to control production rate and emissions in unreliable manufacturing systems. *International Journal of Production Research* 53 (2):435–450.

Bertalanffy, Ludwig von. 1972. *General System Theory: Foundations, Development, Applications.* New York: Braziller.

Beyer, Hans-Georg. 2001. *The Theory of Evolution Strategies.* Berlin: Springer.

Biela, Christoph. 2015. Concept development for steering of decoupling elements in assembly. Master's thesis, University of Applied Sciences Munich.

Bielecki, Tomasz R. and Prabhat Ranjith Kumar. 1988. Optimality of zero-inventory policies for unreliable manufacturing systems. *Operations Research* 36 (4):532–541.

Blum, Christian and Andrea Roli. 2008. Hybrid Metaheuristics: An Introduction. In *Hybrid Metaheuristics: An Emerging Approach to Optimization*, edited by Christian Blum, Maria José Blesa Aguilera, Andrea Roli and Michael Sampels, 1–30. Berlin: Springer.

BMW AG. 2015. *Historie - Meilensteine: Wegweisende Ereignisse der BMW Group*. Available from <http://www.bmwgroup.com/d/0_0_www_bmwgroup_com/unternehmen/historie/meilensteine/meilensteine.html>.

Boyd, Stephen P. and Lieven Vandenberghe. 2004. *Convex Optimization.* Cambridge, UK: Cambridge University Press.

Brühl, Volker. 2015. *Wirtschaft des 21. Jahrhunderts: Herausforderungen in der Hightech-Ökonomie.* Wiesbaden: Springer Gabler.

Burman, Mitchell H. 1995. New Results in Flow Line Analysis. Doctoral thesis, Massachusetts Institute of Technology.

Buzacott, John A. 1982. "Optimal" Operating Rules for Automated Manufacturing Systems. *IEEE Transactions on Automatic Control* AC-27 (1):80–86.

Černý, V. 1985. Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm. *Journal of Optimization Theory and Applications* 45 (1):41–51.

Chwif, Leonardo, Marcos Ribeiro Pereira-Barretto, and Ray J. Paul. 2000. On simulation model complexity. In *Proceedings of the 2000 Winter Simulation Conference*, edited by Paul A. Fishwick, Keebom Kang, Jeffrey A. Joines and Russell R. Barton. San Diego, CA: Societey for Computer Simuation International.

Collette, Yann and Patrick Siarry. 2003. *Multiobjective Optimization: Principles and Case Studies.* Berlin: Springer.

Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. *Introduction to Algorithms.* Cambridge, Massachusetts: MIT Press.

Daimler AG. 2015a. *Bremen, Mercedes-Benz Werk.* Available from <http://www.daimler.com/bremen>.

Daimler AG. 2015b. *Rastatt, Mercedes-Benz Werk*. Available from <http://www.daimler.com/dccom/0-5-1037140-49-1333008-1-0-0-0-0-0-9506-0-0-0-0-0-0-0-0.html>.

Daimler AG. 2015c. *Sindelfingen, Mercedes-Benz Werk*. Available from <http://www.daimler.com/dccom/0-5-1037141-49-1363599-1-0-0-0-0-0-9506-0-0-0-0-0-0-0-0.html>.

Dallery, Yves, Rene David, and Xiao-Lan Xie. 1988. An Efficient Algorithm for Analysis of Transfer Lines With Unreliable Machines and Finite Buffers. *IIE Transactions* 20 (3):280–283.

Dallery, Yves and Stanley B. Gershwin. 1992. Manufacturing flow line systems: a review of models and analytical results. *Queueing Systems* 12 (1-2):3–94.

Dean, Angela M. and Daniel Voss. 1999. *Design and analysis of experiments.* New York: Springer.

Deb, Kalyanmoy. 2004. *Multi-Objective Optimization using Evolutionary Algorithms.* Chichester: Wiley.

Deb, Kalyanmoy, Samir Agrawal, Amrit Pratap, and T. Meyarivan. 2000. A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II. In *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, edited by Marc Schoenauer, Kalyanmoy Deb, Günther Rudolph, Xin Yao, Evelyne Lutton, Juan Julian Merelo and Hans-Paul Schwefel, 849–858. Berlin: Springer.

Demir, Leyla, Semra Tunali, and Deniz Tursel Eliiyi. 2014. The state of the art on buffer allocation problem: A comprehensive survey. *Journal of Intelligent Manufacturing* 25 (3):371–392.

Demir, Leyla, Semra Tunali, and Deniz Türsel Eliiyi. 2010. An adaptive tabu search approach for buffer allocation problem in unreliable production lines. In *24th Mini EURO Conference on Continuous Optimization and Information-Based Technologies in the Financial Sector*, 207–212.

Demir, Leyla, Semra Tunali, and Deniz Türsel Eliiyi. 2012. An adaptive tabu search approach for buffer allocation problem in unreliable non-homogenous production lines. *Computers & Operations Research* 39 (7):1477–1486.

Demir, Leyla, Semra Tunali, and Arne Løkketangen. 2011. A tabu search approach for buffer allocation in production lines with unreliable machines. *Engineering Optimization* 43 (2):213–231.

DIN IEC 60050-351. *Internationales Elektrotechnisches Wörterbuch – Teil 351: Leittechnik.* Berlin: Beuth.

Domschke, Wolfgang, Andreas Drexl, Robert Klein, and Armin Scholl. 2015. *Einführung in Operations Research.* Berlin: Springer Gabler.

Dorigo, Marco and Gianni Di Caro. 1999. The Ant Colony Optimization Meta-Heuristic. In *New Ideas in Optimization*, edited by David Corne, Marco Dorigo and Fred Glover, 11–32. London: McGraw-Hill.

Dowsland, Kathryn A. and Jonathan M. Thompson. 2012. Simulated Annealing. In *Handbook of Natural Computing*, edited by Grzegorz Rozenberg, Thomas Bäck and Joost N. Kok, 1623–1655. Berlin: Springer.

Dr. Ing. h.c. F. Porsche AG. 2014. *Das Porsche-Stammwerk im Wandel der Zeit*: *Unternehmen 28.07.2015*. Available from <http://newsroom.porsche.com/de/unternehmen/porsche-stammwerk-zuffenhausen-geschichte-wandel-der-zeit-10208.html>.

Dr. Ing. h.c. F. Porsche AG. 2015. *Porsche Leipzig*: *Meilensteine*. Available from <https://www.porsche-leipzig.com/porsche-in-leipzig/meilensteine/>.

Dréo, Johann, Alain Pétrowski, Patrick Siarry, and Eric Taillard. 2006. *Metaheuristics for Hard Optimization*: *Simulated Annealing, Tabu Search, Evolutionary and Genetic Algorithms, Ant Colonies,… Methods and Case Studies.* Berlin: Springer.

Eckart Zitzler. 2012. Evolutionary Multiobjective Optimization. In *Handbook of Natural Computing*, edited by Grzegorz Rozenberg, Thomas Bäck and Joost N. Kok. Berlin: Springer.

Ford Werke GmbH. 2015a. *Ford in Deutschland - Tradition mit Zukunft*: *Im Überblick - Standorte (Köln)*. Available from <http://www.ford.de/UeberFord/FordinDeutschland/Standorte>.

Ford Werke GmbH. 2015b. *Ford in Deutschland - Tradition mit Zukunft*: *Im Überblick - Standorte (Saarlouis)*. Available from <http://www.ford.de/UeberFord/FordinDeutschland/Standorte/tabid=tab2>.

Furmans, Kai, Thorsten Schmidt, Ingolf Meinhardt, Martin Epp, and Marion Rimmele. Analytical computation of sojourn time distributions in large-scale conveyer systems. In *10th Conference on Stochastic Models of Manufacturing and Service Operations (SMMSO 2015)*: *June 1-6, 2015, Volos Greece*, edited by George Liberopoulos, 75–82: University of Thessaly Press.

Gendreau, Michel and Jean-Yves Potvin. 2005. Tabu Search. In *Search Methodologies*: *Introductory Tutorials in Optimization and Decision Support Techniques*, edited by Edmund K. Burke and Graham Kendall, 165–186. New York: Springer US.

Gershwin, Stanley B. 1987. An Efficient Decomposition Method for the Approximate Evaluation of Tandem Queues with Finite Storage Space and Blocking. *Operations Research* 35 (2):291–305.

Gershwin, Stanley B. 1994. *Manufacturing Systems Engineering.* Englewood Cliffs, NJ: Prentice Hall.

Gershwin, Stanley B. and James E. Schor. 2000. Efficient algorithms for buffer space allocation. *Annals of Operations Research* 93 (1-4):117–144.

Gershwin, Stanley B., Bariş Tan, and Michael H. Veatch. 2009. Production control with backlog-dependent demand. *IIE Transactions* 41 (6):511–523.

Gharbi, Ali, Adnene Hajji, and Karem Dhouib. 2011. Production rate control of an unreliable manufacturing cell with adjustable capacity. *International Journal of Production Research* 49 (21):6539–6557.

Giordano, M. and F. Martinelli. Optimal safety stock for unreliable, finite buffer, single machine manufacturing systems. *Proceedings of the 2002 IEEE International Conference on Robotics & Automation*:2339–2344.

Glover, Fred. 1986. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research* 13 (5):533–549.

Glover, Fred and Manuel Laguna. 1998. *Tabu search.* Boston, MA: Kluwer.

Goldberg, David E. and Kalyanmoy Deb. 1991. A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. In *FOGA-1*, edited by Gregory J. E. Rawlins, 69–93. San Mateo, CA: Morgan Kaufmann Publishers, Inc.

Goldberg, David Edward. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*: Addison-Wesley.

Goss, S., S. Aron, J. L. Deneubourg, and J. M. Pasteels. 1989. Self-organized Shortcuts in the Argentine Ant. *Naturwissenschaften* 76 (12):579–581.

Groover, Mikell P. 1987. *Automation, Production Systems, and Computer Integrated Manufacturing.* Englewood Cliffs, NJ: Prentice Hall.

Günther, Hans-Otto and Horst Tempelmeier. 2012. *Produktion und Logistik.* Berlin: Springer.

Gutjahr, Walter J. 2004. S-ACO: An Ant-Based Approach to Combinatiorial Optimization Under Uncertainty. In *Ant Colony Optimization and Swarm Intelligence*: *4th International Workshop, ANTS 2004, Brussels, Belgium, September 5-8, 2004. Proceeding*, edited by Marco Dorigo, Mauro Birattari, Christian Blum, Luca Maria Gambardella, Francesco Mondada and Thomas Stützle. Berlin: Springer.

Harris, John H. and Stephen G. Powell. 1999. An algorithm for optimal buffer placement in reliable serial lines. *IIE Transactions* 31 (4):287–302.

Hegenscheidt, Matthias. 2003. *Kennliniengestützte Leistungsprognose verketteter Produktionssysteme*. Dissertation, Universität Hannover. Düsseldorf: VDI-Verl.

Ho, Y. C., M. A. Eyler, and T. T. Chien. 1979. A gradient technique for general buffer storage design in a production line. *International Journal of Production Research* 17 (6):557–580.

Holland, John H. 1975. *Adaptation in natural and artificial systems*: *An introductory analysis with applications to biology, control, and artificial intelligence.* Ann Arbor: University of Michigan Press.

Khuri, André I. 2011. Response Surface Methodology. In *International encyclopedia of statistical science*, edited by Miodrag Lovric, 1229–1231. Berlin: Springer.

Kimemia, Joseph and Stanley B. Gershwin. 1983. An Algorithm for the Computer Control of a flexible Manufacturing System. *IIE Transactions* 15 (4):353–362.

Kirkpatrick, Scott R., C. D. Gelatt, and Mario P. Vecchi. 1983. Optimization by Simulated Annealing. *Science* 220 (4598):671–680.

Klampfl, Erica, Oleg Gusikhin, and Giuseppe Rossi. 2006. Optimization of workcell layouts in a mixed-model assembly line environment. *International Journal of Flexible Manufacturing Systems* 17 (4):277–299.

Kletti, Jürgen and Jochen Schumacher. 2011. *Die perfekte Produktion*: *Manufacturing Excellence durch Short Interval Technology (SIT).* Berlin: Springer.

Kose, Simge Yelkenci and Ozcan Kilincci. 2015. Hybrid approach for buffer allocation in open serial production lines. *Computers & Operations Research* 60:67–78.

Kuhn, Heinrich. 2002. Analyse des Nutzungsgrads verketteter Produktionsanlagen. *Zeitschrift für wirtschaftlichen Fabrikbetrieb* 97 (3):116–120.

Lee, Hsu-Tung, Shao-Kai Chen, and Shunder Chang. 2009. A Meta-heuristic Approach to Buffer Allocation in Production Line. *Journal of C.C.I.T.* 38 (1):167–178.

1992. *Lexikon der Produktionsplanung und -steuerung*: *Begriffszusammenhänge und Begriffsdefinitionen.* Düsseldorf: VDI Verlag.

Lotter, Bruno. 1992. *Wirtschaftliche Montage*: *Ein Handbuch für Elektrogerätebau und Feinwerktechnik.* Düsseldorf: VDI Verlag.

Lotter, Bruno, and Hans-Peter Wiendahl, eds. 2012. *Montage in der industriellen Produktion: Ein Handbuch für die Praxis.* Berlin: Springer Vieweg.

Martí, Rafael and Gerhard Reinelt. 2011. *The Linear Ordering Problem: Exact and Heuristic Methods in Combinatorial Optimization.* Berlin: Springer.

Martinelli, F. 2007. Optimality of a Two-Threshold Feedback Control for a Manufacturing System With a Production Dependent Failure Rate. *IEEE Transactions on Automatic Control* 52 (10):1937–1942.

Matzka, Judith M. 2011. *Discrete Time Analysis of Multi-Server Queueing Systems in Material Handling and Service*. Dissertation, Karslruher Institut für Technologie. Karlsruhe: KIT Scientific Publishing.

Metropolis, Nicholas, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. 1953. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics* 21 (6):1087.

Michalewicz, Zbigniew. 1996. *Genetic Algorithms + Data Structures = Evoltuion Programs.* Berlin: Springer.

Michalewicz, Zbigniew and David B. Fogel. 2004. *How to Solve It: Modern Heuristics.* Berlin: Springer.

Mok, Pikyin Y. and B. Porter. 2006. Evolutionary optimisation of hedging points for unreliable manufacturing systems. *International Journal of Advanced Manufacturing Technology* 28 (1-2):205–214.

Montgomery, Douglas C. 2013. *Design and Analysis of Experiments.* Hoboken, NJ: Wiley.

Myers, Raymond H., Douglas C. Montgomery, and Christine M. Anderson-Cook. 2009. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments.* Hoboken, NJ: Wiley.

Nahas, Nabil, Daoud Ait-Kadi, and Mustapha Nourelfath. 2006. A new approach for buffer allocation in unreliable production lines. *International Journal of Production Economics* 103 (2):873–881.

Nourelfath, Mustapha, Nabil Nahas, and Daoud Ait-Kadi. 2005. Optimal design of series production lines with unreliable machines and finite buffers. *Journal of Quality in Maintenance Engineering* 11 (2):121–138.

Nyhuis, Peter, Gregor von Cieminski, Andreas Fischer, and Klaus Feldmann. 2005. Applying Simulation and Analytical Models for Logistic Performance Prediction. *CIRP Annals - Manufacturing Technology* 54 (1):417–422.

Nyhuis, Peter and Hans-Peter Wiendahl. 2006. Logistic Production Operating Curves – Basic Model of the Theory of Logistic Operating Curves. *CIRP Annals - Manufacturing Technology* 55 (1):441–444.

Ostermeier, Andreas, Andreas Gawelczyk, and Nikolaus Hansen. 1994a. A Derandomized Approach to Self-Adaptation of Evolution Strategies. *Evolutionary Computation* 2 (4):369–380.

Ostermeier, Andreas, Andreas Gawelczyk, and Nikolaus Hansen. 1994b. Step-Size Adaptation Based on Non-Local Use Selection Information. In *Proceedings of the 3rd International Conference on Parallel Problem Solving from Nature (PPSN III)*, edited by Yuval Davidor, Hans-Paul Schwefel and Reinhard Männer, 189–198. Berlin: Springer.

Oyman, A. I. and H. G. Beyer. 2000. Analysis of the (μ/μ, λ)-ES on the Parabolic Ridge. *Evolutionary Computation* 8 (3):267–289.

Papadimitriou, Christos H. and Kenneth Steiglitz. 1998. *Combinatorial Optimization: Algorithms and Complexity.* Mineola, NY: Dover.

Papadopoulos, H. T. and Cathal Heavey. 1996. Queueing theory in manufacturing systems analysis and design: A classification of models for production and transfer lines. *European Journal of Operational Research* 92 (1):1–27.

Park, Taeho. 1993. A two-phase heuristic algorithm for determining buffer sizes of production lines. *International Journal of Production Research* 31 (3):613–631.

Perkins, James R. and Rayadurgam Srikant. 1997. Scheduling Multiple Part-Types in an Unreliable Single-Machine Manufacturing System. *IEEE Transactions on Automatic Control* 42 (3):364–377.

Pidd, M. 1996. Five Simple Principles of Modelling. In *Proceedings of the 1996 Winter Simulation Conference*, edited by John M. Charnes, Douglas J. Morrice, Daniel T. Brunner and James J. Swain. Piscataway, NJ: IEEE Computer Society.

Rabe, Markus, Sven Spieckermann, and Sigrid Wenzel, eds. 2008. *Verifikation und Validierung für die Simulation in Produktion und Logistik: Vorgehensmodelle und Techniken.* Berlin: Springer.

Rechenberg, Ingo. 1994. *Evolutionsstrategie '94.* Stuttgart: Frommann-Holzboog.

Robinson, Stewart. 2007. *Simulation: The Practice of Model Development and Use.* Chichester: Wiley.

Rudolph, Günter. 2012. Evolutionary Strategies. In *Handbook of Natural Computing*, edited by Grzegorz Rozenberg, Thomas Bäck and Joost N. Kok, 673–698. Berlin: Springer.

Sabuncuoglu, I., E. Erel, and Y. Gocgun. 2006. Analysis of serial production lines: Characterisation study and a new heuristic procedure for optimal buffer allocation. *International Journal of Production Research* 44 (13):2499–2523.

Salt, John D. 1993. Keynote Address: Simulation Should Be Easy and Fun! In *Proceedings of the 1993 Winter Simulation Conference*, edited by Gerald W. Evans, Mansooreh Mollaghasemi, Edward C. Russell and William E. Biles, 1–5: ACM Press.

Schleyer, Marc. 2007. *Discrete Time Analysis of Batch Processes in Material Flow Systems*. Dissertation, Universität Karlsruhe (TH). Karlsruhe: Universitätsverlag Karlsruhe.

Schleyer, Marc and Kai Furmans. 2007. An analytical method for the calculation of the waiting time distribution of a discrete time G/G/1-queueing system with batch arrivals. *OR Spectrum* 29 (4):745–763.

Schlick, Christopher, Ralph Bruder, and Holger Luczak, eds. 2010. *Arbeitswissenschaft.* Berlin: Springer.

Schor, James E. 1995. Efficient algorithms for buffer allocation. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA.

Schwefel, Hans-Paul and Günter Rudolph. 1995. Contemporary Evolution Strategies. In *Advances in Artificial Life: Third European Conference on Artificial Life Granada, Spain, June 4-6, 1995 proceedings*, edited by Federico Morán, Alvaro Moreno, Juan Julián Merelo and Pablo Chacón, 891–907. Berlin: Springer.

Semery, A. 1986. *Exemples de lignes pour lesquelles l'algorithme de Gershwin ne converge pas*: Note Interne Renault, AF/562.

Seong, D., S. Y. Chang, and Y. Hong. 1995. Heuristic algorithms for buffer allocation in a production line with unreliable machines: International Journal of Production Research. *International Journal of Production Research* 33 (7):1989–2005.

Shi, Leyuan and Shuli Men. 2003. Optimal buffer allocation in production lines. *IIE Transactions* 35 (1):1–10.

Spieckermann, Sven and Dirk Wortmann. 2003. Die Simulation von Lackieranlagen in der Automobilindustrie. In *Simulation in der Automobilproduktion*, edited by Johann Bayer, Thomas Collisi and Sigrid Wenzel, 47–59. Berlin: Springer.

Tempelmeier, Horst. 2003. Practical considerations in the optimization of flow production systems. *International Journal of Production Research* 41 (1):149–170.

ten Hompel, Michael, Thorsten Schmidt, and Lars Nagel. 2007. *Materialflusssysteme: Förder- und Lagertechnik.* Berlin: Springer.

VDI 3423. *Verfügbarkeit von Maschinen und Anlagen.* Berlin: Beuth.

VDI 3581. *Verfügbarkeit von Transport- und Lageranlagen sowie deren Teilsysteme und Elemente.* Berlin: Beuth.

VDI 3633 Blatt 1. *Simulation von Logistik-, Materialfluss- und Produktionssystemen.* Berlin: Beuth.

VDI 3633 Entwurf. *Simulation von Logistik-, Materialfluss- und Produktionssystemen - Begriffe.* Berlin: Beuth.

VDI 3649. *Anwendung der Verfügbarkeitsrechnung für Förder- und Lagersysteme.* Berlin: Beuth.

Vergara, Hector A. and David S. Kim. 2009. A new method for the placement of buffers in serial production lines. *International Journal of Production Research* 47 (16):4437–4456.

Vitanov, Ivan V., Valentin I. Vitanov, and David K. Harrison. 2009. Buffer capacity allocation using ant colony optimisation algorithm. In *Proceedings of the 2009 Winter Simulation Conference*, edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls, 3158–3168. Austin, TX: WSC.

Volkswagen AG. 2015a. *Volkswagen AG Werk Emden*. Available from <https://www.volkswagen-media-services.com/detailpage/-/detail/Volkswagen-AG-Werk-Emden/view/2085717/6e1e015af7bda8f2a4b42b43d2dcc9b5?p_p_auth=gqDxNm8b>. Accessed 29 September 2015.

Volkswagen AG. 2015b. *Volkswagen AG Werk Wolfsburg*. Available from <https://www.volkswagen-media-services.com/detailpage/-/detail/Volkswagen-AG-Werk-Wolfsburg/view/442911/6e1e015af7bda8f2a4b42b43d2dcc9b5?p_p_auth=gqDxNm8b>. Accessed 29 September 2015.

Volkswagen AG. 2015c. *Volkswagen Osnabrück GmbH*. Available from <https://www.volkswagen-media-services.com/detailpage/-/detail/Volkswagen-Osnabrck-

GmbH/view/443381/1aeeea96db365662765ca55af97d0da9?p_p_auth=gqDxNm8b>. Accessed 29 September 2015.

Volkswagen AG. 2015d. *Volkswagen Sachsen GmbH*: *Die Gläserne Manufaktur Dresden*. Available from <https://www.volkswagen-media-services.com/detailpage/-/detail/Volkswagen-Sachsen-GmbH-Die-Glserne-Manufaktur-Dresden/view/2083689/1aeeea96db365662765ca55af97d0da9?p_p_auth=gqDxNm8b>. Accessed 29 September 2015.

Volkswagen AG. 2015e. *Volkswagen Sachsen GmbH*: *Standort Zwickau*. Available from <https://www.volkswagen-media-services.com/detailpage/-/detail/Volkswagen-Sachsen-GmbH-Standort-Zwickau/view/2083674/6e1e015af7bda8f2a4b42b43d2dcc9b5?p_p_auth=gqDxNm8b>. Accessed 29 September 2015.

Walenda, Harry. 1991. *Analyse des Betriebsverhaltens elastisch verketteter automatischer Montageanlagen*. Dissertation, Universität Hannover. Düsseldorf: VDI-Verl.

Wang, Zheng and Stanley B. Gershwin. 2015. Heuristic Production and Sale Policy for a Two-Product-Type Manufacturing System With Downward Substitution. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45 (6):929–942.

Warnecke, Hans-Jürgen, Hans-Günter Löhr, and Waldemar Kiener. 1975. *Montagetechnik*: *Schwerpunkt der Rationalisierung.* Mainz: Krausskopf.

Wenzel, Sigrid, Matthias Weiß, Simone Collisi-Böhmer, Holger Pitsch, and Oliver Rose. 2008. *Qualitätskriterien für die Simulation in Produktion und Logistik*: *Planung und Durchführung von Simulationsstudien.* Berlin: Springer.

Westkämper, Engelbert. 2008. Fabriken sind komplexe langlebige Systeme. In *Beiträge zu einer Theorie der Logistik*, edited by Peter Nyhuis, 85–107. Berlin: Springer.

Westkämper, Engelbert, Carmen Constantinescu, and Vera Hummel. 2006. New Paradigm in Manufacturing Engineering: Factory Life Cycle. *Production Engineering* 1 (13):143–146.

Wiendahl, Hans-Peter and Matthias Hegenscheidt. 2003. Flexible Transferstraße: Hochdynamisches Störverhalten reduziert Effekte von Maßnahmen zur Leistungssteigerung. In *Simulation in der Automobilproduktion*, edited by Johann Bayer, Thomas Collisi and Sigrid Wenzel, 71–80. Berlin: Springer.

Zäpfel, Günther, Roland Braune, and Michael Bögl. 2010. *Metaheuristic Search Concepts*: *A Tutorial with Applications to Production and Logistics.* Berlin: Springer.

Die Reihen der Fortschritt-Berichte VDI:

1 Konstruktionstechnik/Maschinenelemente
2 Fertigungstechnik
3 Verfahrenstechnik
4 Bauingenieurwesen
5 Grund- und Werkstoffe/Kunststoffe
6 Energietechnik
7 Strömungstechnik
8 Mess-, Steuerungs- und Regelungstechnik
9 Elektronik/Mikro- und Nanotechnik
10 Informatik/Kommunikation
11 Schwingungstechnik
12 Verkehrstechnik/Fahrzeugtechnik
13 Fördertechnik/Logistik
14 Landtechnik/Lebensmitteltechnik
15 Umwelttechnik
16 Technik und Wirtschaft
17 Biotechnik/Medizintechnik
18 Mechanik/Bruchmechanik
19 Wärmetechnik/Kältetechnik
20 Rechnerunterstützte Verfahren (CAD, CAM, CAE CAQ, CIM ...)
21 Elektrotechnik
22 Mensch-Maschine-Systeme
23 Technische Gebäudeausrüstung

ISBN 978-3-18-3**05813**-6