Technische Universität München
TUM School of Management

Integrated planning of unreliable flow lines with limited buffer capacities and spare parts provisioning

Florian Elias Sachs

Vollständiger Abdruck der von der TUM School of Management der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Wirtschafts- und Sozialwissenschaften (Dr. rer. pol.)

genehmigten Dissertation.

Vorsitzender: Prof. Dr. rer. pol. David A. Wuttke

Prüfende der Dissertation:
1. Prof. Dr. rer. nat. Gudrun P. Kiesmüller
2. Prof. Dr. ir. Geert-Jan J.A.N. van Houtum, Technische Universiteit Eindhoven

Die Dissertation wurde am 30.09.2022 bei der Technischen Universität München eingereicht und von der TUM School of Management am 15.11.2022 angenommen.

To my mother

This page intentionally left blank

Abstract

In this thesis, we consider different models of unreliable flow lines with intermediate buffers decoupling adjacent production stages. Each machine contains exactly one unit of a failure-prone critical component. In case of a component failure, we assume that the component units can be replaced by ready-to-use spare parts, thus achieving a high machine availability. For the stock-keeping of spare parts, we assume a one-for-one replenishment policy.

We present a model of a two-machine flow line with stochastic processing times and develop a continuous-time Markov chain to exactly evaluate the essential system characteristics like average throughput or inventories. Based on that, we present a novel decomposition approach to approximately compute the flow-line characteristics for systems with stochastic processing times consisting of an arbitrary number of machines, limited buffer capacity, and spare parts. By means of extensive numerical examples, we analyze the effects of system parameters on approximation quality. We conclude that our method yields striking accuracy after comparing with results obtained by a Markov approach for smaller lines and discrete-event simulation for longer lines. Our results indicate the existence of complex interaction and partial substitution effects between buffer capacity and spare part base-stock levels.

To answer the question of how to design a flow line with buffers and spares, we study the buffer and spare part allocation problem. Since the buffer allocation problem is NP-hard, we focus on heuristic solutions. We propose three greedy heuristics and apply the two metaheuristics, simulated annealing and a genetic algorithm. We compare all heuristics to complete enumeration for small flow lines in order to find the best-suited algorithm to apply to longer flow lines. We analyze different balanced and unbalanced flow line scenarios using a large-scale numerical study. First, we find that spare parts tend to be more effective when arranged at or near the center of a flow line already known for buffers. Second, we quantify the interaction between buffers and spare parts and amplify the strong cost dependence of the best flow line design. Third, we identify that results from the literature on spare parts planning still apply but that planners need our new approach for quantification because of the mentioned interactions.

Using a model of a two-machine system, we study the impact of deterministic and fixed processing times. We calculate system characteristics with a discrete-time Markov chain and apply complete enumeration because the search space size allows for this optimization procedure. We observe that our results are still applicable and get even more considerable if the processing time variability is not present. Furthermore, we compare two system models: one with different critical components, as before, and one with identical components. By comparing both cases, we can identify the impact of component standardization regularly observed in practice. It turns out that component commonality can render significant cost savings possible. The reason for this is, on the one hand, the pooling effect of safety stocks and, on the other hand, the possibility of reducing buffer capacity.

Acknowledgments

> If man is to survive, he will have learned to take a delight in the essential differences between men and between cultures. He will learn that differences in ideas and attitudes are a delight, part of life's exciting variety, not something to fear.
>
> *Eugene Wesley "Gene" Roddenberry, 1921–1991*

This quote originates from the American screenwriter and producer Gene Roddenberry. He lost several jobs in the 1960s because of his insistence on representing all people in his TV series, irrespective of the color of their skin. Later on, Gene Roddenberry created not only a famous TV series but a whole science-fiction world called STAR TREK. The universe he excogitated consists of a multitude of species, different cultures, and — maybe most importantly — a human race that has learned to praise differences instead of despising them and using words as the primary weapon to fight conflicts out. Furthermore, he exemplified how valuable striving for new knowledge is — may it be between the stars, regarding human relations, or for everyday problems on earth. The latter will be the case for this thesis. It is my hope that science — through its way of thinking and reasoning — will eventually help humanity stand together.

Writing this dissertation is only one step in continuing a journey that started long ago. Although I am now the person writing this thesis, all of this would not have been possible without several people I would like to thank. All of your contributions to this work and my life will not be forgotten.

First and foremost, I want to thank my supervisor Professor Gudrun Kiesmüller. Her ongoing feedback, guidance, and support were incredibly helpful. I want to express my gratitude for her initial research idea and her profound advice in scientific and vocational affairs. Her immense experience was always a source of inspiration. It was a pleasure to be part of her teams in Magdeburg and Heilbronn, broadening my view and deepening my knowledge in applying mathematical methods to operations management problems.

I thank Professor Stefan Helber for the fruitful collaboration and detailed discussions. It was a great pleasure to work with him and partake in his experience. Furthermore, I want to thank Professor Geert-Jan van Houtum for being part of the examination committee and Professor David Wuttke for chairing it.

I want to wholeheartedly thank all my current and former colleagues at the Center for Digital Transformation of the Technical University of Munich and the Chair of Operations Management at the Otto-von-Guericke University Magdeburg. My particular thanks go to my colleagues and dear friends Christoph, Danja, Jana, and Julia as well as Charlotte, Dai, Mrunal, and Sai, for the discussions and fun during office times and beyond. Thank you to Anne, Andreas, Ayse, Jana, Julia, and Renate for their invaluable support not only in administrative purposes.

Moreover, I am thankful to all my colleagues at the Chair of Economic Policy and the Chair in E-Business at the University of Magdeburg. Many thanks to Professor Joachim Weimann and Professor Karim Sadrieh for their support, discussions, and laughter. I am so grateful for all the hours spent at the Magdeburg Experimental Laboratory of Economic Research — as a subject, researcher, and lab manager. Thank you to my co-authors, Professor Martin Fochmann and Professor Astrid Dannenberg. Thank you so much for the learning and fun time in the lab to Alex, Dima, Harald, and René, as well as to my colleagues from Kassel, Sven, Elina, and Sonja.

Thank you to my tutor and math teacher, Andrea Hagemann, for sparking an even greater curiosity about mathematics. Thank you also to Henry Haegebarth, Klaus Thormeyer, Martin Wesch, and so many other inspiring teachers and lecturers. Thanks to my fellow students, in particular, my study group and dear friends Alex, Jonathan, and Philipp. Many thanks go to all my close friends for their support, lending a sympathetic ear, and providing distraction whenever needed. Thanks to my proofreaders, Annette, Jana, Sebastian, and Ulli.

Special thanks go to my partner, Ulli, for her support and helpful critique during the final sprint of my doctorate and for making me smile even in moments of disheartenment and strain.

Lastly, I want to express my heartfelt gratitude to my mother for her constant strength and continuous encouragement. Her enduring and unequivocal belief in me and my abilities was always invaluable and an inexhaustible source of power.

Contents

This page intentionally left blank

List of Figures

List of Tables

This page intentionally left blank

List of Algorithms

This page intentionally left blank

List of Abbreviations and Symbols

Abbreviations

ADDX	Advanced DDX decomposition (Burman, 1995)
BAP	Buffer allocation problem
BAP A	Dual buffer allocation problem
BAP B	Primal buffer allocation problem
BAS	Blocking-after-service
BBS	Blocking-before-service
BSAP	Buffer and spare part allocation problem
CBM	Condition-based maintenance
CM	Corrective maintenance
COF	Conservation of flow, equation type of the decomposition approach
CRS	Continuous review systems
DDX	Dallery-David-Xie decomposition (Dallery et al., 1988)
DOM	Design-out maintenance
ENUM	Complete enumeration (brute-force optimization)

FBM	Failure-based maintenance
FRIT	Flow rate idle time, equation type of the decomposition approach
GA	Genetic algorithm
GD	Greedy heuristic with high starting values and decreasing steps
GI	Greedy heuristic with low starting values and increasing steps
GID	Greedy heuristic with low starting values and increasing steps followed by decreasing steps
IOF	Interruption of flow, equation type of the decomposition approach
IP	Inventory position
OBM	Opportunity-based maintenance
ODF	Operation-dependent failures
OEM	Original equipment manufacturer
PM	Preventive maintenance
PRS	Periodic review systems
ROF	Resumption of flow, equation type of the decomposition approach
SAD	Simulated annealing with high starting values
SAI	Simulated annealing with low starting values
SKU	Stock-keeping unit
TDF	Time-dependent failures
TTF	Time to failure

TTR	Time to repair
UBM	Use-based maintenance
WIP	Work-in-process, also known as buffer level

Symbols

AI_i	Average spare part inventory for component i, $i \in \mathcal{I}$
B	Buffer between the machines M_1 and M_2 in the two-machine system
B_j	Buffer between the machines M_j and $M_{j+1}, j \in \mathcal{J}$
BL	Average total extended buffer level
BL_j	Average total extended buffer level of the buffer between M_j and M_{j+1}, $j \in \mathcal{J}$
C	Capacity of buffer B, which is located between the machines M_1 and M_2 in the two-machine system
c	Cost-value ratio for a spare part for the common component with respect to the cost for a workpiece in the system
$\mathbf{C}$	Vector representation of $C_j, j \in \mathcal{J}, \mathbf{C} := (C_1, \ldots, C_J)$
c_1	Cost-value ratio for a spare part for component 1 with respect to the cost for a workpiece in the system
C_j	Capacity of buffer B_j, which is located between the machines M_j and M_{j+1}, $j \in \mathcal{J}$
c_2	Cost-value ratio for a spare part for component 2 with respect to the cost for a workpiece in the system
c_i^{spare}	Costs per unit of component $i, i \in \mathcal{I}$
c_j^{buffer}	Costs per unit of capacity of buffer $B_j, j \in \mathcal{J}$

$C_j^{\max}$	Upper bound for capacity of buffer B_j, $j \in \mathcal{J}$
$C_j^{\min}$	Lower bound for capacity of buffer B_j, $j \in \mathcal{J}$
$C^{\max}$	Upper bound for capacity of buffer B in the two-machine system
C^*	Optimal capacity of buffer B in the discrete-time model
HW	Half-width of the 95% confidence interval of the simulation
$HW(k)$	Half-width of the 95% confidence interval after k simulation runs
$\mathbb{1}(\cdot)$	Indicator function
I	Number of machines
$I(C,S)$	Average number of spare parts for the common component in stock (depending on parameters p, and r and decision variables C, S)
$\mathcal{I}$	Set of machine indices, $\mathcal{I} := \{1, \ldots, I\}$
$I_1(C,S_1,S_2)$	Average number of spare parts for component 1 in stock (depending on parameters p_i, and $r_i, i \in \mathcal{I}$, and decision variables C, S_1, S_2)
$I_2(C,S_1,S_2)$	Average number of spare parts for component 2 in stock (depending on parameters p_i, and $r_i, i \in \mathcal{I}$, and decision variables C, S_1, S_2)
iAV_i	Isolated availability of M_i of the flow line, $i \in \mathcal{I}$
iTP_i	Isolated expected throughput of M_i of the flow line, $i \in \mathcal{I}$
J	Number of buffers, $J := I - 1$
$\mathcal{J}$	Set of buffer indices, $\mathcal{J} := \{1, \ldots, J\} = \{1, \ldots, I-1\}$
$\mathcal{K}$	Set of combined indices for buffers and spare part base-stock

	levels, $\mathcal{K} := \{1, \ldots, 2 \cdot I - 1\}$
$M_a(l)$	Aggregated machine of aggregation step $l \geq 1$
$M_d(l)$	Upstream machine of the virtual line $l \geq 1$ from the decomposition approach
M_i	Machine $i, i \in \mathcal{I}$
$M_u(l)$	Upstream machine of the virtual line $l \geq 1$ from the decomposition approach
C	Extended capacity of buffer B, which is located between the machines M_1 and M_2 in the two-machine system, $N := C + 2$
$\mathbb{N}$	Natural numbers, $\mathbb{N} = \{1, 2, 3, \ldots\}$
$\mathbb{N}_0$	Natural numbers including zeros, $\mathbb{N}_0 = \{0, 1, 2, 3, \ldots\}$
N_j	Extended capacity of buffer B_j; equals the maximum number of workpieces completely processed by machine M_j and not already completely processed by machine $M_{j+1}, j \in \mathcal{J}, N_j = C_j + 2$
n_j	Number of workpieces completely processed by machine M_j and not already completely processed by machine $M_{j+1}, j \in \mathcal{J}$
$\mathcal{P}(\cdot, \cdot, \cdot)$	Steady-state probabilities in the continuous-time model with two machines and machine-specific critical components (depending on parameters μ_i, λ_i, and $\gamma_i, i \in \mathcal{I}$, and decision variables C, Q_1, Q_2)
$p(\cdot)$	Steady-state probability of a state ξ
$\mathcal{P}_D(\cdot, \cdot)$	Steady-state probabilities in the discrete-time model with identical critical component (depending on parameters p, and r and decision variables C, S)
$\mathcal{P}_D(\cdot, \cdot, \cdot)$	Steady-state probabilities in the discrete-time model with machine-specific critical components (depending on parameters

	p_i, and $r_i, i \in \mathcal{I}$, and decision variables C, S_1, S_2)
$\mathcal{P}_i(\cdot,\cdot,\cdot)$	Steady-state probabilities of the virtual line i from the decomposition approach, $i \in \mathcal{I}$
p_i	Failure probability of component i in the discrete-time model, component lifetimes are geometrically distributed, $i \in \mathcal{I}$
$PTC(\mathbf{C}, \mathbf{Q})$	Penalized total costs of the flow line design (depending on $TC(\mathbf{C}, \mathbf{Q})$ and penalizing term)
$\mathbf{Q}$	Vector representation of $Q_i, i \in \mathcal{I}, \mathbf{Q} := (Q_1, \ldots, Q_I)$
Q_i	Maximum number of units of component $i, i \in \mathcal{I}, Q_i = S_i + 1$
Q_d	Maximum number of units of the component of the downstream machine in the continuous-time model with two machines and machine-specific critical components, $i \in \mathcal{I}$
$Q_i^{\max}$	Upper bound for the number of units of component i, $i \in \mathcal{I}$
$Q_i^{\min}$	Lower bound for the number of units of component i, $i \in \mathcal{I}$
Q_u	Maximum number of units of the component of the upstream machine in the continuous-time model with two machines and machine-specific critical components, $i \in \mathcal{I}$
r_i	Replenishment probability for units of component i in the discrete-time model, replenishment times are geometrically distributed, $i \in \mathcal{I}$
S	Base-stock level for units of the common component
$S^2(k)$	Sample variance after k simulation runs
S_1^*	Optimal base-stock level for component 1 in the discrete-time model with machine-specific critical components
S_2^*	Optimal base-stock level for component 2 in the discrete-time model with machine-specific critical components

S_i	Base-stock level for units of component $i, i \in \mathcal{I}$
$S^{\max}$	Upper bound for base-stock levels in the two-machine system
$\overline{SS}$	State space in the discrete-time model with machine-specific critical components
SS	State space in the discrete-time model with identical critical components
S^*	Optimal base-stock level in the discrete-time model with identical critical components
$T(n)$	Temperature used for simulated annealing in iteration $n \in \mathbb{N}$
TBC	Total buffer capacity for a given system design
$TC(\mathbf{C}, \mathbf{Q})$	Total costs of the flow line design (depending on parameters $c_j^{\text{buffer}}, j \in \mathcal{J}$, and $c_j^{\text{spare}}, i \in \mathcal{I}$, and decision variables $\mathbf{C}, \mathbf{Q}$)
$TC_D(C, S)$	Total costs of the flow line design (depending on parameter c and decision variables C, S)
$TC_D(C, S_1, S_2)$	Total costs of the flow line design (depending on parameters $c1$ and c_2 and decision variables C, S_1, S_2)
TBU	Total number of component units for a given system design
TP	Expected throughput of the flow line (depending on parameters μ_i, λ_i, and γ_i and decision variables $C_j, j \in \mathcal{J}$, and $Q_i, i \in \mathcal{I}$)
$TP(\mathbf{C}, \mathbf{Q})$	Expected throughput of the flow line (depending on parameters μ_i, λ_i, and $\gamma_i, i \in \mathcal{I}$, and decision variables $\mathbf{C}, \mathbf{Q}$)
TP^T	Target throughput of the flow line
$TP_D(C, S)$	Expected throughput of the flow line (depending on parameters p, and r and decision variables C, S)
$TP_D(C, S_1, S_2)$	Expected throughput of the flow line (depending on parameters p_i, and $r_i, i \in \mathcal{I}$, and decision variables C, S_1, S_2)

TP_{dec}	Expected throughput of the flow line obtained by decomposition
TP_{exact}	Expected throughput of the flow line obtained by the Markov approach
TP_i	Expected throughput of M_i in the original the flow line, $i \in \mathcal{I}$
$TP(i)$	Expected throughput of the virtual line i from the decomposition approach, $i \in \mathcal{I}$
TP_{sim}	Expected throughput of the flow line obtained by simulation
$WIP(C, S)$	Average work-in-process inventory (depending on parameters p, and r and decision variables C, S)
$WIP(C, S_1, S_2)$	Average work-in-process inventory (depending on parameters p_i, and $r_i, i \in \mathcal{I}$, and decision variables C, S_1, S_2)
$\overline{X}(k)$	Sample mean after k simulation runs
α_d	Number of units of the component of the downstream machine in the continuous-time model with two machines and machine-specific critical components, $i \in \mathcal{I}$
α_i	Number of functional units of component $i \in \mathcal{I}$
α_u	Number of units of the component of the upstream machine in the continuous-time model with two machines and machine-specific critical components, $i \in \mathcal{I}$
Δ_{costs}	Relative deviation of the best system design's total costs obtained by two different algorithms
Δ_{dec}	Deviation of the throughput obtained by decomposition compared to Markov approach
$\Delta_{\text{dec}}^{\text{rel}}$	Relative deviation of the throughput obtained by decomposition compared to Markov approach
$\Delta_{\text{dec/sim}}$	Deviation of the throughput obtained by decomposition com-

	pared to simulation
$\Delta^{\mathsf{rel}}_{\mathsf{dec/sim}}$	Relative deviation of the throughput obtained by decomposition compared to simulation
Δ_{sim}	Deviation of the throughput obtained by simulation compared to Markov approach
$\Delta^{\mathsf{rel}}_{\mathsf{sim}}$	Relative deviation of the throughput obtained by simulation compared to Markov approach
$\Delta_{TP}(p,r)$	Impact of first spare part on throughput depending on p and r
$\tilde{\Delta}_{TP}(p,r)$	Impact of additional buffer capacity on throughput depending on p and r
Δ^{rel}_{TP}	Relative throughput deviation between the exact Markov results with geometrically distributed lead times and the simulated results in the discrete-time model with machine-specific critical components
ε	Desired accuracy of the decomposition approach.
γ_i	Replenishment rate for units of component i in the continuous-time model, replenishment times are exponentially distributed, $i \in \mathcal{I}$
$\gamma_d(i)$	Replenishment rate for units of the component of the downstream machine of the virtual line i from the decomposition approach, $i \in \mathcal{I}$
$\gamma_u(i)$	Replenishment rate for units of the component of the upstream machine of the virtual line i from the decomposition approach, $i \in \mathcal{I}$
λ_i	Failure rate of component i in the continuous-time model, component lifetimes are exponentially distributed, $i \in \mathcal{I}$
$\lambda_d(i)$	Failure rate of the component of the downstream machine of the virtual line i from the decomposition approach, $i \in \mathcal{I}$

$\lambda_u(i)$	Failure rate of the component of the upstream machine of the virtual line i from the decomposition approach, $i \in \mathcal{I}$
μ_i	Processing rate of machine i in the continuous-time model, processing times are exponentially distributed, $i \in \mathcal{I}$
$\mu_d(i)$	Processing rate of the downstream machine of the virtual line i from the decomposition approach, $i \in \mathcal{I}$
$\mu_u(i)$	Processing rate of the upstream machine of the virtual line i from the decomposition approach, $i \in \mathcal{I}$
ξ	State of the state space for a flow line of I machines, $(2 \cdot I - 1)$-tuple

Chapter

1 Introduction

Coffee: the finest organic suspension ever devised.

Captain Kathryn Janeway

Although it has been the focus of researchers for more than 60 years, the design of stochastic manufacturing systems still raises unanswered questions. Koenigsberg (1959) reviewed the fundamental problems arising from internal storage in production lines. He advocated for putting more effort into the scientific analysis of production systems and their effectiveness. We can observe these efforts via the vast amount of literature published on evaluating and designing manufacturing systems. Recently, Gershwin (2018) stated that new technologies will always cause the need for new solutions. Specifically, reducing random effects, their impact, and the propagation of these influences are the main issues for further research.

A variety of sources for stochasticity exists. The main problem in stochastic manufacturing systems is linked to breakdowns, which typically prevent the failed machine from further production steps or, occasionally, negatively impact the production quality. The time between such events (time to failure, TTF) is often assumed to be random as a breakdown can be caused by many incidents: a malfunctioning piece of equipment, an operator error, or likewise. Similarly, the time to repair (TTR) is also typically assumed to be uncertain since, e.g., repair capacity may be limited or the time for troubleshooting is just problem-dependent. Another source of uncertainty is variability in processing times due to differences in raw-material quality, the time a human operator needs at a specific station, or similar. These various random effects make the analysis of stochastic manufacturing systems even more challenging.

The necessity of analyzing manufacturing does not only come from changes in technology or known sources of uncertainty. Another driver is the enormous amount of associated costs with its planning and operation. Liberopoulos (2018) analyzed data from a manufacturing body shop. He reported that each unit of buffer capacity could require an investment of $\$5,000$–$\$10,000$ and that additional holding costs of around $\$500$ per year per workpiece accrue. For this example and many more production systems, buffers are inevitable in order to ensure high throughput.

These interstage buffers, on the one hand, help to reduce the impact of processing time variability. On the other hand, they only cope indirectly with machine downtime. If a machine fails, accumulated workpieces in a downstream buffer allow subsequent machines to continue operations; concurrently, an empty upstream buffer enables antecedent machines to keep up processing for a limited time. The higher the buffer capacities, the more work-in-process (WIP) can be stored. But higher WIP inventories are equivalent to a high amount of tied-up capital, and the failed machine still does not produce any output.

We adumbrated how buffers can tackle downtime of the individual machines. However, another strategy is often used in practice: Spare parts are held in stock in order to ensure fast repairs and increase the machines' availability. For instance, Lamghari-Idrissi (2021) reported downtime costs of up to €$72,000$ per hour for a front-end wafer fabrication process that incorporates machines of ASML, a Dutch original equipment manufacturer (OEM). Companies are investing millions of dollars in spare parts (Basten & Van Houtum, 2014), with a single part costing up to thousands of dollars (Cohen et al., 1997). These investments are reasonable since enhancing the machines' availability can be beneficial. Smets et al. (2012) indicated that increasing the system availability by one percentage point could lead to savings of €$12,000$ per week.

In contrast to buffers, spare parts can directly deal with machine failures. If a machine suffers from breakdowns of a critical component, ready-to-use spare parts in stock allow for fast repairs by implementing a repair-by-replacement strategy. In doing so, additional spare parts help to reduce machine downtime by avoiding the necessary replenishment time. In contrast, when using spare parts, the question arises of how many units should be kept in stock and how the replenishment should be organized.

These considerations reinforce that planners should bear in mind buffers and spare parts when planning an unreliable manufacturing system. Moreover, we identify a lack of models in the scientific literature covering both aspects. Models for separate planning of buffers (see, e.g., Weiss et al., 2019) and spare parts (see, e.g., Basten & Van Houtum, 2014) exist. However, these models differ drastically in their main assumptions. For simultaneously planning a manufacturing system with both buffers and spare parts, a decision-maker could, until now, only rely on rules of thumb derived from anecdotal knowledge or more or less closely connected models.

The literature mainly considers two classes of approaches for spare parts: item-oriented and system-oriented, both of which regard a target service level. Item-oriented approaches strive to reach a target service level for each stock-keeping unit (SKU), causing high costs compared to system-oriented methods. A system-oriented service level considers the target service level together for all SKUs. Analyzing on a system level was introduced by Sherbrooke (1968). A possible advantage of system-level approaches is that inventories of expensive SKUs can be reduced by increasing inventories of cheaper SKUs. In doing so, planners can achieve a higher service level of the system, reduce costs, or a combination of both compared to item-level approaches. Unfortunately, these methods are not conferrable because they cannot factor in the influence of the buffers.

Building on these ideas, we develop several new mathematical models of manufacturing systems. In doing so, we strive to build a common ground of assumptions that make our solutions applicable to the area of flow-line planning. Specifically, this work focuses on flow lines where processing steps must be carried out in a specific order on sequentially arranged machines. Between adjacent machines, interstage buffers store WIP to decouple the production processes. Each of these machines contains one unit of a critical but failure-prone component. The stock-holding is organized with a one-for-one replenishment policy. Kiesmüller and Zimmermann (2018) were the first to consider this kind of setup in a two-machine system where both machines contain the same critical component and hence just one spare-part stock is used. They illustrated and quantified the possible benefits of simultaneous buffer capacity and base-stock level optimization.

1.1 Terminology

In order to point out the aim of this thesis, we discuss the relevant terminology next. We start with focusing on the flow-line model and therefore rely on the review of Dallery and Gershwin (1992).

The subject of consideration of this thesis is *manufacturing flow lines*. We use the terms *manufacturing systems*, *production systems*, *flow lines*, *transfer lines*, or simply *lines* synonymously. The flow lines we consider consist of *machines*, also called *stations*. We label the storage areas between these machines as *interstage buffers*, *intermediate buffers*, or just *buffers*. The aim of evaluating flow lines is to calculate the maximum rate of flow of material through the production system in the long run, typically referred to as *production rate*, or *throughput*.

A typical representation of a four-machine flow line with three interstage buffers is depicted in Figure 1. Important performance measures of flow lines include mainly throughput and WIP inventories, but also probabilities for starving and blocking.

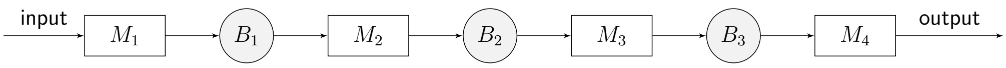


Figure 1. Illustration of a four-machine flow line

Besides machine failures leading to production stops, two sources of non-operating machines exist: *starving* and *blocking*. A machine is *starving* if its production process finishes and the preceding buffer is empty. In this case, the machine lacks another workpiece and suspends operation. If the subsequent buffer of a machine is already full and the machine finishes processing, this machine gets *blocked*. This specific case is labeled *blocking-after-service* (BAS) in contrast to *blocking-before-service* (BBS) where the machine is *blocked* before it starts the processing of a new workpiece. The latter is used, e.g., for perishable products in the food industry. In general, buffers facilitate the decoupling of production processes and reduce starvation and blocking probabilities.

Most flow-line models analyze *unreliable* machines where the machines suffer from any breakdown. In contrast, machines can also be assumed to be *reliable*. In the case of *unreliable* machines, the literature differentiates between *operation-dependent failures* (ODF) and *time-dependent failures* (TDF). ODF means a machine cannot fail while idle due to blocking or starving; this is not valid for TDF.

In order to operate, the first machine of a flow line needs raw material. Additionally, the last machine's output needs to be stored. Most publications consider *saturated* systems where the supply is guaranteed and final products can always be kept. However, there are plenty of reasons why the supply of raw materials or the demand for the final product may be disrupted. An *unsaturated* system covers these sources of uncertainty explicitly. Saturated systems allow focusing on the production process in isolation in order to compute the best-case throughput. Furthermore, it is possible to think of a flow line's first machine's production process as the arrival process and that the last machine is equivalent to the product's demand process. Thereby, a model of a saturated system can describe the unsaturated variant by adding two additional machines if the assumptions for the supply and demand process are suitable.

The models under consideration can be *discrete* or *continuous* regarding time and parts. *Continuous-time* models assume that time goes by continuously. Models assuming *discrete time* are based upon time steps being days, hours, the length of a production cycle, or similar. Flow-line models typically assume that raw material, WIP, and end product are *discrete parts*. When modeling production steps for liquids, continuous-flow models are appropriate. They can, however, also be used to approximate discrete-part systems because a continuous flow of material may be advantageous for optimization.

Manufacturing systems can work *synchronously*. That is, the machines start and stop operations at the same time. However, most flow lines perform their production steps *asynchronously* because their processing times differ, e.g., they are random, and buffers decouple their production processes. Even if the machines' processing times are deterministic and identical, there is no need for synchronous operations if buffers are present.

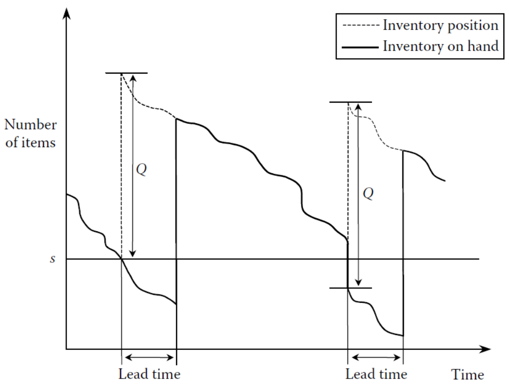
In a nutshell, this thesis focuses on asynchronous, saturated, unreliable flow lines with ODF. We apply BAS as the blocking policy and assume discrete parts. The first model we study operates in continuous time, whereas the second one is a discrete-time model. We highlight possible maintenance strategies to know how to cope with the flow line's unreliability.

Pintelon and Gelders (1992) stated that maintenance management is crucial for industrial equipment in general and that there are five basic maintenance policies: failure- (FBM), use- (UBM), condition- (CBM), and opportunity-based maintenance (OBM), as well as design-out maintenance (DOM). FBM, also known as corrective maintenance (CM), takes place after a breakdown. If more information on the failure behavior is available, UBM can be carried out. It may be performed after a specific number of processed units or after a fixed time interval. When it is possible to monitor the machine state using sensors or other methods, planners can implement CBM. After reaching a predefined machine state, the machine undergoes maintenance. OBM means that maintenance activities are carried out when a chance arises. This may be a failure of a machine component presenting the opportunity to revise other components of the same machine as well or maintenance during blocking or starving. UBM and CBM are often summarized as preventive maintenance (PM). OBM requires multiple different down-states of the machines, which most publications do not consider. DOM aims to improve equipment to increase reliability or ease maintenance operations. In this thesis, OBM and DOM will not be considered further.

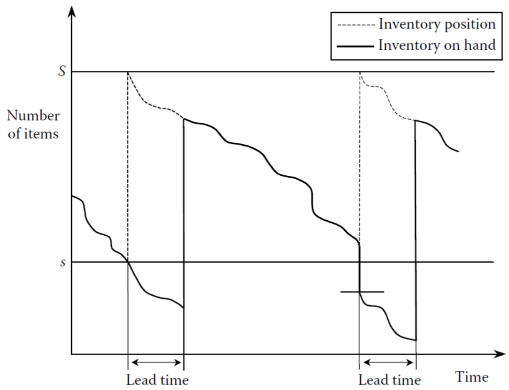
When using spare parts for any kind of maintenance, we need to consider the question of how to replenish them. According to Blumenfeld (2001), the basic replenishment policies are either continuous (CRS) or periodic review systems (PRS). Placing an order depends on the inventory position (IP), which is defined as stock on hand plus the number of ordered items. We need four quantities to understand the replenishment policies: the order quantity Q, the reorder point s, the base-stock level S (also known as order-up-to level), and the review period T, which is the time interval between reviewing the inventory. CRS do not depend on a review period but check the inventory position permanently. For CRS, there are the (s, Q) and the (s, S) policy. An order is placed whenever the IP reaches or undercuts the reorder point s. For the (s, Q) policy, this order always equals the order quantity Q, whereas the order quantity may differ in the (s, S) policy because we always order enough so that our IP equals S units. Similarly, we have two common policies for PRS: the (T, S) and the (T, s, S) policy. Here, we review the IP after time intervals of length T. When applying the (T, S) policy, we always order enough to bring the IP up to the given level S. In contrast, we only place an order when reaching or undercutting the reorder point s if we apply a (T, s, S) policy. Figure 2 depicts the application of the four different replenishment policies.

The literature is not consistent with labeling the variables for these different policies.

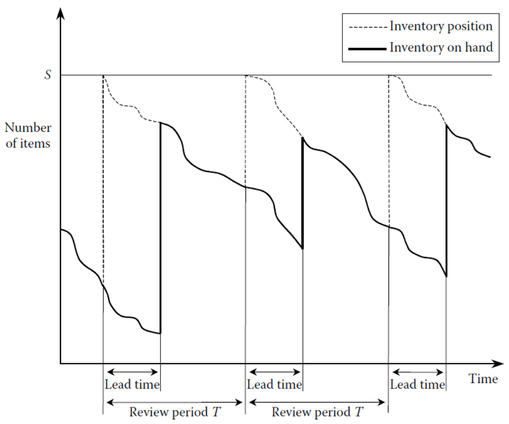
Furthermore, despite their categorization, we can recognize strong connections between the different policies. For instance, the (T, S) policy is a special case of the (T, s, S) policy when $s = S$. If the fixed ordering costs are low in comparison to the holdings costs — which is often the case for spare parts —, a $(S - 1, S)$ is typically applied (Basten & Van Houtum, 2014; Moinzadeh & Lee, 1986; Muckstadt, 2005). Since for each used unit, a new one is directly ordered, this is also called a one-for-one replenishment policy. Since the assumption of a $(S - 1, S)$ policy is reasonable for spare parts, we will not consider the others.



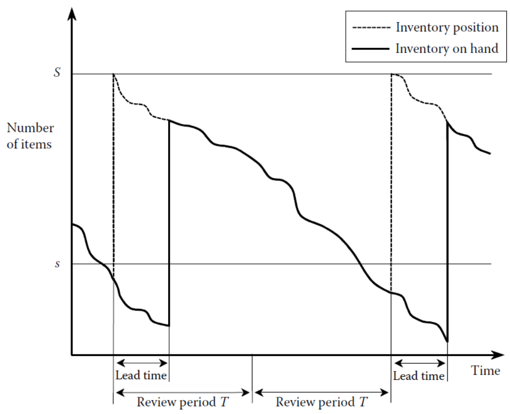
(a) (s, Q) policy



(b) (s, S) policy



(c) (T, S) policy



(d) (T, s, S) policy

Figure 2. Inventory replenishment policies (Blumenfeld, 2001)

1.2 Contribution and methodology

Maintenance activities for manufacturing systems can be included in different ways. With the increasing availability of historical data, planners can schedule UBM more reliably.

Kyriakidis and Dimitrakos (2006) and Meller and Kim (1996) analyzed production systems with UBM and buffers, but there are many more. Another possibility involves sensors that monitor the state of a machine in order to enable CBM (see, e.g., Fitouhi et al., 2017). Both maintenance activities can profit from spare part availability (Wang, 2002).

However, not all types of maintenance operations can be carried out in advance. Even with historical and sensor data, there is still the possibility of unexpected breakdowns and the resulting necessity for CM. These maintenance activities were typically assumed to be carried out by repair persons, including a variety of repair time distributions. Nevertheless, there are plenty of cases where breakdowns cannot be fixed just by an on-site repair since a spare part is required. In this case, the need for spare part replenishment would drastically increase the overall repair time, diminish the machine's availability, and, eventually, the flow line's throughput.

Consequently, the main objective of this thesis is to generate new insights into the integrated design of manufacturing systems regarding both buffers with limited capacities and spare part stocks. This constitutes the overall research question:

> *How should a flow line with buffers and spare parts provisioning for corrective maintenance be designed?*

We aspire to tackle this problem with newly developed mathematical models. We start with a two-machine system consisting of one buffer and two spare part stocks and use it as a building block to analyze systems of arbitrary length (Chapter 3). We model the two-machine line as a continuous-time Markov chain, enabling us to determine different performance measures. Based on this, we develop an approximative decomposition approach extending the general idea proposed by Gershwin (1987). We use discrete event simulation to validate this new evaluative method, as its trustiness is vital for answering the raised design question. In the next step, we apply different heuristics to find efficient solutions (Chapter 4). We use two greedy-type heuristics and two meta-heuristics (simulated annealing and a genetic algorithm). We compare their results for small systems to optimal solutions obtained by complete enumeration in order to emphasize their applicability. Finally, we develop two other two-machine models to tackle some of the critical assumptions (Chapter 5). These systems are modeled as discrete-time Markov chains. The corresponding optimization problem uses complete enumeration to find optimal system designs.

Using these methods, we continue with a clear distinction between the underlying research questions.

1.3 Research questions and outline

We introduce a new model of a flow line of arbitrary length where each machine has a machine-specific failure-prone critical component. Based on this model, we review the relevant literature in Chapter 2. Our review shows that available methods cannot determine system characteristics for flow lines with buffers and spare parts. Hence, our first question arises:

Research Question 1. *What is the throughput of a flow line consisting of an arbitrary number of machines with limited buffer capacities and spare part stocks?*

In Chapter 3, we present a mathematical model for flow lines of arbitrary length. We deduce how to evaluate small systems consisting of two or three machines exactly and derive a decomposition approach to evaluate longer lines approximately. In doing so, we target answering Research Question 1. This chapter is published as Sachs et al. (2022a). With an evaluative procedure at hand, we can start by analyzing the system design:

Research Question 2. *What is the optimal flow-line design regarding buffer capacities and spare part base-stock levels? How do both decisions interact?*

Chapter 4 aims at answering Research Question 2. We introduce the buffer and spare part allocation problem; it comprises the trade-off between both to reach the target throughput. An extensive numerical study illustrates how different system and cost parameters change the best-found allocation. The results in this chapter constitute the working paper Sachs et al. (2022b).

To emphasize the applicability of our results, we relax the assumption of exponentially distributed processing times and consider fixed cycle times. Such a situation is more closely connected to practical settings because the exponential distribution comes with relatively high variability. In contrast, practitioners try to synchronize the processing times of different production steps if possible. This results in the following question:

Research Question 3. *Do the results of systems with stochastic processing times carry over to transfer lines with deterministic cycle times?*

Standardization is known to reduce costs in general and can also be applied in the context of manufacturing systems. Technical appliances, e.g., machines or production robots, consist of components where some of the components might be identical. Systems with commonality in spare parts or machines were observed by Kahan et al. (2009) and Kranenburg and Van Houtum (2007). A setup with identical critical components could also be the organizational goal if cost savings are realizable. These thoughts lead to the following question:

Research Question 4. *What is the impact of component standardization comparing a two-machine system with machine-specific failure-prone critical components to a system with identical ones?*

In Chapter 5, we strive to answer Research Questions 3 and 4 by studying a two-machine transfer line with constant cycle times. In doing so, we can eliminate the influence of processing time variability and focus on the effect of machine downtime. This analysis broadens the view of our results since a high processing time variability is rarely observed in practice. These benefits, however, come at the cost of a higher complexity of the solution approach since we have to rely on methods in discrete time. In addition, we compare two models: one with distinct critical components and two stock points, and the other one with identical critical components and, thus, one stock point. We show that spare part commonality enables a pooling effect, resulting in reduced spare part base-stock levels, decreased buffer capacities, or both — depending on the system specifics. This chapter is based on the publication by Kiesmüller and Sachs (2020).

Lastly, Chapter 6 concludes with the main insights of our research. Moreover, we discuss some general remarks, limitations, and future research opportunities.

Chapter

2 Literature Review

It is the struggle itself that is most important. We must strive to be more than we are. It does not matter that we will never reach our ultimate goal. The effort yields its own rewards.

Lieutenant Commander Data

The bedrock of this work is the scientific analysis of stochastic production systems, which Koenigsberg (1959) advocated for when he first described the buffer allocation problem (BAP). Building upon that, an extensive body of research developed. This literature review focuses on four distinct yet interrelated branches: evaluating manufacturing systems, the BAP, spare parts management, and maintenance for manufacturing systems. Each of these branches is well developed; hence, we will only focus on the most crucial contributions in order to embed and understand the questions handled in this thesis.

2.1 Evaluating manufacturing systems

For a detailed overview of existing models and methods, we refer to the reviews by Dallery and Gershwin (1992) on manufacturing systems under various assumptions, Papadopoulos and Heavey (1996) for queueing-based approaches, Li et al. (2009) regarding throughput-analysis methods for different settings, and Papadopoulos et al. (2019) for a review of Markov models. Another fruitful reading is the monograph of Gershwin (1994) that provides a deep dive into many models and how to obtain evaluative solutions.

Many publications focus on how to assess the throughput of a manufacturing system. In their review, Li et al. (2006) pointed out that the analysis of two-machine systems is still crucial as it constitutes the building block for any further drill-down of larger systems. Hence, we will summarize some essential contributions on how to obtain characteristics, like throughput or average WIP inventories, of a short flow line consisting of two or three machines.

2.1.1 Evaluation of short flow lines

The following contributions have in common that they assume CM after breakdowns since this is the elementary approach. Section 2.4 provides an overview of more sophisticated maintenance policies for manufacturing systems.

Buzacott (1967) proposed a general view of production system effectiveness by using a discrete-time Markov chain to compute characteristics of two- and thee-machine systems. He examined the limiting cases where either no or unlimited buffer capacity is available and inferred that buffers are most effective if the throughput in both cases differs substantially. Okamura and Yamashina (1977) extended Buzacott's model to generate further insights into the effects of buffers and their actual usage for different parameter scenarios. Among others, they found that differences in breakdown rates reduce the positive influence of buffer installation, which does not hold for repair rates. In the same vein, Buzacott and Hanifin (1978) discussed different system designs and compared simulated to analytic results. It turned out that the former suffered from inaccuracies such that Buzacott and Hanifin interceded for further development of analytical methods and careful validations of simulated results.

Some effort was devoted to algorithmic improvements (see, e.g., Buzacott & Kostelski, 1987) or different (processing time) distributions like Coxian-2 (Vidalis & Papadopoulos, 1999) or general phase-type (Colledani & Tolio, 2009). Furthermore, the possible setup broadened. Shanthikumar and Tien (1983) introduced scrapping of currently processed units in case of a machine failure, which is relevant in the food industry when working with perishable products. The flow of material could also be continuous instead of handling discrete parts (Tan & Gershwin, 2009; Wijngaard, 1979). We refer to different contributions on new models like Kim and Gershwin (2005) regarding quality failures in processing steps, Gebennini and Gershwin (2013) for waste production, Gebennini et al. (2013) considering a restart policy where the production of a blocked machine is halted until the buffer is empty again, or Gebennini and Grassi (2015) regarding a buffer-bypass when both machines are functional as well as a constrained production if one machine is down. Other advancement opportunities are different objectives such as incorporating energy consumption or workforce constraints (Su et al., 2016) or state-dependent policies where, e.g., the buffer inventory level determines the system operations (Gebennini et al., 2017).

After glancing at the general models, we want to highlight how prominent models of unreliable production systems developed. Again, Buzacott's (1967) model of a two-machine system, including one interstage buffer, laid the basis for the scientific analysis and became the reference point for further contributions. He assumed deterministic processing times and geometric failure and repair time distributions. More critically, Buzacott required that the probability of two or more machines failing in the same cycle is neglectable. Gershwin and Schick (1983) relaxed this assumption, changed the order of events in one cycle, and presented Markov chain approaches for two- and three-machine lines. Similarly, Gershwin and Berman (1981) used a Markov process model to describe the behavior of a two-machine flow line with exponentially distributed processing times. They further assumed blocking before service and that machines stop processing if the preceding buffer runs empty, leaving one part unprocessed. In using the exponential distribution, they increased the mathematical tractability and were able to contrive a closed-form solution of the Markov chain's steady-state probabilities. Hong and Seong (1993) proposed another model and an efficient Markov chain-based solution algorithm relying on blocking after service (in contrast to Gershwin and Berman's model). Some publications analyzed models with multiple failure modes where machines can fail in more than one way (Colledani & Tolio, 2009; Tolio et al., 2002), the contemporaneous occurrence of operation- and time-dependent failures (Matta & Simone, 2016), or multiple up- and down-states including buffer thresholds (Tolio & Ratti, 2018).

We mainly rely on discrete- and continuous-time Markov chain approaches for small flow lines consisting of two or three machines. The available computational power allows for a fast determination of steady-state probabilities by numerically solving the system of balance equations. Our results indicate that a closed-form formula would require computing the roots of a high-degree polynomial, rendering this approach unfavorable. Furthermore, we apply a discrete-event simulation to inspect the influence of arbitrary distributions for spare part replenishment.

2.1.2 Evaluation of longer flow lines

Hillier and Boling (1967) used a continuous-time Markov chain to model a manufacturing system of I machines with exponentially (or Erlang) distributed service times needing the solution of a linear equation system. Using queuing theory, they developed a more efficient algorithm to approximate the system throughput. The exact Markov approaches, suitable for small systems, cannot be used for longer flow lines because the state space explodes and renders the solution incomputable. There are some approaches to trying to overcome this issue. Papadopoulos et al. (1989) used successive overrelaxation to compute steady-state probabilities of a Markov chain model for a reliable manufacturing system with Erlang distributed processing times and buffers. Heavey et al. (1993) extended this approach to unreliable lines. However, they achieved only solutions for five- and six-machine systems, respectively, and relatively limited buffer sizes. Hence, the need for another method becomes obvious.

There are two approximative approaches for analyzing unreliable flow lines of arbitrary length: aggregation and decomposition. Additionally, one could, of course, conduct a simulation study. However, simulations are computationally expensive and unsuitable for optimization problems where many evaluations need to be done (Gershwin & Schor, 2000; Helber et al., 2011). Another possibility is the usage of artificial neural networks to allow for fast evaluations (Altiparmak et al., 2002; Südbeck et al., 2022). However, these networks are still in need of an evaluative procedure for training. These are the main reasons why approximative methods are so important. The main ideas of both approaches were developed in the 1960s, but it took until the 1980s when efficient algorithms were published for manufacturing systems (Li et al., 2009) for them to be implemented.

Aggregation approaches combine two neighboring machines (including their interstage buffer if applicable) to a virtual machine and reduce the number of machines in a flow line by one per step. Accordingly, this procedure is repeated until a two-machine line is obtained, which can then be evaluated using a Markov chain or similar. The order of this aggregation can be arbitrarily chosen. De Koster (1987) was one of the first to use this algorithmic procedure. He started with aggregating the first two stations and continued in this direction until the end of the flow line. Figure 3 illustrates this approach. Jafari and Shanthikumar (1987) refined this procedure. However, it can be further improved by iteratively repeating the aggregation a) starting from the first machine to the end of the line in a forward pass and b) starting with the last machine stepwise to the start of the line in a backward pass (Jacobs & Meerkov, 1995a, 1995b, 1995c; Lim et al., 1990). This improved version has been proven to be convergent, and it has been shown that several system-theoretic properties hold. However, the accuracy of the aggregation still does not match that of the decomposition in all cases (Li et al., 2009). Therefore, improvements to the aggregation algorithm continue to be the subject of current research (Bai et al., 2020).

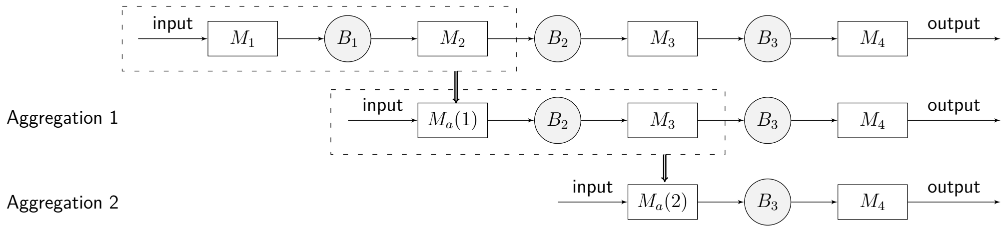


Figure 3. Illustration of the aggregation approach for a four-machine flow line. $M_a(l)$ describes the aggregated machine in step l.

The decomposition approach relies on the idea that a I-machine line is decomposed into a sequence of $I-1$ virtual two-machine-one-buffer systems. Gershwin (1987) published the seminal paper and established a system of equations ensuring that the flow of material in the two-machine lines mimics the behavior of the original system. All buffer capacities of the virtual systems are identical to the original buffers, and the equation system determines each virtual machine's set of parameters. Figure 4 depicts the general procedure of the decomposition approach. Again, the actual algorithm is based on iterative repetitions where the decomposition happens a) in a forward pass starting from the first machine and b) starting with the last machine in a backward pass. To the best of our knowledge, no proof of convergence for this class of approximations exists. However, there is a large body of literature on algorithms and models which deliver accurate results and provide valuable insights.

Gershwin (1987) started the analysis of flow lines with deterministic processing times. Although Gershwin's original algorithm converged in many cases, it revealed some numerical weaknesses. Hence, Dallery et al. (1988) modified the equation system leading to the Dallery-David-Xie (DDX) algorithm. Some further improvements came from Dallery et al. (1989), including a course of action to reduce inhomogeneous flow lines with processing-time differences to homogeneous complements. Eventually, Burman (1995) deduced the accelerated DDX (ADDX) algorithm, relying on a closed-form solution of the decomposition equations. His efforts led to a numerically very stable and fast evaluation of flow lines with a convergence rate of about 99.9%. However, the

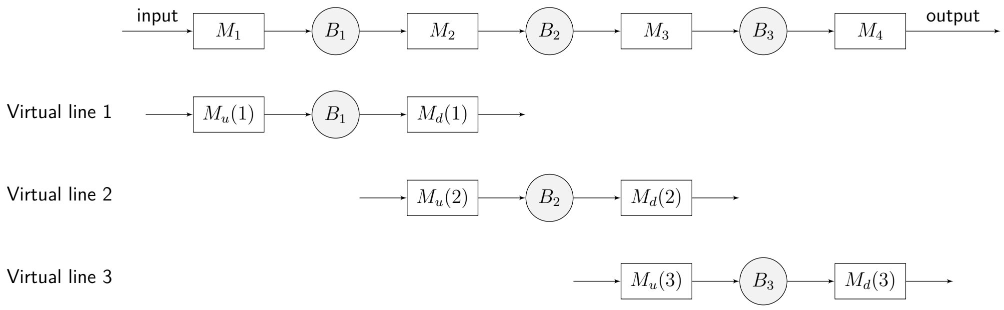


Figure 4. Illustration of the decomposition approach for a four-machine flow line. $M_u(l)$ ($M_d(l)$) describes the upstream (downstream) machine in the virtual line l.

main intricacy of the decomposition is that a new set of equations needs to be derived for each different model. The results do not simply carry over.

Besides the algorithmic improvements, decomposition approaches were derived for other sets of assumptions. Choong and Gershwin (1987) derived an analysis for exponentially distributed processing times. To cover more general repair time distributions, Dallery (1999) and Le Bihan and Dallery (2000) applied two- and three-moment fits. The decomposition was also applied to assembly-disassembly systems: for fixed processing times (Gershwin, 1991; Liu & Buzacott, 1990; Mascolo et al., 1991), random processing times (Helber, 1998, 1999), split in the material flow (Helber, 2000), merges in material flow (Helber & Jusić, 2004), and reworking and scrapping of workpieces (Li, 2005).

We consider two approaches when it comes to evaluating longer flow lines. Based on the general idea of the decomposition (Gershwin, 1987), we develop a new procedure to evaluate flow lines with buffers and spare parts. Furthermore, we use a discrete-event simulation in order to validate the results obtained by our decomposition.

2.2 Design of flow lines and the buffer allocation problem

We refer to three recent reviews for a detailed overview of cost-efficient flow-line designs with a focus on the BAP. Demir et al. (2014) summarized and structured the existing literature, Hudson et al. (2015) focused on results for unbalanced flow lines, and Weiss et al. (2019) conducted a comprehensive review of the BAP. Moreover, Weiss et al. (2019) provided a general survey of the literature on flow lines with detailed overviews of modeling assumptions. They further introduced a differentiation of solution techniques. In the following, we briefly discuss important methods for and insights into the BAP.

The literature on the BAP consists of many different problem formulations. These could be differences in the objective function or regarding the constraint of the optimization problem. Weiss et al. (2019) identified two main problem formulations which are altered to different extents:

- the primal BAP (also called BAP B) aims to minimize costs subject to the condition that the expected target throughput reaches a specific threshold, and
- the dual BAP (also called BAP A) aims to maximize the expected target throughput subject to the condition that the total buffer capacity equals a specific value.

We adhere to the notification and nomenclature by Weiss et al. (2019).

Since exact evaluations of flow lines are rare, exact analytical results for the BAP are even more challenging to obtain. Enginarlar et al. (2002) used a closed-form expression for the throughput and were able to derive an explicit formula for the optimal buffer capacity of an unreliable two-machine flow line. Other analytical results are based on the approximative evaluation of flow lines and using regression (Martin, 1990) or approximate closed-form solutions (Basu, 1977; Mak, 1986) in combination with restrictive assumptions like equal capacity of all buffers.

Another way of solving the BAP is called integrated optimization, where the stochasticity of the flow line is captured by either sampling the random variables or analytical expressions. After that, general solving techniques can handle this now deterministic optimization problem. Soyster et al. (1979) used analytical results followed by mixed-integer linear programming, whereas Matta et al. (2012) developed a kriging meta-model followed by a nonlinear solver. Yet, the majority of publications consider sampling, which we now focus on. In the case of a continuous-flow formulation, the buffer capacities do not need to take integer values, and nonlinear methods can be applied for optimization (Kolb & Göttlich, 2015). However, if the buffer integrality cannot be relaxed, mixed-integer linear programs need to be solved (Helber et al., 2011; Matta, 2008). In this setup, even correlations and general types of probability distributions can be considered (Weiss & Stolletz, 2015). The number of stations to be solved by integrated optimization is still somewhat limited. Zhang et al. (2022) reported results for unreliable lines. They were able to solve a 30-machines system with a total buffer capacity of 37 units, contemporaneously showing an example of an eight-machine system that cannot be solved within five hours of computational time.

The vast majority of publications make use of iterative optimization methods. The central principle for solving the buffer allocation problem iteratively consists of an evaluative method to compute the characteristics of a specific flow-line configuration and a generative method to determine the optimal solution. After discussing the evaluative approaches in Section 2.1, we can now focus on the generative methods. They can roughly be classified into three different groups: enumeration, meta-heuristics, and search algorithms. An enumeration of all feasible solutions guarantees optimality. However, it can only be used for small systems as the only optimization approach (see, e.g., Kiesmüller & Zimmermann, 2018; Powell & Pyke, 1996) or as a validation technique for other solution approaches (see, e.g., Papadopoulos et al., 2013). Hillier (2000) suggested only enumerating the subset of the solution space containing the most promising solutions. There are many meta-heuristics applicable to the BAP. Among others, genetic algorithms (Bulgak et al., 1995; Spinellis & Papadopoulos, 2000b), simulated annealing (Spinellis & Papadopoulos, 2000a, 2000b), or tabu search (Papadopoulos et al., 2013) were applied. Search algorithms include bottleneck-based approaches (Park, 1993), which determine bottlenecks iteratively and cope with their impact by allocating buffer capacity, gradients if the buffer integrality can be relaxed (Gershwin & Schor, 2000; Tempelmeier, 2003), and pseudo-gradients (Jafari & Shanthikumar, 1989).

The main result of allocating buffer capacity was coined the *storage bowl phenomenon* (Conway et al., 1988; Hillier et al., 1993). It means that buffer capacity near or at the center of a flow line is more effective than locating it at the edges. Hence, with a fixed buffer capacity available, the throughput of a flow line can be improved by an unbalanced allocation. Numerous studies confirmed this finding in different settings (see, e.g., Hillier, 2000). This result is a direct consequence of the *bowl phenomenon* described by Hillier and Boling (1979) for the optimal work allocation in flow lines.

The approach in this thesis is the application of iterative optimization approaches. Meta-heuristics and search algorithms are applied to find efficient buffer and spare part allocations. As the literature suggests, we will use enumeration as a validation approach for our algorithms. In the next section, we focus on how spare parts are used and the methods usually applied to allocate them.

2.3 Spare parts management

There is a large body of literature dedicated to spare parts management. We refer to the reviews by Kennedy et al. (2002) on spare parts inventories, Basten and Van Houtum (2014) regarding system-oriented models, Driessen et al. (2015) focusing on decision-making for spare parts planning, and Hu et al. (2018) proposing a wider view for spare parts management including forecasting and optimizing stocking policies. The monographs by Muckstadt (2005) and Van Houtum and Kranenburg (2015) are also valuable readings.

We can categorize the causes of spare parts demand according to the maintenance activities classification: both PM and CM can require spare parts in stock. In the case of deteriorating machines, PM activities can avoid failures (Wang, 2002). Some publications already considered simultaneous PM and buffer planning (Gan & Shi, 2014; Gan et al., 2013, 2015; Meller & Kim, 1996). For a production system, the crucial question is as follows: How much buffer for WIP is needed to ensure only slight effects on the production process? However, these models did not comprise that spare parts could also be required for unexpected breakdowns. Depending on the applied PM strategy, meeting the correct timing is more straightforward than for CM. PM usually takes place after a fixed amount of time or processed units (Pintelon & Gelders, 1992), whereas unexpected breakdowns are difficult to predict. Due to the high downtime costs, spare parts for CM need to be stocked. If a failure occurs, a repair-by-replacement strategy is commonly applied (Van Houtum & Kranenburg, 2015).

Even in the context of maintenance, there are many different replenishment policies for SKUs available (Wang, 2002). Vaughan (2005) investigated (s, S) policies for spare parts in the context of PM with deterministic lead times. The one-for-one $(S-1, S)$ replenishment policy is typically applied and optimal if fixed ordering costs are relatively low compared to holdings costs (Basten & Van Houtum, 2014; Moinzadeh & Lee, 1986; Muckstadt, 2005). Many papers seek the optimal base-stock level under different assumptions (see, e.g., Feeney & Sherbrooke, 1966; Lamghari-Idrissi et al., 2020b; Mirzahosseinian & Piplani, 2011; Moinzadeh & Schmidt, 1991).

These solutions usually rely on either a system- or an item-level approach. In a system-oriented approach, a breakdown of one machine causes downtime of the whole manufacturing system. Consequently, the system's availability equals the time where the system is producing (cf. Basten & Van Houtum, 2014; Rustenburg et al., 2000). This implication is no longer valid for a production system with interstage buffers where a machine downtime may not cause any stoppage of the whole production process. Hence, we cannot apply system-oriented methods.

As a consequence, item-level approaches could be used. However, they rely on knowing the spare parts demand process. This process is complex in a manufacturing system as blocking and starving influence the fraction of time where a machine is operating. In the case of time-dependent failures, it might not be a problem to come up with the demand process, but it is for the more realistic assumption of operation-dependent failures. Typically, analyzing the machine in isolation would be a solution to eliminate the influence of the rest of the manufacturing system. This would lead to overestimating the spare parts demand because of neglecting demand-reducing influences. This effect is called the *passivation effect* (Lau et al., 2006). Although Lau et al. (2006) tackled this problem, their study analyzes the influence of passivation on the system availability. Thus, item-level approaches would likely lead to overestimating the required base-stock level.

A solution to this dilemma is a new approach for determining the spare parts base-stock level. The critical component inside a machine is not critical on the system level but only at the item level. Recently, Lamghari-Idrissi et al. (2019) and Lamghari-Idrissi et al. (2020a) addressed the influence of various spare part service measures on the performance of a wafer production process. They took the manufacturing system design as given and analyzed how to increase the throughput in a given setup. Their approaches illustrate the necessity of developing new solutions for planning spare parts in manufacturing systems.

Algorithmically, greedy heuristics are often used if closed-form solutions are unavailable.

They deliver solutions reliably and efficiently (Basten & Van Houtum, 2014; Wong et al., 2005). We observe the application of greedy-type heuristics in many cases. Basten and Arts (2017) analyzed the joint provision of spare assets and spare parts, Drent and Arts (2021) determined optimal policies in a multi-item two-echelon spare parts inventory system, and Rippe and Kiesmüller (2022) identified which spare parts to pick on a tour of a repair-person.

Given that the existent methods cannot be applied for designing a manufacturing system with buffers, we include the characteristics of the spare parts in newly developed mathematical models. We especially consider greedy heuristics for solving the optimization problem because of the widespread success in applying greedy heuristics for spare parts planning.

2.4 Maintenance for manufacturing systems

The majority of publications in the area of stochastic flow lines consider failure and repair processes as black boxes. The authors argue for reasonable choices of lifetime and repair-time distributions but do not specifically state how failures and repairs occur. Some publications elaborated in more detail on this topic and discussed the failure process in more detail by analyzing deteriorating machines (see, e.g., Colledani & Tolio, 2012; Meller & Kim, 1996) or focused on the repair process by assuming limited repair capacity (see, e.g., Dudick, 1979; Li, 1987). Our work differs in both directions. First, we aim to specify the failure process and assume that failures are caused by critical components. Second, the repair process depends on spare parts availability. The latter is comparable to a situation with limited repair capacity as we may encounter limited spare parts availability.

The first publications considering limited repair capacity presented Markov chain solutions for two-machine systems in discrete (Dudick, 1979) and continuous time (Li, 1987). Kuhn (2003) used a coupled queuing system to evaluate flow lines of arbitrary length, including a limited number of repairpersons inducing an additional waiting time. Göttlich et al. (2012) applied a scheduling approach for allocating limited repair personnel to stations of a production system.

Including different maintenance activities for manufacturing systems pictures the repair process more accurately. Maintenance directly increases the machines' availability instead of only handling downtime consequences by buffers. A first approach to cope with machine downtime was introducing backup machines to circumvent stops in production (Kubat & Sumita, 1985). In the same vein, other setups increase the number of machines by parallel downstream machines (Diamantidis & Papadopoulos, 2006), parallel machines in general (Papadopoulos et al., 2009), or batch machines processing multiple workpieces simultaneously (Chang & Gershwin, 2010). Using a discrete-time Markov decision process, Van der Duyn Schouten and Vanneste (1995) determined near-optimal control-limit policies for a deteriorating machine with a buffer. Under some conditions, they even proved the optimality of their policies. Kyriakidis and Dimitrakos (2006) generalized this setup to non-stationary deterioration, i.e., the deterioration probabilities depend not only on the current system state but also on the machine's age. In continuous time, Meller and Kim (1996) determined the optimal buffer level in a two-machine system triggering PM. There are further publications on how to include PM (Colledani & Tolio, 2012; Karamatsoukis & Kyriakidis, 2010), PM in a Kanban-controlled assembly system using decomposition (Ruifeng & Subramaniam, 2012), or shared repair capacity (Colledani et al., 2012).

Recently, Fitouhi et al. (2017) introduced CBM for a two-machine, one-buffer flow line. They assumed that the condition of the machine is perfectly observable, which is often not the case. Cheng et al. (2017) considered a two-machine, one-buffer flow line with the first machine containing a component that deteriorates over time. After several repairs, however, the component needs to be replaced. The authors do not consider stock-holding of this component, but determine a policy at which buffer level and after how many failures the replacement should take place.

The three streams of the literature regarding manufacturing system evaluation, buffer allocation, and spare parts management were finally connected by a recent publication of Kiesmüller and Zimmermann (2018). They discussed a two-machine, one-buffer flow line where a failure-prone component inside every machine induces machine failures. Both machines have identical components. Hence, they needed only one base-stock level. Kiesmüller and Zimmermann modeled the system as a continuous-time Markov chain determining optimal values for buffer capacity and base-stock level. Further, they quantified the possible savings of a simultaneous optimization approach to an item-level approach with separate buffer-level determination. Although not directly tackling a buffer allocation problem, they studied the question of how to design a flow line in terms of buffer capacity and base-stock level.

This thesis contributes to the literature on stochastic manufacturing systems by extending and combining all four streams of the literature. We present a model for flow lines of arbitrary length with limited interstage buffers and spare parts provisioning for machine-individual failure-prone critical components, where processing times, component lifetimes, and replenishment times are exponentially distributed. Therefore, we model the process of CM more thoroughly and allow for the stock-keeping of spare parts. For the two- and three-machine cases, we show how to compute the steady-state probabilities of a continuous-time Markov chain to obtain performance measures. For longer flow lines, we devise a new decomposition procedure to calculate performance measures. To determine cost-efficient buffer and spare allocations, we exemplify three greedy heuristics and show how metaheuristics used for the BAP perform in comparison. Eventually, we present two new two-machine models with fixed, deterministic processing times, a buffer of limited capacity, and spare parts provisioning for failure-prone critical components, where component lifetimes and replenishment times are geometrically distributed. By comparing the cases for machine-specific and common components, we generate new insights into the effects of spare part commonality and its impact on required buffer capacity for unreliable flow lines.

Chapter

3 Evaluation of Flow Lines With Stochastic Processing Times

There is a way out of every box, a solution to every puzzle; it's just a matter of finding it.

Captain Jean-Luc Picard

The following chapter is published in the *European Journal of Operational Research* (Sachs et al., 2022a). In the next sections, we lay the groundwork for various analyses of unreliable flow lines with buffers and spare parts. We start with a general problem and model description in Section 3.1. Specifically, we discuss how to model the flow line as a continuous-time Markov chain and how to obtain performance measures such as throughput or work-in-process inventories. In Section 3.2, we apply different solution techniques. First, we introduce the solution of the Markov chain via its balance equations. The two-machine system we discuss is most closely connected to the model introduced by Gershwin and Berman (1981). The main difference is that we assume machine failures to be caused by failure-prone critical components. Thus, corrective maintenance is carried out as a repair-by-replacement strategy with stock-holding of spare parts. Second, we show how to approximate flow lines of arbitrary length by extending the decomposition approach first introduced by Gershwin (1987). In doing so, we can approximately evaluate system designs considering finite buffers and spare parts provisioning. We demonstrate in Section 3.3 that the decomposition delivers reliable and accurate results via a comparison with exact performance measures of a Markov chain approach for smaller lines and for longer lines discrete-event simulation, which is slow but arbitrarily accurate. Further, we present the first insights into the performance of different buffer and spare part allocations in flow lines. Finally, Section 3.4 concludes with some remarks.

3.1 Problem and model description

3.1.1 Problem description

We consider a flow line for discrete products consisting of I machines in series as depicted in Figure 5. Interstage buffers B_j, each with capacity C_j ($j \in \mathcal{J}$), are installed between adjacent machines to decouple the processes on the machines.

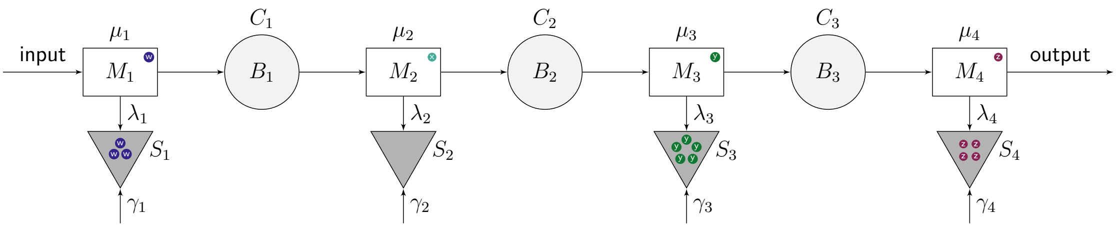


Figure 5. Model of the production system

Workpieces enter the system at machine M_1 and leave it at machine M_I. We consider blocking after service, i.e., if a machine completes a part and finds the downstream buffer full, that part remains on the machine, which is now blocked. (Note that the analysis for blocking *before* service is very similar (Dallery & Gershwin, 1992).) A machine is called "starving" if it finishes production and the upstream buffer is empty, which prevents the machine from starting to process a new workpiece.

We assume an infinite supply of raw material in front of the first machine and infinite space for storing end products behind the last machine. Hence, the system can be analyzed in isolation because the first machine can never be starving, and the last machine can never be blocked.

The processing times of all machines are assumed to be exponentially distributed with processing rates μ_i ($i \in \mathcal{I}$). This is a common assumption that considerably facilitates the analysis; see Papadopoulos et al. (2019). The great variability of exponentially distributed processing times can be interpreted as covering various sources of variability other than that caused by failures of the critical components.

Each of the machines contains one unit of a machine-specific, failure-prone component. If that component fails, the machine stops. We assume operation-dependent failures of the components, i.e., a component can only fail while the machine is working on a workpiece, i.e., being neither starving nor blocked. The times to failure of the components are assumed to follow exponential distributions with failure rates λ_i $(i \in \mathcal{I})$. This can be a reasonable model, e.g., of failures of electronic components that are not affected by wear-out effects; see Birolini (2012). Note that this model leads to a Poisson demand process for spare parts, a typical demand model for such slow-moving items; see Syntetos et al. (2012).

After a failure occurs, production will be resumed if the machine is repaired. We consider a repair-by-replacement strategy to reduce downtimes. Spare parts are held in stock near the production line such that a failed component can be replaced immediately if a spare part is available. When no spare part is on hand, the machine is down and can only resume production after a new spare part has been delivered and replaced with the failed critical component. Workpieces remain at machines while a machine is in a down state.

Whenever a spare part is used to replace a failed unit of the critical component, a new part is ordered directly to replenish the spare part inventory. In other words, a one-for-one replenishment policy with base-stock level S_i is implemented. Following Muckstadt (2005), such a policy is reasonable for valuable spare parts since the fixed ordering costs can be neglected compared to the substantial holding costs of valuable spare parts and the opportunity costs of a machine or potentially an entire flow line being down.

The delivery times of all spare parts are assumed to be exponentially distributed with replenishment rates γ_i $(i \in \mathcal{I})$. The assumption of exponential repair and delivery times is also common in the spare parts and maintenance literature (Sherbrooke, 2006; Wong et al., 2006). It is known that the type of distribution does not have a large effect on the performance of the system. According to the Palm–Khintchine theorem, only the mean of the distribution has a major influence; see Alfredsson and Verrijdt (1999), Kiesmüller and Sachs (2020), and Tan (1998). In manufacturing facilities, machines for different production steps are located close together, and conveyors connect different machines to transport work-in-process. Covering these short time intervals would drastically increase model complexity without generating other insides. Hence, transportation times can be assumed to be negligible. We furthermore consider all random times to be mutually stochastically independent.

As an element of the system state description, we denote the maximum number of units of the respective critical component at hand for machine i as Q_i. As we assume that there is also one unit of that component inside of every machine, we have $Q_i = S_i + 1$.

To analyze the system, several performance measures are of interest, particularly the system throughput, the probability of each of the machines being down, blocked, or starving, and the average inventory for each of the spare parts. Furthermore, the average work-in-process and the average buffer levels of the workpieces are essential characteristics. Such a performance analysis helps quantify the effect of buffer sizes between machines and spare part stocks on the overall system's performance.

This chapter shows how those performance measures can be computed. For ease of exposition, we denote the critical component of machine i as component i.

3.1.2 Mathematical model

The system can be modeled as a continuous-time Markov chain. Each state of the system is described by the number of workpieces that have already been processed completely at the different machines and the number of functional units of the critical component of a machine, either installed inside the machine or in stock as a spare part. The following notation is in line with the established symbols introduced in Gershwin and Berman (1981), Choong and Gershwin (1987), and subsequent publications.

We define n_j as the number of workpieces in the system that have already been completely processed by machine j but not yet by machine $j+1$ for $j \in \mathcal{J}$ as follows:

$$n_j = \begin{cases} 0, & M_{j+1} \text{ is starving, } B_j \text{ is empty} \\ 1, \ldots, C_j + 1, & n_j - 1 \text{ workpieces are in } B_j \text{ and one is on } M_{j+1} \\ C_j + 2, & B_j \text{ is full, } M_{j+1} \text{ is busy or blocked, } M_j \text{ is blocked} \end{cases} \tag{1}$$

We denote the maximum number of workpieces that have already been processed completely on machine j but not yet on machine $j+1$ as the *extended* buffer size $N_j = C_j + 2$, hence including the space on machines j and $j+1$, and denote $n_j = 0, \ldots, N_j$ as the extended buffer level.

We further define α_i as the number of functional units of component $i \in \mathcal{I}$ as follows:

$$\alpha_i = \begin{cases} 0, & M_i \text{ is down and there are no component } i \text{ spare parts available} \\ 1, \ldots, Q_i & M_i \text{ is up and there are } \alpha_i - 1 \text{ component } i \text{ spare parts available} \end{cases} \quad (2)$$

With S_i as the base-stock level of component i spare parts, we define the maximum number of functional units of component i as $Q_i = S_i + 1$. Accordingly, $Q_i - \alpha_i$ is the number of components in resupply.

Given these definitions of the extended buffer level n_i and the number of functional units α_i, we can describe each possible system state ξ as a $(2 \cdot I - 1)$-tuple:

$$\xi = (n_1, n_2, \ldots, n_{I-1}, \alpha_1, \alpha_2, \ldots, \alpha_I) \quad (3)$$

Note, however, that not each tuple describes a possible system state. For example, the tuple $(n_1 = N_1, n_2 = N_2, \alpha_1 = 1, \alpha_2 = 1, \alpha_3 = 1)$ for a three-machine line would describe a state in which the second machine is blocked $(n_2 = N_2 = C_2 + 2)$ with a workpiece already processed by the second machine still being on that second machine. However, entry $n_1 = N_1 = C_1 + 2$ denotes a situation in which the first machine is blocked, and a workpiece is already processed by the first, but *not* the second machine is still on the second machine. If that second machine can hold only one workpiece, we cannot simultaneously have both $n_1 = N_1$ and $n_2 = N_2$, as a workpiece on a machine has either been processed completely or not, but not both. Another example of a state that cannot be reached for that three-machine line is the tuple $(n_1 = 0, n_2 = 0, \alpha_1 = 1, \alpha_2 = 0, \alpha_3 = 0)$. This indicates a situation in which, on the one hand, both the second and third machines are starving. However, they are also down, which violates the assumption of operation-dependent failures that machines cannot fail when they are not operating and cannot be operating if they are starving.

We now denote by

$$p(\xi) = p(n_1, n_2, \ldots, n_{I-1}, \alpha_1, \alpha_2, \ldots, \alpha_I) \tag{4}$$

the probability of finding the system in state ξ with the understanding that some tuples describe states ξ that we will never observe and hence have zero probability.

If we set up the generator matrix for this monolithic Markov model of the system and determine the steady-state probabilities, we can calculate performance measures. For example, we can determine the throughput of the system via the throughput of the final machine I as

$$TP = \sum_{n_1=0}^{N_1} \cdots \sum_{n_{I-2}=0}^{N_{I-2}} \sum_{n_{I-1}=1}^{N_{I-1}} \sum_{\alpha_1=0}^{Q_1} \cdots \sum_{\alpha_{I-1}=0}^{Q_{I-1}} \sum_{\alpha_I=1}^{Q_I} p(n_1, n_2, \ldots, n_{I-1}, \alpha_1, \alpha_2, \ldots, \alpha_I) \cdot \mu_I \tag{5}$$

by multiplying the sum of the probabilities of all the states in which the last machine is neither starving $(n_{I-1} \geq 1)$ nor down $(\alpha_I \geq 1)$ with its processing rate μ_I.

In a similar way, we can define the average total extended buffer level (BL) in the system as

$$BL = \sum_{n_1=0}^{N_1} \cdots \sum_{n_{I-1}=0}^{N_{I-1}} \sum_{\alpha_1=0}^{Q_1} \cdots \sum_{\alpha_I=0}^{Q_I} p(n_1, n_2, \ldots, n_{I-1}, \alpha_1, \alpha_2, \ldots, \alpha_I) \cdot (n_1 + n_2 + \ldots n_{I-1}), \tag{6}$$

the average extended level BL_j of the buffer between machines j and $j+1$ $(j \in \mathcal{J})$ as

$$BL_j = \sum_{n_1=0}^{N_1} \cdots \sum_{n_j=1}^{N_j} \cdots \sum_{n_{I-1}=0}^{N_{I-1}} \sum_{\alpha_1=0}^{Q_1} \cdots \sum_{\alpha_I=0}^{Q_I} p(n_1, n_2, \ldots, n_{I-1}, \alpha_1, \alpha_2, \ldots, \alpha_I) \cdot n_j, \tag{7}$$

and the average inventory AI_i of spare parts (waiting to replace the currently functional unit) of component i ($i \in \mathcal{I}$) as

$$AI_i = \sum_{n_1=0}^{N_1} \cdots \sum_{n_{I-1}=0}^{N_{I-1}} \sum_{\alpha_1=0}^{Q_1} \cdots \sum_{\alpha_i=2}^{Q_i} \cdots \sum_{\alpha_I=0}^{Q_I} p(n_1, n_2, \ldots, n_{I-1}, \alpha_1, \alpha_2, \ldots, \alpha_I) \cdot (\alpha_i - 1). \quad (8)$$

Note that if only one critical component is available for each machine, i.e., $Q_i = 1, i \in \mathcal{I}$, then the replenishment lead time can be interpreted as the repair time, and the model reduces to the system analyzed by Choong and Gershwin (1987). In the remainder of the chapter, we derive approximate formulas to compute these performance measures.

3.2 Solution methods

The state space of this Markov chain expands dramatically with a growing number of machines, buffer sizes, and stock levels of the critical components. However, steady-state probabilities can be determined for flow lines with two machines. If a flow line consists of more than two machines, the exact evaluation of practice-relevant systems is typically impossible. For this reason, a heuristic technique is needed to evaluate longer flow lines at least approximately, i.e., to determine performance measures for a given configuration. These measures are a necessary prerequisite for any approach intended to optimize such a system.

Gershwin (1987) introduced the decomposition approach as a heuristic performance evaluation method for longer flow lines in the context of discrete-time Markov chains. It is mainly based on the idea of decomposing long flow lines into a set of coupled virtual two-machine lines, each of them representing the perspective of a myopic observer next to a specific buffer in the original system. The idea is illustrated with the help of a four-machine line example depicted in Figure 6, which is decomposed into three two-machine lines, each corresponding to a buffer in the original system.

As a flow line composed of I machines is decomposed into $I-1$ virtual two-machine systems, we can exploit the fact that the state space of such a two-machine line is still manageable, and its steady-state probabilities can be determined exactly. For this

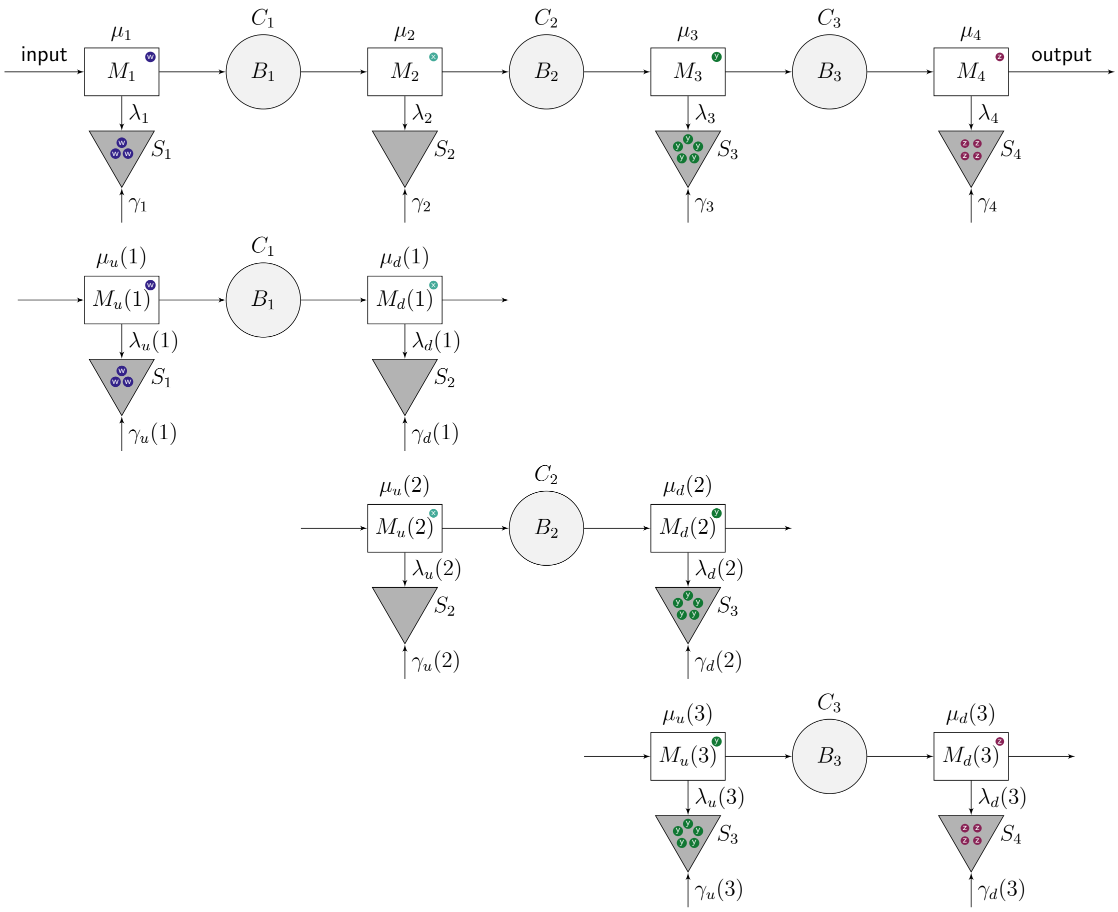


Figure 6. Flow line with four machines and the three corresponding virtual two-machine lines

reason, we now first describe the exact analysis of a two-machine line for our type of system.

3.2.1 Exact analysis of the two-machine line

The state of a two-machine system can be denoted via a tuple (n, α_u, α_d) with α_u representing the number of operational units of the critical component at the upstream machine. The interpretation of α_d is analogous, and n is the number of workpieces already processed by the upstream machine but not yet processed by the downstream machine. In (4), we introduce the notation $p(n_1, n_2, \ldots, n_{I-1}, \alpha_1, \alpha_2, \ldots, \alpha_I)$ to denote steady-state probabilities for the original line. To denote steady-state probabilities for two-machine lines, we use the slightly different notation $\mathcal{P}(n, \alpha_u, \alpha_d)$ to keep the perspectives on the original system and the virtual two-machine lines separate. To determine those steady-state probabilities, we have to state the balance equations that describe the probabilistic and dynamic behavior of the system. To this end, we first introduce an indicator function $\mathbb{1}(\mathcal{L})$ operating on a logical proposition $\mathcal{L}$ as follows:

$$\mathbb{1}(\mathcal{L}) = \begin{cases} 1, & \text{if } \mathcal{L} \text{ is true} \\ 0, & \text{if } \mathcal{L} \text{ is false.} \end{cases} \tag{9}$$

Using this indicator function, we can now state the balance equations for the two-machine system being in steady state for $n = 0, \ldots, N$, $\alpha_u = 0, \ldots, Q_u$, and $\alpha_d = 0, \ldots, Q_d$:

$$\begin{aligned} &\mathcal{P}(n, \alpha_u, \alpha_d) \cdot \Big((\mu_u + \lambda_u) \cdot \mathbb{1}(n < N,\, \alpha_u > 0) + (\mu_d + \lambda_d) \cdot \mathbb{1}(n > 0,\, \alpha_d > 0) + \\ &\qquad \gamma_u \cdot (Q_u - \alpha_u) \cdot \mathbb{1}(\alpha_u < Q_u) + \gamma_d \cdot (Q_d - \alpha_d) \cdot \mathbb{1}(\alpha_d < Q_d) \Big) \\ =& \mathcal{P}(n-1, \alpha_u, \alpha_d) \cdot \mu_u \cdot \mathbb{1}(n > 0,\, \alpha_u > 0) \\ +& \mathcal{P}(n+1, \alpha_u, \alpha_d) \cdot \mu_d \cdot \mathbb{1}(n < N,\, \alpha_d > 0) \\ +& \mathcal{P}(n, \alpha_u + 1, \alpha_d) \cdot \lambda_u \cdot \mathbb{1}(n < N,\, \alpha_u < Q_u) \\ +& \mathcal{P}(n, \alpha_u, \alpha_d + 1) \cdot \lambda_d \cdot \mathbb{1}(n > 0,\, \alpha_d < Q_d) \\ +& \mathcal{P}(n, \alpha_u - 1, \alpha_d) \cdot \gamma_u \cdot (Q_u - (\alpha_u - 1)) \cdot \mathbb{1}(\alpha_u > 0) \\ +& \mathcal{P}(n, \alpha_u, \alpha_d - 1) \cdot \gamma_d \cdot (Q_d - (\alpha_d - 1)) \cdot \mathbb{1}(\alpha_d > 0) \end{aligned} \tag{10}$$

For example, a state (n, α_u, α_d) in which the upstream machine is neither blocked nor down ($n < N$, $\alpha_u > 0$) can be left because the upstream machine completes the processing of a workpiece (with the rate μ_u) or because its critical component fails (with the rate λ_u). State (n, α_u, α_d) can be reached, e.g., from state $(n-1, \alpha_u, \alpha_d)$ as the upstream machine completes its process on a part, provided that in the new (receiving) state, i.e., after receiving that part, the buffer is not empty and the upstream machine is not down ($n > 0$, $\alpha_u > 0$). Note that (10) provides a complete and very compact description that can be used directly within a program to set up the generator matrix for this continuous-time Markov chain. Some few states are transient and hence have a steady-state probability of zero. An example is state $(n = 0, \alpha_u = 1, \alpha_d = 0)$ in which the second machine is both starving and down. This state cannot be reached naturally via the protocol of the system, as a machine can only fail while it is working.

Together with the normalization equation,

$$\sum_{n=0}^{N} \sum_{\alpha_u=0}^{Q_u} \sum_{\alpha_d=0}^{Q_d} \mathcal{P}(n, \alpha_u, \alpha_d) \;=\; 1, \tag{11}$$

we obtain a system of linear equations that determines the transition rate matrix and thus all the steady-state probabilities of the Markov process as well as the zero probabilities of the few transient states.

This system of equations can be solved numerically very quickly using standard software such as MATLAB. We hence obtain the steady-state probabilities of the two-machine system from which we can compute all necessary performance measures for a two-machine line. This two-machine model serves as a building block to analyze larger systems via an approximate decomposition to be described below. We also attempted to develop a closed-form solution for the steady-state probabilities along the lines in Gershwin and Berman (1981) and Gershwin and Schick (1983) and Tolio et al. (2002). However, this led to the problem of finding the roots of a polynomial, the degree of which increases with the number of spare parts, thus rendering this approach less attractive than the direct solution of the system of linear equations.

3.2.2 Decomposition approach

Each of the virtual two-machine lines in a decomposition, as depicted in the lower part of Figure 6, is characterized by three parameters for the upstream machine and three parameters for the downstream machine: processing rates $\mu_u(i)$ and $\mu_d(i)$, failure rates $\lambda_u(i)$ and $\lambda_d(i)$, and spare part replenishment rates $\gamma_u(i)$ and $\gamma_d(i)$ with $i = 1, \ldots, I-1$. The buffer sizes C_i and the base-stock levels in the virtual lines agree with their respective counterparts in the original line, e.g., in Figure 6 S_2 is first the base-stock level of the second machine in the original line, second of the downstream machine of the first virtual line, and third of the upstream machine of the second virtual line. Analogously, C_3 is both the buffer size between the third and the fourth machine in the original line and the buffer size between the up- and downstream machines of the third virtual line, as shown in Figure 6.

The first mathematical relationship used in the decomposition developed by Gershwin (1987) is denoted as the *conservation of flow* (COF) equation. Since the material can neither be destroyed nor created in any step of the production process, the throughput must be equal for all machines in the original flow line. In addition, it must be equal in all virtual lines for the decomposition results to be consistent. We use $TP(i)$ to refer to the throughput of the virtual line i and denote by TP_i the throughput of machine i in the original line. Furthermore, we postulate this equality condition as follows:

$$TP_1 = TP_2 = \cdots = TP_I = TP(1) = TP(2) = \cdots = TP(I-1) \tag{12}$$

This equality condition is used in the derivation of equations to determine the virtual machine parameters. Following the basic approach originally presented by Gershwin (1987), three types of decomposition equations have to be derived in order to determine numerical values for the virtual machines in the decomposition:

1. The *Flow Rate Idle Time* (FRIT) equations address the effects of blocking and starving on the throughput and serve to determine processing rates $\mu_u(i)$ and $\mu_d(i)$ for virtual machines.

2. The *Resumption of Flow* (ROF) equations deal with the delivery of spare parts and are used to determine spare part arrival rates $\gamma_u(i)$ and $\gamma_d(i)$ for virtual machines.

3. The *Interruption of Flow* (IOF) equations characterize the failure of critical components and are employed to compute failure rates $\lambda_u(i)$ and $\lambda_d(i)$ for virtual machines.

The derivation of those equations requires a major analytical effort that reflects the specific assumptions about the manufacturing systems, particularly the newly studied aspect of spare part provisioning. All proofs of the following statements can be found in the appendix (Section 3.5).

To refer to the steady-state probabilities of a *specific* virtual two-machine line i, we still use the previous notation for two-machine line state probabilities but append a subscript representing that line. The steady-state probability of a specific state (n, α_u, α_d) of virtual line i with $0 \leq n \leq N_i, 0 \leq \alpha_u \leq Q_i, 0 \leq \alpha_d \leq Q_{i+1}$ is hence referred to as $\mathcal{P}_i(n, \alpha_u, \alpha_d)$. When we refer to the sum of multiple steady-state probabilities, we use a probability-like notation, e.g., $\mathcal{P}_i(n > 0, 1, 0)$ is the probability that the buffer is not empty, the upstream machine is functional with no spare part in stock, and the downstream machine is down (with regard to the virtual line i). It can be computed as

$$\mathcal{P}_i(n > 0, 1, 0) = \sum_{n=1}^{N_i} \mathcal{P}_i(n, 1, 0) \tag{13}$$

For the upcoming approximate analysis of the flow line, we make the following three assumptions:

(A1) The probability that a machine in the original line is starving and blocked at the same time is negligible, i.e.,

$$P(n_{i-1} = 0, n_i = N_i, \alpha_i = j) \approx 0, \quad \forall i = 2, \dots, I-1, \forall j = 1, \dots, Q_i, \tag{14}$$

(Note that a machine that is down ($\alpha_i = 0$) cannot be blocked or starving because we assume operation-dependent failures.)

(A2) The probability that machine i of the original line is starving can be approximated by the corresponding probability of the virtual downstream machine of the virtual line $i-1$, i.e.,

$$P(n_{i-1} = 0, n_i < N_i, \alpha_i = j) \approx \mathcal{P}_{i-1}(0, \alpha_u \geq 0, j), \qquad \forall i = 2, \dots, I-1, \forall j = 1, \dots, Q_i. \tag{15}$$

Similarly, we approximate the probability that machine i in the original line is down via the probability of observing the virtual downstream machine of the line $i-1$ being down:

$$P(n_{i-1} > 0, n_i < N_i, \alpha_i = 0) \approx \mathcal{P}_{i-1}(n > 0,\ \alpha_u \geq 0,\ 0), \quad \forall i = 2, \dots, I-1 \tag{16}$$

(A3) The probability that machine i of the original line is blocked can be approximated by the corresponding probability of the virtual upstream machine of the virtual line i, i.e.,

$$P(n_{i-1} > 0,\, n_i = N_i,\, \alpha_i = j) \approx \mathcal{P}_i(N_i, j,\, \alpha_d \geq 0), \quad \forall i = 2, \ldots, I-1, \forall j = 1, \ldots, Q_i. \tag{17}$$

Similarly, we approximate the probability that machine i in the original line is down via the probability of observing the virtual upstream machine of line i being down:

$$P(n_{i-1} > 0,\, n_i < N_i,\, \alpha_i = 0) \approx \mathcal{P}_i(n < N_i,\, 0,\, \alpha_d \geq 0), \quad \forall i = 2, \ldots, I-1 \tag{18}$$

These assumptions are the basis for a related set of decomposition equations, which we state below as theorems. They are used to determine the parameters of the machines of the virtual two-machine lines. The proofs for all those theorems can be found in the appendix (Section 3.5). As a first step, we provide *Flow Rate Idle Time* equations related to the throughput of machine i in the flow line. The throughput can be determined as the fraction of time that a machine is actually producing, i.e., being neither down nor starving nor blocked, multiplied by its corresponding processing rate.

Theorem 1. *For given values of $\mu_i, \gamma_i, \lambda_i, N_i$ and Q_i of the original line and under Assumptions (A1), (A2) and (A3), we can approximate the throughput of machine $i = 2, \ldots, I-1$ as*

$$TP_i \approx \mu_i \cdot \frac{A_i}{A_i + 1} \cdot \Big(1 - \mathcal{P}_{i-1}(0, \alpha_u \geq 0, \alpha_d \geq 1) - \mathcal{P}_i(N_i, \alpha_u \geq 1, \alpha_d \geq 0)\Big) \tag{19}$$

where A_i is defined as

$$A_i = \sum_{j=1}^{Q_i} \prod_{m=0}^{j-1} (Q_i - m) \cdot \left(\frac{\gamma_i}{\lambda_i}\right)^j + \sum_{j=1}^{Q_i} \sum_{k=1}^{j-1} \prod_{m=k}^{j-1} (Q_i - m) \cdot \left(\frac{\gamma_i}{\lambda_i}\right)^{j-k} \cdot \left(\frac{\mathcal{P}_{i-1}(0, \alpha_u \geq 0, k)}{\mathcal{P}_{i-1}(n > 0, \alpha_u \geq 0, 0)} + \frac{\mathcal{P}_i(N_i, k, \alpha_d \geq 0)}{\mathcal{P}_i(n < N_i, 0, \alpha_d \geq 0)}\right). \tag{20}$$

The interpretation of formula (19) is that the throughput of machine i of the original line is the product of that machine's processing rate μ_i, the fraction of time $\frac{A_i}{A_i+1}$ with at least one operational unit of the critical component, and the probability of machine i being neither starved nor blocked. Note that if the base-stock level of spare parts is $S_i = 0$ and hence $Q_i = 1$, i.e., at most one functional copy of the critical unit is available, then we find that $A_i = \frac{\gamma_i}{\lambda_i}$ from (20) and have $\frac{A_i}{A_i+1} = \frac{\gamma_i}{\gamma_i+\lambda_i}$, the standard availability expression for machine i. An equivalent approach can be used to analyze the throughput of a virtual two-machine line's upstream and downstream machines. This enables us to connect the throughput from Equation (19) of the original line with the processing rates of the virtual upstream machine from virtual line i and the downstream machine of the line $i-1$ as expressed in the following corollary:

Corollary 2. *For given values of $\mu_i, \gamma_i, \lambda_i, N_i$ and Q_i of the original line, with an analogous definition of $A_u(i+1)$ and $A_d(i-1)$ to A_i in (20) given in the appendix (Section 3.5) as (70) and (71) and under Assumptions (A1), (A2) and (A3), the processing rate of the virtual upstream machine $i = 2, \ldots, I-1$ can be calculated as*

$$\mu_u(i) = \left(\frac{1}{\mu_i} \cdot \frac{A_i+1}{A_i} - \frac{1}{\mu_d(i-1)} \cdot \frac{1+A_d(i-1)}{A_d(i-1)} + \frac{1}{TP(i-1)}\right)^{-1} \cdot \frac{1+A_u(i)}{A_u(i)} \tag{21}$$

and for the virtual downstream machine $i = 1, \ldots, I-2$ as

$$\mu_d(i) = \left(\frac{1}{\mu_{i+1}} \cdot \frac{A_{i+1}+1}{A_{i+1}} - \frac{1}{\mu_u(i+1)} \cdot \frac{1+A_u(i+1)}{A_u(i+1)} + \frac{1}{TP(i+1)}\right)^{-1} \cdot \frac{1+A_d(i)}{A_d(i)}. \tag{22}$$

In a second step, we present the *Interruption of Flow* equations that characterize the failure rates of the virtual machines in terms of parameters of both the original and virtual machines as given in the following theorem:

Theorem 3. *For given values of $\mu_i, \gamma_i, \lambda_i, N_i$ and Q_i of the original line and under Assumptions (A1), (A2) and (A3), the failure rate for the virtual upstream machine $i = 2, \ldots, I-1$ can be calculated as*

$$\lambda_u(i) = \lambda_i + \frac{\mathcal{P}_{i-1}(1,0,1)\,\mu_d(i-1) + \mathcal{P}_{i-1}(0,1,\alpha_d \geq 1)\,\lambda_u(i-1)}{\frac{TP(i-1)}{\mu_u(i)} - \mathcal{P}_i(n < N_i, \alpha_u \geq 2, \alpha_d \geq 0)} \tag{23}$$

and for the virtual downstream machine $i = 1, \ldots, I-2$ *as*

$$\lambda_d(i) = \lambda_{i+1} + \frac{\mathcal{P}_{i+1}(N_{i+1}-1, 1, 0)\, \mu_u(i+1) + \mathcal{P}_{i+1}(N_{i+1}, \alpha_u \geq 1, 1)\, \lambda_d(i+1)}{\frac{TP(i+1)}{\mu_d(i)} - \mathcal{P}_i(n > 0, \alpha_u \geq 0, \alpha_d \geq 2)}. \tag{24}$$

Similarly, the *Resumption of Flow* equations describe the rates at which individual outstanding spare parts arrive for virtual machines:

Theorem 4. *For given values of* $\mu_i, \gamma_i, \lambda_i, N_i$ *and* Q_i *of the original line and under Assumptions (A1), (A2) and (A3), the replenishment rate for an outstanding critical component of the virtual upstream machine of virtual line* $i = 2, \ldots, I-1$ *can be calculated as*

$$\gamma_u(i) = \gamma_i + \frac{\left(\frac{Q_{i-1}}{Q_i}\gamma_u(i-1) - \gamma_i\right)\mathcal{P}_{i-1}(0, 0, \alpha_d \geq 1)}{\frac{\lambda_u(i)}{Q_i \gamma_u(i)}\left(\frac{TP(i-1)}{\mu_u(i)} - \mathcal{P}_i(n < N_i, \alpha_u \geq 2, \alpha_d \geq 0)\right)} \tag{25}$$

and for the virtual downstream machine of virtual line $i = 1, \ldots, I-2$ *as*

$$\gamma_d(i) = \gamma_{i+1} + \frac{\left(\frac{Q_{i+2}}{Q_{i+1}}\gamma_d(i+1) - \gamma_{i+1}\right)\mathcal{P}_{i+1}(N_{i+1}, \alpha_u \geq 1, 0)}{\frac{\lambda_d(i)}{Q_{i+1}\gamma_d(i)}\left(\frac{TP(i+1)}{\mu_d(i)} - \mathcal{P}_i(n > 0, \alpha_u \geq 0, \alpha_d \geq 2)\right)}. \tag{26}$$

3.2.3 Algorithmic notes

In the decomposition approach, we seek for all virtual lines $i = 1, \ldots, I-1$ numerical values for all virtual machine parameters $\mu_u(i)$, $\mu_d(i)$, $\lambda_u(i), \lambda_d(i), \gamma_u(i)$ and $\gamma_d(i)$ so that (21) to (26) hold simultaneously. We observed some numerical problems during the initial implementation of the iterative algorithm used to solve those equations. The updates for failure and replenishment rates ((23) to (26)) contain quotients that represent conditional probabilities, as can be seen in the derivation in the appendix (Section 3.5). Being (conditional) probabilities, they are known to be bound from above by 1. The derivation also indicates that they are as well bound from below by a positive value. We can hence impose those bounds in the iterative updating process.

To this end, we define a function

$$f(x,y) := \min\left\{1, \max\left\{x, \frac{x}{y}\right\}\right\} \tag{27}$$

and use the following updating formulas:

$$\begin{aligned}\lambda_u(i) = \lambda_i &+ f\left(\mathcal{P}_{i-1}(1,0,1), \frac{TP(i-1)}{\mu_u(i)} - \mathcal{P}_i(n < N_i, \alpha_u \geq 2, \alpha_d \geq 0)\right) \cdot \mu_d(i-1) \\ &+ f\left(\mathcal{P}_{i-1}(0,1,\alpha_d \geq 1), \frac{TP(i-1)}{\mu_u(i)} - \mathcal{P}_i(n < N_i, \alpha_u \geq 2, \alpha_d \geq 0)\right) \cdot \\ &\lambda_u(i-1)\end{aligned} \tag{28}$$

$$\begin{aligned}\lambda_d(i) = \lambda_{i+1} &+ f\left(\mathcal{P}_{i+1}(N_{i+1}-1,1,0), \frac{TP(i+1)}{\mu_d(i)} - \mathcal{P}_i(n > 0, \alpha_u \geq 0, \alpha_d \geq 2)\right) \\ &\cdot \mu_u(i+1) + f\Big(\mathcal{P}_{i+1}(N_{i+1}, \alpha_u \geq 1, 1), \\ &\frac{TP(i+1)}{\mu_d(i)} - \mathcal{P}_i(n > 0, \alpha_u \geq 0, \alpha_d \geq 2)\Big) \cdot \lambda_d(i+1)\end{aligned} \tag{29}$$

$$\begin{aligned}\gamma_u(i) = \gamma_i &+ \left(\frac{Q_{i-1}}{Q_i}\gamma_u(i-1) - \gamma_i\right) \cdot f\Bigg(\mathcal{P}_{i-1}(0,0,\alpha_d \geq 1), \\ &\frac{\lambda_u(i)}{Q_i\gamma_u(i)} \cdot \left(\frac{TP(i-1)}{\mu_u(i)} - \mathcal{P}_i(n < N_i, \alpha_u \geq 2, \alpha_d \geq 0)\right)\Bigg)\end{aligned} \tag{30}$$

$$\begin{aligned}\gamma_d(i) = \gamma_{i+1} &+ \left(\frac{Q_{i+2}}{Q_{i+1}}\gamma_d(i+1) - \gamma_{i+1}\right) \cdot f\Bigg(\mathcal{P}_{i+1}(N_{i+1}, \alpha_u \geq 1, 0), \\ &\frac{\lambda_d(i)}{Q_{i+1}\gamma_d(i)} \cdot \left(\frac{TP(i+1)}{\mu_d(i)} - \mathcal{P}_i(n > 0, \alpha_u \geq 0, \alpha_d \geq 2)\right)\Bigg).\end{aligned} \tag{31}$$

inside of the iterative Algorithm 1. The algorithm was implemented in MATLAB. To determine new parameters, e.g., of the upstream machine of virtual two-machine line i in the forward pass of the algorithm, the nonlinear system of (21), (23), and (25) in the unknowns $\mu_u(i), \gamma_u(i)$, and $\lambda_u(i)$ has to be solved. As we did not find a general analytic solution for arbitrary numbers of spare parts, we apply a fixed-point iteration to solve the nonlinear system of equations numerically.

We observe situations in which the decomposition approach did not converge to a final set of parameters and performance measures to any given measuring accuracy. In these situations, it cycles between different values with neighboring performance measures infinitely long. Hence, we first try to increase ε by one order of magnitude to achieve a

Algorithm 1. Decomposition algorithm

Initialize $\varepsilon := 0.001$;
for *all virtual lines* $i = 1, 2, \dots, I-1$ **do**
  Initialize $\lambda_u(i) := \lambda_i,\ \gamma_u(i) := \gamma_i,\ \mu_u(i) := \mu_i, \lambda_d(i) := \lambda_{i+1},\ \gamma_d(i) := \gamma_{i+1},$ $\mu_d(i) := \mu_{i+1}$, and determine steady-state probabilities $\mathcal{P}_i(n, \alpha_u, \alpha_d)$ and throughput $TP(i)$;
while $|TP(1) - TP(I-1)| > \varepsilon$ **do**
  for $i = 2, 3, \dots, I-1$ **do**
    Forward pass: Solve for line i (21), (28), and (30) iteratively via fixed-point iteration to determine new values for rates $\mu_u(i), \gamma_u(i)$, and $\lambda_u(i)$, updating steady-state probabilities $\mathcal{P}_i(n, \alpha_u, \alpha_d)$ and throughput $TP(i)$ after each iteration of the fixed-point iteration;
  for $i = I-2, I-3, \dots, 1$ **do**
    Backward pass: Solve for line i (22), (29), and (31) iteratively via fixed-point iteration to determine new values for rates $\mu_d(i), \gamma_d(i)$, and $\lambda_d(i)$, updating steady-state probabilities $\mathcal{P}_i(n, \alpha_u, \alpha_d)$ and throughput $TP(i)$ after each iteration of the fixed-point iteration.

result with slightly lower accuracy. If this does not solve the problem, we abort the iterations and use the latest values as final performance measures.

3.3 Numerical analysis

Our numerical analysis aims to understand the behavior of our algorithm in terms of both accuracy and reliability, as well as the behavior of the specific manufacturing system. To this end, we studied a series of systematically generated artificial systems via discrete-event simulation, via our approximate decomposition approach, and, for the special case of short flow lines with only three machines, via the exact analysis of a monolithic continuous-time Markov chain model. Additionally, real-world examples of manufacturing systems are used to examine the applicability of the proposed decomposition. Tempelmeier (2003) lists several systems he observed in practice. These systems contain between 8 and 23 machines or, more generally, stations. He states that algorithms must be capable of managing 50 up to 100 stations for practical purposes.

Each of the simulation studies contains a warm-up period of 1,000 time units. After that, 100,000 time units are simulated and used for the calculation of the system throughput.

The 95% confidence interval (i.e., $\alpha = 0.05$) of the throughput is computed after each run following Law (2014, p. 235) as

$$\left[\overline{X}(k) - HW(k)\ ,\ \overline{X}(k) + HW(k)\right] \tag{32}$$

with sample mean, sample variance, and half-width defined as

$$\overline{X}(k) = \frac{1}{k}\sum_{l=1}^{k} X_l, \tag{33}$$

$$S^2(k) = \frac{1}{k-1}\sum_{l=1}^{k}\left(X_l - \overline{X}(k)\right)^2 \tag{34}$$

$$HW(k) = t_{k-1,1-\frac{\alpha}{2}}\sqrt{\frac{1}{k}\,S^2(k)} \tag{35}$$

where $t_{k-1,1-\frac{\alpha}{2}}$ is the upper $1-\frac{\alpha}{2}$ critical point of the t distribution with $k-1$ degrees of freedom and X_l refers to the throughput of simulation run l, $l = 1, \ldots, k$ when k runs are already through. We conduct at least 10 runs of the simulation but follow a sequential procedure (as described by, e.g., Law, 2014, p. 505) such that we conduct additional simulation runs and recalculate all statistical values as long as the half-width of the 95% confidence interval is less than or equal to a specified small value

$$HW(k) \leq 0.01. \tag{36}$$

For each simulation, we use the abbreviated notation HW for the half-width of the final 95% confidence interval.

All algorithms were implemented in MATLAB. The calculations took place on a machine with MATLAB R2021a on an Intel Xeon Platinum 8280 processor with a base frequency of 2.70 GHz. Run times were measured using single-core calculations.

For our further analysis, we define the terms “balanced” and “unbalanced” flow lines. A balanced flow line consists of similar machines, spare parts, and buffers, i.e., for a given number of machines I, it holds that

$$C_j = C_l \qquad \forall j, l = 1, \ldots, I-1 \tag{37}$$

$$Q_i = Q_k, \quad \mu_i = \mu_k, \quad \lambda_i = \lambda_k, \quad \gamma_i = \gamma_k \qquad \forall i, k = 1, \ldots, I. \tag{38}$$

Consequently, a flow line is called unbalanced if one of these equalities does not apply.

3.3.1 Short lines

Short flow lines consisting of three machines are studied because the exact analysis is still possible for these lines, and the performance measures obtained by the decomposition approach can be compared with exact results. To this end, we also developed a monolithic Markov chain model for the three-machine lines (which is not presented in this thesis). We additionally include results from a simulation to validate this simulation approach. Table 1 lists cases for different parameter constellations of the three-machine line. These cases provide insights into a wide variety of three-machine systems. Although the presented cases are synthetic, they show the characteristics of real-world flow lines. Importantly, the failure and replenishment rates as well as the production rates, are realistically chosen compared to real-world systems, such as those studied by Tempelmeier (2003). These parameters are given in the appendix (Section 3.5) and are discussed later in more detail (Tables 14 and 15).

Case 1 is a system design equivalent to those studied in Gershwin (1987) because there are no spare parts for critical components (i.e., $Q_1 = Q_2 = Q_3 = 1$). By comparing cases 2 and 3, we can identify the effects of buffer capacity on our algorithm in a situation with spare parts. Comparing the results of cases 1, 2, and 4 helps to understand the effect of the spare part base-stock level on the algorithm. For cases 5 to 8, case 4 must be considered as the baseline case. Case 5 indicates longer replenishment lead times. Cases 6 to 8 show the impact of unbalanced lines. In this way, two virtual flow lines of the decomposition are affected, and hence, this has a relatively strong influence on the algorithmic behavior. We consider a higher processing rate for the second machine (case 6) as well as a higher failure rate (case 7) and a lower replenishment rate (case 8) for the first and third machines. Cases 6 to 8 have in common that they lead, relative to baseline case 4, to a higher probability of simultaneously starving and blocking (in the three-machine line only possible at the second machine). This situation is expected to pose a potential problem for the algorithm, as we assume in Assumption (A1) that its probability is negligible.

The corresponding results are listed in Table 2. We compare the throughput obtained by the exact Markov analysis (TP_{exact}) with the throughput of the decomposition approach (TP_{dec}) and our simulation (TP_{sim}). Furthermore, the actual and relative deviations of decomposition and simulation compared to the exact solution

$$\Delta_{\text{dec}} := TP_{\text{dec}} - TP_{\text{exact}}, \qquad \Delta^{\text{rel}}_{\text{dec}} := \frac{TP_{\text{dec}} - TP_{\text{exact}}}{TP_{\text{exact}}} \tag{39}$$

$$\Delta_{\text{sim}} := TP_{\text{sim}} - TP_{\text{exact}}, \qquad \Delta^{\text{rel}}_{\text{sim}} := \frac{TP_{\text{sim}} - TP_{\text{exact}}}{TP_{\text{exact}}} \tag{40}$$

as well as the half-width of the 95% confidence interval of our simulation study are given. Additionally, we provide computation times, which are abbreviated as T_{exact}, T_{dec}, and T_{sim} for exact Markov analysis, decomposition approach, and simulation, respectively. In order to better understand the impact of spare parts on the specific machines' performance, we introduce the isolated machine availability and the isolated machine throughput as follows

$$iAV_i := 1 - \left(\sum_{k=0}^{Q_i} \left(\frac{\gamma_i}{\lambda_i} \right)^k \cdot \frac{Q_i!}{(Q_i - k)!} \right)^{-1}, \tag{41}$$

$$iTP_i := iAV_i \cdot \mu_i. \tag{42}$$

The proof for (41) can be found in the appendix (Section 3.5). As a consequence, all availabilities of the different machines are identical in case of balanced flow lines.

Table 1. Cases for the three-machine line

Case	N_1	N_2	Q_1	Q_2	Q_3	μ_1	μ_2	μ_3	λ_1	λ_2	λ_3	γ_1	γ_2	γ_3	iAV_1	iAV_2	iAV_3
1	10	10	1	1	1	1	1	1	0.005	0.005	0.005	0.1	0.1	0.1	0.9524	0.9524	0.9524
2	10	10	2	2	2	1	1	1	0.005	0.005	0.005	0.1	0.1	0.1	0.9988	0.9988	0.9988
3	20	20	2	2	2	1	1	1	0.005	0.005	0.005	0.1	0.1	0.1	0.9988	0.9988	0.9988
4	10	10	3	3	3	1	1	1	0.005	0.005	0.005	0.1	0.1	0.1	1.0000	1.0000	1.0000
5	10	10	3	3	3	1	1	1	0.005	0.005	0.005	0.01	0.01	0.01	0.9873	0.9873	0.9873
6	10	10	3	3	3	1	1.1	1	0.005	0.005	0.005	0.1	0.1	0.1	1.0000	1.0000	1.0000
7	10	10	3	3	3	1	1	1	0.05	0.005	0.05	0.1	0.1	0.1	0.9873	1.0000	0.9873
8	10	10	3	3	3	1	1	1	0.005	0.005	0.005	0.01	0.1	0.01	0.9873	1.0000	0.9873

The values given in Table 2 clearly show that the decomposition approach delivers very accurate results in all cases. The relative deviation between the throughput of the exact and decomposition approaches is considerably less than 1% in all cases. As expected, the accuracy is diminished in the cases with an increased probability of simultaneous starving and blocking (case 4 compared to case 6). However, the accuracy in cases 7 and 8 is still very good. The same is true (in an extenuated manner) if we increase the number of spare parts since this reduces the downtimes and accordingly increases the probability of simultaneous starving and blocking (comparison of cases 1, 2, and 4). Furthermore, the algorithm is more accurate for a larger buffer capacity (case 2

Table 2. Results for the three-machine line cases. Asterisks indicate that TP_{exact} is included in the 95% confidence interval of the simulation result.

Case	TP_{exact}	TP_{dec}	Δ_{dec}	$\Delta^{\text{rel}}_{\text{dec}}$	TP_{sim}	Δ_{sim}	$\Delta^{\text{rel}}_{\text{sim}}$	HW
1	0.8133	0.8124	−0.0009	−0.0011	0.8142	0.0009	0.0012	0.0027*
2	0.8927	0.8915	−0.0012	−0.0013	0.8906	−0.0021	−0.0024	0.0022*
3	0.9381	0.9377	−0.0004	−0.0004	0.9372	−0.0009	−0.0009	0.0018*
4	0.8944	0.8932	−0.0012	−0.0014	0.8957	0.0013	0.0014	0.0026*
5	0.8715	0.8717	0.0002	0.0002	0.8693	−0.0022	−0.0025	0.0026*
6	0.9216	0.9250	0.0034	0.0037	0.9228	0.0012	0.0013	0.0023*
7	0.8840	0.8842	0.0002	0.0002	0.8827	−0.0013	−0.0015	0.0017*
8	0.8791	0.8783	−0.0008	−0.0009	0.8784	−0.0006	−0.0007	0.0045*

compared to case 3). This is in line with previous findings for similar decomposition approaches. The throughput calculated with the decomposition algorithm is always within the simulation's very tight 95% confidence intervals.

Additionally, the results show how well our simulation model reproduces the system behavior. Again, we only have minor throughput deviations of less than 1% in all cases. Moreover, the exact throughput of the considered flow line is always inside the very tight 95% confidence intervals from the simulation.

The computation times for the different cases are listed in Table 3. As expected, the time consumed by the exact Markov approach using a monolithic three-machine model depends mainly on the buffer capacity and the spare part base-stock levels because they increase the size of the state space. In all cases, the decomposition approach is faster than the exact solution method. Additionally, the decomposition is much faster than the simulation in all instances. Its computational speed seems to be nearly independent of the machine parameters.

In addition, Table 4 shows further performance measures obtained by the decomposition approach: The average extended buffer levels and the average spare part inventories. The decomposition approximates both measures accurately. We observe a maximum deviation of about 2% for the average buffer level and about 8% for the average stock on hand. In this respect, the differences are slightly higher than the throughput results. Previous studies of decomposition approaches reported similar findings with less accurate approximated buffer levels compared to throughput. However, there are system characteristics that positively affect the approximation. Balanced and small

Table 3. Computation times for the three-machine line cases

Case	T_{exact} (s)	T_{dec} (s)	T_{sim} (s)
1	0.33	0.15	45.12
2	2.94	0.18	50.04
3	78.29	0.18	51.33
4	38.39	0.24	52.75
5	35.34	0.19	47.50
6	37.49	0.22	59.33
7	34.94	0.27	51.85
8	38.58	1.50	51.40

systems and systems with larger buffer capacities show a good overall accuracy (see, e.g., Dallery et al., 1989; Helber, 2000; Le Bihan & Dallery, 2000). We tested about 20,000 instances of unbalanced three-machine flow lines (whose results are not presented in this thesis). Even though unbalanced systems negatively influence the decomposition performance, we confirmed a minor mean absolute percentage error of about 1.60% for the average buffer levels and 1.78% for the average spare part stocks on hand.

Table 4. Average extended buffer levels and spare part inventories obtained by the Markov approach (exact) and by the proposed decomposition (dec) along with the relative deviation (Δ)

Case	BL_1			BL_2			AI_1			AI_2			AI_3		
	exact	dec	Δ	exact	dec	Δ	exact	dec	Δ	exact	dec	Δ	exact	dec	Δ
1	6.93	6.90	0.00	5.07	5.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	6.82	6.80	0.00	5.18	5.18	0.00	0.96	0.96	0.00	0.96	0.95	−0.01	0.96	0.96	0.00
3	12.44	12.36	−0.01	9.56	9.58	0.00	0.95	0.95	0.00	0.95	0.95	0.00	0.95	0.95	0.00
4	6.81	6.79	0.00	5.19	5.19	0.00	1.96	1.96	0.00	1.96	1.95	0.00	1.96	1.96	0.00
5	6.85	6.85	0.00	5.15	5.13	0.00	1.57	1.57	0.00	1.57	1.51	−0.04	1.57	1.57	0.00
6	5.98	5.92	−0.01	6.02	6.06	0.01	1.95	1.95	0.00	1.96	1.95	0.00	1.95	1.95	0.00
7	6.74	6.73	0.00	5.26	5.26	0.00	1.57	1.57	0.00	1.96	1.80	−0.08	1.57	1.57	0.00
8	6.79	6.78	0.00	5.21	5.10	−0.02	1.57	1.57	0.00	1.96	1.93	−0.01	1.57	1.57	0.00

In a further experiment, we deliberately provoke in Figure 7 a situation with simultaneous starving and blocking of the second machine of our three-machine line. To this end, we fix all parameters but μ_2 to show how much the probability of concurrently starving and blocking increases with an increasing machine M_2 processing rate μ_2. Even in this case, the accuracy of the production rate approximation of the decomposition diminishes only up to a certain point. The probability of concurrently starving and blocking as well as the relative error compared to the exact Markov chain approach, are depicted in Figure 7. Although the deviation increases with increasing values of μ_2, it does not become worse than 5% even for this unrealistically unbalanced flow line.

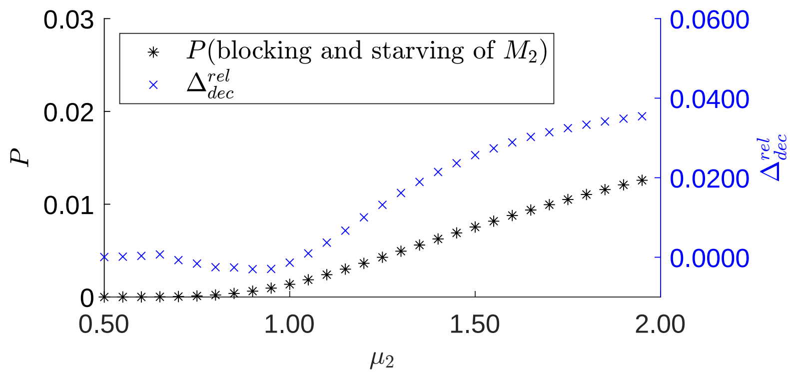


Figure 7. Performance of the decomposition approach with varying μ_2; $\mu_1 = \mu_3 = 1, C_1 = C_2 = 10, Q_i = 3, \lambda_i = 0.005, \gamma_i = 0.05\ (i = 1, 2, 3)$

The presented results clearly show that the decomposition approach can generate highly accurate production rate estimates. Furthermore, the results validate our simulation study. We will therefore compare our decomposition results to those obtained via simulation to analyze longer lines.

3.3.2 Longer lines

In order to illustrate the effectiveness of the proposed decomposition on a broader base, we analyzed a large number of unbalanced flow lines.

We first randomly generated 1,000 sets of parameters for a five-machine flow line. The parameters were chosen from a uniform distribution with the following intervals.

$$\mu_i \in [0.8, 1.2], \quad \lambda_i \in [0.004, 0.006], \quad \gamma_i \in [0.04, 0.06] \qquad \forall i \in \mathcal{I} \tag{43}$$

Thus, we altered the parameters used before by a maximum of 20% up and down. As a baseline case we use only one spare part, i.e., $Q_i = 2, i \in \mathcal{I}$, and identical buffer capacities, i.e., $C_j = 20, j = 1, \ldots, 4$, for all instances. In this setting, we altered only one specific parameter in five different values. We did this for both design parameters (buffer capacities and spare part base-stock levels) and machine parameters. All parameter ranges are given in Table 5. Figure 8 depicts the results as boxplots. Each of the boxplots represents 1,000 instances. This yields 25,000 solved instances for the whole figure. The actual and relative deviations of the throughput are calculated as

$$\Delta_{\text{dec/sim}} := TP_{\text{dec}} - TP_{\text{sim}}, \qquad \Delta^{\text{rel}}_{\text{dec/sim}} := \frac{TP_{\text{dec}} - TP_{\text{sim}}}{TP_{\text{sim}}} \tag{44}$$

In each of the presented cases, the parameters in Figure 8 are altered in a way that the throughput of the system increases from the left to the right, e.g., from $\mu_i = 0.8$ over $\mu_i = 0.9$ until $\mu_i = 1.2$. All in all, the relative deviation decreases slightly if the system parameters change towards a better throughput. However, this effect is limited due to the good overall accuracy of the proposed decomposition.

Table 5. Parameter ranges for the different examined variables in the boxplots; $i = 1, \ldots, 5, j = 1, \ldots, 4$

Examined variable / Parameter range of	C_j	Q_i	μ_i	λ_i	γ_i
C_j	$\{10, 15, 20, 25, 30\}$	20	20	20	20
Q_i	2	$\{1, 2, 3, 4, 5\}$	2	2	2
μ_i	$\mathcal{U}_{[0.8,1.2]}$	$\mathcal{U}_{[0.8,1.2]}$	$\{0.8, 0.9, 1.0, 1.1, 1.2\}$	$\mathcal{U}_{[0.8,1.2]}$	$\mathcal{U}_{[0.8,1.2]}$
λ_i	$\mathcal{U}_{[0.004,0.006]}$	$\mathcal{U}_{[0.004,0.006]}$	$\mathcal{U}_{[0.004,0.006]}$	$\{0.007, 0.006, 0.005, 0.004, 0.003\}$	$\mathcal{U}_{[0.004,0.006]}$
γ_i	$\mathcal{U}_{[0.04,0.06]}$	$\mathcal{U}_{[0.04,0.06]}$	$\mathcal{U}_{[0.04,0.06]}$	$\mathcal{U}_{[0.04,0.06]}$	$\{0.03, 0.04, 0.05, 0.06, 0.07\}$

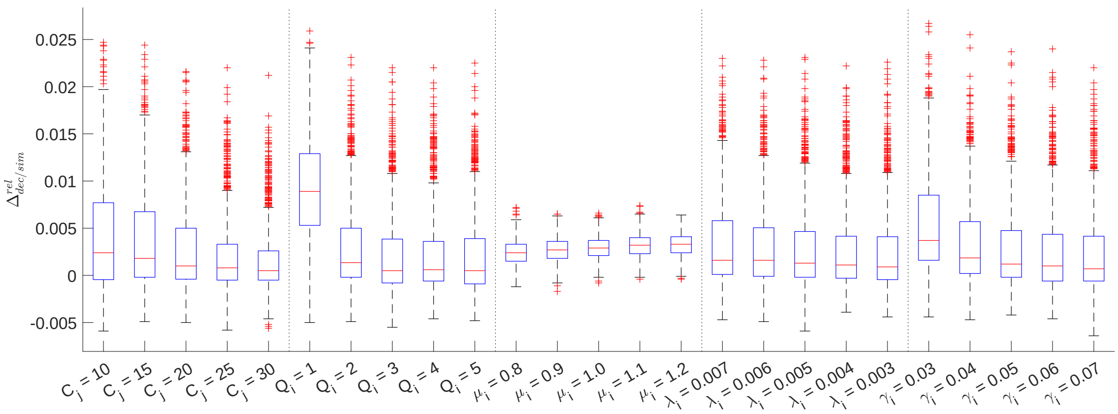


Figure 8. Boxplots of relative throughput deviations for in total 25,000 randomly generated instances of unbalanced flow lines

Moreover, we find that a good proportion of the results are within the 95% confidence intervals from the simulation. Table 6 shows the number of instances where the decomposition results are inside the confidence interval. As for the general accuracy, the numbers increase with an increasing system throughput. Hence, the decomposition works better for more effective systems, which are, after all, the practically relevant instances. Only for the processing rates can we observe a decrease. However, the maximum deviation from the confidence interval in the instances with $\mu_i = 1.2$ is no more than 0.0045 (on average, even only 0.0015) at an average throughput level obtained by simulation of 1.0970.

Table 6. Number of instances where the resulting throughput of the decomposition is inside the 95% confidence interval of the simulation (in total 1,000 instances per parameter); $i = 1, \ldots, 5, j = 1, \ldots, 4$

Parameter	Parameter value (number of instances with result inside 95% confidence interval)				
C_j	10	15	20	25	30
	(468)	(531)	(600)	(675)	(710)
Q_i	1	2	3	4	5
	(325)	(596)	(632)	(645)	(620)
μ_i	0.8	0.9	1	1.1	1.2
	(631)	(504)	(415)	(314)	(247)
λ_i	0.007	0.006	0.005	0.004	0.003
	(587)	(572)	(613)	(630)	(629)
γ_i	0.03	0.04	0.05	0.06	0.07
	(453)	(579)	(628)	(630)	(638)

Tables 7 and 8 list different cases for longer flow lines with more than five machines to further study the influence of the different parameters (Table 13 in the appendix, Section 3.5, shows all values including availabilities). We alter the buffer capacity between adjacent machines as $C_j \in \{10, 30\}$ and the number of available units of the critical components per machine as $Q_i \in \{2, 3, 4\}$ and additionally vary the number of machines $I \in \{5, 25, 45\}$ $(i \in \mathcal{I}, j \in \mathcal{J})$. The instances in both tables differ only in their replenishment rate. Table 7 uses $\gamma_i = 0.1$, whereas we have only one-tenth of the original replenishment rate in Table 8. The throughput is given for the simulation and the decomposition approach, as well as the actual and relative error compared to the simulation results. Additionally, we indicate whether the confidence interval contains the result of the decomposition approach. This is the case for most instances with five machines. The results clearly show that the throughput generated by the decomposition approach is very close to the simulated value in all cases with a higher replenishment rate (Table 7). The deviation is even less than 2% in the worst case. The deviation increases with the number of machines, which is in line with prior research.

One should also know under which conditions the method fails to give accurate throughput estimates. We observed that long lines consisting of machines with a very low availability seem to be difficult for the method to evaluate. Such a case is presented with the parameters in Table 8. For cases with only one spare part $(Q_i = 2)$ in combination with a relatively large number of machines (25 or 45), we observed relative deviations to the simulated throughput, e.g., of up to 28.1% for the case $C_j = 10, Q_i = 2, I = 45$. The simulated throughput here was 0.4801, which is substantially below the processing rates $\mu_i = 1.0$, while the decomposition computed 0.6149. In other words, we can say that the method becomes inaccurate for systems that show poor performance, and it gets quite accurate for systems that show a good performance due to sufficiently large buffer sizes and a sufficient number of spare parts. This behavior could be caused by the assumption that starving and blocking cannot happen concurrently for each of the machines. On the one hand, improved machine availabilities realized by a higher number of spare parts would be able to face this issue since the only remaining source of starving and blocking would be the variability in processing times. On the other hand, a higher buffer capacity also reduces the influence of failed machines, as starving and blocking will occur less frequently. We observe both when we compare Tables 7 and 8.

Table 7. TP_{dec} and $\Delta^{\text{rel}}_{\text{dec/sim}}$ (in brackets) for longer flow lines in case of high replenishment rates. Asterisks indicate that TP_{dec} is included in the 95% confidence interval of the simulation result; $\mu_i = 1, \lambda_i = 0.005, \gamma_i = 0.1, i \in \mathcal{I}, j \in \mathcal{J}$.

C_j	10			30		
Q_i	2	3	4	2	3	4
$I = 5$	0.8678 (−0.0005)*	0.8696 (0.0013)*	0.8696 (0.0020)*	0.9465 (0.0010)*	0.9481 (0.0008)*	0.9482 (0.0020)*
$I = 25$	0.8434 (0.0122)	0.8453 (0.0125)	0.8454 (0.0117)	0.9366 (0.0078)	0.9383 (0.0060)	0.9383 (0.0097)
$I = 45$	0.8366 (0.0101)	0.8386 (0.0086)	0.8386 (0.0071)	0.9294 (0.0047)	0.9311 (0.0042)	0.9312 (0.0043)

Table 8. TP_{dec} and $\Delta^{\text{rel}}_{\text{dec/sim}}$ (in brackets) for longer flow lines in case of low replenishment rates. Asterisks indicate that TP_{dec} is included in the 95% confidence interval of the simulation result; $\mu_i = 1, \lambda_i = 0.005, \gamma_i = 0.01, i \in \mathcal{I}, j \in \mathcal{J}$.

C_j	10			30		
Q_i	2	3	4	2	3	4
$I = 5$	0.7021 (0.0151)	0.8440 (0.0065)	0.8667 (0.0007)*	0.7711 (0.0360)	0.9202 (0.0083)	0.9449 (0.0015)*
$I = 25$	0.6220 (0.2091)	0.8145 (0.0699)	0.8419 (0.0166)	0.7226 (0.1624)	0.9068 (0.0561)	0.9347 (0.0126)
$I = 45$	0.6149 (0.2808)	0.8071 (0.1029)	0.8351 (0.0183)	0.7184 (0.1913)	0.9002 (0.0661)	0.9276 (0.0107)

In order to emphasize the relevance and applicability of the proposed decomposition, the previous synthetic results are broadened to some real-world examples documented by Tempelmeier (2003). Table 9 shows the performance of the decomposition. All relevant parameters as well as the availability are provided in the appendix (Section 3.5: Tables 14 and 15). System C has relatively low availabilities. Although the system is a transfer line with deterministic processing times, it is useful to generate insights regarding the class of less reliable flow lines. Hence, we take the given processing time as the mean processing time. One spare part was introduced for all adapted cases (C1, C2, C3, D1, D2, and D3). Additionally, the replenishment rates were adjusted to include that replenishment will usually take longer than an on-site repair. Thus, we have the same value for γ_i as in the baseline case for C1 and D1, $\frac{1}{2}\gamma_i$ for C2 and D2, and $\frac{1}{4}\gamma_i$ for C3 and D3. Comparing the throughput to the baseline cases clarifies that the introduction of spare parts increases the throughput.

Table 9. Results for adapted real-world flow lines with baseline cases C and D from Tempelmeier (2003). Asterisks indicate that TP_{dec} is included in the 95% confidence interval of the simulation result.

Case	TP_{dec}	TP_{sim}	$\Delta_{\text{dec/sim}}$	$\Delta^{\text{rel}}_{\text{dec/sim}}$	HW	Relative TP increase compared to baseline case
C	0.1905	0.1899	−0.0006	−0.0030	0.0008*	
$C1$	0.2081	0.2079	−0.0002	−0.0011	0.0012*	0.0926
$C2$	0.2057	0.2055	−0.0002	−0.0008	0.0010*	0.0800
$C3$	0.1951	0.1938	−0.0013	−0.0067	0.0009	0.0246
D	1.1994	1.1894	−0.0100	−0.0083	0.0035	
$D1$	1.2905	1.2757	−0.0148	−0.0115	0.0028	0.0760
$D2$	1.2848	1.2748	−0.0099	−0.0077	0.0025	0.0712
$D3$	1.2688	1.2700	0.0012	0.0009	0.0045*	0.0579

Overall, the decomposition tends to work better in cases with a realistically high throughput of the flow line, as the addressed effects all tend to negatively influence the throughput of the flow line and the accuracy of the decomposition. Hence, the presented decomposition can be expected to work well for cases that are of practical relevance.

3.3.3 Managerial insights

Our newly developed method enables us to evaluate different system designs. Since this is the first study where spare part stocks are allowed for flow lines of any length with intermediate buffers, we focus on the impact of the number of spare parts on the throughput. Moreover, in this preliminary study, we generate insights about the best position of an additional spare part, and we illustrate that less buffer capacity is necessary if spare parts are kept in stock. We start our discussion with balanced systems concerning all parameters except the base-stock level.

Previous research on stochastic flow lines suggested that buffers should be installed at the center or near the center of balanced flow lines (see, e.g., Hillier and Boling, 1979 or Conway et al., 1988). Table 10 lists different cases for the allocation of spare parts in a balanced flow line. Cases 1 through 6 illustrate how the first additional spare part should be allocated most efficiently. It unfolds that a spare part at or near the center of the flow line increases the throughput more than spare parts located at the beginning or end of the flow line.

Observation 3.1. *Additional spares should be located near the center of the flow line to improve throughput the most if the flow line is balanced. The spare part allocation tends to show a bowl effect.*

This observation holds for the situation where each machine already has one spare part in stock, and we wish to add an additional part (cases 7 to 12). A closer examination of the situation with two parts in stock for all machines (cases 13 to 18) reveals that the throughput can only be increased slightly by simply adding additional spare parts. The spare parts have already increased the isolated availability of the single machines to such an extent that the effect of more parts is negligible. In this situation, the throughput can only be increased by adding additional buffer capacity.

By analyzing cases 13 and 19 to 21, we observe that the equal distribution of a constant number of spare parts is beneficial (all cases have 15 spare parts in total). In summary, the bowl allocation (case 20), where we have more spare parts at the center and fewer at the boundaries, is here slightly less efficient than an equal allocation that allocates the rest of the spare parts equally near the center. This effect is caused by the availability of the center machine, as it has already reached 0.9928 and, therefore, additional spare parts are useless.

We now consider the case of unbalanced flow lines. The results in Table 11 tend in a similar direction. To obtain the results, we increased the failure rate of the second machine by 20% such that it becomes the bottleneck in the flow line. The first two groups of cases (1 to 6 and 7 to 12) indicate that the best position (in terms of throughput increase) for an additional spare would be the second machine. Only in the third group (cases 13 to 18), we observe that the center machine should receive an additional spare part. However, the isolated availabilities are already very similar.

The observations above can be shown to apply to a large set of similar situations. For this purpose, we have again created random sets of flow lines. A random sample of all stochastic parameters also means a massive sample of positions for different

Table 10. Different spare part allocations along the flow line. Asterisks indicate that TP_{dec} is included in the 95% confidence interval of the simulation result; $I = 5, C_j = 10, \mu_i = 1, \lambda_i = 0.02, \gamma_i = 0.05$ with $i \in \mathcal{I}, j \in \mathcal{J}$.

Case	Q_1	Q_2	Q_3	Q_4	Q_5	iAV_1	iAV_2	iAV_3	iAV_4	iAV_5	TP_{sim}	TP_{dec}	$\Delta_{\text{dec/sim}}$	$\Delta^{\text{rel}}_{\text{dec/sim}}$	HW
1	1	1	1	1	1	0.7143	0.7143	0.7143	0.7143	0.7143	0.4194	0.4163	−0.0031	−0.0074	0.0045*
2	2	1	1	1	1	0.9459	0.7143	0.7143	0.7143	0.7143	0.4434	0.4411	−0.0023	−0.0051	0.0037*
3	1	2	1	1	1	0.7143	0.9459	0.7143	0.7143	0.7143	0.4558	0.4678	0.0121	0.0264	0.0053
4	1	1	2	1	1	0.7143	0.7143	0.9459	0.7143	0.7143	0.4583	0.4658	0.0075	0.0164	0.0023
5	1	1	1	2	1	0.7143	0.7143	0.7143	0.9459	0.7143	0.4553	0.4748	0.0195	0.0428	0.0027
6	1	1	1	1	2	0.7143	0.7143	0.7143	0.7143	0.9459	0.4451	0.4412	−0.0039	−0.0089	0.0045*
7	2	2	2	2	2	0.9459	0.9459	0.9459	0.9459	0.9459	0.7617	0.7709	0.0092	0.0120	0.0038
8	3	2	2	2	2	0.9928	0.9459	0.9459	0.9459	0.9459	0.7752	0.7806	0.0054	0.0069	0.0023
9	2	3	2	2	2	0.9459	0.9928	0.9459	0.9459	0.9459	0.7815	0.7889	0.0074	0.0095	0.0022
10	2	2	3	2	2	0.9459	0.9459	0.9928	0.9459	0.9459	0.7829	0.7896	0.0067	0.0085	0.0032
11	2	2	2	3	2	0.9459	0.9459	0.9459	0.9928	0.9459	0.7790	0.7861	0.0071	0.0091	0.0043
12	2	2	2	2	3	0.9459	0.9459	0.9459	0.9459	0.9928	0.7730	0.7804	0.0074	0.0096	0.0041
13	3	3	3	3	3	0.9928	0.9928	0.9928	0.9928	0.9928	0.8567	0.8577	0.0011	0.0013	0.0025*
14	4	3	3	3	3	0.9993	0.9928	0.9928	0.9928	0.9928	0.8577	0.8591	0.0014	0.0016	0.0029*
15	3	4	3	3	3	0.9928	0.9993	0.9928	0.9928	0.9928	0.8574	0.8605	0.0032	0.0037	0.0024
16	3	3	4	3	3	0.9928	0.9928	0.9993	0.9928	0.9928	0.8578	0.8607	0.0030	0.0035	0.0017
17	3	3	3	4	3	0.9928	0.9928	0.9928	0.9993	0.9928	0.8605	0.8607	0.0002	0.0002	0.0013*
18	3	3	3	3	4	0.9928	0.9928	0.9928	0.9928	0.9993	0.8591	0.8591	0.0000	0.0000	0.0023*
19	4	4	3	2	2	0.9993	0.9993	0.9928	0.9459	0.9459	0.8153	0.8197	0.0044	0.0053	0.0039
20	2	3	5	3	2	0.9459	0.9928	0.9999	0.9928	0.9459	0.8193	0.8316	0.0123	0.0151	0.0031
21	2	2	3	4	4	0.9459	0.9459	0.9928	0.9993	0.9993	0.8160	0.8188	0.0028	0.0034	0.0026

Table 11. Different spare part allocations along the flow line with second machine as bottleneck. Asterisks indicate that TP_{dec} is included in the 95% confidence interval of the simulation result; $I = 5, C_j = 10, \mu_i = 1, \lambda_k = 0.02, \gamma_i = 0.05$ with $i \in \mathcal{I}, j \in \mathcal{J}, k = 1, 3, 4, 5$ and $\lambda_2 = 0.024$.

Case	Q_1	Q_2	Q_3	Q_4	Q_5	iAV_1	iAV_2	iAV_3	iAV_4	iAV_5	TP_{sim}	TP_{dec}	$\Delta_{\text{dec/sim}}$	$\Delta^{\text{rel}}_{\text{dec/sim}}$	HW
1	1	1	1	1	1	0.7143	0.6757	0.7143	0.7143	0.7143	0.4073	0.4083	0.0010	0.0024	0.0048*
2	2	1	1	1	1	0.9459	0.6757	0.7143	0.7143	0.7143	0.4358	0.4326	−0.0032	−0.0074	0.0042*
3	1	2	1	1	1	0.7143	0.9278	0.7143	0.7143	0.7143	0.4542	0.4641	0.0098	0.0216	0.0042
4	1	1	2	1	1	0.7143	0.6757	0.9459	0.7143	0.7143	0.4499	0.4607	0.0108	0.0241	0.0042
5	1	1	1	2	1	0.7143	0.6757	0.7143	0.9459	0.7143	0.4446	0.4623	0.0176	0.0396	0.0028
6	1	1	1	1	2	0.7143	0.6757	0.7143	0.7143	0.9459	0.4338	0.4304	−0.0034	−0.0079	0.0049*
7	2	2	2	2	2	0.9459	0.9278	0.9459	0.9459	0.9459	0.7568	0.7632	0.0065	0.0085	0.0026
8	3	2	2	2	2	0.9928	0.9278	0.9459	0.9459	0.9459	0.7662	0.7735	0.0073	0.0096	0.0029
9	2	3	2	2	2	0.9459	0.9886	0.9459	0.9459	0.9459	0.7764	0.7925	0.0162	0.0208	0.0039
10	2	2	3	2	2	0.9459	0.9278	0.9928	0.9459	0.9459	0.7729	0.7815	0.0086	0.0111	0.0024
11	2	2	2	3	2	0.9459	0.9278	0.9459	0.9928	0.9459	0.7726	0.7776	0.0050	0.0064	0.0026
12	2	2	2	2	3	0.9459	0.9278	0.9459	0.9459	0.9928	0.7655	0.7719	0.0064	0.0084	0.0033
13	3	3	3	3	3	0.9928	0.9886	0.9928	0.9928	0.9928	0.8541	0.8560	0.0020	0.0023	0.0030*
14	4	3	3	3	3	0.9993	0.9886	0.9928	0.9928	0.9928	0.8581	0.8574	−0.0007	−0.0008	0.0026*
15	3	4	3	3	3	0.9928	0.9986	0.9928	0.9928	0.9928	0.8590	0.8605	0.0015	0.0018	0.0016*
16	3	3	4	3	3	0.9928	0.9886	0.9993	0.9928	0.9928	0.8571	0.8611	0.0040	0.0047	0.0025
17	3	3	3	4	3	0.9928	0.9886	0.9928	0.9993	0.9928	0.8566	0.8582	0.0017	0.0019	0.0021*
18	3	3	3	3	4	0.9928	0.9886	0.9928	0.9928	0.9993	0.8579	0.8574	−0.0005	−0.0006	0.0018*
19	4	4	3	2	2	0.9993	0.9986	0.9928	0.9459	0.9459	0.8143	0.8209	0.0066	0.0081	0.0029
20	2	3	5	3	2	0.9459	0.9886	0.9999	0.9928	0.9459	0.8185	0.8312	0.0128	0.0156	0.0024
21	2	2	3	4	4	0.9459	0.9278	0.9928	0.9993	0.9993	0.8049	0.8083	0.0034	0.0042	0.0032

bottleneck machines. Because of the evaluative nature of the method presented in this chapter, an additional optimization methodology cannot be reasonably described and analyzed in our setting and shall be subject to future research. Hence, we must reduce the complexity by randomly changing one specific machine's parameters. Thereby, the bottleneck can be identified and enables us to investigate the impact of different bottleneck positions on reasonable spare part allocations. The parameter ranges are as follows for $I = 5$ and a bottleneck $i_b \in \{1, \dots, 5\}$

$$C_j = 20, Q_i = 2, \mu_i = 1, \lambda_i = 0.005, \gamma_i = 0.05 \qquad \forall j \in \mathcal{J}, i \in \mathcal{I} \setminus \{i_b\} \quad (45)$$

$$\mu_{i_b} \in [0.8, 0.9], \lambda_{i_b} \in [0.0055, 0.006], \gamma_{i_b} \in [0.04, 0.045] \quad (46)$$

To produce noticeable changes, we worsen (increase or decrease, respectively) the parameters by a minimum of ten percent and again up to 20 percent. Thus, we have five machines, whereby four machines are similar, and one machine is the bottleneck. We created 500 sets of parameters for each bottleneck machine, i.e., 2,500 instances in total. We used the same random set for each of the machines and evaluated which spare part allocations yield the highest throughput. We compare which of the cases with one additional spare part at one of the five machines performs best, e.g., $Q_1 = 3$ and $Q_i = 2, i = 2, 3, 4, 5$. In all cases, the additional spare part allocated at the bottleneck position yields the highest throughput.

Observation 3.2. *The best position for an additional spare part for a flow line with a bottleneck is the stock point near it. This positioning leads to the highest increase in throughput.*

This outcome alone is not surprising. However, quantifying the system behavior creates the possibility of later including cost considerations for optimization purposes.

Finally, the new possibilities to achieve target throughputs are investigated with the help of the examples of five-machine lines shown in Table 12. Cases A1 and A2 plus B1 to B3 are chosen to guarantee a target throughput of 80% and 86%, respectively. Small deviations can occur due to the discrete values of the buffer levels and spare part stocks. There are two main results. First, it is clear that spare parts drastically influence the performance of the flow line. In the case A1, not a single spare part is in stock, whereas in the case A2, the base-stock level for the spare parts is increased to one. As a result, the buffer capacity can be reduced by more than 86% from 45 to 6 units per buffer,

Table 12. Different flow lines with similar throughput. Asterisks indicate that TP_{dec} is included in the 95% confidence interval of the simulation result; $I = 5, \mu_i = 1, \lambda_i = 0.005, \gamma_i = 0.05$ with $i \in \mathcal{I}, j \in \mathcal{J}$.

Case	C_j	Q_i	iAV_i	TP_{sim}	TP_{dec}	$\Delta_{\mathsf{dec/sim}}$	$\Delta^{\mathsf{rel}}_{\mathsf{dec/sim}}$	HW
$A1$	45	1	0.9091	0.7951	0.8016	0.0065	0.0082	0.0033
$A2$	6	2	0.9955	0.8065	0.8044	−0.0021	−0.0026	0.0029*
$B1$	120	1	0.9091	0.8581	0.8603	0.0021	0.0025	0.0052*
$B2$	10	2	0.9955	0.8592	0.8614	0.0022	0.0026	0.0022*
$B3$	10	3	0.9998	0.8685	0.8693	0.0008	0.0010	0.0023*

while a slightly higher throughput can be achieved. Similarly, cases B1 to B3 support this finding for a higher target throughput. Second, the latter cases also indicate that situations exist in which spare parts and buffers should be considered simultaneously. The additional spare part (case B2 compared to B3) facilitates no buffer reduction at all. Obviously, a combination of buffers and spare parts is needed to guarantee specific throughput levels. Spare part provisioning increases machine availability, e.g., from approximately 0.91 to over 0.99 (case A1 compared to A2), but does not hedge against processing times variability. Therefore, some buffers are still inevitable. These findings lead to our final observation.

Observation 3.3. *There exists a limited trade-off between the buffer capacity and the spare part base-stock levels. Both are only substitutable to a limited extent, which heavily depends on the system characteristics.*

However, to determine the optimal design of a manufacturing system in terms of optimal buffer capacities and spare part stock levels, cost considerations must also be taken into account, which will be a topic for future research.

3.4 Summary

In this chapter, we investigated how performance measures of long flow lines with buffers as well as spare parts for critical machine components can be determined. To facilitate the analysis, processing times, component lifetimes, and spare part replenishment times are assumed to be exponentially distributed. The two-machine flow line is modeled as a continuous-time Markov chain, such that the steady-state behavior and the relevant performance measures can be determined numerically. Using this building block, a decomposition approach is presented to analyze longer flow lines and to determine relevant performance measures. We examine the algorithmic behavior as a function of the flow line parameters. Furthermore, the influence of buffer capacity, spare part stock sizes, and distribution parameters on the flow line performance are studied.

For the cases we studied, the results of the decomposition approach are very accurate compared to the exact results from the Markov chain approach for three-machine lines. The deviations of the throughput are considerably less than 1% in all instances, and the approximation is fast. Even for worst-case scenarios with unrealistically high probabilities of simultaneous blocking and starving, the resulting performance measures differ only to a limited extent. Similar results hold for longer flow lines compared to a simulation study: The deviations of the throughput are slight for many cases of practical relevance. Overall, the proposed solution appears to be promising for further applications and performs very well, especially for economically efficient balanced flow lines.

We were able to generate some first insights into the allocation of spare parts in longer flow lines with interstage buffers. When aiming for maximum throughput in balanced flow lines, spare parts tend to show a bowl effect as buffers do: A centered position for additional spare parts is more beneficial than positioning it at the beginning or end of the flow line. Furthermore, we illustrated that situations exist where buffer capacities and spare parts are substitutable for reaching target throughput levels. However, this substitution is limited depending on the system's properties.

This chapter lays the foundation to focus on the design problem: How many units of buffer capacity and spare parts should be provided to achieve a specific throughput? The proposed model and solution technique are useful elements for answering this question in the next chapter.

3.5 Appendix

3.5.1 Proofs for FRIT

Proof for Theorem 1. For machine i, we can write the throughput as

$$
\begin{aligned}
TP_i =& \mu_i \cdot P(n_{i-1} > 0, n_i < N_i, \alpha_i \geq 1) && (47)\\
=& \mu_i \cdot P(n_{i-1} > 0, n_i < N_i, \alpha_i \geq 1) \cdot \frac{P(n_{i-1} > 0, n_i < N_i, \alpha_i \geq 0)}{P(n_{i-1} > 0, n_i < N_i, \alpha_i \geq 0)} && (48)\\
=& \mu_i \cdot \frac{P(n_{i-1} > 0, n_i < N_i, \alpha_i \geq 1)}{P(n_{i-1} > 0, n_i < N_i, \alpha_i \geq 0)} && (49)\\
& \cdot \Big(1 - P(n_{i-1} = 0, n_i < N_i, \alpha_i \geq 1) - P(n_{i-1} > 0, n_i = N_i, \alpha_i \geq 1)\\
& + P(n_{i-1} = 0, n_i = N_i, \alpha_i \geq 1)\Big)\\
\overset{(A1)}{\approx}& \mu_i \cdot \frac{P(n_{i-1} > 0, n_i < N_i, \alpha_i \geq 1)}{P(n_{i-1} > 0, n_i < N_i, \alpha_i \geq 0)} && (50)\\
& \cdot \Big(1 - P(n_{i-1} = 0, n_i < N_i, \alpha_i \geq 1) - P(n_{i-1} > 0, n_i = N_i, \alpha_i \geq 1)\Big)\\
=& \mu_i \cdot \frac{P(n_{i-1} > 0, n_i < N_i, \alpha_i \geq 1)}{P(n_{i-1} > 0, n_i < N_i, \alpha_i = 0) + P(n_{i-1} > 0, n_i < N_i, \alpha_i \geq 1)} && (51)\\
& \cdot \Big(1 - P(n_{i-1} = 0, n_i < N_i, \alpha_i \geq 1) - P(n_{i-1} > 0, n_i = N_i, \alpha_i \geq 1)\Big)
\end{aligned}
$$

and with the auxiliary quantity

$$
A_i := \frac{P(n_{i-1} > 0, n_i < N_i, \alpha_i \geq 1)}{P(n_{i-1} > 0, n_i < N_i, \alpha_i = 0)} \quad (52)
$$

this reduces to

$$
\begin{aligned}
TP_i \quad =& \quad \mu_i \cdot \frac{A_i}{A_i + 1} \cdot \Big(1 - P(n_{i-1} = 0, n_i < N_i, \alpha_i \geq 1)\\
& \quad - P(n_{i-1} > 0, n_i = N_i, \alpha_i \geq 1)\Big) && (53)\\
\overset{(A2),(A3)}{\approx}& \quad \mu_i \cdot \frac{A_i}{A_i + 1} \cdot \Big(1 - \mathcal{P}_{i-1}(0, \alpha_u \geq 0, \alpha_d \geq 1) - \mathcal{P}_i(N_i, \alpha_u \geq 1, \alpha_d \geq 0)\Big)\\
& && (54)
\end{aligned}
$$

Since the probabilities used for the definition of A_i are not known, they are approximated in terms of probabilities of the virtual two-machine line. First, by analyzing the two-machine Markov chain, we find $P(n_{i-1} > 0, n_i < N_i, \alpha_i = j)$ for $j = 1, 2, \ldots, Q_i$ and $i = 2, \ldots, I-1$. We start with a level crossing argument in terms of states in which either no or exactly one functional unit of machine i's critical component is available. The "boundary" between those two levels ($\alpha_i = 0$ and $\alpha_i = 1$) must be crossed equally often in the long term:

$$\lambda_i \cdot P(n_{i-1} > 0, n_i < N_i, \alpha_i = 1) = Q_i \cdot \gamma_i \cdot P(n_{i-1} > 0, n_i < N_i, \alpha_i = 0) \quad (55)$$

$$\Leftrightarrow \quad P(n_{i-1} > 0, n_i < N_i, \alpha_i = 1) = Q_i \cdot \frac{\gamma_i}{\lambda_i} \cdot P(n_{i-1} > 0, n_i < N_i, \alpha_i = 0). \quad (56)$$

We further find a balance equation for transitions between states with two and one available component(s) by inserting the previous equation that

$$\begin{aligned}
& \lambda_i \cdot P(n_{i-1} > 0, n_i < N_i, \alpha_i = 2) && (57)\\
& = (Q_i - 1)\gamma_i \cdot P(n_{i-1} > 0, n_i < N_i, \alpha_i = 1)\\
& + (Q_i - 1)\gamma_i \cdot P(n_{i-1} = 0, n_i < N_i, \alpha_i = 1)\\
& + (Q_i - 1)\gamma_i \cdot P(n_{i-1} > 0, n_i = N_i, \alpha_i = 1)\\
\Leftrightarrow \quad & P(n_{i-1} > 0, n_i < N_i, \alpha_i = 2) && (58)\\
& = (Q_i - 1)\frac{\gamma_i}{\lambda_i} \cdot P(n_{i-1} > 0, n_i < N_i, \alpha_i = 1)\\
& + (Q_i - 1)\frac{\gamma_i}{\lambda_i} \cdot P(n_{i-1} = 0, n_i < N_i, \alpha_i = 1)\\
& + (Q_i - 1)\frac{\gamma_i}{\lambda_i} \cdot P(n_{i-1} > 0, n_i = N_i, \alpha_i = 1)\\
\Leftrightarrow \quad & P(n_{i-1} > 0, n_i < N_i, \alpha_i = 2) && (59)\\
& = Q_i(Q_i - 1)\left(\frac{\gamma_i}{\lambda_i}\right)^2 \cdot P(n_{i-1} > 0, n_i < N_i, \alpha_i = 0)\\
& + (Q_i - 1)\frac{\gamma_i}{\lambda_i} \cdot \Big(P(n_{i-1} = 0, n_i < N_i, \alpha_i = 1) + P(n_{i-1} > 0, n_i = N_i, \alpha_i = 1)\Big)
\end{aligned}$$

Eventually, one can formulate the following induction hypothesis for $j = 1, 2, \ldots, Q_i - 1$

$$\begin{aligned}
&P(n_{i-1} > 0, n_i < N_i, \alpha_i = j) \\
&= \prod_{m=0}^{j-1} (Q_i - m) \cdot \left(\frac{\gamma_i}{\lambda_i}\right)^j P(n_{i-1} > 0, n_i < N_i, \alpha_i = 0) \\
&\quad + \sum_{k=1}^{j-1} \prod_{m=k}^{j-1} (Q_i - m) \cdot \left(\frac{\gamma_i}{\lambda_i}\right)^{j-k} \cdot \\
&\qquad \left(P(n_{i-1} = 0, n_i < N_i, \alpha_i = k) + P(n_{i-1} > 0, n_i = N_i, \alpha_i = k)\right),
\end{aligned} \tag{60}$$

with $\sum_{m=1}^{0} := 0$. The base case was already done with $j = 1, 2$ since $j = 1$ is a special case. Then, the inductive step is as follows:

$$\begin{aligned}
&\lambda_i \cdot P(n_{i-1} > 0, n_i < N_i, \alpha_i = j+1) \\
&= (Q_i - j)\gamma_i \cdot P(n_{i-1} > 0, n_i < N_i, \alpha_i = j) \\
&+ (Q_i - j)\gamma_i \cdot P(n_{i-1} = 0, n_i < N_i, \alpha_i = j) \\
&+ (Q_i - j)\gamma_i \cdot P(n_{i-1} > 0, n_i = N_i, \alpha_i = j)
\end{aligned} \tag{61}$$

$$\begin{aligned}
\Leftrightarrow \quad &P(n_{i-1} > 0, n_i < N_i, \alpha_i = j+1) \\
&= (Q_i - j)\frac{\gamma_i}{\lambda_i} \cdot P(n_{i-1} > 0, n_i < N_i, \alpha_i = j) \\
&+ (Q_i - j)\frac{\gamma_i}{\lambda_i} \cdot P(n_{i-1} = 0, n_i < N_i, \alpha_i = j) \\
&+ (Q_i - j)\frac{\gamma_i}{\lambda_i} \cdot P(n_{i-1} > 0, n_i = N_i, \alpha_i = j).
\end{aligned} \tag{62}$$

The induction hypothesis (60) yields

$$\begin{aligned}
&P(n_{i-1} > 0, n_i < N_i, \alpha_i = j+1) \\
=&(Q_i - j)\frac{\gamma_i}{\lambda_i} \cdot \Bigg(\prod_{m=0}^{j-1} (Q_i - m) \cdot \left(\frac{\gamma_i}{\lambda_i}\right)^j P(n_{i-1} > 0, n_i < N_i, \alpha_i = 0) \\
&+ \sum_{k=1}^{j-1} \prod_{m=k}^{j-1} (Q_i - m) \cdot \left(\frac{\gamma_i}{\lambda_i}\right)^{j-k} \cdot \\
&\qquad \left(P(n_{i-1} = 0, n_i < N_i, \alpha_i = k) + P(n_{i-1} > 0, n_i = N_i, \alpha_i = k)\right) \Bigg) \\
&+ (Q_i - j)\frac{\gamma_i}{\lambda_i} \cdot P(n_{i-1} = 0, n_i < N_i, \alpha_i = j) \\
&+ (Q_i - j)\frac{\gamma_i}{\lambda_i} \cdot P(n_{i-1} > 0, n_i = N_i, \alpha_i = j),
\end{aligned}$$

whereat the factors can be incorporated into the products such that we obtain

$$
\begin{aligned}
&P(n_{i-1} > 0, n_i < N_i, \alpha_i = j+1) \\
&= \prod_{m=0}^{(j+1)-1} (Q_i - m) \cdot \left(\frac{\gamma_i}{\lambda_i}\right)^{(j+1)} P(n_{i-1} > 0, n_i < N_i, \alpha_i = 0) \\
&\quad + \sum_{k=1}^{j-1} \prod_{m=k}^{(j+1)-1} (Q_i - m) \cdot \left(\frac{\gamma_i}{\lambda_i}\right)^{(j+1)-k} \\
&\qquad (P(n_{i-1} = 0, n_i < N_i, \alpha_i = k) + P(n_{i-1} > 0, n_i = N_i, \alpha_i = k)) \\
&\quad + (Q_i - j)\frac{\gamma_i}{\lambda_i} \cdot P(n_{i-1} = 0, n_i < N_i, \alpha_i = j) \\
&\quad + (Q_i - j)\frac{\gamma_i}{\lambda_i} \cdot P(n_{i-1} > 0, n_i = N_i, \alpha_i = j).
\end{aligned}
$$

Finally, the last two summands are put into the sum, and the induction is finished:

$$
\begin{aligned}
&P(n_{i-1} > 0, n_i < N_i, \alpha_i = j+1) \\
&= \prod_{m=0}^{(j+1)-1} (Q_i - m) \cdot \left(\frac{\gamma_i}{\lambda_i}\right)^{(j+1)} P(n_{i-1} > 0, n_i < N_i, \alpha_i = 0) \\
&\quad + \sum_{k=1}^{(j+1)-1} \prod_{m=k}^{(j+1)-1} (Q_i - m) \cdot \left(\frac{\gamma_i}{\lambda_i}\right)^{(j+1)-k} \\
&\qquad (P(n_{i-1} = 0, n_i < N_i, \alpha_i = k) + P(n_{i-1} > 0, n_i = N_i, \alpha_i = k)) \Big)
\end{aligned}
$$

Because (60) was shown to hold for all $Q_i \geq 1$ and $j = 1, 2, \ldots, Q_i$, we can reformulate

$$
A_i = \frac{P(n_{i-1} > 0, n_i < N_i, \alpha_i \geq 1)}{P(n_{i-1} > 0, n_i < N_i, \alpha_i = 0)} \tag{63}
$$

$$
= \sum_{j=1}^{Q_i} \frac{P(n_{i-1} > 0, n_i < N_i, \alpha_i = j)}{P(n_{i-1} > 0, n_i < N_i, \alpha_i = 0)} \tag{64}
$$

and further

$$A_i = \sum_{j=1}^{Q_i} \prod_{m=0}^{j-1} (Q_i - m) \cdot \left(\frac{\gamma_i}{\lambda_i}\right)^j \cdot \frac{P(n_{i-1} > 0, n_i < N_i, \alpha_i = 0)}{P(n_{i-1} > 0, n_i < N_i, \alpha_i = 0)} \tag{65}$$

$$+ \sum_{j=1}^{Q_i} \sum_{k=1}^{j-1} \prod_{m=k}^{j-1} (Q_i - m) \cdot \left(\frac{\gamma_i}{\lambda_i}\right)^{j-k}$$

$$\cdot \left(\frac{P(n_{i-1} = 0, n_i < N_i, \alpha_i = k)}{P(n_{i-1} > 0, n_i < N_i, \alpha_i = 0)} + \frac{P(n_{i-1} > 0, n_i = N_i, \alpha_i = k)}{P(n_{i-1} > 0, n_i < N_i, \alpha_i = 0)}\right)$$

$$= \sum_{j=1}^{Q_i} \prod_{m=0}^{j-1} (Q_i - m) \cdot \left(\frac{\gamma_i}{\lambda_i}\right)^j \tag{66}$$

$$+ \sum_{j=1}^{Q_i} \sum_{k=1}^{j-1} \prod_{m=k}^{j-1} (Q_i - m) \cdot \left(\frac{\gamma_i}{\lambda_i}\right)^{j-k}$$

$$\cdot \left(\frac{P(n_{i-1} = 0, n_i < N_i, \alpha_i = k)}{P(n_{i-1} > 0, n_i < N_i, \alpha_i = 0)} + \frac{P(n_{i-1} > 0, n_i = N_i, \alpha_i = k)}{P(n_{i-1} > 0, n_i < N_i, \alpha_i = 0)}\right)$$

$$\overset{(A2),(A3)}{\approx} \sum_{j=1}^{Q_i} \prod_{m=0}^{j-1} (Q_i - m) \cdot \left(\frac{\gamma_i}{\lambda_i}\right)^j + \sum_{j=1}^{Q_i} \sum_{k=1}^{j-1} \prod_{m=k}^{j-1} (Q_i - m) \cdot \left(\frac{\gamma_i}{\lambda_i}\right)^{j-k} \cdot \tag{67}$$

$$\left(\frac{\mathcal{P}_{i-1}(0, \alpha_u \geq 0, k)}{\mathcal{P}_{i-1}(n > 0, \alpha_u \geq 0, 0)} + \frac{\mathcal{P}_i(N_i, k, \alpha_d \geq 0)}{\mathcal{P}_i(n < N_i, 0, \alpha_d \geq 0)}\right)$$

□

Proof for Corollary 2. Applying (19) for virtual flow line i yields

$$\frac{TP(i)}{\mu_u(i)} \cdot \frac{1 + A_u(i)}{A_u(i)} = 1 - \mathcal{P}_i(N_i, \alpha_u \geq 1, \alpha_d \geq 0) \tag{68}$$

because virtual upstream machine $M_u(i)$ can never starve. For the virtual downstream machine of line $i-1$, i.e., $M_d(i-1)$, we obtain

$$\frac{TP(i-1)}{\mu_d(i-1)} \cdot \frac{1 + A_d(i-1)}{A_d(i-1)} = 1 - \mathcal{P}_{i-1}(0, \alpha_u \geq 0, \alpha_d \geq 1) \tag{69}$$

since this machine cannot be blocked. The constants for the virtual machines are given as

$$A_u(i) = \sum_{j=1}^{Q_i} \prod_{m=0}^{j-1} (Q_i - m) \cdot \left(\frac{\gamma_u(i)}{\lambda_u(i)}\right)^j + \sum_{j=1}^{Q_i} \sum_{k=1}^{j-1} \prod_{m=k}^{j-1} (Q_i - m) \cdot \left(\frac{\gamma_u(i)}{\lambda_u(i)}\right)^{j-k} \cdot \left(\frac{\mathcal{P}_i(N_i, k, \alpha_d \geq 0)}{\mathcal{P}_i(n < N_i, 0, \alpha_d \geq 0)}\right), \tag{70}$$

$$A_d(i-1) = \sum_{j=1}^{Q_i} \prod_{m=0}^{j-1} (Q_i - m) \cdot \left(\frac{\gamma_d(i-1)}{\lambda_d(i-1)}\right)^j + \sum_{j=1}^{Q_i} \sum_{k=1}^{j-1} \prod_{m=k}^{j-1} (Q_i - m) \cdot \left(\frac{\gamma_d(i-1)}{\lambda_d(i-1)}\right)^{j-k} \cdot \left(\frac{\mathcal{P}_{i-1}(0, \alpha_u \geq 0, k)}{\mathcal{P}_{i-1}(n > 0, \alpha_u \geq 0, 0)}\right). \tag{71}$$

due to (20). Therefore, we use the result of Theorem 1 and reformulate with the help of (68) and (69)

$$\frac{TP}{\mu_i} \cdot \frac{A_i + 1}{A_i} = 1 - \mathcal{P}_{i-1}(0, \alpha_u \geq 0, \alpha_d \geq 1) + 1 - \mathcal{P}_i(N_i, \alpha_u \geq 1, \alpha_d \geq 0) - 1 \tag{72}$$

$$= \frac{TP(i)}{\mu_u(i)} \cdot \frac{1 + A_u(i)}{A_u(i)} + \frac{TP(i-1)}{\mu_d(i-1)} \cdot \frac{1 + A_d(i-1)}{A_d(i-1)} - 1 \tag{73}$$

The results are then given by applying (12). □

3.5.2 Proof for IOF

Proof for Theorem 3. **Definition of virtual machine states.** To derive the *Interruption of Flow* equations for failure rates and the *Resumption of Flow* equations for the replenishment rates, we need to define when the upstream and downstream machines of the virtual machine lines are said to have a certain number of operational units of their respective failure-prone component, thus being either up or down.

(D1) $M_u(i)$ is down if no material is flowing into the buffer B_i due to a failure of an upstream machine. Hence, $M_u(i)$ is down if M_i is down or M_i is not down, but $n_{i-1} = 0$ (i.e., $M_d(i-1)$ is starving) and $M_u(i-1)$ is down:

$$\begin{aligned} &\alpha_u(i,t) = 0 \\ \Leftrightarrow\ &(n_{i-1}(t) > 0 \text{ and } \alpha_i(t) = 0) \text{ or} \\ &(n_{i-1}(t) = 0 \text{ and } \alpha_i(t) \geq 1 \text{ and } \alpha_u(i-1,t) = 0) \end{aligned} \tag{74}$$

Note that $\alpha_i(t) = 0$ implies $n_{i-1}(t) > 0$ since machine M_i cannot fail while it is starved. Similarly, $\alpha_u(i-1,t) = 0$ and $n_{i-1}(t) = 0$ implies that $\alpha_i(t) \geq 1$, i.e., we can have more than one operational unit at machine M_i.

A single replenishment at either M_i (if we have $\alpha_i(t) = 0$) or $M_u(i)$ (if we have $\alpha_u(i-1,t) = 0$ and $n_{i-1}(t) = 0$) is sufficient to leave a state with $\alpha_u(i,t) = 0$ again.

(D2) $M_u(i)$ is up if we have either $\alpha_u(i,t) = 1$ or $\alpha_u(i,t) > 1$. A state with $\alpha_u(i,t) = 1$ is one in which a single failure of real machine M_i or virtual machine $M_u(i-1)$ is sufficient to reach a state with $\alpha_u(i,t) = 0$:

$$\begin{aligned} &\alpha_u(i,t) = 1 \\ \Leftrightarrow\ &(n_{i-1}(t) > 0 \text{ and } \alpha_i(t) = 1) \text{ or} \\ &(n_{i-1}(t) = 0 \text{ and } \alpha_i(t) \geq 1 \text{ and } \alpha_u(i-1,t) = 1) \end{aligned} \tag{75}$$

There are other states for which a single failure is not sufficient to reach a state with $\alpha_u(i,t) = 0$ and that cannot be reached via a single replacement arrival from states with $\alpha_u(i,t) = 0$:

$$\alpha_u(i,t) > 1 \Leftrightarrow \alpha_u(i,t) \neq 0 \text{ and } \alpha_u(i,t) \neq 1 \tag{76}$$

(D3) Analogously, $M_d(i)$ is down if no material is flowing out of the buffer B_i due to a failure of a downstream machine. Hence, $M_d(i)$ is down if M_{i+1} is down or M_{i+1} is not down, but $n_{i+1} = N_{i+1}$ (i.e., $M_u(i+1)$ is blocked) and $M_d(i+1)$

is down:

$$\begin{aligned} & \alpha_d(i,t) = 0 \\ \Leftrightarrow\ & (n_{i+1}(t) < N_{i+1} \text{ and } \alpha_{i+1}(t) = 0) \text{ or} \\ & (n_{i+1}(t) = N_{i+1} \text{ and } \alpha_{i+1}(t) \geq 1 \text{ and } \alpha_d(i+1,t) = 0) \end{aligned} \tag{77}$$

Note that $\alpha_{i+1}(t) = 0$ implies $n_{i+1}(t) < N_{i+1}$ since machine M_{i+1} cannot fail while it is blocked. Similarly, $\alpha_d(i+1,t) = 0$ and $n_{i+1}(t) = N_{i+1}$ implies that $\alpha_{i+1}(t) \geq 1$, i.e., we can have more than one operational unit at machine M_{i+1}.

(D4) $M_d(i)$ is up if we have either $\alpha_d(i,t) = 1$ or $\alpha_d(i,t) > 1$. A state with $\alpha_d(i,t) = 1$ is one in which a single failure of the real machine M_{i+1} or the virtual machine $M_d(i+1)$ is sufficient to reach a state with $\alpha_d(i,t) = 0$:

$$\begin{aligned} & \alpha_d(i,t) = 1 \\ \Leftrightarrow\ & (n_{i+1}(t) < N_{i+1} \text{ and } \alpha_{i+1}(t) = 1) \text{ or} \\ & (n_{i+1}(t) = N_{i+1} \text{ and } \alpha_{i+1}(t) \geq 1 \text{ and } \alpha_d(i+1,t) = 1) \end{aligned} \tag{78}$$

There are other states for which a single failure is not sufficient to reach a state with $\alpha_d(i,t) = 0$ and that cannot be reached via a single replacement arrival from states with $\alpha_d(i,t) = 0$:

$$\alpha_d(i,t) > 1 \Leftrightarrow \alpha_d(i,t) \neq 0 \text{ and } \alpha_d(i,t) \neq 1 \tag{79}$$

We start with the IOF condition for $M_u(i)$ and apply definition (D1) such that

$$\lambda_u(i)\delta t = P(\alpha_u(i,t+\delta t)=0|n_i(t)<N_i,\alpha_u(i,t)=1) \tag{80}$$

$$\overset{(74)}{=} P(n_{i-1}(t+\delta t)>0,\alpha_i(t+\delta t)=0|n_i(t)<N_i,\alpha_u(i,t)=1) \tag{81}$$

$$+P(n_{i-1}(t+\delta t)=0,\alpha_i(t+\delta t)\geq 1,\alpha_u(i-1,t+\delta t)=0| \\ n_i(t)<N_i,\alpha_u(i,t)=1)$$

$$\overset{(75)}{=} P(n_{i-1}(t+\delta t)>0,\alpha_i(t+\delta t)=0|n_i(t)<N_i, \tag{82}$$

$$(n_{i-1}(t)>0,\alpha_i(t)=1) \text{ or } (n_{i-1}(t)=0,\alpha_i(t)\geq 1,\alpha_u(i-1,t)=1))$$

$$+P(n_{i-1}(t+\delta t)=0,\alpha_i(t+\delta t)\geq 1,\alpha_u(i-1,t+\delta t)=0| \\ n_i(t)<N_i,\alpha_u(i,t)=1)$$

$$=\lambda_i\delta t+P(n_{i-1}(t+\delta t)=0,\alpha_i(t+\delta t)\geq 1,\alpha_u(i-1,t+\delta t)=0| \\ n_i(t)<N_i,\alpha_u(i,t)=1). \tag{83}$$

To handle the latter probability, we note that there are two possibilities to reach the state $n_{i-1}(t+\delta t)=0,\alpha_i(t+\delta t)\geq 1,\alpha_u(i-1,t+\delta t)=0$ subject to the condition that $\alpha_u(i,t)=1$. First, machine $M_d(i-1)$ may finish processing during the time interval $[t,t+\delta t]$ with rate $\mu_d(i-1)\delta t$. Second, machine $M_u(i-1)$ was up at time point t with no available spare part ($\alpha_u(i-1,t)=1$), and the machine fails, which occurs with rate $\lambda_u(i-1)\delta t$. For small values of δt, the probability that more events occur is zero.

$$P(n_{i-1}(t+\delta t)=0,\alpha_i(t+\delta t)\geq 1,\alpha_u(i-1,t+\delta t)=0|n_i(t)<N_i,\alpha_u(i,t)=1) \tag{84}$$

$$=P(n_{i-1}(t)=1,\alpha_i(t)\geq 1,\alpha_u(i-1,t)=0|n_i(t)<N_i,\alpha_u(i,t)=1)\cdot\mu_d(i-1)\delta t \tag{85}$$

$$+P(n_{i-1}(t)=0,\alpha_i(t)\geq 1,\alpha_u(i-1,t)=1|n_i(t)<N_i,\alpha_u(i,t)=1)\cdot\lambda_u(i-1)\delta t$$

Both conditional probabilities in (85) can be simplified by applying the definition of conditional probabilities and the definition (75) of the virtual machine being in a state with $\alpha_u(i,t)=1$ and then using the steady-state probabilities of the virtual two-machine line $i-1$. We obtain

$$\begin{aligned}
&P(n_{i-1}(t)=1,\alpha_i(t)\geq 1,\alpha_u(i-1,t)=0|n_i(t)<N_i,\alpha_u(i,t)=1)\\
=&\frac{P(n_{i-1}(t)=1,n_i(t)<N_i,\alpha_i(t)\geq 1,\alpha_u(i-1,t)=0,\alpha_u(i,t)=1)}{P(n_i(t)<N_i,\alpha_u(i,t)=1)} && (86)\\
\overset{(75)}{=}&\frac{P(n_{i-1}(t)=1,n_i(t)<N_i,\alpha_i(t)\geq 1,\alpha_u(i-1,t)=0,(n_{i-1}(t)>0,\alpha_i(t)=1))}{P(n_i(t)<N_i,\alpha_u(i,t)=1)}\\
& && (87)\\
=&\frac{P(n_{i-1}(t)=1,n_i(t)<N_i,\alpha_i(t)=1,\alpha_u(i-1,t)=0)}{P(n_i(t)<N_i,\alpha_u(i,t)=1)} && (88)\\
\overset{(78)}{=}&\frac{P(n_{i-1}(t)=1,\alpha_u(i-1,t)=0,\alpha_d(i-1,t)=1)}{P(n_i(t)<N_i,\alpha_u(i,t)=1)} && (89)\\
=&\frac{\mathcal{P}_{i-1}(1,0,1)}{P(n_i(t)<N_i,\alpha_u(i,t)=1)} && (90)
\end{aligned}$$

for the first conditional probability in (85). In a similar way, we proceed for the second conditional probability in (85):

$$\begin{aligned}
&P(n_{i-1}(t)=0,\alpha_i(t)\geq 1,\alpha_u(i-1,t)=1|n_i(t)<N_i,\alpha_u(i,t)=1)\\
=&\frac{P(n_{i-1}(t)=0,n_i(t)<N_i,\alpha_i(t)\geq 1,\alpha_u(i-1,t)=1,\alpha_u(i,t)=1)}{P(n_i(t)<N_i,\alpha_u(i,t)=1)} && (91)\\
\overset{(75)}{=}&\frac{1}{P(n_i(t)<N_i,\alpha_u(i,t)=1)}\cdot P(n_{i-1}(t)=0,n_i(t)<N_i,\alpha_i(t)\geq 1,\\
&\qquad\alpha_u(i-1,t)=1,(n_{i-1}(t)=0,\alpha_i(t)\geq 1,\alpha_u(i-1,t)=1)) && (92)\\
=&\frac{P(n_{i-1}(t)=0,n_i(t)<N_i,\alpha_i(t)\geq 1,\alpha_u(i-1,t)=1)}{P(n_i(t)<N_i,\alpha_u(i,t)=1)} && (93)\\
\overset{(79),(78)}{=}&\frac{P(n_{i-1}(t)=0,\alpha_u(i-1,t)=1,\alpha_d(i-1,t)\geq 1)}{P(n_i(t)<N_i,\alpha_u(i,t)=1)} && (94)\\
=&\frac{\mathcal{P}_{i-1}(0,1,\alpha_d\geq 1)}{P(n_i(t)<N_i,\alpha_u(i,t)=1)}. && (95)
\end{aligned}$$

Consequently, (85) can be reformulated by applying (90) and (95) such that (83) leads to

$$\lambda_u(i)\delta t=\lambda_i\delta t+\frac{\mathcal{P}_{i-1}(1,0,1)\,\mu_d(i-1)\delta t+\mathcal{P}_{i-1}(0,1,\alpha_d\geq 1)\,\lambda_u(i-1)\delta t}{P(n_i(t)<N_i,\alpha_u(i,t)=1)}. \quad (96)$$

Finally, the denominator must be expressed in terms of steady-state probabilities of the virtual two-machine line i. We use the FRIT condition

$$TP(i) = \mu_u(i) \cdot \Big(P(n_i(t) < N_i, \alpha_u(i,t) \geq 1)\Big) \tag{97}$$
$$= \mu_u(i) \cdot \Big(P(n_i(t) < N_i, \alpha_u(i,t) = 1) + P(n_i(t) < N_i, \alpha_u(i,t) > 1)\Big) \tag{98}$$

and by rearranging the terms and imposing conservation of flow (i.e., $TP(i) = TP(i-1)$) and replacing the probabilities, we obtain the following expression:

$$P(n_i(t) < N_i, \alpha_u(i,t) = 1) = \frac{TP(i)}{\mu_u(i)} - P(n_i(t) < N_i, \alpha_u(i,t) > 1) \tag{99}$$
$$= \frac{TP(i-1)}{\mu_u(i)} - \mathcal{P}_i(n < N_i, \alpha_u \geq 2, \alpha_d \geq 0). \tag{100}$$

Finally, (100) can be used to evaluate the denominator of (96) such that we obtain after dividing by δt

$$\lambda_u(i) = \lambda_i + \frac{\mathcal{P}_{i-1}(1,0,1)\,\mu_d(i-1) + \mathcal{P}_{i-1}(0,1,\alpha_d \geq 1)\,\lambda_u(i-1)}{\frac{TP(i-1)}{\mu_u(i)} - \mathcal{P}_i(n < N_i, \alpha_u \geq 2, \alpha_d \geq 0)} \tag{101}$$

as the failure rate of the upstream machine in the virtual flow line i. Analogously, we can start with the IOF condition for $M_d(i)$ and apply definition (D2) such that

$$\lambda_d(i)\delta t = P(\alpha_d(i, t+\delta t) = 0 | n_i(t) > 0, \alpha_d(i,t) = 1) \tag{102}$$
$$\overset{(77)}{=} P(n_{i+1}(t+\delta t) < N_{i+1}, \alpha_{i+1}(t+\delta t) = 0 | n_i(t) > 0, \alpha_d(i,t) = 1) \tag{103}$$
$$+ P(n_{i+1}(t+\delta t) = N_{i+1}, \alpha_{i+1}(t) \geq 1, \alpha_d(i+1, t+\delta t) = 0 | n_i(t) > 0, \alpha_d(i,t) = 1)$$
$$\overset{(78)}{=} P(n_{i+1}(t+\delta t) < N_{i+1}, \alpha_{i+1}(t+\delta t) = 0 | n_i(t) > 0, (n_{i+1}(t) < N_{i+1}, \alpha_{i+1}(t) = 1) \text{ or } (n_{i+1}(t) = N_{i+1}, \alpha_{i+1}(t) \geq 1, \alpha_d(i+1,t) = 1)) \tag{104}$$
$$+ P(n_{i+1}(t+\delta t) = N_{i+1}, \alpha_{i+1}(t) \geq 1, \alpha_d(i+1, t+\delta t) = 0 | n_i(t) > 0, \alpha_d(i,t) = 1).$$
$$= \lambda_i \delta t + P(n_{i+1}(t+\delta t) = N_{i+1}, \alpha_{i+1}(t) \geq 1, \alpha_d(i+1, t+\delta t) = 0 | n_i(t) > 0, \alpha_d(i,t) = 1). \tag{105}$$

Eventually, we obtain a similar result for the failure rate of the downstream machine in the virtual flow line i:

$$\lambda_d(i) = \lambda_{i+1} + \frac{\mathcal{P}_{i+1}(N_{i+1}-1,1,0)\,\mu_u(i+1) + \mathcal{P}_{i+1}(N_{i+1},\alpha_u \geq 1,1)\,\lambda_d(i+1)}{\frac{TP(i+1)}{\mu_d(i)} - \mathcal{P}_i(n>0,\alpha_u \geq 0,\alpha_d \geq 2)}. \tag{106}$$

□

3.5.3 Proof for ROF

Proof for Theorem 4. First, we want to find expressions for the ROF condition for virtual upstream machine $M_u(i)$. We can reformulate the equation by applying (D1) and obtain

$$\begin{aligned} Q_i\gamma_u(i)\delta t &= P(\alpha_u(i,t+\delta t)=1|n_i(t)<N_i,\alpha_u(i,t)=0) && (107)\\ &\overset{(74)}{=} P(\alpha_u(i,t+\delta t)=1|n_i(t)<N_i, && (108)\\ &\qquad (n_{i-1}(t)>0,\alpha_i(t)=0) \text{ or } (n_{i-1}(t)=0,\alpha_i(t)\geq 1,\alpha_u(i-1,t)=0)).\\ &= P(\alpha_u(i,t+\delta t)=1|(n_{i-1}(t)>0,n_i(t)<N_i,\alpha_i(t)=0) && (109)\\ &\qquad \text{or } (n_{i-1}(t)=0,n_i(t)<N_i,\alpha_i(t)\geq 1,\alpha_u(i-1,t)=0)).\end{aligned}$$

Let $\mathcal{A},\mathcal{B}$ and $\mathcal{C}$ be events with $\mathcal{B}$ and $\mathcal{C}$ mutually exclusive, i.e., $P(\mathcal{B}\cap\mathcal{C})=0$. Then, one will find that

$$P(\mathcal{A}|\mathcal{B}\cup\mathcal{C}) = \frac{P(\mathcal{A}\cap(\mathcal{B}\cup\mathcal{C}))}{P(\mathcal{B}\cup\mathcal{C})} = \frac{P(\mathcal{A}\cap\mathcal{B}\cup\mathcal{A}\cap\mathcal{C})}{P(\mathcal{B}\cup\mathcal{C})} = \frac{P(\mathcal{A}\cap\mathcal{B})}{P(\mathcal{B}\cup\mathcal{C})} + \frac{P(\mathcal{A}\cap\mathcal{C})}{P(\mathcal{B}\cup\mathcal{C})} \tag{110}$$

$$= \frac{P(\mathcal{A}|\mathcal{B})\cdot P(\mathcal{B})}{P(\mathcal{B}\cup\mathcal{C})} + \frac{P(\mathcal{A}|\mathcal{C})\cdot P(\mathcal{C})}{P(\mathcal{B}\cup\mathcal{C})} \tag{111}$$

$$= \frac{P(\mathcal{A}|\mathcal{B})\cdot P(\mathcal{B}\cap(\mathcal{B}\cup\mathcal{C}))}{P(\mathcal{B}\cup\mathcal{C})} + \frac{P(\mathcal{A}|\mathcal{C})\cdot P(\mathcal{C}\cap(\mathcal{B}\cup\mathcal{C}))}{P(\mathcal{B}\cup\mathcal{C})} \tag{112}$$

and finally

$$P(\mathcal{A}|\mathcal{B}\cup\mathcal{C}) = P(\mathcal{A}|\mathcal{B})\cdot P(\mathcal{B}|\mathcal{B}\cup\mathcal{C}) + P(\mathcal{A}|\mathcal{C})\cdot P(\mathcal{C}|\mathcal{B}\cup\mathcal{C}). \tag{113}$$

Accordingly, to rewrite (109) as this kind of concatenation of probabilities, we define

$$A := P(\alpha_u(i, t+\delta t) = 1 | n_{i-1}(t) > 0, n_i(t) < N_i, \alpha_i(t) = 0) \tag{114}$$

$$B := P(n_{i-1}(t) > 0, n_i(t) < N_i, \alpha_i(t) = 0 | n_i(t) < N_i, \alpha_u(i,t) = 0) \tag{115}$$

$$C := P(\alpha_u(i, t+\delta t) = 1 | n_{i-1}(t) = 0, n_i(t) < N_i, \alpha_i(t) \geq 1, \alpha_u(i-1,t) = 0) \tag{116}$$

$$D := P(n_{i-1}(t) = 0, n_i(t) < N_i, \alpha_i(t) \geq 1, \alpha_u(i-1,t) = 0 | n_i(t) < N_i, \alpha_u(i,t) = 0) \tag{117}$$

and obtain by applying (113)

$$Q_i \gamma_u(i) \delta t = A \cdot B + C \cdot D. \tag{118}$$

The probabilities A and C can easily be rewritten with the replenishment rates

$$A = Q_i \gamma_i \delta t \tag{119}$$

$$C = Q_{i-1} \gamma_u(i-1) \delta t \tag{120}$$

and it is clear that

$$B = 1 - D. \tag{121}$$

For D, we use the definition of conditional probability and definition (D1) to obtain

$$D = \frac{P(n_{i-1}(t) = 0, n_i(t) < N_i, \alpha_i(t) \geq 1, \alpha_u(i-1,t) = 0, \alpha_u(i,t) = 0)}{P(n_i(t) < N_i, \alpha_u(i,t) = 0)} \tag{122}$$

$$\overset{(74)}{=} \frac{1}{P(n_i(t) < N_i, \alpha_u(i,t) = 0)} \cdot \tag{123}$$

$$P(n_{i-1}(t) = 0, n_i(t) < N_i, \alpha_i(t) \geq 1, \alpha_u(i-1,t) = 0,$$
$$((n_{i-1}(t) > 0, \alpha_i(t) = 0) \text{ or } (n_{i-1}(t) = 0, \alpha_i(t) \geq 1, \alpha_u(i-1,t) = 0)))$$

$$= \frac{P(n_{i-1}(t) = 0, n_i(t) < N_i, \alpha_i(t) \geq 1, \alpha_u(i-1,t) = 0)}{P(n_i(t) < N_i, \alpha_u(i,t) = 0)} \tag{124}$$

$$\overset{(79),(78)}{=} \frac{P(n_{i-1}(t) = 0, \alpha_u(i-1,t) = 0, \alpha_d(i-1,t) \geq 1)}{P(n_i(t) < N_i, \alpha_u(i,t) = 0)}. \tag{125}$$

The numerator of this expression can directly be reformulated via the steady-state probabilities of the virtual two-machine line $i-1$

$$P(n_{i-1}(t)=0,\alpha_u(i-1,t)=0,\alpha_d(i-1,t)\geq 1)=\mathcal{P}_{i-1}(0,0,\alpha_d\geq 1) \tag{126}$$

For the denominator, we make use of the FRIT condition

$$TP(i)=\mu_u(i)\cdot(P(n_i(t)<N_i,\alpha_u(i,t)=1)+P(n_i(t)<N_i,\alpha_u(i,t)>1)) \tag{127}$$

$$\Leftrightarrow \quad P(n_i(t)<N_i,\alpha_u(i,t)=1)=\frac{TP(i)}{\mu_u(i)}-P(n_i(t)<N_i,\alpha_u(i,t)>1) \tag{128}$$

and the balance equation

$$Q_i\gamma_u(i)\cdot P(n_i(t)<N_i,\alpha_u(i,t)=0)=\lambda_u(i)\cdot P(n_i(t)<N_i,\alpha_u(i,t)=1) \tag{129}$$

$$\Leftrightarrow \qquad P(n_i(t)<N_i,\alpha_u(i,t)=0)=\frac{\lambda_u(i)}{Q_i\gamma_u(i)}\cdot P(n_i(t)<N_i,\alpha_u(i,t)=1). \tag{130}$$

so that (128) together with the explicit steady-state probabilities of the virtual flow line i lead to

$$P(n_i(t)<N_i,\alpha_u(i,t)=0)=\frac{\lambda_u(i)}{Q_i\gamma_u(i)}\left(\frac{TP(i)}{\mu_u(i)}-\mathcal{P}_i(n<N_i,\alpha_u\geq 2,\alpha_d\geq 0)\right). \tag{131}$$

In summary, (118) can be rewritten as

$$Q_i\gamma_u(i)\delta t = A\cdot B+C\cdot D \tag{132}$$

$$= A\cdot(1-D)+C\cdot D \tag{133}$$

$$= A - A\cdot D+C\cdot D \tag{134}$$

$$= A+D(C-A) \tag{135}$$

$$= Q_i\gamma_i\delta t+\frac{(Q_{i-1}\gamma_u(i-1)\delta t-Q_i\gamma_i\delta t)\mathcal{P}_{i-1}(0,0,\alpha_d\geq 1)}{\frac{\lambda_u(i)}{Q_i\gamma_u(i)}\left(\frac{TP(i)}{\mu_u(i)}-\mathcal{P}_i(n<N_i,\alpha_u\geq 2,\alpha_d\geq 0)\right)} \tag{136}$$

and finally after dividing by δt and Q_i and using conservation of flow (i.e., $TP(i)=$

$TP(i-1))$

$$\gamma_u(i) = \gamma_i + \frac{\left(\frac{Q_{i-1}}{Q_i}\gamma_u(i-1) - \gamma_i\right)\mathcal{P}_{i-1}(0,0,\alpha_d \geq 1)}{\frac{\lambda_u(i)}{Q_i\gamma_u(i)}\left(\frac{TP(i-1)}{\mu_u(i)} - \mathcal{P}_i(n < N_i, \alpha_u \geq 2, \alpha_d \geq 0)\right)}. \tag{137}$$

Equivalently, we obtain

$$\gamma_d(i) = \gamma_{i+1} + \frac{\left(\frac{Q_{i+2}}{Q_{i+1}}\gamma_d(i+1) - \gamma_{i+1}\right)\mathcal{P}_{i+1}(N_{i+1}, \alpha_u \geq 1, 0)}{\frac{\lambda_d(i)}{Q_{i+1}\gamma_d(i)}\left(\frac{TP(i+1)}{\mu_d(i)} - \mathcal{P}_i(n > 0, \alpha_u \geq 0, \alpha_d \geq 2)\right)} \tag{138}$$

for the downstream machine of the virtual line i. □

3.5.4 Proof for isolated availability and throughput

Proof for (41). In the following, we omit the index i as we only have one machine. Thus, we have for a specific machine in total Q units of the critical component, a replenishment rate γ, and a failure rate λ. Analyzed in isolation, we can describe this situation by a Markov chain $(Y(t))_{t\geq 0}$ with a one-dimensional state space, which can be identified as $\{0, 1, \dots, Q\}$. $Y(t) = \alpha \in \{1, 2, \dots, Q\}$ means that a total of α units of the component are available: one inside of the machine and the rest (if applicable) in stock. $Y(t) = 0$ represents the situation where the machine is down, and we have Q outstanding orders. The transitions are similar to the two-machine system described by (10) and (11). We find two possible transition probabilities (for time-points $t \geq 0$ and small $\delta t > 0$):

1. When a failure happens, one unit of the component is used. Hence, we get for $\alpha \in \{1, \dots, Q\}$

$$P(Y(t+\delta t) = \alpha - 1 | Y(t) = \alpha) = \lambda \tag{139}$$

2. A spare part is replenished with cumulative rate for $\alpha \in \{0, 1, \dots, Q-1\}$

$$P(Y(t+\delta t) = \alpha + 1 | Y(t) = \alpha) = (Q - \alpha) \cdot \gamma \tag{140}$$

The steady-state probabilities of this Markov chain are represented by

$$\pi = (\pi_0, \pi_1, \ldots, \pi_Q) \tag{141}$$

where $\pi_\alpha, \alpha \in \{0, 1, \ldots, Q\}$, is the proportion of time the machine stays in state α in the long run. With a level crossing argument between every two neighboring states and the normalization equation, one can find the following system of equations to determine the steady-state probabilities:

$$\lambda \cdot \pi_k = (Q - (k-1)) \cdot \gamma \cdot \pi_{k-1} \qquad \forall k = 1, \ldots, Q \tag{142}$$

$$\sum_{k=0}^{Q} \pi_k = 1 \tag{143}$$

Isolating π_k in (142) and solving the emerging recursion yields $\forall k = 1, \ldots, Q$

$$\pi_k = \frac{\gamma}{\lambda} \cdot (Q - (k-1)) \cdot \pi_{k-1} \tag{144}$$

$$= \left(\frac{\gamma}{\lambda}\right)^k \cdot \frac{Q!}{(Q-k)!} \cdot \pi_0 \tag{145}$$

This expression for $\pi_k, k = 1, \ldots, Q$, is inserted in (143) to find that

$$1 = \pi_0 + \sum_{k=1}^{Q} \left(\frac{\gamma}{\lambda}\right)^k \cdot \frac{Q!}{(Q-k)!} \cdot \pi_0$$

$$\Leftrightarrow \qquad 1 = \sum_{k=0}^{Q} \left(\frac{\gamma}{\lambda}\right)^k \cdot \frac{Q!}{(Q-k)!} \cdot \pi_0$$

$$\Leftrightarrow \qquad \pi_0 = \left(\sum_{k=0}^{Q} \left(\frac{\gamma}{\lambda}\right)^k \cdot \frac{Q!}{(Q-k)!}\right)^{-1}$$

Finally, the availability results as

$$iAV := 1 - \pi_0 = 1 - \left(\sum_{k=0}^{Q} \left(\frac{\gamma}{\lambda}\right)^k \cdot \frac{Q!}{(Q-k)!}\right)^{-1} \tag{146}$$

□

Table 13. Details for the longer flow lines given in Tables 7 and 8. Asterisks indicate that TP_{dec} is included in the 95% confidence interval of the simulation result; $\mu_i = 1, \lambda_i = 0.005$ (per row: $i \in \mathcal{I}, j \in \mathcal{J}$).

Case	I	C_j	Q_i	γ_i	iAV_i	TP_{sim}	TP_{dec}	$\Delta_{\text{dec/sim}}$	$\Delta^{\text{rel}}_{\text{dec/sim}}$	HW
1	5	10	2	0.1	0.9988	0.8682	0.8678	−0.0004	−0.0005	0.0026*
2	25	10	2	0.1	0.9988	0.8332	0.8434	0.0102	0.0122	0.0024
3	45	10	2	0.1	0.9988	0.8283	0.8366	0.0083	0.0101	0.0017
4	5	30	2	0.1	0.9988	0.9456	0.9465	0.0009	0.0010	0.0019*
5	25	30	2	0.1	0.9988	0.9293	0.9366	0.0072	0.0078	0.0026
6	45	30	2	0.1	0.9988	0.9251	0.9294	0.0043	0.0047	0.0027
7	5	10	3	0.1	1.0000	0.8684	0.8696	0.0011	0.0013	0.0032*
8	25	10	3	0.1	1.0000	0.8349	0.8453	0.0105	0.0125	0.0017
9	45	10	3	0.1	1.0000	0.8314	0.8386	0.0072	0.0086	0.0027
10	5	30	3	0.1	1.0000	0.9474	0.9481	0.0007	0.0008	0.0019*
11	25	30	3	0.1	1.0000	0.9326	0.9383	0.0056	0.0060	0.0019
12	45	30	3	0.1	1.0000	0.9272	0.9311	0.0039	0.0042	0.0022
13	5	10	4	0.1	1.0000	0.8679	0.8696	0.0017	0.0020	0.0020*
14	25	10	4	0.1	1.0000	0.8356	0.8454	0.0098	0.0117	0.0020
15	45	10	4	0.1	1.0000	0.8327	0.8386	0.0060	0.0071	0.0019
16	5	30	4	0.1	1.0000	0.9462	0.9482	0.0019	0.0020	0.0023*
17	25	30	4	0.1	1.0000	0.9293	0.9383	0.0090	0.0097	0.0026
18	45	30	4	0.1	1.0000	0.9272	0.9312	0.0040	0.0043	0.0025
19	5	10	2	0.01	0.9231	0.6917	0.7021	0.0105	0.0151	0.0098
20	25	10	2	0.01	0.9231	0.5145	0.6220	0.1076	0.2091	0.0042
21	45	10	2	0.01	0.9231	0.4801	0.6149	0.1348	0.2808	0.0048
22	5	30	2	0.01	0.9231	0.7443	0.7711	0.0268	0.0360	0.0077
23	25	30	2	0.01	0.9231	0.6217	0.7226	0.1009	0.1624	0.0048
24	45	30	2	0.01	0.9231	0.6031	0.7184	0.1154	0.1913	0.0063
25	5	10	3	0.01	0.9873	0.8385	0.8440	0.0055	0.0065	0.0041
26	25	10	3	0.01	0.9873	0.7613	0.8145	0.0532	0.0699	0.0055
27	45	10	3	0.01	0.9873	0.7318	0.8071	0.0753	0.1029	0.0040
28	5	30	3	0.01	0.9873	0.9126	0.9202	0.0076	0.0083	0.0047
29	25	30	3	0.01	0.9873	0.8586	0.9068	0.0482	0.0561	0.0019
30	45	30	3	0.01	0.9873	0.8443	0.9002	0.0558	0.0661	0.0020
31	5	10	4	0.01	0.9984	0.8661	0.8667	0.0006	0.0007	0.0022*
32	25	10	4	0.01	0.9984	0.8281	0.8419	0.0138	0.0166	0.0031
33	45	10	4	0.01	0.9984	0.8201	0.8351	0.0150	0.0183	0.0020
34	5	30	4	0.01	0.9984	0.9435	0.9449	0.0014	0.0015	0.0025*
35	25	30	4	0.01	0.9984	0.9231	0.9347	0.0116	0.0126	0.0031
36	45	30	4	0.01	0.9984	0.9178	0.9276	0.0098	0.0107	0.0025

Table 14. System designs based on system C from Tempelmeier (2003). Slight deviations due to rate conversion and rounding occur.

i	C_i	μ_i	λ_i	Q_i				γ_i				iTP_i			
				C	C1	C2	C3	C	C1	C2	C3	C	C1	C2	C3
1	21	0.2575	0.0247	1	2	2	2	0.2	0.2	0.1	0.05	0.8901	0.2558	0.2514	0.2381
2	31	0.2575	0.0273	1	2	2	2	0.2	0.2	0.1	0.05	0.8799	0.2554	0.2502	0.2349
3	20	0.2564	0.0286	1	2	2	2	0.2	0.2	0.1	0.05	0.8749	0.2541	0.2485	0.2322
4	24	0.2778	0.0299	1	2	2	2	0.2	0.2	0.1	0.05	0.8699	0.2751	0.2686	0.2498
5	19	0.2830	0.0500	1	2	2	2	0.2	0.2	0.1	0.05	0.8000	0.2761	0.2612	0.2264
6	4	0.2727	0.0299	1	2	2	2	0.2	0.2	0.1	0.05	0.8699	0.2701	0.2636	0.2453
7	12	0.2353	0.0083	1	2	2	2	0.2	0.2	0.1	0.05	0.9602	0.2351	0.2346	0.2326
8		0.2335	0.0083	1	2	2	2	0.2	0.2	0.1	0.05	0.9602	0.2333	0.2328	0.2308

Table 15. System designs based on system D from Tempelmeier (2003). Slight deviations due to rate conversion and rounding occur.

i	C_i	μ_i	λ_i	Q_i				γ_i				iTP_i			
				D	D1	D2	D3	D	D1	D2	D3	D	D1	D2	D3
1	8	2.4000	0.0010	1	2	2	2	0.05	0.05	0.025	0.0125	0.9804	2.3995	2.3982	2.3929
2	25	1.7647	0.0015	1	2	2	2	0.05	0.05	0.025	0.0125	0.9709	1.7639	1.7617	1.7534
3	1	1.7910	0.0005	1	2	2	2	0.05	0.05	0.025	0.0125	0.9901	1.7909	1.7906	1.7896
4	2	1.6901	0.0010	1	2	2	2	0.05	0.05	0.025	0.0125	0.9804	1.6898	1.6888	1.6851
5	16	2.7273	0.0005	1	2	2	2	0.05	0.05	0.025	0.0125	0.9901	2.7272	2.7268	2.7252
6	7	1.6667	0.0015	1	2	2	2	0.05	0.05	0.025	0.0125	0.9709	1.6660	1.6639	1.6561
7	32	2.7273	0.0015	1	2	2	2	0.05	0.05	0.025	0.0125	0.9709	2.7261	2.7227	2.7099
8	1	1.7647	0.0015	1	2	2	2	0.05	0.05	0.025	0.0125	0.9709	1.7639	1.7617	1.7534
9	8	2.3077	0.0015	1	2	2	2	0.05	0.05	0.025	0.0125	0.9709	2.3067	2.3038	2.2930
10	20	2.0000	0.0015	1	2	2	2	0.05	0.05	0.025	0.0125	0.9709	1.9991	1.9966	1.9872
11	9	2.2642	0.0010	1	2	2	2	0.05	0.05	0.025	0.0125	0.9804	2.2638	2.2625	2.2575
12	21	1.8182	0.0015	1	2	2	2	0.05	0.05	0.025	0.0125	0.9709	1.8174	1.8151	1.8066
13	16	2.0339	0.0015	1	2	2	2	0.05	0.05	0.025	0.0125	0.9709	2.0330	2.0305	2.0209
14		1.7143	0.0005	1	2	2	2	0.05	0.05	0.025	0.0125	0.9901	1.7142	1.7140	1.7130

Chapter

4 Design of Flow Lines With Stochastic Processing Times

Things are only impossible until they are not.

Captain Jean-Luc Picard

The following chapter is based on a working paper (Sachs et al., 2022b). Based on the problem formulation, the mathematical model, and solution techniques developed in Chapter 3, we can continue to answer the design question of how flow lines with buffers and spare parts should be built optimally. Section 4.1 introduces the optimization problem. After that, Section 4.2 explains the applied algorithms. We develop two greedy heuristics, which are later on combined to get a third heuristic. Furthermore, we apply two common metaheuristics to our problem: simulated annealing and a genetic algorithm. The numerical study in Section 4.3 starts with validating results for our heuristics. We compare the (meta)heuristics to optimal results obtained by complete enumeration for a two-machine line. Besides, we amplify how the different algorithms perform and which of them are worth further consideration. It turns out that the greedy approaches outperform the metaheuristics. Finally, Section 4.3 also delivers managerial insights on the optimal buffer and spare part allocations in flow lines. We can confirm well-known phenomena from spare parts planning and present novel findings for interaction effects and joint allocation patterns. Section 4.4 concludes with the most crucial findings.

4.1 Description of the optimization problem

Regarding the general description of the flow line and the corresponding assumptions, we refer to Section 3.1. Building on this, we continue with the optimization problem. According to Weiss et al., 2019, we formulate the primal *buffer and spare part allocation problem* (BSAP) based on the primal BAP as

$$\min_{\substack{C_j, j \in \mathcal{J} \\ Q_i, i \in \mathcal{I}}} TC(\mathbf{C}, \mathbf{Q}) := \sum_{j \in \mathcal{J}} c_j^{\text{buffer}} \cdot C_j + \sum_{i \in \mathcal{I}} c_i^{\text{spare}} \cdot Q_i \tag{147}$$

$$\text{s.t.} \quad TP(\mathbf{C}, \mathbf{Q}) \geq TP^T, \tag{148}$$

$$0 \leq C_j^{\min} \leq C_j \leq C_j^{\max} \qquad \forall j \in \mathcal{J} \tag{149}$$

$$1 \leq Q_i^{\min} \leq Q_i \leq Q_i^{\max} \qquad \forall i \in \mathcal{I} \tag{150}$$

$$C_j \in \mathbb{N}_0, Q_i \in \mathbb{N} \qquad \forall j \in \mathcal{J}, i \in \mathcal{I} \tag{151}$$

where $TC(\mathbf{C}, \mathbf{Q})$ is the objective function, specifically the total costs, $TP(\mathbf{C}, \mathbf{Q})$ is the expected throughput, TP^T is the target throughput, $C_j^{\min}, j \in \mathcal{J}$, and $Q_i^{\min}, i \in \mathcal{I}$, ($C_j^{\max}$ and $Q_i^{\max}$) are the lower (upper) bounds for buffer capacities and the number of units. We use $\mathbf{C} := (C_1, \ldots, C_J)$ and $\mathbf{Q} := (Q_1, \ldots, Q_I)$ as vector representations of $C_j, j \in \mathcal{J}$, and $Q_i, i \in \mathcal{I}$. Naturally, we have $Q_i^{\min} \geq 1, \forall i \in \mathcal{I}$, as each machine contains exactly one unit of its specific component, whereas additional spare parts are optional.

4.2 Solution methods

The proposed optimization problem (primal BSAP) is NP-hard since it is an extension of the BAP, known to be NP-hard (Weiss et al., 2019). Thus, we apply different heuristic approaches. First, we implement two pseudo-gradient-based greedy procedures, inspired by both greedy approaches from spare parts planning and pseudo-gradient methods, to solve the BAP. Second, we use two common metaheuristics: simulated annealing and a genetic algorithm.

4.2.1 Greedy heuristics

Greedy heuristics are iterative procedures that start with initial values $C_j(0), j \in \mathcal{J}$, and $Q_i(0), i \in \mathcal{I}$. Then, these values are changed iteratively until a termination criterion is reached. The corresponding values in iteration $n \in \mathbb{N}$ are $C_j(n), j \in \mathcal{J}$, and $Q_i(n), i \in \mathcal{I}$.

The first greedy heuristic (greedy with increasing steps — GI) is presented in Algorithm 2. It starts with a minimally equipped system $\mathbf{C}^{\text{start}} := \mathbf{C}^{\min}, \mathbf{Q}^{\text{start}} := \mathbf{Q}^{\min}$. During the iterations, the current solution's neighborhood is searched for a better solution. Each neighbor originates from the current solution by increasing one single decision variable by one. Therefore, we define the index $k \in \mathcal{K} := \{1, \ldots, 2 \cdot I - 1\}$ where $k < I$ means that the capacity of buffer B_k is increased by one and $k \geq I$ implies that the base-stock level S_{k-I+1} is increased by one. This determines the new state for given k as

$$C_j(n,k) := C_j(n-1) + \mathbb{1}(k < I \wedge j = k), \qquad \forall n \geq 1, k \in \mathcal{K}, j \in \mathcal{J} \quad (152)$$

$$Q_i(n,k) := Q_i(n-1) + \mathbb{1}(k \geq I \wedge i = k - I + 1) \quad \forall n \geq 1, k \in \mathcal{K}, i \in \mathcal{I}. \quad (153)$$

As before, we use the vectorized notations

$$\mathbf{C}(n) := (C_1(n), \ldots, C_{\mathcal{J}}(n)) \qquad \forall n \geq 0 \quad (154)$$

$$\mathbf{Q}(n) := (Q_1(n), \ldots, Q_{\mathcal{I}}(n)) \qquad \forall n \geq 0 \quad (155)$$

$$\mathbf{C}(n,k) := (C_1(n,k), \ldots, C_{\mathcal{J}}(n,k)) \qquad \forall n \geq 1, k \in \mathcal{K} \quad (156)$$

$$\mathbf{Q}(n,k) := (Q_1(n,k), \ldots, Q_{\mathcal{I}}(n,k)) \qquad \forall n \geq 1, k \in \mathcal{K} \quad (157)$$

We choose to increment either one buffer level or the spare part stock by one to facilitate the maximum throughput-per-cost increase. The following function reflects this decision:

$$f(k) := \frac{TP(\mathbf{C}(n,k), \mathbf{Q}(n,k)) - TP(\mathbf{C}(n-1), \mathbf{Q}(n-1))}{\mathbb{1}(k < I) \cdot c_k^{\text{buffer}} + \mathbb{1}(k \geq I) \cdot c_{k-I+1}^{\text{spare}}} \quad \forall n \geq 1, k \in \mathcal{K} \quad (158)$$

The numerator consists of the throughput change, whereas the denominator considers the difference in the objective function, which is of this form since only one decision variable is changed. Eventually, the iterative procedure is based on the following

Algorithm 2. Greedy heuristic with low starting values and increasing steps (GI)

Result: $TP = TP(n), \mathbf{C} = \mathbf{C}(n), \mathbf{Q} = \mathbf{Q}(n)$
Input: Target throughput TP^T, starting values $\mathbf{C}^{\text{start}}$, $\mathbf{Q}^{\text{start}}$, boundaries $\mathbf{C}^{\max}, \mathbf{Q}^{\max}$, costs $c_j^{\text{buffer}}, j \in \mathcal{J}, c_i^{\text{spares}}, i \in \mathcal{I}$

/* initialize */
$n \leftarrow 0$
adjust $\mathbf{Q}^{\text{start}}$ such that the isolated machine's throughput $\geq TP^T$
$\mathbf{C}(0) \leftarrow \mathbf{C}^{\text{start}}$
$\mathbf{Q}(0) \leftarrow \mathbf{Q}^{\text{start}}$
/* iterate until feasibility is reached */
while $TP(n) < TP^T$ **do**
    $n \leftarrow n + 1$
    /* check all feasible neighboring increments */
    for $k \leftarrow 1$ **to** $2 \cdot I - 1$ **do**
        $\mathbf{C}(n-1,k) \leftarrow \mathbf{C}(n-1)$
        $\mathbf{Q}(n-1,k) \leftarrow \mathbf{Q}(n-1)$
        if $k < I$ **then**
            $C_k(n-1,k) \leftarrow C_k(n-1,k) + 1$
            if $C_k(n-1,k) > C_k^{\max}$ **then continue**
        else
            $Q_{k-I+1}(n-1,k) \leftarrow Q_{k-I+1}(n-1,k) + 1$
            if $Q_{k-I+1}(n-1,k) > Q_{k-I+1}^{\max}$ **then continue**
        use the Markov approach or decomposition to compute $TP(\mathbf{C}(n-1,k), \mathbf{Q}(n-1,k))$
        $TP(n-1,k) \leftarrow TP(\mathbf{C}(n-1,k), \mathbf{Q}(n-1,k))$
        evaluate $f(k)$
    /* find the best increment (if feasible steps exist) */
    index $\leftarrow \arg\max_k f(k)$
    if *index does not exist* **then return**
    /* store new solution */
    $\mathbf{C}(n) \leftarrow \mathbf{C}(n-1, \text{index})$
    $\mathbf{Q}(n) \leftarrow \mathbf{Q}(n-1, \text{index})$
    $TP(n) \leftarrow TP(n-1, \text{index})$

optimization problem:

$$k^* := \arg\max_{k \in \mathcal{K}} f(k). \tag{159}$$

Based upon this, we have the state for the next iteration given as

$$\mathbf{C}(n) := \mathbf{C}(n, k^*) \text{ and } \mathbf{Q}(n) := \mathbf{Q}(n, k^*) \qquad \forall n \geq 1, k^* \in \mathcal{K} \tag{160}$$

The algorithm stops as soon as the throughput overshoots the target throughput.

Besides the actual algorithm, we make two adjustments for our implementation. These are necessary because the decomposition approach suffers from rare numerical problems for poorly performing system designs. This is the case for the starting values of the greedy heuristic with increasing steps. First, we allow step sizes of one to five for the increasing steps if such a numerical issue arises. Second, we modify the starting values and rerun the algorithm if it does not come up with a feasible solution. We start at $C_j^{\text{start}} = C_j^{\min}$ and allow up to $C_j^{\text{start}} = 10, j \in \mathcal{J}$.

Similarly, we propose the second greedy heuristic with the opposite search direction (greedy with decreasing steps — GD). We list the pseudo-code in the appendix (Algorithm 3). It starts with a maximally equipped system $\mathbf{C}^{\text{start}} := \mathbf{C}^{\max}, \mathbf{Q}^{\text{start}} := \mathbf{Q}^{\max}$ and decreases either a single buffer level or a single component base-stock level by one in each of its iterations. Similarly, we choose the decision variable with the minimum throughput-per-cost decrease and apply the following formulas instead of (152), (153), and (159):

$$C_j(n, k) := C_j(n-1) - \mathbb{1}(k < I \wedge j = k), \qquad \forall n \geq 1, k \in \mathcal{K}, j \in \mathcal{J} \tag{161}$$

$$Q_i(n, k) := Q_i(n-1) - \mathbb{1}(k \geq I \wedge i = k - I + 1), \quad \forall n \geq 1, k \in \mathcal{K}, i \in \mathcal{I} \tag{162}$$

$$k^* := \arg\min_{k \in \mathcal{K}} f(k). \tag{163}$$

The algorithm stops as soon as the throughput undercuts the target and gives the last feasible solution as output.

In addition, we apply simulated annealing and a genetic algorithm to check whether we had become stuck in a local optimum due to the integer optimization problem. This is a problem the proposed greedy approaches could suffer from in contrast to the metaheuristics.

4.2.2 Metaheuristics

Simulated annealing and genetic algorithms are widely used metaheuristics to solve the BAP (cf. Bulgak et al., 1995; Papadopoulos et al., 2013; Weiss et al., 2019). Since both metaheuristics are known to work better for unconstrained optimization problems, we include the target throughput constraint in the objective function (147) by adding a penalty term and obtain the penalized total cost (PTC) as follows:

$$PTC(\mathbf{C}, \mathbf{Q}) := \begin{cases} TC(\mathbf{C}, \mathbf{Q}) + 1,000 \cdot \dfrac{TP^T - TP(\mathbf{C}, \mathbf{Q})}{TP^T} & \text{if } TP(\mathbf{C}, \mathbf{Q}) < TP^T \\ TC(\mathbf{C}, \mathbf{Q}), & \text{else.} \end{cases} \tag{164}$$

We implement simulated annealing using a logarithmic cooling scale

$$T(n) := \frac{T(0)}{\log(1+n)} \qquad \forall n \geq 1 \tag{165}$$

which cools sufficiently slowly to achieve — theoretically — globally optimal solutions (cf. Geman & Geman, 1984; Hajek, 1988). However, this theoretical property comes at a high computational cost (Papadopoulos et al., 2013). To react to this issue, we use a start temperature of $T(0) := 20 \cdot \log(2)$ degrees and stop when the temperature reaches $T^{\min} := 2$ degrees or after $n_{\max} = (2 \cdot I - 1) \cdot 200$ iterations. We obtain a similar neighborhood as for our greedy approaches:

a) increase a single buffer level that is not yet maximal by one ($I - 1$ neighbors),

b) increase a single base-stock level that is not yet maximal by one (I neighbors),

c) decrease a single buffer level that is not yet minimal by one ($I - 1$ neighbors),

d) decrease a single base-stock level that is not yet minimal by one (I neighbors),

e) increase all buffer levels which are not yet maximal by five (1 neighbor),

f) decrease all buffer levels which are not yet minimal by five (1 neighbor),

g) increase all base-stock levels which are not yet maximal by one (1 neighbor), and

h) decrease all base-stock levels which are not yet minimal by one (1 neighbor).

Each of these neighbors has an equal probability of being chosen. There is one noteworthy difference: we design the latter four neighbors in order to make it easier to leave local optima. Furthermore, the main idea of simulated annealing is that not only are improvements selected as the next solution but also worse solutions may be accepted with a certain probability. This probability depends negatively on the cost difference: the worse the neighboring solution, the less its probability of being accepted. Let l^* be the chosen neighbor in iteration $n \geq 1$ with associated values of the decision variables $\mathbf{C}(n, l^*)$ and $\mathbf{Q}(n, l^*)$. Then, the acceptance probability equals

$$\exp\left(\frac{PTC(\mathbf{C}(n,l^*), \mathbf{Q}(n,l^*)) - PTC(\mathbf{C}(n-1), \mathbf{Q}(n-1))}{T(n)}\right) \quad \forall n \geq 1. \tag{166}$$

Based on this procedure, we apply simulated annealing twice: with low starting values $\mathbf{C}^{\text{start}} := \mathbf{C}^{\min}, \mathbf{Q}^{\text{start}} := \mathbf{Q}^{\min}$ such that we need to increase the decision variables per iteration to gain feasibility (SAI), and with high starting values $\mathbf{C}^{\text{start}} := \mathbf{C}^{\max}, \mathbf{Q}^{\text{start}} := \mathbf{Q}^{\max}$ similar to GD with decreasing steps (SAD).

To choose the parameters of our genetic algorithm (GA), we adapt the suggestions by Spinellis and Papadopoulos (2000b). We iterate over 100 generations and allow 32 individual solutions to exist in each generation. Each individual consists of $2 \cdot I - 1$ genes: the first $I - 1$ encoding buffer levels, the next I encoding number of component units. When mating a new individual, we use two solutions as parents for the new solution. At this point, a one-point crossover is applied, i.e., an index $k \in \mathcal{K} = \{1, \ldots, 2 \cdot I - 1\}$ is chosen randomly and specifies the crossover point for the genes. Additionally, one of the genes is randomly mutated to take a value in the allowed boundaries. Per generation, we keep one parent and apply a steady-state selection rule which means that we keep the individuals with the highest fitness and remove the ones with the lowest fitness.

We start our numerical study by setting up the instance parameters and validating all of the proposed algorithms against a complete enumeration of the search space.

4.3 Numerical study

Our numerical experiments show that the greedy heuristic with low starting values and increasing steps suffers from a problem-specific drawback. Due to the expected interaction between buffer capacities and base-stock levels, a late base-stock-level increase may overshoot the target throughput. In order to counter this issue, we augment the GI algorithm with a phase of the GD algorithm, which starts with the GI solution. In doing so, we can reduce the costs while still obtaining a feasible solution. An overview of the implemented algorithms is provided in Table 16. We implemented all algorithms except the genetic algorithm in MATLAB R2022a Update 3 (MATLAB, 2022). For the genetic algorithm, we relied on Python 3.9.13. (Van Rossum & Drake, 2009) and used the package PyGAD 2.17.0 (Gad, 2021).

Table 16. Implemented algorithms and their abbreviation

Algorithm	Abbreviation
Complete enumeration (brute-force optimization)	ENUM
Greedy heuristic with low starting values and increasing steps	GI
Greedy heuristic with high starting values and decreasing steps	GD
Greedy heuristic with low starting values and increasing steps followed by decreasing steps	GID
Simulated annealing with low starting values	SAI
Simulated annealing with high starting values	SAD
Genetic algorithm	GA

For all parts of this numerical study, we choose the following parameters as our baseline scenario:

$$\begin{aligned} c_j^{\text{buffer}} &= 1, \quad C_j^{\text{min}} = 1, \quad C_j^{\text{max}} = 25 && \forall j \in \mathcal{J} \\ c_i^{\text{spare}} &= 1, \quad Q_i^{\text{min}} = 1, \quad Q_i^{\text{max}} = 5 && \forall i \in \mathcal{I} \end{aligned} \tag{167}$$

Our further analysis distinguishes between balanced and unbalanced flow lines. A balanced flow line in this context is a flow line where the conditions

$$\mu_i = \mu_k, \quad \lambda_i = \lambda_k, \quad \gamma_i = \gamma_k \qquad \forall i, k \in \mathcal{I} \tag{168}$$

hold. Hence, our baseline scenario for balanced flow lines sets

$$\mu_i = 1, \quad \lambda_i = 0.005, \quad \gamma_i = 0.05 \qquad \forall i \in \mathcal{I} \tag{169}$$

We will refer to the situation reflected in (167) and (169) as the *balanced scenario*. Since we want to generate generalizable insights, we mainly consider unbalanced systems whose parameters are derived from the *balanced scenario*. For these unbalanced flow lines, we allow deviations of up to 5% (processing rates) or 10% (failure and replenishment rates), respectively. This yields the following range of parameters:

$$\mu_i \in [0.95, 1.05], \quad \lambda_i \in [0.0045, 0.0055], \quad \gamma_i \in [0.045, 0.055] \qquad \forall i \in \mathcal{I} \tag{170}$$

The actual parameters for each instance are drawn randomly from a continuous uniform distribution over these intervals. We label these parameter settings given in (167) and (170) the *unbalanced scenario*.

We conduct several numerical experiments with this setup to validate our algorithmic approaches and generate numerical insights. Table 17 lists these experiments, including the number of instances and highlighting the parameter differences per scenario. Our experiments will focus on the optimal or the best-found solutions. We will primarily measure and report the relative cost deviation as

$$\Delta_{\text{costs}} := \frac{TC(\mathbf{C}^{\text{alg}}, \mathbf{Q}^{\text{alg}}) - TC(\mathbf{C}^{\text{rp}}, \mathbf{Q}^{\text{rp}})}{TC(\mathbf{C}^{\text{rp}}, \mathbf{Q}^{\text{rp}})} \tag{171}$$

where $\mathbf{C}^{\text{alg}}, \mathbf{Q}^{\text{alg}}$ is the solution of a specific algorithm, and $\mathbf{C}^{\text{rp}}, \mathbf{Q}^{\text{rp}}$ describes the reference-point solution. Furthermore, we measure the total buffer capacity (TBC) and the total number of component units (TCU) for a given solution as follows

$$TBC := TBC(\mathbf{C}) := \sum_{j \in \mathcal{J}} C_j, \qquad TCU := TCU(\mathbf{Q}) := \sum_{i \in \mathcal{I}} Q_i, \tag{172}$$

where we omit the function inputs when they are clear from the context.

This section starts by comparing the different heuristics to optimal results obtained via complete enumeration. In doing so, we can validate the approximate optimization procedures' results and decide which algorithms to consider for further instances. In all cases, we report results for different target throughput levels. However, the results carry over to arbitrarily but reasonably chosen target throughput levels. For simplicity, we present an unbiased choice of target throughput levels that aim to be sufficiently high to be of practical relevance.

Table 17. Overview of numerical experiments

Experiment	Aim/setting	Number of instances	Parameters
1	Validation	500 flow lines	$I = 2$, unbalanced scenario
2	Algorithm selection	500 flow lines	$I = 5$, unbalanced scenario
3	Illustrate integrated planning	100 flow lines × 3 planning modes (integrated, only buffers, or only spares)	$I = 3$, unbalanced scenario with only buffers: $Q_i^{\text{max}} = 1, \forall i \in \mathcal{I}$, only spares: $C_j^{\text{max}} = 1, \forall j \in \mathcal{J}$,
4	Influence of different cost ratios of all spare parts compared to buffers	100 flow lines × 5 cost scenarios	$I = 5$, unbalanced scenario with $c_i^{\text{spare}} \in \{0.01, 0.1, 1, 10, 100\}, \forall i \in \mathcal{I}$
5	Influence of different cost ratios of just one spare part compared to buffers and other spares	100 flow lines × 5 cost scenarios	$I = 5$, unbalanced scenario with $c_3^{\text{spare}} \in \{0.01, 0.1, 1, 10, 100\}$
6	Influence of bottlenecks	100 flow lines × 5 bottleneck scenarios (bottleneck M_b, $b \in \mathcal{I}$) and 1 balanced flow line for comparison	$I = 5$, balanced scenario with $\mu_b \in [0.90, 0.95]$, $\lambda_b \in [0.0040, 0.0045]$, $\gamma_b \in [0.040, 0.045]$

4.3.1 Validation

The validation considers small systems with $I = 2$ machines. Sachs et al. (2022a) presented an exact numerical solution based on steady-state probabilities of a continuous-time Markov chain. We implement this evaluative procedure using MATLAB and Python. The latter implementation is only used for the genetic algorithm.

We generate 500 instances of unbalanced two-machine lines using the *unbalanced scenario* (see Table 17, Experiment 1). Furthermore, we can increase $C_j^{\text{max}} = 100, \forall j \in \mathcal{J}$, in this small setting. Table 18 shows the results of the optimization approaches for different target throughput levels and compares the relative deviation to the optimal solution found by complete enumeration. For each target throughput level, all 500 instances are solved by each of the proposed algorithms. Besides the genetic algorithm, all approaches were able to find feasible solutions for all instances. Furthermore, the greedy approaches always found the optimal solution. Both simulated annealing procedures found the optimal solution in the vast majority of instances. The genetic algorithm exhibits the poorest performance. In the appendix (Tables 23 and 24), we provide further evidence of the algorithms' performance by showing the fraction of cases where each algorithm found the optimal solution and where each algorithm was outperformed by another.

Table 18. Results for Experiment 1 (see Table 17): 500 instances of randomly generated, unbalanced five-machine lines

		# infeasible	# optimal	Computational time (s)				Δ_{costs} compared to ENUM			
TP^T	Algorithm	solutions	solutions	Mean	Std. error	Min	Max	Mean	Std. error	Min	Max
0.80	ENUM	0	500	3.87	0.02	2.93	5.00				
	GD	0	500	67.20	0.41	42.57	98.76	0.0000	0.0000	0.0000	0.0000
	GI	0	500	6.27	0.03	4.84	8.97	0.0000	0.0000	0.0000	0.0000
	GID	0	500	6.64	0.03	5.06	9.42	0.0000	0.0000	0.0000	0.0000
	SAD	0	493	22.60	0.08	18.72	28.79	0.0020	0.0007	0.0000	0.1667
	SAI	0	492	1.40	0.01	0.78	2.37	0.0022	0.0008	0.0000	0.1667
	GA	17	463	9.47	0.08	5.36	24.41	0.0066	0.0015	0.0000	0.2857
0.90	ENUM	0	500	4.06	0.02	3.23	5.29				
	GD	0	500	66.43	0.41	41.97	97.87	0.0000	0.0000	0.0000	0.0000
	GI	0	500	12.20	0.05	9.42	16.25	0.0000	0.0000	0.0000	0.0000
	GID	0	500	12.59	0.06	9.73	16.71	0.0000	0.0000	0.0000	0.0000
	SAD	0	497	26.90	0.14	19.95	39.80	0.0004	0.0003	0.0000	0.0769
	SAI	0	486	4.18	0.10	1.43	16.92	0.0022	0.0006	0.0000	0.0909
	GA	71	426	10.94	0.11	7.52	28.46	0.0006	0.0003	0.0000	0.1000

Nearly all algorithms revealed promising results in this first experiment. In the next step, we shed light on the algorithmic behavior for longer flow lines. Given that the performance of the genetic algorithm cannot keep up with the others, we do not consider it further.

4.3.2 Algorithm selection

To decide which algorithms to use eventually, we conduct another numerical experiment using a similar setting but increase the number of machines to five ($I = 5$). Again, we randomly generate 500 flow line instances using the *unbalanced scenario* (see Table 17, Experiment 2). All 500 instances are solved for the different target throughput levels with each algorithm. Table 19 summarizes the results. The three greedy heuristics still outperform the two simulated annealing approaches. There are only very few instances where SAD or SAI find better solutions than GD or GID, e.g., for $TP^T = 0.80$, there is just one instance with SAD finding a better solution than GD and two instances for SAI. Additionally, these cases may be due to rare numeric issues of the decomposition approach described in detail in Sachs et al., 2022a. Tables 25 and 26 in the appendix list the fraction of cases where each algorithm found the optimal solution and where each algorithm was outperformed by another one, providing further support for our findings.

Table 19. Results for Experiment 2 (see Table 17): 500 instances of randomly generated, unbalanced five-machine lines

		# infeasible	# best	Computational time (s)				Δ_{costs} compared to GD			
TP^T	Algorithm	solutions	solutions	Mean	Std. error	Min	Max	Mean	Std. error	Min	Max
0.80	GD	0	497	4622.84	39.21	2978.09	9320.24				
	GI	0	391	910.48	40.70	97.31	11331.99	0.0074	0.0007	−0.0294	0.1333
	GID	0	493	915.31	40.71	100.53	11335.58	0.0003	0.0002	−0.0294	0.0625
	SAD	0	36	1601.91	26.00	925.27	10117.67	0.0543	0.0013	−0.0606	0.1563
	SAI	0	36	1389.99	29.14	714.69	11147.63	0.0573	0.0014	−0.0294	0.1563
0.90	GD	0	500	4125.11	39.91	2285.86	8956.80				
	GI	0	493	1855.37	122.37	291.00	37791.21	0.0002	0.0001	0.0000	0.0179
	GID	0	498	1872.80	122.52	296.78	37876.05	0.0001	0.0000	0.0000	0.0161
	SAD	0	62	4383.43	48.21	2092.36	8737.69	0.0192	0.0005	0.0000	0.0811
	SAI	2	59	4317.01	57.71	1862.81	9854.94	0.0203	0.0005	0.0000	0.1099

Since GD and GID demonstrate exceptional performance, we use these two approaches for further analyses. We call the best solution found by running both GD and GID an "efficient" solution since we cannot prove its optimality nor determine the optimality gap.

4.3.3 Managerial insights

Up to now, there is only limited evidence about how buffer capacities and spare part base-stock levels interact. This section aims to widen the understanding of how decision-makers should simultaneously plan buffers and spare parts for manufacturing systems. Additionally, we are the first to report efficient allocation patterns of spare parts and buffer capacity for longer flow lines. Due to the lack of analytic results, we base our findings on numerical results of several thousand flow-line instances.

The first experiment stresses the differences between buffers and spare parts in the context of manufacturing systems and quantifies their impact. We decide to solve the same instances in three different modes: a) *integrated* planning, as proposed by this study, b) *only buffers*, and c) *only spares*. We generate 100 instances of unbalanced three-machine flow lines (see Table 17, Experiment 3) and solve them with each mode and different target throughput levels. Table 20 shows the results in terms of the best solutions' mean TC, mean TBC, and mean TCU. Furthermore, we compare the costs with integrated planning. On the one hand, using solely spare parts is only suitable for a relatively low target throughput. For higher values, *only spares* can often not reach the throughput constraint because spare parts alone cannot cope with differences in processing times. This effect intensifies for longer flow lines because of their increasing processing times interference. On the other hand, using *only buffers* delivers feasible solutions for higher required throughput. But still, the highest throughput can seldom be obtained by isolated approaches. Buffers have inherent flexibility and can tackle breakdowns as well as processing time differences. This flexibility, however, comes at a cost: compared to the *integrated* approach, we find not only higher costs but also a much higher TBC. Hence, we also need much more space to operate such a manufacturing system. This yields our first observation:

Observation 4.1. *Integrated planning of buffer capacities and spare parts can drastically reduce the buffer capacity while giving a higher throughput.*

Table 20. Results for Experiment 3 (see Table 17): 100 instances of randomly generated, unbalanced three-machine lines under three different planning modes

						Δ_{costs} compared to Integrated			
TP^T	Case	# solvable	Mean TBC	Mean TCU	Mean TC	Mean	Std. error	Min	Max
0.65	Integrated	100	2.40	5.94	8.34				
	Only buffers	100	7.94	3.00	10.94	0.31	0.01	0.11	0.56
	Only spares	81	2.00	6.41	8.41	0.02	0.01	0.00	0.33
0.75	Integrated	100	5.74	6.00	11.74				
	Only buffers	100	27.39	3.00	30.39	1.58	0.03	0.73	2.46
	Only spares	0							
0.85	Integrated	100	13.90	6.00	19.90				
	Only buffers	7	88.71	3.00	91.71	4.43	0.24	3.31	5.06
	Only spares	0							

We want to amplify this effect for one instance of Experiment 3 listed in Table 21. Table 22 shows the results in detail. For this instance, all three planning modes meet the target throughput constraint. However, *integrated* planning saves costs of 30% (25%) compared to *only buffers* (*only spares*). Concurrently, the integrated approach achieves a slightly higher throughput. This example illustrates the possible gains of *integrated* planning: by just adding one spare part per stock point (three in total), we can save seven units of buffer capacity compared to planning with *buffers only*.

For our next experiment, we focus on the influence of spare part costs. This setup allows the first insights into the efficient allocation of buffer space and spare parts for longer flow lines. We generate 100 unbalanced five-machine instances and solve them for different target throughput levels and cost parameters from cheap, i.e., $c_i^{\text{spare}} = 0.01$, to expensive spares, i.e., $c_i^{\text{spare}} = 100, \forall i \in \mathcal{I}$, (see Table 17, Experiment 4). Figure 9

Table 21. One instance of Experiment 3 (see Table 17): randomly generated, unbalanced three-machine line

Machine i	μ_i	λ_i	γ_i
1	0.963881652	0.005115477	0.046924179
2	0.982347920	0.005102008	0.045660356
3	1.023938779	0.005098789	0.047708449

Table 22. Results for one instance (see Table 21) of Experiment 3 (see Table 17) and $TP^T = 0.65$

Case	TP	TC	TBC	TCU	C_1	C_2	Q_1	Q_2	Q_3
Integrated	0.6782	9	3	6	2	1	2	2	2
Only buffers	0.6560	13	10	3	5	5	1	1	1
Only spares	0.6500	12	2	10	1	1	3	4	3

displays the results. We can observe several things: First, buffer space strongly tends to be arranged in a bowl pattern, i.e., buffers at or near the center of a flow line are more effective (cf. Conway et al., 1988 or Hillier et al., 1993). Second, spare parts reveal the same effect. Its smaller magnitude is caused by the immense impact of an additional spare in combination with the discrete step sizes. Third, specific interactions exist between the number of component units and the buffer sizes. Comparing both allocations for the cases $TP^T = 0.80$ and (very) expensive spares, we notice that a strong bowl effect for spare parts diminishes the bowl effect for buffers. That concludes our next observation:

Observation 4.2. *Additional spare parts should be allocated near or at the center of a flow line. Their allocation strongly influences optimal buffer allocations.*

Our two experiment highlights efficient flow line designs for cases where units of one component differ from the others regarding their costs. The most interesting trade-off is created when we choose the centered machine's component to be the unit with different prices, i.e., $c_3^{\text{spare}} \in \{0.01, 0.1, 1, 10, 100\}$. Again, we generate 100 unbalanced five-machine instances and solve them for different target throughput levels and the specified cost parameters (see Table 17, Experiment 5). Figure 10 depicts the results. We still observe the bowl pattern for buffers and spares — the latter at least for cheap units of component 3. In this case, the effect even increases, and the cheap spare is used as a supplement for the proportionately more expensive spares. More interestingly,

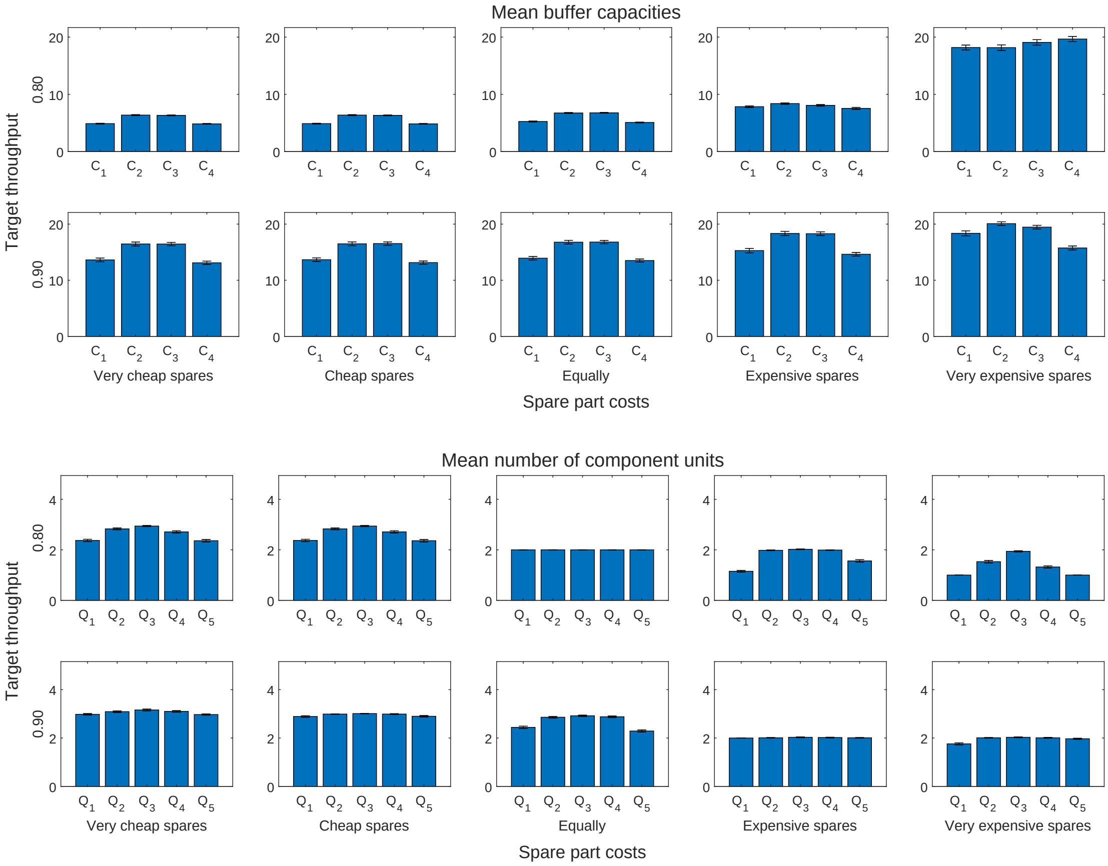


Figure 9. Mean buffer capacities and mean number of component units, including standard errors for Experiment 4 (see Table 17): 100 instances of randomly generated, unbalanced five-machine lines combined with different spare part costs $c_i^{\text{spare}} \in \{0.01, 0.1, 1, 10, 100\}, i \in \mathcal{I}$

we notice that cheap spares are used more frequently, whereas expensive spares are avoided, which is a well-known effect in spare parts planning. Beyond that, we extend the established knowledge as we can quantify the substitution effects. On the one hand, more neighboring spare parts compensate for the lesser usage of expensive spares, e.g., for $TP^T = 0.80$ when moving from $c_3 = 10$ to $c_3 = 100$ or for $TP^T = 0.90$ when the costs change from $c_3 = 1$ to $c_3 = 10$. On the other hand, additional buffer space can counterbalance the expensive units of component 3, e.g., for $TP^T = 0.80$ when the costs increase from $c_3 = 10$ to $c_3 = 100$. We summarize these findings as follows:

Observation 4.3. *Cheaper spares are stocked more often, and expensive spares less often. Both cases cause substitution effects for buffer sizes and other base-stock levels.*

Our last experiment emphasizes the influence of bottlenecks. The literature reports

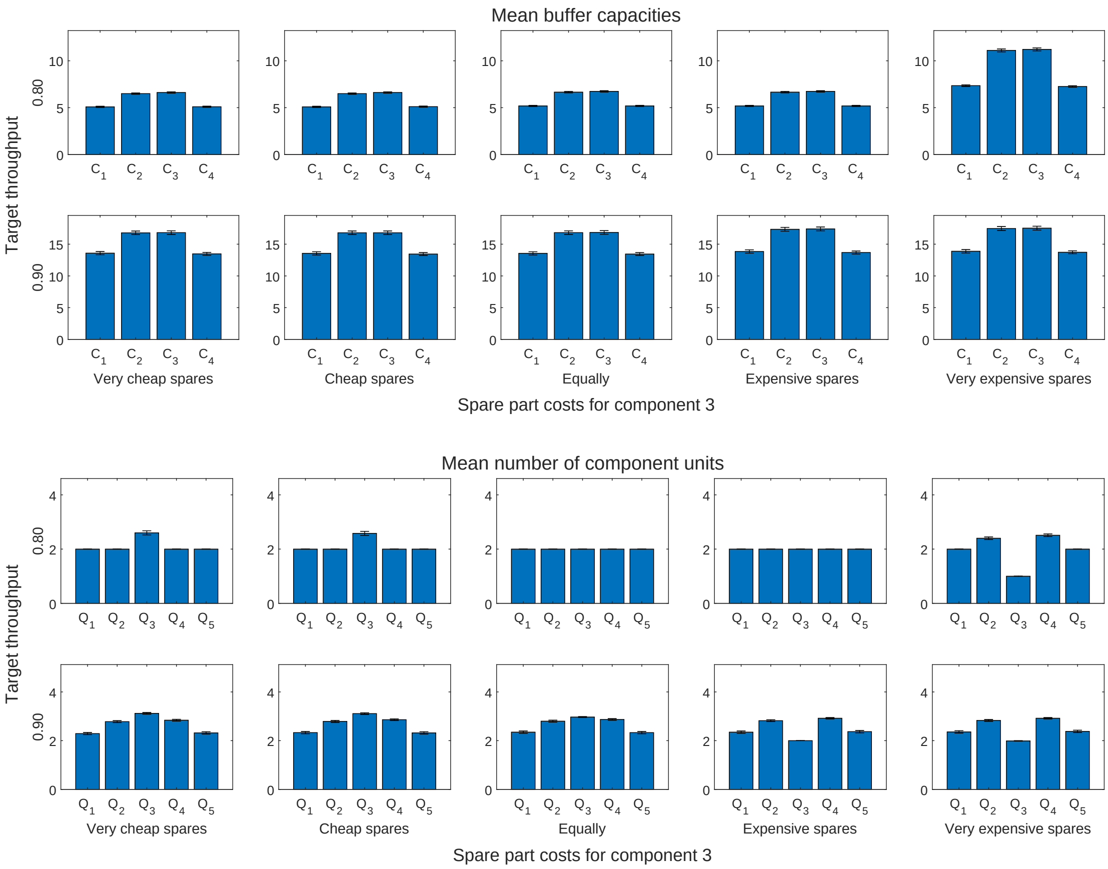


Figure 10. Mean buffer capacities and mean number of component units, including standard errors for Experiment 5 (see Table 17): 100 instances of randomly generated, unbalanced five-machine lines combined with different spare part costs $c_3^{\text{spare}} \in \{0.01, 0.1, 1, 10, 100\}$

that bottlenecks drag attention towards them (Weiss et al., 2019). In order to focus on the bottleneck itself, we use the *balanced scenario* of a five-machine flow line. From this setup, we deduce 100 instances per bottleneck position $b \in \mathcal{I}$. The bottleneck parameters are randomly generated (see Table 17, Experiment 6). Then these 501 instances (1 balanced flow line without any bottleneck and 100 flow lines for each bottleneck position) are solved for different target throughput levels. The results are depicted in Figure 11. For both target throughput levels, we can observe that the allocation of buffer capacities shifts considerably towards the bottleneck. The same effect holds for the spare parts, especially for a high target throughput. This leads us to the following result:

Observation 4.4. *The occurrence of bottlenecks changes efficient system designs to have more buffer capacity and spare parts located close to the bottleneck.*

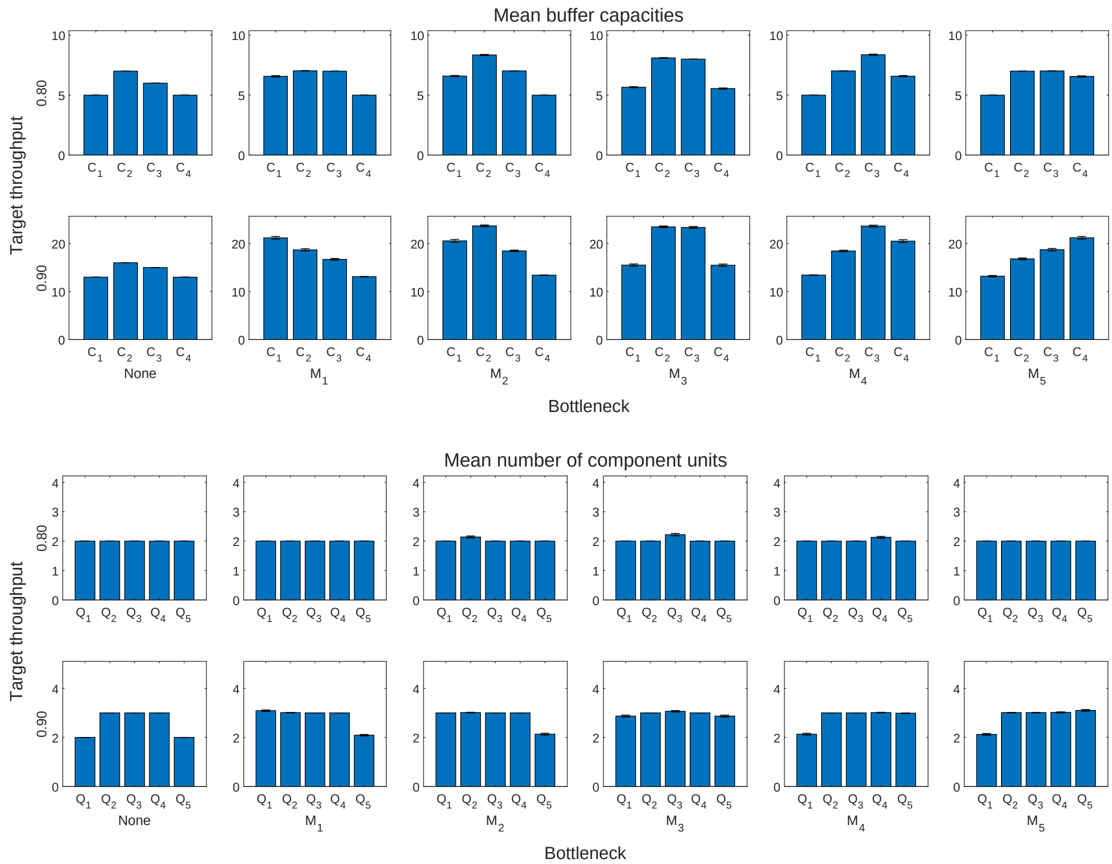


Figure 11. Mean buffer capacities and mean number of component units, including standard errors for Experiment 6 (see Table 17): 100 instances of randomly generated, unbalanced five-machine lines combined with different bottleneck scenarios (bottleneck: *none* or M_b, $b \in \mathcal{I}$)

In summary, we provide new insights on efficient joint allocations of buffer capacity and spare part stocks in manufacturing systems. Although some previous evidence carries over to the new design problem, there are complex interaction effects that make a joint analysis necessary.

4.4 Summary

This study analyzed the efficient joint allocations of buffers and spare parts in a manufacturing system with stochastic influences. Processing times, lifetimes of the machine-specific critical components, and spare part replenishment times were assumed to be exponentially distributed. Amplifying the existing terminology, we coin the proposed optimization problem primal *buffer and spare part allocation problem* (primal

BSAP). To solve this NP-hard problem, we developed three greedy heuristics, which are related to pseudo-gradient-based approaches from the buffer planning literature. Additionally, we implemented two metaheuristics (simulated annealing and a genetic algorithm). We validated all algorithms using complete enumeration as a benchmark for two-machine systems and compared them for larger systems. Therefore, we randomly generated and solved hundreds of flow line instances. The results reveal the well-suited behavior of the greedy approaches, which deliver superior performance compared to the metaheuristics. As the evaluative procedure, we rely on the decomposition approach by Sachs et al. (2022a).

Our research fits well in the body of literature as we developed new insights and observed common knowledge. Our results explicitly show that the bowl allocation of buffers is beneficial in most situations. We also observed exceptions as already known for bottlenecks. In addition, we verified the results from spare parts planning, indicating that expensive spares are avoided, whereas cheap spares are included numerously.

We studied several situations and based our novel findings on thousands of instances of unbalanced flow lines. We observed that planners should install additional spares at or near the center of the flow line even if buffers are already arranged similarly. In our analysis, this effect was only dominated by the bottleneck's need for additional spares. Furthermore, we were able to illustrate and quantify the substitution effects for expensive spare parts as they can be substituted by buffers, spares, or a combination of both.

The most questionable assumption is probably about the exponentially distributed processing times. Typically, processing times are not concerned with high variability as induced by the exponential distribution (cf. Inman, 1999). Practitioners try to achieve similar cycle times, which could be modeled by identical processing times of all flow-line machines.

Additionally, an OEM could also opt for component standardization as it is known to make cost savings possible (cf. Kranenburg, 2006). Hence, all or some of the machines could contain an identical critical component. In this case, there would be no need to have separate stock points for different machines.

We address these two important issues in the next chapter by modeling a two-machine flow line with deterministic processing times in discrete time. Moreover, we provide two models: one with different critical components per machine, whereas the other one yields standardized components.

4.5 Appendix

4.5.1 Pseudo code

Algorithm 3. Greedy heuristic with high starting values and decreasing steps (GD)

Result: $TP = TP(n-1), \mathbf{C} = \mathbf{C}(n-1), \mathbf{Q} = \mathbf{Q}(n-1)$
Input: Target throughput TP^T, starting values $\mathbf{C}^{\text{start}}, \mathbf{Q}^{\text{start}}$, boundaries $\mathbf{C}^{\min}, \mathbf{Q}^{\min}$, costs $c_j^{\text{buffer}}, j \in \mathcal{J}, c_i^{\text{spares}}, i \in \mathcal{I}$

```
/* initialize                                                  */
```
$n \leftarrow 0$
$\mathbf{C}(0) \leftarrow \mathbf{C}^{\text{start}}$
$\mathbf{Q}(0) \leftarrow \mathbf{Q}^{\text{start}}$
```
/* iterate as long as feasibility is not lost                  */
```
while $TP(n) > TP^T$ **do**
    $n \leftarrow n+1$
    `/* check all feasible neighboring decrements */`
    for $k \leftarrow 1$ **to** $2 \cdot I - 1$ **do**
        $\mathbf{C}(n-1,k) \leftarrow \mathbf{C}(n-1)$
        $\mathbf{Q}(n-1,k) \leftarrow \mathbf{Q}(n-1)$
        if $k < I$ **then**
            $C_k(n-1,k) \leftarrow C_k(n-1,k) - 1$
            if $C_k(n-1,k < C_k^{\min}$ **then continue**
        else
            $Q_{k-I+1}(n-1,k) \leftarrow Q_{k-I+1}(n-1,k) - 1$
            if $Q_{k-I+1}(n-1,k) < Q_{k-I+1}^{\min}$ **then continue**
        use the Markov approach or decomposition to compute $TP(\mathbf{C}(n-1,k), \mathbf{Q}(n-1,k))$
        $TP(n-1,k) \leftarrow TP(\mathbf{C}(n-1,k), \mathbf{Q}(n-1,k))$
        evaluate $f(k)$
    `/* find the best decrement (if feasible steps exist) */`
    index $\leftarrow \arg\min_k f(k)$
    if *index does not exist* **then return**
    `/* store new solution */`
    $\mathbf{C}(n) \leftarrow \mathbf{C}(n-1, \text{index})$
    $\mathbf{Q}(n) \leftarrow \mathbf{Q}(n-1, \text{index})$
    $TP(n) \leftarrow TP(n-1, \text{index})$

4.5.2 Further results

Table 23. Results for each algorithm on how many optimal solutions were found and the proportion of these that the other algorithms also found. The results are based on 500 instances of randomly generated, unbalanced two-machine lines.

			Proportion of also found optimal solutions						
TP^T	Algorithm	# optimal solutions	ENUM	GD	GI	GID	SAD	SAI	GA
0.80	ENUM	500		1.0000	1.0000	1.0000	0.9860	0.9840	0.9260
	GD	500	1.0000		1.0000	1.0000	0.9860	0.9840	0.9260
	GI	500	1.0000	1.0000		1.0000	0.9860	0.9840	0.9260
	GID	500	1.0000	1.0000	1.0000		0.9860	0.9840	0.9260
	SAD	493	1.0000	1.0000	1.0000	1.0000		0.9980	0.9249
	SAI	492	1.0000	1.0000	1.0000	1.0000	1.0000		0.9248
	GA	463	1.0000	1.0000	1.0000	1.0000	0.9849	0.9827	
0.90	ENUM	500		1.0000	1.0000	1.0000	0.9940	0.9720	0.8520
	GD	500	1.0000		1.0000	1.0000	0.9940	0.9720	0.8520
	GI	500	1.0000	1.0000		1.0000	0.9940	0.9720	0.8520
	GID	500	1.0000	1.0000	1.0000		0.9940	0.9720	0.8520
	SAD	497	1.0000	1.0000	1.0000	1.0000		0.9759	0.8511
	SAI	486	1.0000	1.0000	1.0000	1.0000	0.9979		0.8477
	GA	426	1.0000	1.0000	1.0000	1.0000	0.9930	0.9671	

Table 24. Results for each algorithm on the proportion of better solutions found by the other algorithms. The results are based on 500 instances of randomly generated, unbalanced two-machine lines.

			Proportion of found better solutions						
TP^T	Algorithm	# solvable instances	ENUM	GD	GI	GID	SAD	SAI	GA
0.80	ENUM	500		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	GD	500	0.0000		0.0000	0.0000	0.0000	0.0000	0.0000
	GI	500	0.0000	0.0000		0.0000	0.0000	0.0000	0.0000
	GID	500	0.0000	0.0000	0.0000		0.0000	0.0000	0.0000
	SAD	500	0.0140	0.0140	0.0140	0.0140		0.0000	0.0140
	SAI	500	0.0160	0.0160	0.0160	0.0160	0.0020		0.0160
	GA	500	0.0400	0.0400	0.0400	0.0400	0.0400	0.0400	
0.90	ENUM	500		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	GD	500	0.0000		0.0000	0.0000	0.0000	0.0000	0.0000
	GI	500	0.0000	0.0000		0.0000	0.0000	0.0000	0.0000
	GID	500	0.0000	0.0000	0.0000		0.0000	0.0000	0.0000
	SAD	500	0.0060	0.0060	0.0060	0.0060		0.0020	0.0060
	SAI	500	0.0280	0.0280	0.0280	0.0280	0.0240		0.0280
	GA	500	0.0060	0.0060	0.0060	0.0060	0.0060	0.0060	

Table 25. Results for each algorithm on how many optimal solutions were found and the proportion of these that the other algorithms also found. The results are based on 500 instances of randomly generated, unbalanced five-machine lines.

			Proportion of also found best solutions				
TP^T	Algorithm	# best solutions	GD	GI	GID	SAD	SAI
0.80	GD	497		0.7847	0.9899	0.0704	0.0704
	GI	391	0.9974		1.0000	0.0818	0.0767
	GID	493	0.9980	0.7931		0.0710	0.0710
	SAD	36	0.9722	0.8889	0.9722		0.0833
	SAI	36	0.9722	0.8333	0.9722	0.0833	
0.90	GD	500		0.9860	0.9960	0.1240	0.1180
	GI	493	1.0000		1.0000	0.1258	0.1197
	GID	498	1.0000	0.9900		0.1245	0.1185
	SAD	62	1.0000	1.0000	1.0000		0.2258
	SAI	59	1.0000	1.0000	1.0000	0.2373	

Table 26. Results for each algorithm on the proportion of better solutions found by the other algorithms. The results are based on 500 instances of randomly generated, unbalanced five-machine lines.

			Proportion of found better solutions				
TP^T	Algorithm	# solvable instances	GD	GI	GID	SAD	SAI
0.80	GD	500		0.0020	0.0020	0.0020	0.0020
	GI	500	0.2140		0.2140	0.0220	0.0220
	GID	500	0.0100	0.0000		0.0020	0.0040
	SAD	500	0.9280	0.8500	0.9200		0.3240
	SAI	500	0.9280	0.8620	0.9220	0.3760	
0.90	GD	500		0.0000	0.0000	0.0000	0.0000
	GI	500	0.0140		0.0100	0.0000	0.0000
	GID	500	0.0040	0.0000		0.0000	0.0000
	SAD	500	0.8760	0.8720	0.8740		0.2780
	SAI	500	0.8780	0.8700	0.8740	0.3520	

4.5.3 Numeric values for figures

Table 27. Data for Figure 9, based on Experiment 4 (see Table 17)

		Mean buffer capacities (std. error)				Mean number of component units (std. error)				
TP^T	Case	C_1	C_2	C_3	C_4	Q_1	Q_2	Q_3	Q_4	Q_5
0.80	Very cheap spares	4.9 (0.1)	6.4 (0.1)	6.3 (0.1)	4.8 (0.1)	2.4 (0.0)	2.8 (0.0)	2.9 (0.0)	2.7 (0.0)	2.4 (0.0)
	Cheap spares	4.9 (0.1)	6.4 (0.1)	6.3 (0.1)	4.8 (0.1)	2.4 (0.0)	2.8 (0.0)	2.9 (0.0)	2.7 (0.0)	2.4 (0.0)
	Equally	5.3 (0.1)	6.7 (0.1)	6.8 (0.1)	5.1 (0.1)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)
	Expensive spares	7.8 (0.1)	8.4 (0.1)	8.1 (0.1)	7.5 (0.2)	1.1 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)	1.6 (0.0)
	Very expensive spares	18.2 (0.4)	18.2 (0.5)	19.1 (0.5)	19.7 (0.5)	1.0 (0.0)	1.5 (0.1)	1.9 (0.0)	1.3 (0.0)	1.0 (0.0)
0.90	Very cheap spares	13.6 (0.3)	16.5 (0.3)	16.4 (0.3)	13.1 (0.3)	3.0 (0.0)	3.1 (0.0)	3.2 (0.0)	3.1 (0.0)	3.0 (0.0)
	Cheap spares	13.7 (0.3)	16.5 (0.3)	16.5 (0.3)	13.1 (0.3)	2.9 (0.0)	3.0 (0.0)	3.0 (0.0)	3.0 (0.0)	2.9 (0.0)
	Equally	13.9 (0.3)	16.8 (0.3)	16.8 (0.3)	13.5 (0.3)	2.4 (0.0)	2.9 (0.0)	2.9 (0.0)	2.9 (0.0)	2.3 (0.0)
	Expensive spares	15.3 (0.4)	18.3 (0.4)	18.3 (0.3)	14.6 (0.3)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)
	Very expensive spares	18.4 (0.4)	20.1 (0.3)	19.5 (0.3)	15.8 (0.4)	1.8 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)

Table 28. Data for Figure 10, based on Experiment 5 (see Table 17)

		Mean buffer capacities (std. error)				Mean number of component units (std. error)				
TP^T	Case	C_1	C_2	C_3	C_4	Q_1	Q_2	Q_3	Q_4	Q_5
0.80	Very cheap spare	5.1 (0.1)	6.5 (0.1)	6.6 (0.1)	5.1 (0.1)	2.0 (0.0)	2.0 (0.0)	2.6 (0.1)	2.0 (0.0)	2.0 (0.0)
	Cheap spare	5.1 (0.1)	6.5 (0.1)	6.6 (0.1)	5.1 (0.1)	2.0 (0.0)	2.0 (0.0)	2.6 (0.1)	2.0 (0.0)	2.0 (0.0)
	Equally	5.2 (0.1)	6.7 (0.1)	6.7 (0.1)	5.2 (0.1)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)
	Expensive spare	5.2 (0.1)	6.7 (0.1)	6.7 (0.1)	5.2 (0.1)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)
	Very expensive spare	7.3 (0.1)	11.1 (0.2)	11.2 (0.2)	7.3 (0.1)	2.0 (0.0)	2.4 (0.0)	1.0 (0.0)	2.5 (0.1)	2.0 (0.0)
0.90	Very cheap spare	13.6 (0.3)	16.8 (0.3)	16.8 (0.3)	13.5 (0.2)	2.3 (0.0)	2.8 (0.0)	3.1 (0.0)	2.8 (0.0)	2.3 (0.0)
	Cheap spare	13.6 (0.3)	16.8 (0.3)	16.8 (0.3)	13.4 (0.2)	2.3 (0.0)	2.8 (0.0)	3.1 (0.0)	2.9 (0.0)	2.3 (0.0)
	Equally	13.6 (0.2)	16.8 (0.3)	16.9 (0.3)	13.4 (0.2)	2.4 (0.0)	2.8 (0.0)	3.0 (0.0)	2.9 (0.0)	2.3 (0.0)
	Expensive spare	13.8 (0.3)	17.4 (0.3)	17.4 (0.3)	13.7 (0.2)	2.4 (0.0)	2.8 (0.0)	2.0 (0.0)	2.9 (0.0)	2.4 (0.0)
	Very expensive spare	13.9 (0.3)	17.5 (0.3)	17.5 (0.3)	13.7 (0.2)	2.4 (0.0)	2.8 (0.0)	2.0 (0.0)	2.9 (0.0)	2.4 (0.0)

Table 29. Data for Figure 11, based on Experiment 6 (see Table 17)

		Mean buffer capacities (std. error)				Mean number of component units (std. error)				
TP^T	Bottleneck	C_1	C_2	C_3	C_4	Q_1	Q_2	Q_3	Q_4	Q_5
0.80	None	5.0 (0.0)	7.0 (0.0)	6.0 (0.0)	5.0 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)
	M_1	6.6 (0.1)	7.0 (0.0)	7.0 (0.0)	5.0 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)
	M_2	6.6 (0.0)	8.3 (0.0)	7.0 (0.0)	5.0 (0.0)	2.0 (0.0)	2.1 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)
	M_3	5.7 (0.0)	8.1 (0.0)	8.0 (0.0)	5.5 (0.1)	2.0 (0.0)	2.0 (0.0)	2.2 (0.0)	2.0 (0.0)	2.0 (0.0)
	M_4	5.0 (0.0)	7.0 (0.0)	8.4 (0.0)	6.6 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)	2.1 (0.0)	2.0 (0.0)
	M_5	5.0 (0.0)	7.0 (0.0)	7.0 (0.0)	6.6 (0.1)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)
0.90	None	13.0 (0.0)	16.0 (0.0)	15.0 (0.0)	13.0 (0.0)	2.0 (0.0)	3.0 (0.0)	3.0 (0.0)	3.0 (0.0)	2.0 (0.0)
	M_1	21.2 (0.3)	18.7 (0.3)	16.7 (0.2)	13.1 (0.1)	3.1 (0.0)	3.0 (0.0)	3.0 (0.0)	3.0 (0.0)	2.1 (0.0)
	M_2	20.6 (0.3)	23.7 (0.2)	18.5 (0.2)	13.4 (0.1)	3.0 (0.0)	3.0 (0.0)	3.0 (0.0)	3.0 (0.0)	2.1 (0.0)
	M_3	15.5 (0.2)	23.5 (0.2)	23.3 (0.2)	15.5 (0.2)	2.9 (0.0)	3.0 (0.0)	3.1 (0.0)	3.0 (0.0)	2.9 (0.0)
	M_4	13.4 (0.1)	18.4 (0.2)	23.6 (0.2)	20.5 (0.3)	2.1 (0.0)	3.0 (0.0)	3.0 (0.0)	3.0 (0.0)	3.0 (0.0)
	M_5	13.2 (0.1)	16.8 (0.2)	18.7 (0.3)	21.2 (0.3)	2.1 (0.0)	3.0 (0.0)	3.0 (0.0)	3.0 (0.0)	3.1 (0.0)

This page intentionally left blank

Chapter

5 Two-Machine Flow Line With Fixed Cycle Times

The soufflé will either rise or it won't, and there's not a damn thing you can do about it.

Captain Benjamin Sisko

The following chapter is published in the *European Journal of Operational Research* (Kiesmüller & Sachs, 2020). Its main focus is widening the view on the design problem of flow lines with buffers and spare parts. We want to tackle two important issues: Get rid of the assumption of exponentially distributed processing times and analyze the impact of component standardization. The exponential distribution is questionable when using it for processing times (cf. Inman, 1999). Hence, this effort can be seen as a robustness check for the previously obtained results. We know that component standardization can enable cost savings (cf. Kranenburg, 2006). However, we strive to generate insights into the impact on manufacturing systems. For both issues, we use two-machine flow lines. Section 5.1 describes the general problem. In Section 5.2, we introduce a transfer-line model with two machines suffering from breakdowns of an identical failure-prone critical component. We present the discrete-time Markov chain model and how to compute steady-state probabilities. Furthermore, we show the optimization problem, solve it via complete enumeration, and present optimal system designs regarding buffer capacity and base-stock level. Similarly, Section 5.3 introduces the mathematical model for a situation with two different critical components. Again, we generate insights on optimal system designs. Moreover, we illustrate that the lead-time distribution only impacts the results to a very limited extent and derive managerial insights on the effects of component standardization. Lastly, in Section 5.4, we summarize the fundamental results.

5.1 Problem description

We consider a transfer line consisting of two machines in series, which is used to produce a single discrete product, where the first production step is executed on machine 1, and afterward, the workpiece has to be processed on machine 2. The processing times on both machines are constant and equal such that the processing time can be interpreted as a time unit. Although perfectly identical processing times are hard to achieve, it is one of the goals of lean manufacturing. Efforts should be invested to achieve balanced processing times (Linck & Cochran, 1999). Furthermore, the idea of balanced cycle times is to simultaneously decrease WIP inventories and adjust the production rates to the customer's demand rates (Frandson et al., 2013). This approach can realize remarkable savings (Frandson & Tommelein, 2014).

In a perfect environment, the transfer line would be able to produce one unit per time unit. However, it is usually impossible to achieve this throughput due to unexpected events. In this chapter, we assume that each machine contains one unit of a critical component, which means that failure of this component leads to a stoppage of the related machine. We consider operation-dependent failures of the component such that a component can only fail if the machine is producing. The lifetime of a component is assumed to be geometrically distributed, and the probability of failure of the component during the processing of a unit at machine i is denoted as $p_i, (i = 1, 2)$. This type of distribution is especially applicable for electronic components because they are not affected by wear-out effects (Birolini, 2012). Additionally, Bernoulli demand models for slow-moving spare parts, which are induced by geometrically distributed failure times, are supported by empiric data (see Syntetos et al., 2012, for an overview of spare parts demand distributions).

In order to prevent the production stop of one machine from affecting the other machine, a buffer with a capacity of C can be installed between the two machines. Thus, if the second machine stops working due to a failed component, the first machine can continue to work, and processed workpieces can be temporarily stored in the buffer until machine 2 is repaired. The first machine only has to stop producing when the buffer is full for it to get blocked. In this chapter, we consider blocking after service. This means that the first machine continues the production of the workpiece even if the buffer is full and gets blocked after the production process is finished. However, the analysis for blocking before service is similar. In addition, the buffer helps to reduce the starving of machine 2 if the first machine fails and there are still workpieces in the buffer available. We assume that the transportation times for the workpieces from the machine to the buffer and the other way around are negligible.

After the failure of a component, the machine can only continue to work if the machine is repaired. In this chapter, a repair-by-replacement strategy is considered, which means that spare parts are kept close to the transfer line such that the broken part can be immediately replaced with a new part if there is one in stock. If a spare part is available, it is assumed that the repair time is negligible such that failure of the component does not lead to a stoppage of a machine and expensive downtime due to production loss. Only if no spare part is available does the machine stop working in case of a failed component, and it remains down until a new part is delivered. During this period, the workpiece stays at the machine, and processing is resumed after repair.

If a failed component has been replaced by a spare part, a new spare part has to be ordered to replenish the inventory. Fixed ordering costs are assumed to be negligible compared to the holding cost of a spare part such that a one-for-one replenishment policy (Muckstadt, 2005) with base-stock level S_i is applied. It is assumed that the replenishment lead time for a spare part for machine i can be modeled with a geometrically distributed random variable with parameter $r_i, (i = 1, 2)$. This choice of the distribution is in line with earlier assumptions regarding the repair times (see, e.g., Buzacott, 1967; Dallery & Gershwin, 1992) and therefore ensures comparability. Furthermore, it is known as a conclusion of the Palm–Khintchine theorem that the type of the distribution does not have a large effect on the performance of the system but especially the mean value does (Alfredsson & Verrijdt, 1999; Zimmermann & Kiesmüller, 2019). This issue is also addressed in Section 5.3.3.

We assume the following sequence of events: at the beginning of a time unit, failures are observed and spare parts delivered if appropriate, while buffer level changes occur at the end of a time unit. We further assume that if both machines are waiting for a spare part, machine 2 is repaired first because the output can only be produced if machine 2 is working.

In order to obtain a high throughput of the unreliable transfer line, a designer can consider two different design options for the manufacturing system. On the one hand, he has to determine the size C of the interstage buffer, which helps to decouple adjacent machines. On the other hand, he has to determine how many spare parts to keep in stock. This decision has an influence on the isolated production rate of a machine, which can be increased with higher stock levels due to higher machine availability. Hence, both options can improve the throughput of the system. However, a large buffer capacity can lead to high WIP, and a large number of spare parts can lead to high inventory levels, both of which result in high tied-up capital costs. Thus, the planner has to resolve the trade-off between production output and the high cost of tied-up capital.

5.2 Analysis of a system with one critical component

Component part standardization can lead to a situation where the same critical component is built into each machine of the transfer line. This situation can be encountered in manufacturing systems using production robots such that sequential processing steps are carried out using very similar robots (for systems with commonality in spare parts or machines see, e.g., Kranenburg and Van Houtum (2007) and Kahan et al. (2009) and for production systems including robots and incorporating cycle times, e.g., Lopes et al. (2017) and Weckenborg et al. (2019)). According to Okamura and Yamashina (1977), differences in breakdown rates reduce the utility of buffers. Thus, roughly identical breakdown rates should be achieved. This is far easier to obtain by incorporating identical critical components. Therefore, we assume that both machines suffer from the same failure mode such that we can use $p = p_1 = p_2$ in the following. Since there is only one type of spare part, there is only one stock point with one base-stock level S, which also results in $r = r_1 = r_2$. The system is illustrated in Figure 12.

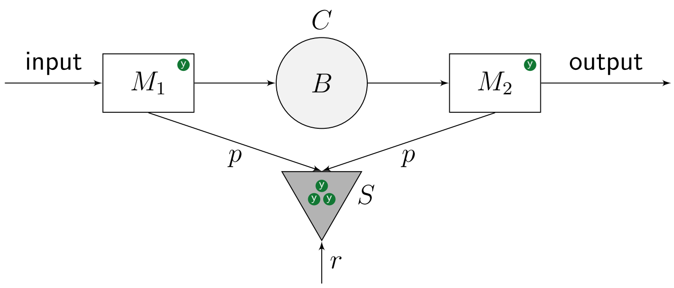


Figure 12. Transfer line with two machines and one critical component

5.2.1 Mathematical model

The system can be modeled as a discrete-time Markov chain, where the state of the system is denoted by the quadruple $(n, s, \alpha_1, \alpha_2)$. The first component is related to the number of units in the buffer and at machine 2. Since there is a buffer with a capacity of C, we have $n = 0, 1, \ldots, C, C+1$. In addition, we have to model the situation where the first machine is blocked, which is done by setting $n = C+2$. The second component is related to the spare parts and denotes the number of outstanding replenishment orders. At the beginning of the process, there are S units in stock and two units in the machines, resulting in a maximum number of outstanding orders given as $S+2$. Finally, $\alpha_i \;\; (i = 1, 2)$ denotes the state of machine i, where $\alpha_i = 0$ means that machine i is down and waiting for a spare part, while $\alpha_i = 1$ stands for an operational machine. Summing up, all possible quadruples are given as:

$$\overline{SS} = \{(n, s, \alpha_1, \alpha_2) \mid 0 \le n \le C+2, 0 \le s \le S+2, \alpha_1, \alpha_2 \in \{0, 1\}\} \tag{173}$$

Note that not all quadruples included in $\overline{SS}$ are possible due to our assumptions. Hence, the impossible states will be identified and excluded to get the state space of the Markov chain (see (183) below).

In order to derive mathematical expressions for the performance measures, we determine the steady-state probabilities denoted as

$$\mathcal{P}_D(n, s, \alpha_1, \alpha_2). \tag{174}$$

We are interested in the throughput of the system, which is defined as the average number of units produced per time unit. It can be computed by summing up all probabilities for states where machine 2 is working:

$$TP_D(C, S) = \sum_{n=1}^{C+2} \sum_{s=0}^{S} \mathcal{P}_D(n, s, 1, 1) + \sum_{n=1}^{C+2} \mathcal{P}_D(n, S+1, 0, 1). \tag{175}$$

The average number of spare parts in stock is given as

$$I(C, S) = \sum_{n=0}^{C+2} \sum_{s=0}^{S} (S - s) \cdot \mathcal{P}_D(n, s, 1, 1) \tag{176}$$

and the average number of workpieces in the system is given as

$$\begin{aligned} WIP(C,S) = 1 &+ \sum_{n=1}^{C+1}\sum_{s=0}^{S} n\cdot\mathcal{P}_D(n,s,1,1) + \sum_{n=1}^{C+1} n\cdot\mathcal{P}_D(n,S+2,0,0) \\ &+ \sum_{n=1}^{C+1} n\cdot\mathcal{P}_D(n,S+1,0,1) + \sum_{n=1}^{C+1} n\cdot\mathcal{P}_D(n,S+1,1,0) \\ &+ (C+1)\cdot\mathcal{P}_D(C+2,S+1,1,0) \end{aligned} \tag{177}$$

We formulate the following optimization problem where TP^T denotes the target throughput, and c is the cost-value ratio for a spare part with respect to the cost for a workpiece in the system, which is normalized to one. Thus $c > 1$ reflects the situation where spare parts are more expensive than workpieces and $c < 1$ where one spare part is cheaper than a workpiece. In our optimization problem, we also formulate a restriction for the maximum buffer capacity because many companies only have limited space available for buffers.

$$\begin{aligned} \min_{C,S} \quad & TC_D(C,S) := WIP(C,S) + c\cdot I(C,S) \\ \text{s.t.} \quad & TP_D(C,S) \geq TP^T \\ & 0 \leq C \leq C^{\max} \\ & 0 \leq S \leq S^{\max} \\ & C,S \in \mathbb{N}_0 \end{aligned} \tag{178}$$

If the base-stock level is chosen as $S = 0$, then there are no spare parts in stock, and each failure of a component results in a production stop of the corresponding machine and a replenishment order for the spare part. If additionally, both machines are down, then an arriving spare part is used to repair machine 2. Therefore, our system is different from the system studied in Gershwin (1994), where there is no relation between the two repair processes.

5.2.2 Analysis of the system

In order to obtain the steady-state probabilities, we have to formulate a system of equations for all possible states. We start with a discussion of states where both machines are down. This can only happen if the spare part stock is empty and the number of outstanding replenishment orders is equal to $S+2$. Since a failed machine cannot be blocked or suffer from starvation, we obtain $C+1$ states where both machines are down:

$$SS_1 = \{(n, S+2, 0, 0) \mid 1 \leq n \leq C+1\}. \tag{179}$$

When only the second machine is down, and the first is still working, then it can happen that the buffer capacity is reached and the first machine is blocked. In addition, the number of outstanding spare parts is equal to $S+1$. Furthermore, the first machine cannot starve, and we know that a state $(1, S+1, 1, 0)$ cannot be reached because a starving machine cannot fail. This yields:

$$SS_2 = \{(n, S+1, 1, 0) \mid 2 \leq n \leq C+2\}. \tag{180}$$

Similarly, the number of outstanding replenishment orders is equal to $S+1$ if only the first machine is down. In this situation, it can happen that the buffer is running empty, and the second machine is starving. Furthermore, the state $(C+1, S+1, 0, 1)$ is not possible because a blocked machine cannot fail. Thus, we get

$$SS_3 = \{(n, S+1, 0, 1) \mid 0 \leq n \leq C\}. \tag{181}$$

Finally, starving and blocking cannot be observed when both machines are working, and the number of outstanding replenishment orders cannot be larger than S.

$$SS_4 = \{(n, s, 1, 1) \mid 1 \leq n \leq C+1, 0 \leq s \leq S\}. \tag{182}$$

Summing up, the state space of the Markov chain is given as

$$SS = SS_1 \cup SS_2 \cup SS_3 \cup SS_4 \tag{183}$$

and the number of states is given as $|SS| = (S+4)(C+1)$. We formulate the equations for the general situations where $C \geq 2$ and $S \geq 2$. The special cases for smaller values of C and S can be derived in a similar manner. The balance equations can be found in the appendix (Section 5.5).

The equations reveal that there is a large difference between the transitions related to the buffer and the transitions related to the spare parts stock. While only one workpiece can be transferred to or from the buffer during one unit of time, it can be seen that several spare parts can arrive at the stock point at the same time, and up to two spare parts can be taken out of stock to repair failed machines. Consequently, transitions do not only occur between neighboring states, making the system equations intractable for an analytic solution.

5.2.3 Numerical results

Based on the equations formulated above, the steady-state probabilities for each state as well as the performance measures can be computed numerically. We implemented the model in MATLAB to solve the optimization problem (178) and computed the optimal system design (C^{*}, S^{*}) for several instances. We applied a complete enumeration with $C^{\max} = 100$ and $S^{\max} = 6$ because this latter value already leads to a machine availability of almost 100%. For different target values for the throughput, different cost-value ratios c and a failure probability of $p = 0.1$, the optimal system designs are presented in Table 30.

It can be observed that no buffer is installed between the two machines when spare parts are cheap compared to the cost of a workpiece (small value for c). In order to reach the required throughput, only spare parts are held in stock. An interstage buffer is only used for large values of c, which means that the costs for holding spare parts are higher than those for WIP. In addition, if the base-stock level necessary to reach the target throughput leads to much higher throughput than required, it may be more beneficial to install buffer capacity in addition to spare parts stock. This effect can be observed for the instances with $TP^{T} = 0.65, c \in \{10, 50, 100\}$ and $r = 0.06$ or $r = 0.1$, or $TP^{T} = 0.75, c \in \{10, 50, 100\}$ and $r = 0.14$, or $TP^{T} = 0.80, c \in \{1, 10, 50, 100\}$ and $r = 0.1$, or $TP^{T} = 0.90, c \in \{10, 50, 100\}$ and $r = 0.14$. We can conclude that spare parts stock is mainly used to come close to the target throughput, and sometimes, depending on the cost value, buffer capacity is used for fine-tuning. The reason for this is the different effect of an additional buffer capacity or spare part on the throughput. This is also illustrated in Figure 13a, where the impact of one spare part on the throughput $(\tilde{\Delta}_{TP}(p))$ is depicted as a function of the failure probability p

Table 30. Optimal system designs with one critical component

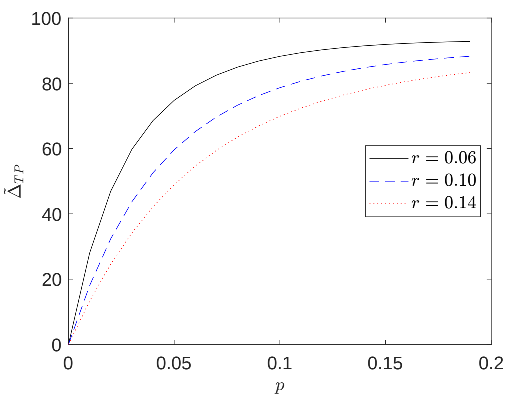
		$r = 0.06$			$r = 0.1$			$r = 0.14$		
TP^T	c	(C^*, S^*)	$TP_D(\cdot)$	$TC_D(\cdot)$	(C^*, S^*)	$TP_D(\cdot)$	$TC_D(\cdot)$	(C^*, S^*)	$TP_D(\cdot)$	$TC_D(\cdot)$
	0.1	$(0, 2)$	0.6209	1.83	$(0, 1)$	0.6061	1.81	$(0, 1)$	0.7108	1.88
	1.0	$(0, 2)$	0.6209	2.13	$(0, 1)$	0.6061	1.99	$(0, 1)$	0.7108	2.13
0.60	10	$(0, 2)$	0.6209	5.04	$(0, 1)$	0.6061	3.75	$(0, 1)$	0.7108	4.69
	50	$(0, 2)$	0.6209	17.99	$(0, 1)$	0.6061	11.56	$(0, 1)$	0.7108	16.04
	100	$(0, 2)$	0.6209	34.18	$(0, 1)$	0.6061	21.32	$(0, 1)$	0.7108	30.23
	0.1	$(0, 3)$	0.7629	1.95	$(0, 2)$	0.7958	1.95	$(0, 1)$	0.7108	1.88
	1.0	$(0, 3)$	0.7629	2.58	$(0, 2)$	0.7958	2.51	$(0, 1)$	0.7108	2.13
0.65	10	$(7, 2)$	0.6522	7.23	$(8, 1)$	0.6525	6.29	$(0, 1)$	0.7108	4.69
	50	$(7, 2)$	0.6522	19.43	$(8, 1)$	0.6525	13.66	$(0, 1)$	0.7108	16.04
	100	$(7, 2)$	0.6522	34.68	$(8, 1)$	0.6525	22.86	$(0, 1)$	0.7108	30.23
	0.1	$(0, 3)$	0.7629	1.95	$(0, 2)$	0.7958	1.95	$(0, 1)$	0.7108	1.88
	1.0	$(0, 3)$	0.7629	2.58	$(0, 2)$	0.7958	2.51	$(0, 1)$	0.7108	2.13
0.70	10	$(0, 3)$	0.7629	8.91	$(0, 2)$	0.7629	8.09	$(0, 1)$	0.7108	4.69
	50	$(0, 3)$	0.7629	37.04	$(0, 2)$	0.7629	21.91	$(0, 1)$	0.7108	16.04
	100	$(3, 3)$	0.7775	71.90	$(1, 2)$	0.8037	63.91	$(0, 1)$	0.7108	30.23
	0.1	$(0, 3)$	0.7629	1.95	$(0, 2)$	0.7958	1.95	$(0, 2)$	0.8829	2.02
	1.0	$(0, 3)$	0.7629	2.58	$(0, 2)$	0.7958	2.51	$(0, 2)$	0.8829	2.80
0.75	10	$(0, 3)$	0.7629	8.91	$(0, 2)$	0.7629	8.09	$(6, 1)$	0.7515	6.74
	50	$(0, 3)$	0.7629	37.04	$(0, 2)$	0.7629	21.91	$(6, 1)$	0.7515	17.71
	100	$(3, 3)$	0.7775	71.90	$(1, 2)$	0.8037	63.91	$(6, 1)$	0.7515	31.43
	0.1	$(0, 4)$	0.8663	2.05	$(0, 3)$	0.9105	2.08	$(0, 2)$	0.8829	2.02
	1.0	$(0, 4)$	0.8663	3.18	$(1, 2)$	0.8037	2.94	$(0, 2)$	0.8829	2.80
0.80	10	$(0, 4)$	0.8663	14.44	$(1, 2)$	0.8037	8.48	$(0, 2)$	0.8829	10.54
	50	$(0, 4)$	0.8663	64.51	$(1, 2)$	0.8037	33.12	$(0, 2)$	0.8829	44.94
	100	$(3, 4)$	0.8765	126.78	$(1, 2)$	0.8037	63.91	$(0, 2)$	0.8829	87.95
	0.1	$(0, 4)$	0.8663	2.05	$(0, 3)$	0.9105	2.08	$(0, 2)$	0.8829	2.02
	1.0	$(0, 4)$	0.8663	3.18	$(0, 3)$	0.9105	3.22	$(0, 2)$	0.8829	2.80
0.85	10	$(0, 4)$	0.8663	14.44	$(0, 3)$	0.9105	14.67	$(0, 2)$	0.8829	10.54
	50	$(0, 4)$	0.8663	64.51	$(0, 3)$	0.9105	65.56	$(0, 2)$	0.8829	44.94
	100	$(3, 4)$	0.8765	126.78	$(0, 3)$	0.9105	129.16	$(0, 2)$	0.8829	87.95
	0.1	$(0, 5)$	0.9330	2.16	$(0, 3)$	0.9105	2.08	$(0, 3)$	0.9623	2.15
	1.0	$(0, 5)$	0.9330	3.92	$(0, 3)$	0.9105	3.22	$(0, 3)$	0.9623	3.64
0.90	10	$(0, 5)$	0.9330	21.56	$(0, 3)$	0.9105	14.67	$(4, 2)$	0.9003	11.91
	50	$(0, 5)$	0.9330	99.96	$(0, 3)$	0.9105	65.56	$(4, 2)$	0.9003	45.97
	100	$(1, 5)$	0.9357	197.87	$(0, 3)$	0.9105	129.16	$(4, 2)$	0.9003	88.54

and compared with the impact of buffer capacity on the throughput ($\Delta_{TP}(p)$) given as

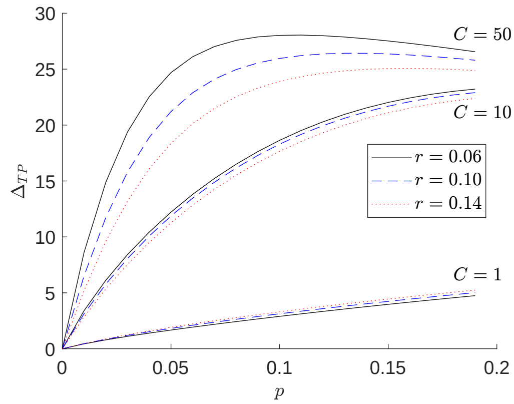
$$\tilde{\Delta}_{TP}(p,r) = \frac{TP_D(0,1) - TP_D(0,0)}{TP_D(0,0)} \cdot 100\% \tag{184}$$

$$\Delta_{TP}(p,r) = \frac{TP_D(C,0) - TP_D(0,0)}{TP_D(0,0)} \cdot 100\% \tag{185}$$

The latter is shown as a function of the failure probability p in Figure 13b for different values of the buffer capacity C.



(a) Impact of one spare part on throughput; $C = 0$



(b) Impact of buffer capacity on throughput; $S = 0$

Figure 13. Impact of decision variables on throughput depending on failure and repair probabilities

It can be seen that keeping one spare part in stock has a much larger effect than having one unit of buffer capacity. However, ten or fifty additional units of buffer capacity also cannot achieve the same throughput increase as just one spare part.

The results in Table 30 also suggest that there is another reason for installing a buffer to guarantee a specific target throughput when spare parts are very expensive. Especially for $r = 0.06$ and $TP^T \geq 0.7$, the optimal design includes positive buffer capacity for $c = 100$. This can be explained by the impact of buffer space on the average spare parts stock. More buffer capacity leads to less starving and blocking and, therefore, to a higher machine productivity. Thus, there are more opportunities to fail, and more spare parts are needed. This reduces the average stock on hand of spare parts. When spare parts are very expensive, this reduction in holding costs also compensates for the higher WIP-related costs and is optimal for designing a system with spare parts and buffers. However, the relative difference in cost to an optimal system without a buffer is quite small in such situations because this effect is small. This is completely different in situations where a buffer is used instead of an additional spare part. In these cases, it can be very expensive if only spare parts are used to avoid long machine downtimes. For example, the relative cost difference between systems $(4, 2)$ and $(0, 3)$ in case of $c = 100$ and $r = 0.14$ is 90%. Thus, we can conclude that it is necessary to optimize the base-stock level and buffer capacity simultaneously.

5.3 Analysis of a system with two critical components

In situations where machines have two diverse critical components, there are different failure modes for each machine, and two stock points are needed with base-stock levels S_1 and S_2. Hence, we use p_i as the probability of a failure of component i during the processing of a unit at machine i and r_i for the probability that a replenishment order arrives at stock point i $(i = 1, 2)$. All other assumptions mentioned in Section 5.1 still hold. The system is presented in Figure 14.

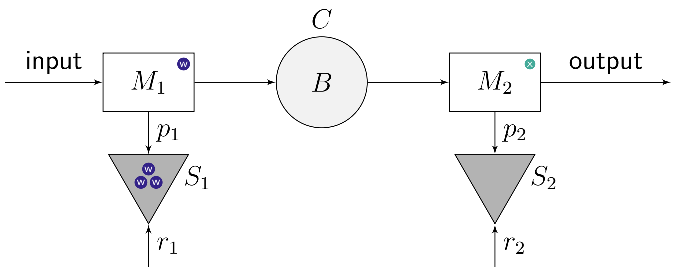


Figure 14. Transfer line with two machines and two critical components

5.3.1 Mathematical model

In line with the former notation, the state of the system is denoted by (n, s_1, s_2). Again, the first component is related to the size of the buffer and denotes the number of workpieces in the system where production on machine 1 is already finished ($n = 0, 1, \ldots, C + 2$). In the case of $n = 0$, the second machine is starving, and the first machine being blocked is reflected by $n = C + 2$. The second and third component s_i denote the number of outstanding replenishment orders of component i $(i = 1, 2)$. At the beginning of the process, there are S_i units of component i in stock, and there is also one additional unit in the corresponding machine. This gives at most $S_i + 1$

outstanding orders for each component. If there are $S_i + 1$ outstanding orders for component i, it implies that machine i is failed and not working. In all other cases, machine i is not failed. Hence, it is not necessary to model a failed machine separately with α_i as required in the situation with one critical component.

Furthermore, not all states are possible. It is easy to see that machine 1 cannot fail while it is blocked:

$$\overline{SS_1} = \{(C+2, S_1+1, s_2) | s_2 \in \{0, 1, \ldots, S_2+1\}\} \tag{186}$$

An analogous finding holds for machine 2, which cannot fail while it is starved:

$$\overline{SS_2} = \{(0, s_1, S_2+1) | s_1 \in \{0, 1, \ldots, S_1+1\}\} \tag{187}$$

Due to the system structure, the states with a starving machine 2 and a producing machine 1 are also impossible because a producing machine 1 would have increased the buffer.

$$\overline{SS_3} = \{(0, s_1, s_2) | s_1 \in \{0, \ldots, S_1\}, s_2 \in \{0, 1, \ldots, S_2+1\}\} \tag{188}$$

Conversely, a blocked machine 1 with a producing machine 2 is also impossible since machine 2 would have decreased the buffer.

$$\overline{SS_4} = \{(C+2, s_1, s_2) | s_1 \in \{0, \ldots, S_1+1\}, s_2 \in \{0, 1, \ldots, S_2\}\} \tag{189}$$

This results in the state space as

$$\begin{aligned} SS = &\{(n, s_1, s_2) | 0 \leq n \leq C+2, 0 \leq s_1 \leq S_1+1, 0 \leq s_2 \leq S_2+1\} \\ &\setminus \left(\overline{SS_1} \cup \overline{SS_2} \cup \overline{SS_3} \cup \overline{SS_4}\right). \end{aligned} \tag{190}$$

We determine the steady-state probabilities of the system and denote them as

$$\mathcal{P}_D(n, s_1, s_2). \tag{191}$$

The throughput of the system can be computed again by summing up all probabilities for states where machine 2 is working:

$$TP_D(C, S_1, S_2) = \sum_{n=1}^{C+2} \sum_{s_1=0}^{S_1+1} \sum_{s_2=0}^{S_2} \mathcal{P}_D(n, s_1, s_2) \tag{192}$$

The average number of spare parts in stock can be determined as

$$I_1(C, S_1, S_2) = \sum_{n=0}^{C+2} \sum_{s_1=0}^{S_1} \sum_{s_2=0}^{S_2+1} (S_1 - s_1)\mathcal{P}_D(n, s_1, s_2) \tag{193}$$

for the first machine and

$$I_2(C, S_1, S_2) = \sum_{n=0}^{C+2} \sum_{s_1=0}^{S_1+1} \sum_{s_2=0}^{S_2} (S_2 - s_2)\mathcal{P}_D(n, s_1, s_2) \tag{194}$$

for the second machine. The average number of workpieces in the system is given as

$$\begin{aligned} WIP(C, S_1, S_2) = 1 &+ \sum_{n=1}^{C+1} \sum_{s_1=0}^{S_1+1} \sum_{s_2=0}^{S_2+1} n\mathcal{P}_D(n, s_1, s_2) \\ &+ \sum_{s_1=0}^{S_1+1} \sum_{s_2=0}^{S_2+1} (C+1)\mathcal{P}_D(C+2, s_1, s_2). \end{aligned} \tag{195}$$

We extend the optimization problem from (178) and introduce the cost-value ratios c_i for a spare part for component i with respect to the cost for a workpiece in the system. Summing up, the problem to be solved is given as

$$\min_{C,S_1,S_2} \quad TC_D(C, S_1, S_2) = WIP(C, S_1, S_2) + c_1 I_1(C, S_1, S_2) + c_2 I_2(C, S_1, S_2) \tag{196}$$

$$\begin{aligned} \text{s.t.} \quad & TP_D(C, S_1, S_2) \geq TP^T \\ & 0 \leq C \leq C^{\max} \\ & 0 \leq S_1, S_2 \leq S^{\max} \\ & C, S_1, S_2 \in \mathbb{N}_0 \end{aligned}$$

If the base-stock level is chosen as $S_i = 0$, there are no spare parts in stock, and each broken component has to be repaired before it can be used again. Thus, each failure of a component implies a machine failure. In this case, the system is equivalent to the model analyzed by Gershwin (1994).

The balance equations are obtained in a similar way as before. They are presented in the appendix (Section 5.5).

5.3.2 Numerical results

The optimal system designs (C^*, S_1^*, S_2^*) were determined as a solution of the optimization problem (196), where a complete enumeration was conducted again with $C^{\max} = 100$ for the buffer capacity and $S^{\max} = 6$ for the base-stock levels. For different target values for the throughput and different cost-value ratios c_1, c_2 the results for a complete symmetric line are presented in Table 31.

Similar conclusions as in the case where both machines contain an identical component can be drawn: spare parts are held in stock to guarantee the target throughput, and the buffer is only installed for fine-tuning or used to increase the operational time of a machine and reduce the spare parts inventory level. The latter effect can mainly be observed for expensive spare parts, i.e., high values for c_1, c_2, because the small changes in the average spare parts inventory caused by increased buffer capacity lead to higher savings for expensive spare parts. The impact of buffer size on the average spare parts inventories is illustrated in Figure 15a.

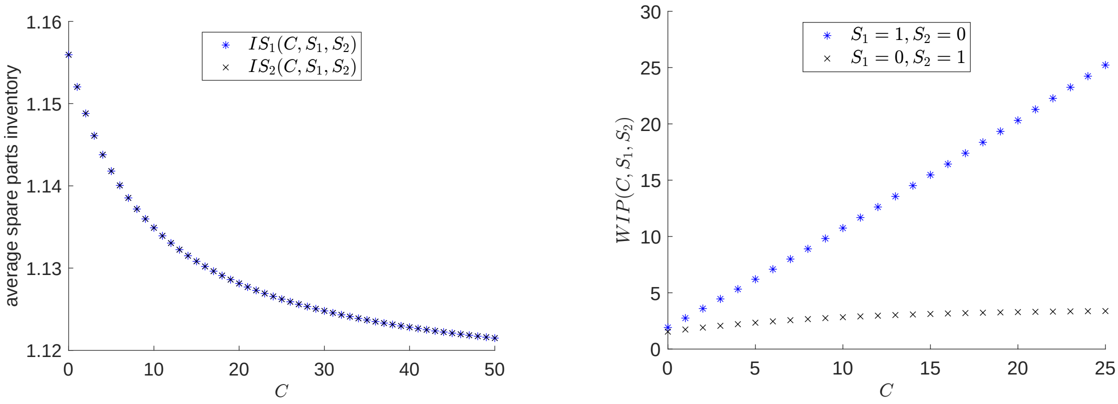
Furthermore, we can observe that the optimal base-stock level for component 2 is always greater than or equal to the one for component 1. To amplify this effect, Figure 15b depicts the average WIP as a function of the buffer size. As the buffer size increases, so does the average WIP; however, this effect is considerably more vital for a system with a higher base-stock level for component 1. The reason for this is the higher availability of machine 1, which leads to higher buffer usage because workpieces are pushed into the buffer at a higher rate than workpieces are pulled out of the buffer. Since a higher WIP results in higher costs, system designs with $S_1 > S_2$ are only reasonable in specific situations with noticeably distinct characteristics of the components.

In the next step, we investigate optimal system designs where the second machine is the bottleneck since the critical component in the machine is less reliable $(p_2 = 0.2)$. The numerical results are presented in Table 32, where the probability of failure of component 2 is doubled compared to component 1.

Table 31. Optimal system designs for symmetric lines

			$p_1 = 0.1000, p_2 = 0.1000$ $r_1 = 0.0600, r_2 = 0.0600$			$p_1 = 0.1000, p_2 = 0.1000$ $r_1 = 0.1000, r_2 = 0.1000$			$p_1 = 0.1000, p_2 = 0.1000$ $r_1 = 0.1400, r_2 = 0.1400$		
TP^T	c_1	c_2	(C^*, S_1^*, S_2^*)	$TP_D(\cdot)$	$TC_D(\cdot)$	(C^*, S_1^*, S_2^*)	$TP_D(\cdot)$	$TC_D(\cdot)$	(C^*, S_1^*, S_2^*)	$TP_D(\cdot)$	$TC_D(\cdot)$
	0.10	0.10	(0, 1, 2)	0.6121	1.83	(0, 1, 1)	0.6906	1.94	(0, 1, 1)	0.7891	2.00
	1.00	1.00	(0, 1, 2)	0.6121	3.05	(0, 1, 1)	0.6906	2.78	(0, 1, 1)	0.7891	2.98
0.60	10.00	10.00	(0, 1, 2)	0.6121	15.20	(0, 1, 1)	0.6906	11.16	(0, 1, 1)	0.7891	12.75
	50.00	50.00	(27, 1, 1)	0.6007	45.10	(1, 1, 1)	0.6990	48.41	(0, 1, 1)	0.7891	56.18
	100.00	100.00	(27, 1, 1)	0.6007	74.80	(7, 1, 1)	0.7311	93.32	(4, 1, 1)	0.8163	109.77
	0.10	0.10	(0, 1, 4)	0.6568	1.98	(0, 1, 1)	0.6906	1.94	(0, 1, 1)	0.7891	2.00
	1.00	1.00	(0, 2, 2)	0.7516	3.62	(0, 1, 1)	0.6906	2.78	(0, 1, 1)	0.7891	2.98
0.65	10.00	10.00	(18, 1, 2)	0.6507	17.55	(0, 1, 1)	0.6906	11.16	(0, 1, 1)	0.7891	12.75
	50.00	50.00	(18, 1, 2)	0.6507	68.05	(1, 1, 1)	0.6990	48.41	(0, 1, 1)	0.7891	56.18
	100.00	100.00	(35, 1, 2)	0.6578	130.78	(7, 1, 1)	0.7311	93.32	(4, 1, 1)	0.8163	109.77
	0.10	0.10	(0, 2, 2)	0.7516	2.05	(0, 1, 2)	0.7759	1.98	(0, 1, 1)	0.7891	2.00
	1.00	1.00	(0, 2, 2)	0.7516	3.62	(0, 1, 2)	0.7759	3.49	(0, 1, 1)	0.7891	2.98
0.70	10.00	10.00	(0, 2, 2)	0.7516	19.32	(2, 1, 1)	0.7062	11.99	(0, 1, 1)	0.7891	12.75
	50.00	50.00	(3, 2, 2)	0.7652	88.83	(2, 1, 1)	0.7062	48.46	(0, 1, 1)	0.7891	56.18
	100.00	100.00	(12, 2, 2)	0.7893	172.30	(7, 1, 1)	0.7311	93.32	(4, 1, 1)	0.8163	109.77
	0.10	0.10	(0, 2, 2)	0.7516	2.05	(0, 1, 2)	0.7759	1.98	(0, 1, 1)	0.7891	2.00
	1.00	1.00	(0, 2, 2)	0.7516	3.62	(0, 1, 2)	0.7759	3.49	(0, 1, 1)	0.7891	2.98
0.75	10.00	10.00	(0, 2, 2)	0.7516	19.32	(14, 1, 1)	0.7503	17.47	(0, 1, 1)	0.7891	12.75
	50.00	50.00	(3, 2, 2)	0.7652	88.83	(14, 1, 1)	0.7503	51.62	(0, 1, 1)	0.7891	56.18
	100.00	100.00	(12, 2, 2)	0.7893	172.30	(14, 1, 1)	0.7503	94.32	(4, 1, 1)	0.8163	109.77
	0.10	0.10	(0, 2, 3)	0.8142	2.10	(1, 1, 3)	0.8003	2.13	(0, 1, 2)	0.8594	2.07
	1.00	1.00	(0, 2, 3)	0.8142	4.33	(0, 2, 2)	0.8962	4.26	(0, 1, 2)	0.8594	3.79
0.80	10.00	10.00	(0, 2, 3)	0.8142	26.58	(12, 1, 2)	0.8004	19.86	(2, 1, 1)	0.8052	13.61
	50.00	50.00	(20, 2, 2)	0.8011	92.54	(12, 1, 2)	0.8004	85.30	(2, 1, 1)	0.8052	56.39
	100.00	100.00	(20, 2, 2)	0.8011	173.13	(16, 1, 2)	0.8022	167.07	(4, 1, 1)	0.8163	109.77
	0.10	0.10	(0, 3, 3)	0.8970	2.26	(0, 2, 2)	0.8962	2.18	(0, 1, 2)	0.8594	2.07
	1.00	1.00	(0, 3, 3)	0.8970	5.06	(0, 2, 2)	0.8962	4.26	(0, 1, 2)	0.8594	3.79
0.85	10.00	10.00	(0, 3, 3)	0.8970	33.08	(0, 2, 2)	0.8962	25.07	(0, 1, 2)	0.8594	21.01
	50.00	50.00	(0, 3, 3)	0.8970	157.62	(0, 2, 2)	0.8962	117.54	(21, 1, 1)	0.8502	63.66
	100.00	100.00	(6, 3, 3)	0.9107	312.12	(3, 2, 2)	0.9074	232.68	(21, 1, 1)	0.8502	114.86
	0.10	0.10	(0, 3, 4)	0.9292	2.34	(0, 2, 3)	0.9328	2.26	(0, 2, 2)	0.9510	2.24
	1.00	1.00	(1, 3, 3)	0.9000	5.56	(1, 2, 2)	0.9006	4.76	(0, 2, 2)	0.9510	4.67
0.90	10.00	10.00	(1, 3, 3)	0.9000	33.49	(1, 2, 2)	0.9006	25.49	(0, 2, 2)	0.9510	28.88
	50.00	50.00	(1, 3, 3)	0.9000	157.66	(1, 2, 2)	0.9006	117.66	(0, 2, 2)	0.9510	136.50
	100.00	100.00	(6, 3, 3)	0.9107	312.12	(3, 2, 2)	0.9074	232.68	(0, 2, 2)	0.9510	271.02

It is not surprising that a comparison of the results in Tables 31 and 32 leads to the conclusion that more spare parts are often needed for component 2 or larger buffer capacity is required or both together if the probability of failure is increasing. However, in the case of long replenishment lead times and cheap spare parts, we can also observe a higher base-stock level for the first component (see, for example, the instance with $TP^T = 0.80, c_1 = c_2 = 1.0$), which may be necessary to avoid starving of the second machine. If an additional spare part for component 2 not only compensates for the higher failure probability but also increases machine availability, it pulls workpieces out of the buffer faster than machine 1 is supplying units. Thus, it is beneficial to avoid starvation and increase the availability of machine 1 by adding one spare part.



(a) Impact of buffer size on both spare part inventories

(b) Impact of buffer size on WIP depending on spare part position

Figure 15. Impact of buffer size on inventories; $r_1 = r_2 = 0.1, p_1 = p_2 = 0.1, S_1 = S_2 = 2$

Furthermore, in some instances we observe the same base-stock levels in Tables 31 and 32 with a reduced buffer capacity (e.g., for $r_1 = r_2 = 0.14$ and $TP^T = 0.60, 0.65, 0.70$ and $c_1 = c_2 = 100$). This counterintuitive effect is again driven by the fact that a decreased buffer size leads to less operational time and a reduced number of failures. Therefore, even with an increasing failure probability, the spare parts inventory levels do not have to be increased.

An increasing failure probability of the first component is usually combined with an increased buffer size, a higher base-stock level for component 1, or a combination of both, as can be seen in Table 33 where the results for the more unreliable component 1 are presented. A comparison of the results in Tables 31 and 33 also implies that for low values of c_1 and c_2 it may be optimal to raise the base-stock level not only for component 1 but also for component 2. This action may be necessary because a larger spare parts inventory level for component 1 can increase the first machine's availability despite the higher failure probability. Thus, the buffer is filled more often, and starvation of the second machine is reduced, which means that the second machine can fail more often. To compensate for the increasing number of failures, keeping more spare parts in stock may be beneficial.

In addition to previous research, we conclude that not only buffer capacity can be used if the line efficiency is low. Installing buffers and stocking spare parts are possible ways to increase the efficiency and, thus, the throughput of a transfer line. For unreliable transfer lines, it is always a good decision to use at least some spare parts, especially for

Table 32. Optimal system designs for an asymmetric line with a more unreliable component 2

			$p_1 = 0.1000, p_2 = 0.2000$ $r_1 = 0.0600, r_2 = 0.0600$			$p_1 = 0.1000, p_2 = 0.2000$ $r_1 = 0.1000, r_2 = 0.1000$			$p_1 = 0.1000, p_2 = 0.2000$ $r_1 = 0.1400, r_2 = 0.1400$		
TP^T	c_1	c_2	(C^*, S_1^*, S_2^*)	$TP_D(\cdot)$	$TC_D(\cdot)$	(C^*, S_1^*, S_2^*)	$TP_D(\cdot)$	$TC_D(\cdot)$	(C^*, S_1^*, S_2^*)	$TP_D(\cdot)$	$TC_D(\cdot)$
	0.10	0.10	(0, 1, 4)	0.6264	1.91	(0, 1, 2)	0.6920	1.97	(0, 1, 1)	0.6625	2.02
	1.00	1.00	(0, 2, 3)	0.6994	3.70	(0, 1, 2)	0.6920	3.08	(0, 1, 1)	0.6625	2.85
0.60	10.00	10.00	(3, 1, 3)	0.6006	17.23	(0, 1, 2)	0.6920	14.21	(0, 1, 1)	0.6625	11.15
	50.00	50.00	(48, 1, 2)	0.6004	64.30	(19, 1, 1)	0.6001	54.29	(0, 1, 1)	0.6625	48.04
	100.00	100.00	(48, 1, 2)	0.6004	97.05	(19, 1, 1)	0.6001	90.25	(2, 1, 1)	0.6784	93.61
	0.10	0.10	(0, 2, 3)	0.6994	2.08	(0, 1, 2)	0.6920	1.97	(0, 1, 1)	0.6625	2.02
	1.00	1.00	(0, 2, 3)	0.6994	3.70	(0, 1, 2)	0.6920	3.08	(0, 1, 1)	0.6625	2.85
0.65	10.00	10.00	(0, 2, 3)	0.6994	19.96	(0, 1, 2)	0.6920	14.21	(0, 1, 1)	0.6625	11.15
	50.00	50.00	(32, 1, 3)	0.6500	72.85	(5, 1, 2)	0.7251	62.70	(0, 1, 1)	0.6625	48.04
	100.00	100.00	(40, 1, 3)	0.6535	135.50	(12, 1, 2)	0.7477	119.52	(2, 1, 1)	0.6784	93.61
	0.10	0.10	(0, 2, 4)	0.7725	2.11	(0, 1, 3)	0.7631	2.02	(0, 1, 2)	0.8006	2.04
	1.00	1.00	(0, 2, 4)	0.7725	4.24	(1, 1, 2)	0.7007	3.61	(0, 1, 2)	0.8006	3.38
0.70	10.00	10.00	(1, 2, 3)	0.7047	20.41	(1, 1, 2)	0.7007	14.56	(0, 1, 2)	0.8006	16.75
	50.00	50.00	(5, 2, 3)	0.7206	91.29	(5, 1, 2)	0.7251	62.70	(9, 1, 1)	0.7015	52.89
	100.00	100.00	(12, 2, 3)	0.7373	175.45	(12, 1, 2)	0.7477	119.52	(9, 1, 1)	0.7015	96.38
	0.10	0.10	(0, 2, 4)	0.7725	2.11	(0, 1, 3)	0.7631	2.02	(0, 1, 2)	0.8006	2.04
	1.00	1.00	(0, 2, 4)	0.7725	4.24	(0, 1, 3)	0.7631	3.77	(0, 1, 2)	0.8006	3.38
0.75	10.00	10.00	(0, 2, 4)	0.7725	25.53	(14, 1, 2)	0.7519	20.34	(0, 1, 2)	0.8006	16.75
	50.00	50.00	(23, 2, 3)	0.7505	98.40	(14, 1, 2)	0.7519	64.47	(3, 1, 2)	0.8229	75.81
	100.00	100.00	(23, 2, 3)	0.7505	178.32	(14, 1, 2)	0.7519	119.64	(7, 1, 2)	0.8386	147.46
	0.10	0.10	(0, 2, 5)	0.8134	2.17	(0, 1, 5)	0.8013	2.19	(0, 1, 2)	0.8006	2.04
	1.00	1.00	(0, 3, 4)	0.8382	4.93	(0, 2, 3)	0.8744	4.45	(0, 1, 2)	0.8006	3.38
0.80	10.00	10.00	(8, 2, 4)	0.8009	28.01	(16, 1, 3)	0.8002	23.42	(0, 1, 2)	0.8006	16.75
	50.00	50.00	(10, 2, 4)	0.8052	117.69	(16, 1, 3)	0.8002	97.75	(3, 1, 2)	0.8229	75.81
	100.00	100.00	(22, 2, 4)	0.8211	226.92	(24, 1, 3)	0.8029	190.55	(7, 1, 2)	0.8386	147.46
	0.10	0.10	(0, 3, 5)	0.8940	2.31	(0, 2, 3)	0.8744	2.20	(0, 1, 3)	0.8553	2.11
	1.00	1.00	(0, 3, 5)	0.8940	5.59	(0, 2, 3)	0.8744	4.45	(0, 1, 3)	0.8553	4.19
0.85	10.00	10.00	(5, 3, 4)	0.8516	34.82	(0, 2, 3)	0.8744	26.98	(13, 1, 2)	0.8507	22.05
	50.00	50.00	(5, 3, 4)	0.8516	151.27	(1, 2, 3)	0.8795	127.06	(13, 1, 2)	0.8507	78.07
	100.00	100.00	(6, 3, 4)	0.8533	296.80	(5, 2, 3)	0.8925	250.62	(13, 1, 2)	0.8507	148.10
	0.10	0.10	(0, 4, 5)	0.9249	2.43	(0, 2, 4)	0.9203	2.28	(0, 2, 3)	0.9444	2.28
	1.00	1.00	(0, 4, 5)	0.9249	6.44	(0, 2, 4)	0.9203	5.26	(0, 2, 3)	0.9444	5.01
0.90	10.00	10.00	(2, 3, 5)	0.9003	39.15	(10, 2, 3)	0.9009	32.82	(0, 2, 3)	0.9444	32.30
	50.00	50.00	(3, 3, 5)	0.9028	183.40	(10, 2, 3)	0.9009	130.10	(0, 2, 3)	0.9444	153.59
	100.00	100.00	(10, 3, 5)	0.9141	361.70	(10, 2, 3)	0.9009	251.70	(1, 2, 3)	0.9482	305.04

the second machine. Beyond that, the decision highly depends on costs. In situations with different machine characteristics, the provisioning of spare parts influences the best decisions. Additional spare parts can be used to cover differences in breakdown rates directly. Thus, the effectiveness of the buffer capacity is not necessarily smaller in systems with different breakdown rates.

These effects illustrate the interdependency of spare part stocks and buffers and explain why a separate optimization of base-stock levels and buffer capacity may not lead to the optimal system design.

Table 33. Optimal system designs for an asymmetric line with a more unreliable component 1

			$p_1 = 0.2000, p_2 = 0.1000$ $r_1 = 0.0600, r_2 = 0.0600$			$p_1 = 0.2000, p_2 = 0.1000$ $r_1 = 0.1000, r_2 = 0.1000$			$p_1 = 0.2000, p_2 = 0.1000$ $r_1 = 0.1400, r_2 = 0.1400$		
TP^T	c_1	c_2	(C^*, S_1^*, S_2^*)	$TP_D(\cdot)$	$TC_D(\cdot)$	(C^*, S_1^*, S_2^*)	$TP_D(\cdot)$	$TC_D(\cdot)$	(C^*, S_1^*, S_2^*)	$TP_D(\cdot)$	$TC_D(\cdot)$
	0.10	0.10	(0, 2, 3)	0.6112	1.87	(0, 1, 3)	0.6027	1.87	(0, 1, 1)	0.6625	1.84
	1.00	1.00	(0, 3, 2)	0.6994	3.61	(0, 2, 1)	0.6920	3.08	(0, 1, 1)	0.6625	2.67
0.60	10.00	10.00	(5, 2, 2)	0.6011	16.88	(19, 1, 1)	0.6001	11.58	(0, 1, 1)	0.6625	10.97
	50.00	50.00	(48, 2, 1)	0.6004	53.05	(19, 1, 1)	0.6001	40.35	(8, 1, 1)	0.6998	46.86
	100.00	100.00	(48, 2, 1)	0.6004	85.80	(31, 1, 1)	0.6034	76.20	(20, 1, 1)	0.7094	90.04
	0.10	0.10	(0, 3, 2)	0.6994	1.99	(0, 2, 1)	0.6920	1.97	(0, 1, 1)	0.6625	1.84
	1.00	1.00	(0, 3, 2)	0.6994	3.61	(0, 2, 1)	0.6920	3.08	(0, 1, 1)	0.6625	2.67
0.65	10.00	10.00	(0, 3, 2)	0.6994	19.87	(0, 2, 1)	0.6920	14.21	(0, 1, 1)	0.6625	10.97
	50.00	50.00	(16, 3, 2)	0.7433	88.10	(6, 2, 1)	0.7294	62.40	(8, 1, 1)	0.6998	46.86
	100.00	100.00	(32, 3, 1)	0.6500	151.30	(13, 2, 1)	0.7499	118.87	(20, 1, 1)	0.7094	90.04
	0.10	0.10	(0, 3, 3)	0.7456	2.03	(0, 2, 2)	0.7729	2.00	(0, 1, 2)	0.7033	1.90
	1.00	1.00	(1, 3, 2)	0.7047	3.94	(1, 2, 1)	0.7007	3.54	(0, 2, 1)	0.8006	3.39
0.70	10.00	10.00	(1, 3, 2)	0.7047	20.01	(1, 2, 1)	0.7007	14.50	(9, 1, 1)	0.7015	12.07
	50.00	50.00	(16, 3, 2)	0.7433	88.10	(6, 2, 1)	0.7294	62.40	(9, 1, 1)	0.7015	46.86
	100.00	100.00	(35, 3, 2)	0.7574	167.68	(13, 2, 1)	0.7499	118.87	(20, 1, 1)	0.7094	90.04
	0.10	0.10	(0, 3, 4)	0.7604	2.11	(0, 2, 2)	0.7729	2.00	(0, 2, 1)	0.8006	2.06
	1.00	1.00	(0, 4, 2)	0.7725	4.27	(0, 2, 2)	0.7729	3.72	(0, 2, 1)	0.8006	3.39
0.75	10.00	10.00	(23, 3, 2)	0.7505	24.37	(14, 2, 1)	0.7519	19.59	(0, 2, 1)	0.8006	16.77
	50.00	50.00	(23, 3, 2)	0.7505	88.30	(14, 2, 1)	0.7519	63.73	(2, 2, 1)	0.8170	75.84
	100.00	100.00	(35, 3, 2)	0.7574	167.68	(14, 2, 1)	0.7519	118.90	(6, 2, 1)	0.8355	147.70
	0.10	0.10	(0, 4, 3)	0.8382	2.18	(0, 3, 2)	0.8744	2.17	(0, 2, 1)	0.8006	2.06
	1.00	1.00	(0, 4, 3)	0.8382	4.85	(0, 3, 2)	0.8744	4.43	(0, 2, 1)	0.8006	3.39
0.80	10.00	10.00	(8, 4, 2)	0.8009	28.71	(0, 3, 2)	0.8744	26.95	(0, 2, 1)	0.8006	16.77
	50.00	50.00	(8, 4, 2)	0.8009	118.46	(16, 3, 1)	0.8002	107.94	(2, 2, 1)	0.8170	75.84
	100.00	100.00	(15, 4, 2)	0.8134	229.25	(16, 3, 1)	0.8002	200.86	(6, 2, 1)	0.8355	147.70
	0.10	0.10	(0, 4, 4)	0.8617	2.26	(0, 3, 2)	0.8744	2.17	(0, 2, 2)	0.8714	2.12
	1.00	1.00	(0, 5, 3)	0.8940	5.58	(0, 3, 2)	0.8744	4.43	(0, 2, 2)	0.8714	4.15
0.85	10.00	10.00	(5, 4, 3)	0.8516	32.28	(0, 3, 2)	0.8744	26.95	(13, 2, 1)	0.8507	22.89
	50.00	50.00	(13, 4, 3)	0.8615	148.13	(4, 3, 2)	0.8900	126.46	(13, 2, 1)	0.8507	78.91
	100.00	100.00	(31, 4, 3)	0.8684	290.44	(10, 3, 2)	0.9009	248.65	(13, 2, 1)	0.8507	148.94
	0.10	0.10	(0, 5, 4)	0.9249	2.38	(0, 3, 3)	0.9069	2.25	(0, 3, 2)	0.9444	2.27
	1.00	1.00	(0, 5, 4)	0.9249	6.39	(0, 3, 3)	0.9069	5.29	(0, 3, 2)	0.9444	5.00
0.90	10.00	10.00	(2, 5, 3)	0.9003	38.90	(10, 3, 2)	0.9009	29.76	(0, 3, 2)	0.9444	32.29
	50.00	50.00	(5, 5, 3)	0.9069	182.98	(10, 3, 2)	0.9009	127.04	(0, 3, 2)	0.9444	153.58
	100.00	100.00	(12, 5, 3)	0.9162	360.61	(10, 3, 2)	0.9009	248.65	(3, 3, 2)	0.9531	304.65

In order to point out that the lead-time assumptions do not reduce the transferability of these results, a detailed analysis of the lead-time distribution influences is carried out.

5.3.3 Influence of the lead-time distribution

We conduct a simulation study to illustrate the influence of the lead-time distribution. In each case, we simulate one million time units and as many runs as necessary such that the half-width of the two-sided 99% confidence intervals of all four performance measures gets below 1% of the simulated values, i.e., if a throughput of 0.8 is achieved by the simulation the half-width has to be smaller than 0.008.

With the use of the exact Markov results, we validate the simulation. The simulated and exact results for geometric lead-time distributions, i.e., completely identical assumptions, only differ in the magnitude of stochastic influences. Hence we can assume the simulated values to be reliable for all analyzed lead-time distributions.

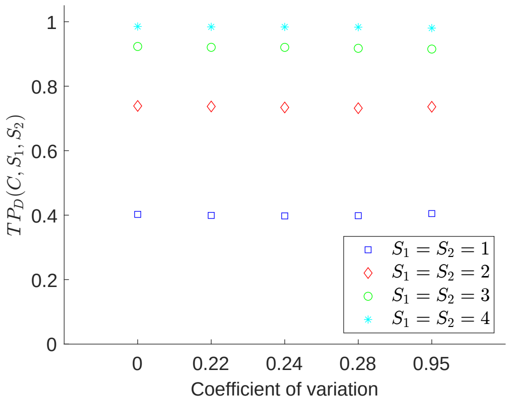
In the following, simulated results for the four main performance measures, which are average throughput, average WIP, and average spare part inventories for each of the two components, are presented. We simulated five different lead-time distributions: constant as well as binomial distributed, and geometric distributed lead times. They all have the same mean value but different variances, which are given in Table 34. Note that the expected value is taken as constant with a value of 10. For reasons of clarity and comprehensibility, we focus on these distributions. However, the results are very similar for different expected values and their corresponding distribution parameters.

Table 34. Different lead-time distributions

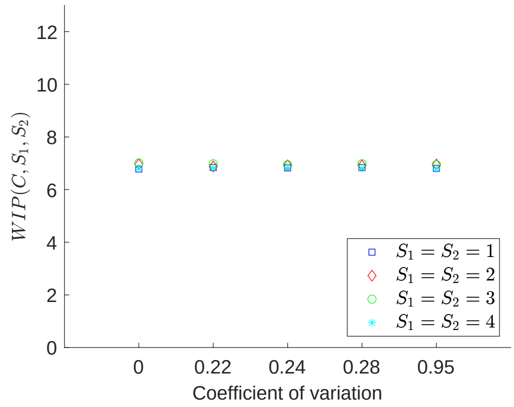
ID	Distribution	Expected value	Variance	Coefficient of variation
1	Constant	10	0	0.00
2	Binomial	10	5	0.22
3	Binomial	10	6	0.25
4	Binomial	10	8	0.28
5	Geometric	10	90	0.95

The graphs in Figure 16 and the boxplots in Figure 17 illustrate the simulation results for the performance measures for different given system designs and different distributions of the replenishment lead-times. The results are similar both on the individual and on the aggregated level. The data clearly shows that the variance of the lead-time distribution does not influence the performance measures. Thus, only the mean of the distribution is relevant. Hence, using geometrically distributed lead times to model our system as a Markov chain does not restrict the informative value of the presented results.

Because the results do not show any systematic differences, even for a relatively small number of spare parts and buffer capacity, we must analyze the situation of the boundary

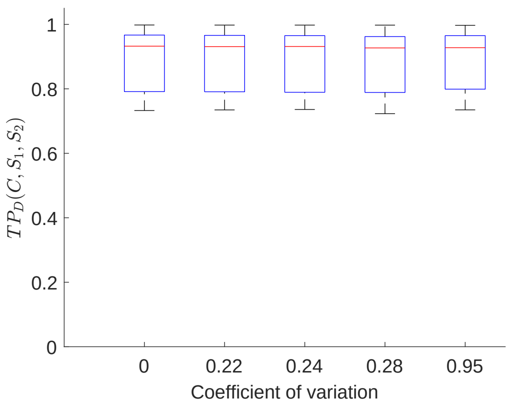


(a) Impact of variance on throughput

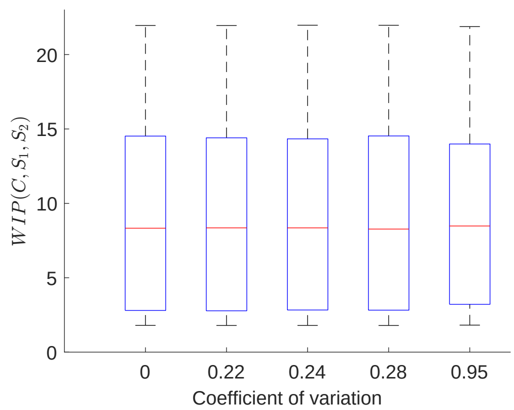


(b) Impact of variance on WIP

Figure 16. Impact of variance for different base-stock levels; $p_1 = p_2 = r_1 = r_2 = 0.1$ and $C = 10$



(a) Impact of variance on throughput



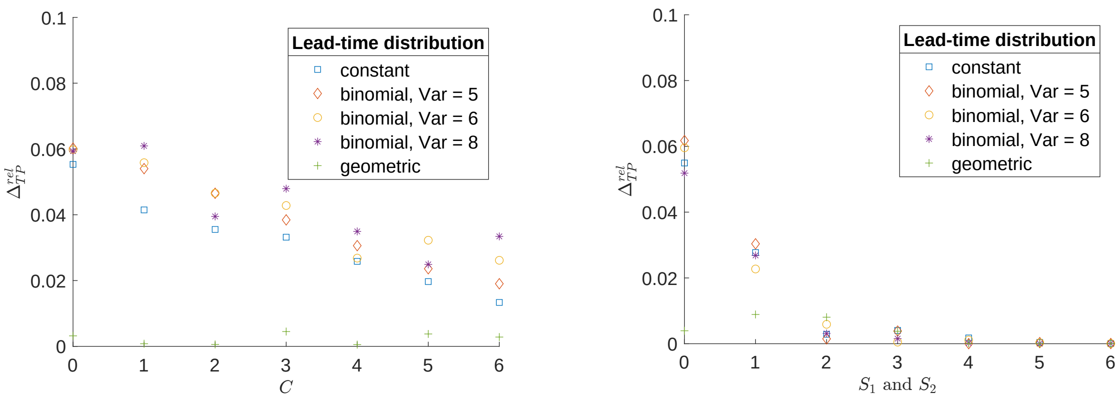
(b) Impact of variance on WIP

Figure 17. Impact of variance for different buffer capacities and base-stock levels; boxplots of 176 instances with $p_1 = p_2 = r_1 = r_2 = 0.1$ and for each combination of $C \in \{10, 11, \ldots, 20\}, S_1, S_2 \in \{1, 2, 3, 4\}$

states. Hence, a system without buffers and spare parts is a good starting point to distinguish between the positive effects of buffer capacity and spare parts. Therefore, we consider both: on the one hand, a system without spare parts and, on the other hand, a system without buffers. Figure 18 depicts the relative throughput deviation between the exact Markov results with geometrically distributed lead times and the simulated results for the different lead-time distributions, which is calculated as

$$\Delta_{TP}^{rel} = \frac{|TP_{\text{sim}} - TP_{\text{exact}}|}{TP_{\text{exact}}} \tag{197}$$

The simulated values get closer to the exact values for increasing buffer levels (without any spare parts) as well as increasing base-stock levels (without any buffers). These results are in line with earlier research. Tan (1997) and Tan (1998) analysed a flow-line with continuous flow of material, zero buffers and time-dependent failures with different repair time distributions. They found that the repair time variance has no influence on so-called first-order measures such as throughput or WIP. These results were also obtained for transfer lines with geometric failure and repair time distributions assuming time-dependent failures and no inter-station buffers (Tan, 1999). The previous findings confirm the Palm–Khintchine theorem, which says that the superposition of a large number of renewal processes, which are not necessarily needed to be Poisson processes, will behave Poissonian. In the case of time-dependent failures and exponentially distributed lead times, this is exactly true as even the single processes are Poissonian. The introduction of operation-dependent failures results in small deviations, which are shown in Figure 18. Decreasing the probabilities of starving, blocking, and downtimes through increasing buffer capacity or base-stock levels directly yields better fitting results as the operation-dependent failures approach time-dependent failures. It is apparent that spare parts have a much higher effect on the deviation, which is caused by different lead-time distributions. Thus, we can conclude that the provisioning of spare parts additionally absorbs variability and flattens the effect of lead-time variability also in case of operation-dependent failures to zero.



(a) Without spare parts; $S_1 = S_2 = 0$

(b) Without buffer space; $C = 0$

Figure 18. Relative throughput deviation; $p_1 = p_2 = r_1 = r_2 = 0.1$

All in all, the lead-time distribution only affects situations where the system performs very poorly because of missing buffer space and no available spare parts. In systems of practical relevance, there is no noticeable difference between the different lead-time distributions.

5.3.4 Managerial insights

In the last part of our discussion, we compare the results of Tables 30 and 31 in order to investigate the impact of component part standardization. In addition to the well-known effects on the unit production cost due to larger lot sizes or lower purchase material prices, we can observe a positive effect on the system designs.

The most apparent effect is that the number of spare parts held in stock is reduced in the situation with only one type of component compared to systems with two different types of critical components. In 82% of the instances, the optimal number of spare parts in stock diminishes. However, there are no instances with a higher number of spare parts. This means that inventory pooling of spare parts positively impacts the optimal system design.

Furthermore, less buffer capacity is needed in a system with one critical component type when spare parts are relatively expensive. In over 90% of the instances, the buffer capacity is less than or equal compared to the optimal design with two different types of components. We can conclude that there is also a pooling effect between the spare parts inventory and the buffer.

Due to the pooling effects, the costs of the optimal system designs in around 97% of the instances are much lower if both machines contain the identical critical component. Thus, our numerical results show the positive effect of component part standardization on the life-cycle costs of flow lines. Therefore, the planner of a manufacturing system should not only determine the locations and capacities of buffers but also consider component commonality and spare parts stocks in order to increase flow line efficiency and decrease costs.

5.4 Summary

In this chapter, we formulate a new problem where the designer of a flow line has to determine the required buffer capacity and spare part stock levels. The situations with one or two different critical components are modeled as a discrete-time Markov chain in order to obtain the relevant performance measures. Optimal system designs are computed, which minimize the average cost for WIP and stocking spare parts while meeting the required target throughput level.

We use numerical experiments to analyze optimal system designs and show that buffers are often not needed, especially when the holding costs for spare parts are lower than those for WIP. The reason for this is the different impact of one spare part or one unit of buffer capacity on the throughput. One additional spare part can lead to a much higher increase in throughput than an additional unit of buffer capacity, which explains why the optimal buffer capacity is equal to zero in 78% of our instances. Thus, we can conclude from our analysis that the optimization of buffer capacity and spare parts stock levels has to be done simultaneously. In a more general sense, this means that the potential of after-sales service should not be neglected when designing a manufacturing system.

Another finding is related to spare parts planning. A larger buffer leads to longer operation times due to less starving and blocking, and thus to more failures of components. This can result in lower spare parts inventory levels such that an increase in buffer capacity can lead to lower holding costs. On the other hand, it may also be necessary to increase the base-stock level in order to guarantee the required target level.

Further insights were gained regarding the situation with two different critical components. Even if both parts have the same stochastic characteristics, different stock levels may be required for an optimal system design. In such situations, the higher stock level is consistently implemented for the critical component of the second machine. This is justified because the higher stock level leads to higher availability of the second machine, such that the workpieces are pulled out of the buffer faster, which positively affects WIP. A higher spare part stock level for the first machine will only push more workpieces into the buffer and increase WIP, which clearly shows that an optimal system design requires an adequate balance of the spare parts stock levels.

Finally, the simultaneous optimization of spare parts base-stock levels and buffer capacity also enables the utilization of pooling effects between spare parts stock and buffers and helps to reduce investments. Pooling effects are the main reason component part standardization leads to significant cost reductions.

5.5 Appendix

5.5.1 Balance equations for Section 5.2

A state where both machines are down can only be reached when there are already at least S outstanding replenishment orders:

$$\begin{aligned}\mathcal{P}_D(C+1,S+2,0,0) = {} & (1-r)^{S+2}\cdot\mathcal{P}_D(C+1,S+2,0,0)\\ & + (1-r)^{S+1}p\cdot\mathcal{P}_D(C+1,S+1,1,0)\\ & + (1-r)^{S}p^2\cdot\mathcal{P}_D(C+1,S,1,1)\end{aligned}\tag{198}$$

For $2 \le n \le C$, we get:

$$\begin{aligned}\mathcal{P}_D(n,S+2,0,0) = {} & (1-r)^{S+2}\cdot\mathcal{P}_D(n,S+2,0,0)\\ & + (1-r)^{S+1}p\cdot\mathcal{P}_D(n,S+1,1,0)\\ & + (1-r)^{S+1}p\cdot\mathcal{P}_D(n,S+1,0,1)\\ & + (1-r)^{S}p^2\cdot\mathcal{P}_D(n,S,1,1)\end{aligned}\tag{199}$$

and for the lower boundary condition, we obtain:

$$\begin{aligned}\mathcal{P}_D(1,S+2,0,0) = {} & (1-r)^{S+2}\cdot\mathcal{P}_D(1,S+2,0,0)\\ & + (1-r)^{S+1}p\cdot\mathcal{P}_D(1,S+1,0,1)\\ & + (1-r)^{S}p^2\cdot\mathcal{P}_D(1,S,1,1).\end{aligned}\tag{200}$$

Machine 1 can only get blocked if the buffer is already full:

$$\begin{aligned}\mathcal{P}_D(C+2,S+1,1,0) = {} & (1-r)^{S+1}\cdot\mathcal{P}_D(C+2,S+1,1,0)\\ & + (1-r)^{S+1}(1-p)\cdot\mathcal{P}_D(C+1,S+1,1,0)\\ & + (1-r)^{S}p(1-p)\cdot\mathcal{P}_D(C+1,S,1,1)\end{aligned}\tag{201}$$

For the internal states where the second machine is down $(3 \le n \le C+1)$, we get:

$$\begin{aligned}\mathcal{P}_D(n,S+1,1,0) = {} & (1-r)^{S+1}(1-p)\cdot\mathcal{P}_D(n-1,S+1,1,0)\\ & + (1-r)^{S}p(1-p)\cdot\mathcal{P}_D(n-1,S,1,1)\end{aligned}\tag{202}$$

and for the lower boundary state, we obtain:

$$\mathcal{P}_D(2, S+1, 1, 0) = (1-r)^S p(1-p) \cdot \mathcal{P}_D(1, S, 1, 1). \tag{203}$$

If the first machine is down, we get the following for the upper boundary state:

$$\begin{aligned}\mathcal{P}_D(C, S+1, 0, 1) = {} & (1-r)^{S+1} r(S+2) \cdot \mathcal{P}_D(C+1, S+2, 0, 0) \\ & + (1-r)^S r(S+1)p \cdot \mathcal{P}_D(C+1, S+1, 1, 0) \\ & + \Big((1-r)^S p(1-p) + r(1-r)^{S-1} S p^2\Big) \cdot \mathcal{P}_D(C+1, S, 1, 1) \\ & + p^2(1-r)^{S-1} \cdot \mathcal{P}_D(C+1, S-1, 1, 1). \end{aligned} \tag{204}$$

For the internal states with $1 \leq n \leq C-1$, we get:

$$\begin{aligned}\mathcal{P}_D(n, S+1, 0, 1) = {} & (1-r)^{S+1} r(S+2) \cdot \mathcal{P}_D(n+1, S+2, 0, 0) \\ & + (1-r)^S r(S+1)p \cdot \mathcal{P}_D(n+1, S+1, 1, 0) \\ & + \Big((1-r)^{S+1}(1-p) + (S+1)r(1-r)^S p\Big) \\ & \quad \cdot \mathcal{P}_D(n+1, S+1, 0, 1) \\ & + \Big((1-r)^S p(1-p) + p^2 r(1-r)^{S-1} S\Big) \cdot \mathcal{P}_D(n+1, S, 1, 1) \\ & + (1-r)^{S-1} p^2 P(n+1, S-1, 1, 1). \end{aligned} \tag{205}$$

The equation for the lower boundary state where machine 2 is starving is given as:

$$\begin{aligned}\mathcal{P}_D(0, S+1, 0, 1) = {} & (1-r)^{S+1} r(S+2) \cdot \mathcal{P}_D(1, S+2, 0, 0) \\ & + (1-r)^{S+1} \cdot \mathcal{P}_D(0, S+1, 0, 1) \\ & + \Big((1-r)^{S+1}(1-p) + p(S+1)r(1-r)^S\Big) \cdot \mathcal{P}_D(1, S+1, 0, 1) \\ & + \Big((1-r)^S p(1-p) + p^2 r(1-r)^{S-1} S\Big) \cdot \mathcal{P}_D(1, S, 1, 1) \\ & + p^2(1-r)^{S-1} \cdot \mathcal{P}_D(1, S-1, 1, 1). \end{aligned} \tag{206}$$

If both machines are working, we must distinguish between the situation where more than one outstanding spare part order exists and the situation with fewer outstanding spare part orders. For $2 \leq s \leq S$, we get the following for a full buffer:

$$
\begin{aligned}
&\mathcal{P}_D(C+1,s,1,1) = \\
&\quad p^2(1-r)^{s-2} \cdot \mathcal{P}_D(C+1,s-2,1,1) \\
&\quad + \left(2p(1-p)(1-r)^{s-1} + p^2(s-1)r(1-r)^{s-2}\right) \cdot \mathcal{P}_D(C+1,s-1,1,1) \\
&\quad + \sum_{i=s}^{S} \left((1-p)^2 \binom{i}{s} r^{i-s}(1-r)^s + 2p(1-p)r^{i-s+1}(1-r)^{s-1}\binom{i}{s-1} \right) \\
&\qquad \cdot \mathcal{P}_D(C+1,i,1,1) \\
&\quad + \sum_{i=s}^{S} \left(p^2 r^{i-s+2}(1-r)^{s-2}\binom{i}{s-2} \right) \cdot \mathcal{P}_D(C+1,i,1,1) \\
&\quad + \left((1-p)\binom{S+1}{s} r^{S+1-s}(1-r)^s + p(1-r)^{s-1} r^{S+1-(s-1)}\binom{S+1}{s-1} \right) \\
&\qquad \cdot \mathcal{P}_D(C+1,S+1,1,0) \\
&\quad + \binom{S+1}{s} r^{S+1-s}(1-r)^s \cdot \mathcal{P}_D(C+2,S+1,1,0) \\
&\quad + \binom{S+2}{s} r^{S+2-s}(1-r)^s \cdot \mathcal{P}_D(C+1,S+2,0,0). \qquad (207)
\end{aligned}
$$

For $2 \leq n \leq C$, we obtain:

$$
\begin{aligned}
&\mathcal{P}_D(n,s,1,1) = \\
&\quad p^2(1-r)^{s-2} \cdot \mathcal{P}_D(n,s-2,1,1) \\
&\quad + \Big(2p(1-p)(1-r)^{s-1} + p^2(s-1)r(1-r)^{s-2}\Big) \cdot \mathcal{P}_D(n,s-1,1,1) \\
&\quad + \sum_{i=s}^{S} \left((1-p)^2 \binom{i}{s} r^{i-s}(1-r)^s + 2p(1-p)r^{i-s+1}(1-r)^{s-1}\binom{i}{s-1} \right) \\
&\qquad \cdot \mathcal{P}_D(n,i,1,1) \\
&\quad + \sum_{i=s}^{S} \left(p^2 r^{i-s+2}(1-r)^{s-2}\binom{i}{s-2} \right) \cdot \mathcal{P}_D(n,i,1,1) \\
&\quad + \left(\binom{S+1}{s} r^{S+1-s}(1-r)^s(1-p) + p\binom{S+1}{s-1} r^{S+1-(s-1)}(1-r)^{s-1} \right) \\
&\qquad \cdot \mathcal{P}_D(n,S+1,0,1) \\
&\quad + \left(\binom{S+1}{s} r^{S+1-s}(1-r)^s(1-p) + p\binom{S+1}{s-1} r^{S+1-(s-1)}(1-r)^{s-1} \right) \\
&\qquad \cdot \mathcal{P}_D(n,S+1,1,0) \\
&\quad + \binom{S+2}{s} r^{S+2-s}(1-r)^s \cdot \mathcal{P}_D(n,S+2,0,0).
\end{aligned} \tag{208}
$$

The lower boundary case is given as:

$$
\begin{aligned}
\mathcal{P}_D(1,s,1,1) &= p^2(1-r)^{s-2} \cdot \mathcal{P}_D(1,s-2,1,1) \\
&\quad + \Big(2p(1-p)(1-r)^{s-1} + p^2(s-1)r(1-r)^{s-2}\Big) \cdot \mathcal{P}_D(1,s-1,1,1) \\
&\quad + \sum_{i=s}^{S} \left((1-p)^2 \binom{i}{s} r^{i-s}(1-r)^s + 2p(1-p)r^{i-s+1}(1-r)^{s-1}\binom{i}{s-1} \right) \\
&\qquad \cdot \mathcal{P}_D(1,i,1,1) \\
&\quad + \sum_{i=s}^{S} \left(p^2 r^{i-s+2}(1-r)^{s-2}\binom{i}{s-2} \right) \cdot \mathcal{P}_D(1,i,1,1) \\
&\quad + \binom{S+1}{s} r^{S+1-s}(1-r)^s \cdot \mathcal{P}_D(0,S+1,0,1) \\
&\quad + \left(\binom{S+1}{s} r^{S+1-s}(1-r)^s(1-p) + p\binom{S+1}{s-1} r^{S+1-(s-1)}(1-r)^{s-1} \right) \\
&\qquad \cdot \mathcal{P}_D(1,S+1,0,1) \\
&\quad + \binom{S+2}{s} r^{S+2-s}(1-r)^s \cdot \mathcal{P}_D(1,S+2,0,0).
\end{aligned} \tag{209}
$$

For $s = 1$, the following equation holds:

$$\begin{aligned}\mathcal{P}_D(C+1,1,1,1) =& \binom{S+2}{S+1} r^{S+1}(1-r) \cdot \mathcal{P}_D(C+1,S+2,0,0) \\ &+ \binom{S+1}{S} r^{S}(1-r) \cdot \mathcal{P}_D(C+2,S+1,1,0) \\ &+ \left(\binom{S+1}{1} r^{S}(1-r)(1-p) + r^{S+1}p\right) \\ &\quad \cdot \mathcal{P}_D(C+1,S+1,1,0) \\ &+ \sum_{s=1}^{S} \left(\binom{s}{1}(1-p)^2(1-r)r^{s-1} + 2(1-p)pr^{s}\right) \\ &\quad \cdot \mathcal{P}_D(C+1,s,1,1) \\ &+ 2(1-p)p \cdot \mathcal{P}_D(C+1,0,1,1). \end{aligned} \tag{210}$$

For $2 \leq n \leq C$, it holds that:

$$\begin{aligned}\mathcal{P}_D(n,1,1,1) =& \binom{S+2}{1} r^{S+1}(1-r) \cdot \mathcal{P}_D(n,S+2,0,0) \\ &+ \left(\binom{S+1}{1} r^{S}(1-r)(1-p) + r^{S+1}p\right) \cdot \mathcal{P}_D(n,S+1,1,0) \\ &+ \left(\binom{S+1}{1} r^{S}(1-r)(1-p) + r^{S+1}p\right) \cdot \mathcal{P}_D(n,S+1,0,1) \\ &+ \sum_{s=1}^{S} \left(\binom{s}{1}(1-p)^2(1-r)r^{s-1} + 2(1-p)pr^{s}\right) \cdot \mathcal{P}_D(n,s,1,1) \\ &+ 2(1-p)p \cdot \mathcal{P}_D(n,0,1,1). \end{aligned} \tag{211}$$

The lower boundary equation in this situation is given as:

$$\begin{aligned}\mathcal{P}_D(1,1,1,1) = &\binom{S+2}{1} r^{S+1}(1-r)\cdot\mathcal{P}_D(1,S+2,0,0)\\&+\left(\binom{S+1}{1} r^{S}(1-r)(1-p)+r^{S+1}p\right)\cdot\mathcal{P}_D(1,S+1,0,1)\\&+\binom{S+1}{1} r^{S}(1-r)\cdot\mathcal{P}_D(0,S+1,0,1)\\&+\sum_{s=1}^{S}\left(\binom{s}{s-1}(1-p)^2 r^{s-1}(1-r)+2(1-p)pr^s\right)\cdot\mathcal{P}_D(1,s,1,1)\\&+2(1-p)p\cdot\mathcal{P}_D(1,0,1,1). \qquad (212)\end{aligned}$$

Finally, we have to derive the equations for the situation without outstanding replenishment orders for the spare part stock. The upper boundary equation is given as:

$$\begin{aligned}\mathcal{P}_D(C+1,0,1,1) = &\, r^{S+2}\cdot\mathcal{P}_D(C+1,S+2,0,0)\\&+r^{S+1}\cdot\mathcal{P}_D(C+2,S+1,1,0)\\&+r^{S+1}(1-p)\cdot\mathcal{P}_D(C+1,S+1,1,0)\\&+(1-p)^2\sum_{s=0}^{S} r^s\cdot\mathcal{P}_D(C+1,s,1,1). \qquad (213)\end{aligned}$$

For $2 \leq n \leq C$, we get:

$$\begin{aligned}\mathcal{P}_D(n,0,1,1) = &\, r^{S+2}\cdot\mathcal{P}_D(n,S+2,0,0)\\&+r^{S+1}(1-p)\cdot\mathcal{P}_D(n,S+1,1,0)\\&+r^{S+1}(1-p)\cdot\mathcal{P}_D(n,S+1,0,1)\\&+(1-p)^2\sum_{s=0}^{S} r^s\cdot\mathcal{P}_D(n,s,1,1). \qquad (214)\end{aligned}$$

For the lower boundary equation, we obtain:

$$\begin{aligned}\mathcal{P}_D(1,0,1,1) = {} & r^{S+2} \cdot \mathcal{P}_D(1, S+2, 0, 0) \\ & + r^{S+1}(1-p) \cdot \mathcal{P}_D(1, S+1, 0, 1) \\ & + r^{S+1} \cdot \mathcal{P}_D(0, S+1, 0, 1) \\ & + (1-p)^2 \sum_{s=0}^{S} r^s \cdot \mathcal{P}_D(1, s, 1, 1). \end{aligned} \tag{215}$$

5.5.2 Balance equations for Section 5.3

The state space leads to a complex variation of balance equations. In particular, the boundary equations are complex and must follow a set of conditions. Hence, we use a combined form of all balance equations, which depends directly on these conditions. This formulation allows for better readability and provides more insights into the problem structure.

Before setting up the balance equations, some of the characteristics that lead to using indicator functions should be explained. As only one failure can happen per machine and period, the number of outstanding orders s_i can only increase by a maximum of one. Conversely, the number of delivered components can take any number from zero to s_i. Hence, the formulation of the balance equations mainly uses $(\cdot, s_1, s_2)$ as the state notation for incoming transitions and $(\cdot, z_1, z_2)$ as the state notation for outgoing transitions with z_1 and z_2 defined accordingly. There are five types of conditions for indicator functions used in the balance equations, which can be combined for concurrent validity. Outstanding orders can only be delivered when $s_i > 0$. Therefore we need:

$$\mathbb{1}(s_i \neq 0). \tag{216}$$

If there are $S_i + 1$ outstanding orders and thus machine i is not working due to this failure, the probability that the machine will not fail has to be adjusted to one, which requires the following function:

$$\mathbb{1}(z_i \neq S_i + 1) \text{ in combination with } (1 - p_i)^{\mathbb{1}(z_i \neq S_i + 1)}. \tag{217}$$

The function

$$\mathbb{1}(0 < z_i < S_i + 1) \tag{218}$$

is used in equations where the number of outstanding orders stays equal. There are four possibilities to guarantee the same number: a) neither machine fails, b) machine 1 fails and machine 2 does not fail, but one outstanding component 1 is delivered, c) vice versa, and d) both machines fail with one outstanding component for each component is delivered. The last three situations require a positive number of outstanding orders $0 < z_i$ and for the corresponding machine to not already be down $z_i < S_i + 1$.

Finally,

$$\mathbb{1}(n \neq 1) \text{ and} \tag{219}$$

$$\mathbb{1}(n \neq C + 1) \tag{220}$$

handle cases, which would lead to blocking and starving when changes in the buffer level are involved, prohibiting some types of transitions.

For $1 \leq n \leq C + 1$, we get with $0 \leq s_1 \leq S_1, 0 \leq s_2 \leq S_2$ the balance equations of internal states:

$$
\begin{aligned}
&\mathcal{P}_D(n, s_1, s_2) = \\
&\quad \mathbb{1}(s_1 \neq 0, s_2 \neq 0)\mathcal{P}_D(n, s_1 - 1, s_2 - 1) \cdot \left(p_1 p_2 \cdot r_1(1 - r_1)^{s_1 - 1} \cdot r_2(1 - r_2)^{s_2 - 1}\right) \\
&\quad + \mathbb{1}(s_1 \neq 0) \sum_{z_2 = s_2}^{S_2 + 1} \mathcal{P}_D(n, s_1 - 1, z_2) \cdot \\
&\qquad \left(p_1(1 - p_2)^{\mathbb{1}(z_2 \neq S_2 + 1)} \cdot (1 - r_1)^{s_1 - 1} \cdot \binom{z_2}{s_2} r_2^{z_2 - s_2}(1 - r_2)^{s_2}\right. \\
&\qquad \left. + \mathbb{1}(0 < z_2 < S_2 + 1) \cdot p_1 p_2 \cdot (1 - r_1)^{s_1 - 1} \cdot \binom{z_2}{s_2 - 1} r_2^{z_2 - s_2 + 1}(1 - r_2)^{s_2 - 1}\right) \\
&\quad + \mathbb{1}(s_2 \neq 0) \sum_{z_1 = s_1}^{S_1 + 1} \mathcal{P}_D(n, z_1, s_2 - 1) \cdot \\
&\qquad \left((1 - p_1)^{\mathbb{1}(z_1 \neq S_1 + 1)} p_2 \cdot \binom{z_1}{s_1} r_1^{z_1 - s_1}(1 - r_1)^{s_1} \cdot (1 - r_2)^{s_2 - 1}\right. \\
&\qquad \left. + \mathbb{1}(0 < z_1 < S_1 + 1) \cdot p_1 p_2 \cdot \binom{z_1}{s_1 - 1} r_1^{z_1 - s_1 + 1}(1 - r_1)^{s_1 - 1} \cdot (1 - r_2)^{s_2 - 1}\right) \\
&\quad + \sum_{z_1 = s_1}^{S_1 + 1} \sum_{z_2 = s_2}^{S_2 + 1} \mathcal{P}_D(n, z_1, z_2) \left((1 - p_1)^{\mathbb{1}(z_1 \neq S_1 + 1)}(1 - p_2)^{\mathbb{1}(z_2 \neq S_2 + 1)} \cdot \right. \\
&\qquad\quad \binom{z_1}{s_1} r_1^{z_1 - s_1}(1 - r_1)^{s_1} \cdot \binom{z_2}{s_2} r_2^{z_2 - s_2}(1 - r_2)^{s_2} \\
&\qquad + \mathbb{1}(0 < z_1 < S_1 + 1) \cdot p_1(1 - p_2)^{\mathbb{1}(z_2 \neq S_2 + 1)} \cdot \\
&\qquad\quad \binom{z_1}{s_1 - 1} r_1^{z_1 - s_1 + 1}(1 - r_1)^{s_1 - 1} \cdot \binom{z_2}{s_2} r_2^{z_2 - s_2}(1 - r_2)^{s_2} \\
&\qquad + \mathbb{1}(0 < z_2 < S_2 + 1) \cdot (1 - p_1)^{\mathbb{1}(z_1 \neq S_1 + 1)} p_2 \cdot \binom{z_1}{s_1} r_1^{z_1 - s_1}(1 - r_1)^{s_1} \cdot \\
&\qquad\quad \binom{z_2}{s_2 - 1} r_2^{z_2 - s_2 + 1}(1 - r_2)^{s_2 - 1} \\
&\qquad + \mathbb{1}(0 < z_1 < S_1 + 1, 0 < z_2 < S_2 + 1) \cdot p_1 p_2 \cdot \\
&\qquad\quad \left. \binom{z_1}{s_1 - 1} r_1^{z_1 - s_1 + 1}(1 - r_1)^{s_1 - 1} \cdot \binom{z_2}{s_2 - 1} r_2^{z_2 - s_2 + 1}(1 - r_2)^{s_2 - 1}\right).
\end{aligned}
\tag{221}
$$

Another set of internal equations with $1 \leq n \leq C + 1$ and $0 \leq s_2 \leq S_2$ but $s_1 = S_1 + 1$

leads to:

$$
\begin{aligned}
&\mathcal{P}_D(n, S_1+1, s_2) = \\
&\quad \mathbb{1}(s_2 \neq 0, n \neq C+1)\mathcal{P}_D(n+1, S_1, s_2-1) \cdot \left(p_1 p_2 \cdot r_1(1-r_1)^{S_1} \cdot r_2(1-r_2)^{s_2-1}\right) \\
&\quad + \mathbb{1}(n \neq C+1) \sum_{z_2=s_2}^{S_2+1} \mathcal{P}_D(n+1, S_1, z_2) \cdot \\
&\qquad\qquad \left(p_1(1-p_2)^{\mathbb{1}(z_2 \neq S_2+1)} \cdot (1-r_1)^{S_1} \cdot \binom{z_2}{s_2} r_2^{z_2-s_2}(1-r_2)^{s_2}\right. \\
&\qquad\qquad \left. + \mathbb{1}(0 < z_2 < S_2+1) \cdot p_1 p_2 \cdot (1-r_1)^{S_1} \cdot \binom{z_2}{s_2-1} r_2^{z_2-s_2+1}(1-r_2)^{s_2-1}\right) \\
&\quad + \mathbb{1}(s_2 \neq 0)\mathcal{P}_D(n+1, S_1+1, s_2-1) \cdot \left(p_2 \cdot (1-r_1)^{S_1+1} \cdot (1-r_2)^{s_2-1}\right) \\
&\quad + \sum_{z_2=s_2}^{S_2+1} \mathcal{P}_D(n+1, S_1+1, z_2) \cdot \\
&\qquad\qquad \left((1-p_2)^{\mathbb{1}(z_2 \neq S_2+1)} \cdot (1-r_1)^{S_1+1} \cdot \binom{z_2}{s_2} r_2^{z_2-s_2}(1-r_2)^{s_2}\right. \\
&\qquad\qquad \left. + \mathbb{1}(0 < z_2 < S_2+1) \cdot p_2 \cdot (1-r_1)^{S_1+1} \cdot \binom{z_2}{s_2-1} r_2^{z_2-s_2+1}(1-r_2)^{s_2-1}\right).
\end{aligned} \tag{222}
$$

Conversely, for $1 \leq n \leq C+1$ and $0 \leq s_1 \leq S_1,\ s_2 = S_2 + 1$ can be fixed and yields:

$$\begin{aligned}
&\mathcal{P}_D(n, s_1, S_2+1) = \\
&\quad \mathbb{1}(s_1 \neq 0, n \neq 1)\mathcal{P}_D(n-1, s_1-1, S_2) \cdot \left(p_1 p_2 \cdot r_1(1-r_1)^{s_1-1} \cdot r_2(1-r_2)^{S_2}\right) \\
&\quad + \mathbb{1}(s_1 \neq 0)\mathcal{P}_D(n-1, S_1+1, s_2-1) \cdot \left(p_2 \cdot (1-r_1)^{S_1+1} \cdot (1-r_2)^{s_2-1}\right) \\
&\quad + \mathbb{1}(n \neq 1) \sum_{z_1=s_1}^{S_1+1} \mathcal{P}_D(n-1, z_1, S_2) \cdot \\
&\qquad \left((1-p_1)^{\mathbb{1}(z_1 \neq S_1+1)} p_2 \cdot \binom{z_1}{s_1} r_1^{z_1-s_1}(1-r_1)^{s_1} \cdot (1-r_2)^{S_2}\right. \\
&\qquad \left. + \mathbb{1}(0 < z_1 < S_1+1) \cdot p_1 p_2 \cdot \binom{z_1}{s_1-1} r_1^{z_1-s_1+1}(1-r_1)^{s_1-1} \cdot (1-r_2)^{S_2}\right) \\
&\quad + \sum_{z_1=s_1}^{S_1+1} \mathcal{P}_D(n-1, z_1, S_2+1) \cdot \\
&\qquad \left((1-p_1)^{\mathbb{1}(z_1 \neq S_1+1)} \cdot \binom{z_1}{s_1} r_1^{z_1-s_1}(1-r_1)^{s_1} \cdot (1-r_2)^{S_2+1}\right. \\
&\qquad \left. + \mathbb{1}(0 < z_1 < S_1+1) \cdot p_1 \cdot \binom{z_1}{s_1-1} r_1^{z_1-s_1+1}(1-r_1)^{s_1-1} \cdot (1-r_2)^{S_2+1}\right).
\end{aligned} \tag{223}$$

Finally, for $1 \leq n \leq C+1$ and $s_1 = S_1+1, s_2 = S_2+1$, we get the last equation for internal values of the buffer level as:

$$\begin{aligned}
\mathcal{P}_D(n, S_1+1, S_2+1) = &\ \mathcal{P}_D(n, S_1, S_2) \cdot \left(p_1 p_2 \cdot (1-r_1)^{S_1} \cdot (1-r_2)^{S_2}\right) \\
&+ \mathcal{P}_D(n, S_1+1, S_2) \cdot \left(p_2 \cdot (1-r_1)^{S_1+1} \cdot (1-r_2)^{S_2}\right) \\
&+ \mathcal{P}_D(n, S_1, S_2+1) \cdot \left(p_1 \cdot (1-r_1)^{S_1} \cdot (1-r_2)^{S_2+1}\right) \\
&+ \mathcal{P}_D(n, S_1+1, S_2+1) \cdot \left((1-r_1)^{S_1+1} \cdot (1-r_2)^{S_2+1}\right).
\end{aligned} \tag{224}$$

As a lower boundary $n = 0$ is set. This directly implies $s_1 = S_1 + 1$ as machine 2 can only be starving if machine 1 is not working. Accordingly, we get with $0 \leq s_2 \leq S_2$:

$$\begin{aligned}
&\mathcal{P}_D(0, S_1+1, s_2) = \\
&\quad \mathbb{1}(s_2 \neq 0)\mathcal{P}_D(1, S_1, s_2-1) \cdot \left(p_1 p_2 \cdot r_1(1-r_1)^{S_1} \cdot r_2(1-r_2)^{s_2-1}\right) \\
&\quad + \sum_{z_2=s_2}^{S_2+1} \mathcal{P}_D(1, S_1, z_2) \cdot \\
&\qquad\quad \left(p_1(1-p_2)^{\mathbb{1}(z_2 \neq S_2+1)} \cdot (1-r_1)^{S_1} \cdot \binom{z_2}{s_2} r_2^{z_2-s_2}(1-r_2)^{s_2}\right. \\
&\qquad\quad \left. + \mathbb{1}(0 < z_2 < S_2+1) \cdot p_1 p_2 \cdot (1-r_1)^{S_1} \cdot \binom{z_2}{s_2-1} r_2^{z_2-s_2+1}(1-r_2)^{s_2-1}\right) \\
&\quad + \mathbb{1}(s_2 \neq 0)\mathcal{P}_D(1, S_1+1, s_2-1) \cdot \left(p_2 \cdot (1-r_1)^{S_1+1} \cdot (1-r_2)^{s_2-1}\right) \\
&\quad + \sum_{z_2=s_2}^{S_2+1} \mathcal{P}_D(1, S_1+1, z_2) \cdot \\
&\qquad\quad \left((1-p_2)^{\mathbb{1}(z_2 \neq S_2+1)} \cdot (1-r_1)^{S_1+1} \cdot \binom{z_2}{s_2} r_2^{z_2-s_2}(1-r_2)^{s_2}\right. \\
&\qquad\quad \left. + \mathbb{1}(0 < z_2 < S_2+1) \cdot p_2 \cdot (1-r_1)^{S_1+1} \cdot \binom{z_2}{s_2-1} r_2^{z_2-s_2+1}(1-r_2)^{s_2-1}\right) \\
&\quad + \sum_{z_2=s_2}^{S_2+1} \mathcal{P}_D(0, S_1+1, z_2) \cdot \left((1-r_1)^{S_1+1} \cdot \binom{z_2}{s_2} r_2^{z_2-s_2}(1-r_2)^{s_2}\right). \qquad (225)
\end{aligned}$$

The upper boundary equations can be obtained with setting $n = C + 2$. Analogously, this implies $s_2 = S_2 + 1$ since machine 1 can only be blocked if machine 2 is not working. With $0 \leq s_1 \leq S_1$ the last equations can be obtained:

$$
\begin{aligned}
&\mathcal{P}_D(C+2, s_1, S_2+1) = \\
&\quad \mathbb{1}(s_1 \neq 0)\mathcal{P}_D(C+1, s_1-1, S_2) \cdot \Big(p_1 p_2 \cdot r_1(1-r_1)^{s_1-1} \cdot r_2(1-r_2)^{S_2}\Big) \\
&\quad + \mathbb{1}(s_1 \neq 0)\mathcal{P}_D(C+1, S_1+1, s_2-1) \cdot \Big(p_2 \cdot (1-r_1)^{S_1+1} \cdot (1-r_2)^{s_2-1}\Big) \\
&\quad + \sum_{z_1=s_1}^{S_1+1} \mathcal{P}_D(C+1, z_1, S_2) \cdot \\
&\qquad\quad \Bigg((1-p_1)^{\mathbb{1}(z_1 \neq S_1+1)} p_2 \cdot \binom{z_1}{s_1} r_1^{z_1-s_1}(1-r_1)^{s_1} \cdot (1-r_2)^{S_2} \\
&\qquad\quad + \mathbb{1}(0 < z_1 < S_1+1) \cdot p_1 p_2 \cdot \binom{z_1}{s_1-1} r_1^{z_1-s_1+1}(1-r_1)^{s_1-1} \cdot (1-r_2)^{S_2}\Bigg) \\
&\quad + \sum_{z_1=s_1}^{S_1+1} \mathcal{P}_D(C+1, z_1, S_2+1) \cdot \\
&\qquad\quad \Bigg((1-p_1)^{\mathbb{1}(z_1 \neq S_1+1)} \cdot \binom{z_1}{s_1} r_1^{z_1-s_1}(1-r_1)^{s_1} \cdot (1-r_2)^{S_2+1} \\
&\qquad\quad + \mathbb{1}(0 < z_1 < S_1+1) \cdot p_1 \cdot \binom{z_1}{s_1-1} r_1^{z_1-s_1+1}(1-r_1)^{s_1-1} \cdot (1-r_2)^{S_2+1}\Bigg) \\
&\quad + \sum_{z_1=s_1}^{S_1+1} \mathcal{P}_D(C+2, z_1, S_2+1) \cdot \left(\binom{z_1}{s_1} r_1^{z_1-s_1}(1-r_1)^{s_1} \cdot (1-r_2)^{S_2+1}\right). \qquad (226)
\end{aligned}
$$

Chapter

6 Summary and Outlook

Logic is the beginning of wisdom, not the end.

Commander Spock

This thesis aimed at integrating spare parts planning into the design process of manufacturing systems. Until now, there were only low-fidelity insights on how planners could incorporate spare parts. Hence, we developed several models of flow lines and corresponding solutions techniques. We presented results amplifying the necessity of simultaneous buffer and spare parts planning. This chapter summarizes the key findings of this thesis by answering the research questions and raises suggestions for future research.

6.1 Summary

> *Research Question 1: What is the throughput of a flow line consisting of an arbitrary number of machines with limited buffer capacities and spare part stocks?*

In Chapter 3, we answered Research Question 1. We presented a continuous-time Markov chain formulation and determined its steady-state probabilities using a system of balance equations. In doing so, we were able to calculate performance measures for small systems consisting of two machines and, to some extent, three-machine systems, exactly. Using the idea of Gershwin's (1987) decomposition approach, we developed an approximate algorithm to compute performance measures for flow lines of arbitrary length.

Our results indicate how fast the exact Markov chain evaluation is for small systems. Applying this as the building block in the decomposition algorithm, we obtained very accurate and prompt evaluations for longer systems. We tested about 20,000 instances for unbalanced three-machine lines in order to validate the decomposition with exact results. We obtained mean absolute deviations of 0.60% for the throughput, 1.60% for the average buffer levels, and 1.78% for the average spare part stocks on hand.

The decomposition algorithm is known to cause convergence problems in some instances. Since proofs for convergence — even for the initially proposed decomposition — are still to be found, we shed light on examples raising concerns. Long lines consisting of machines with low availability seem especially challenging to evaluate. Moreover, system designs with relatively poor performance produce poor results. An explanation may be the assumption that the probability of concurrent blocking and starving is zero. Overall, a good proportion of cases has throughput estimates inside our tight 95% confidence intervals.

Eventually, we were able to get throughput estimates for a wide variety of flow-line instances. We presented some first insights that buffers and spare parts interact and that allocating spare parts with identical costs actually makes a difference in throughput. Therefore, it is vital to analyze optimal allocations to generate robust insights into allocation patterns of spare parts in flow lines with interstage buffers.

> *Research Question 2: What is the optimal flow-line design regarding buffer capacities and spare part base-stock levels? How do both decisions interact?*

Chapter 4 made use of the developed ability to evaluate long flow lines in order to answer Research Question 2. We proposed the *Primal buffer and spare part allocation problem*, which comprises the trade-off between buffers and spare parts in order to reach the target throughput. In order to solve this NP-hard optimization problem, we developed three greedy heuristics and applied the metaheuristics simulated annealing and a genetic algorithm.

We validated all algorithms against complete enumeration. Using results for hundreds of unbalanced flow line instances, we inferred that the greedy heuristics are most promising for further application.

We observed several well-known phenomena. The buffer allocation often follows a bowl pattern, even in the presence of spare parts. Comparable to results for system-level approaches from spare parts planning, cheap spare parts are used numerously, whereas expensive spares are avoided if possible.

Moreover, we identified multiple unknown effects. The allocation of spare parts tends to follow a bowl pattern, too. Also, there are complex interactions between buffers and spare parts. In some cases, even the bowl pattern of spare parts stashes away the bowl allocation for buffers. These results support the necessity of further analysis on how integrated flow line planning should optimally occur. Two interesting special cases were further analyzed in this thesis.

> *Research Question 3: Do the results of systems with stochastic processing times carry over to transfer lines with deterministic cycle times?*

In Chapter 4, we presented two new models of two-machine transfer lines with deterministic and identical processing times: one with machine-specific and the other one with identical critical components. The results related to the former model gave us the answer to Research Question 3. Using discrete-time Markov chains, we obtained performance measures for both.

We observed that the interaction between buffers and spare parts is even more remarkable for deterministic processing times. Since there is no need to cope with processing time variability, the only source of production stoppage is a failure of a critical component. Thus, the impact of costs per buffer or spare is more sweeping, yielding system designs without buffers if they are very expensive and vice versa for spare parts.

Similarly, we find that cheap spares are stocked numerously, whereas expensive spares are avoided. The effect of bottlenecks is comparably causing more spare parts to increase their availability, a higher buffer capacity to overcome its impact, or both.

> *Research Question 4: What is the impact of component standardization comparing a two-machine system with machine-specific failure-prone critical*

components to a system with identical ones?

The comparison of the two different models regarding identical and machine-specific critical components discussed in Chapter 5 enables us to answer Research Question 4. On average, the optimal flow line design with spare part commonality reduces costs by about 38%. Moreover, we find that in 97% of the 105 analyzed instances, a common critical component facilitates a cost reduction. The impact on the achieved throughput was ambiguous. There are instances with a throughput increase as well as a decrease. The mean throughput change is negligible at about -0.005%.

In 82% of the analyzed instances, spare part commonality reduced the number of spare parts in stock. On average, the reduction amounts to 29%. This is in line with the literature on component standardization. However, even more noteworthy, we observe that buffer capacities could be reduced in 40% of the instances. On average, buffer capacities could be reduced by 30%. This result once again amplifies the strong interaction between buffers and spare parts.

How should a flow line with buffers and spare parts provisioning for corrective maintenance be designed?

We shed light on the integrated design for flow lines with limited buffer capacities and spare parts provisioning for corrective maintenance. In our research, it became apparent that substantial cost-savings can be rendered possible if planners consider the decisions on buffers and spare parts jointly instead of treating them separately. We presented results for flow lines of arbitrary length and illustrated that our results become even stronger for more realistic fixed cycle times.

With the models, solutions, and numerical results presented in this thesis, we provide the basis for future research on this topic. Our results indicate the importance of integrated flow-line planning and that it should be pursued.

6.2 Further research opportunities

Future research could focus on the influence of digital transformation. 3D-printed spares are about to become an alternative to traditional replenishment and stock-keeping of spare parts (Song & Zhang, 2020). Hence, 3D-printed spare parts could also be adopted for manufacturing system maintenance. This dual-sourcing approach would pose additional questions that need to incorporate the differences in replenishment times, etc., in 3D printing.

Another issue arises from the inclusion of critical components. It is apparent that neither all failures are caused by critical components nor do complex appliances contain only one component, which is vital for their functioning. Further analysis should include additional failure modes: some are component-related, and some are not.

We amplified in Chapter 5 how component standardization could facilitate cost savings for a two-machine system. However, for longer manufacturing systems, it becomes even more likely that some — or even all — machines share identical critical components. In such a system, component commonality could render even higher cost savings possible. Future research could, therefore, develop models for longer flow lines, including multiple critical components per machine.

In our analysis of the transfer line with deterministic and identical processing times we restricted our analysis to the two-machine case. This research direction may allow for further insights into spare parts provisioning since the impact of spare parts is considerably stronger if no variability in processing times occurs. A new decomposition approach could be developed where the presented Markov chain is a valuable building block.

It has been 63 years since Koenigsberg (1959) reviewed the state regarding buffer allocations in flow lines. This thesis contributes to this stream of research by integrating further decisions into the design process of manufacturing systems. Still, there are open questions requiring scientific analysis and being the subject of future publications.

Bibliography

Alfredsson, P., & Verrijdt, J. (1999). Modeling emergency supply flexibility in a two-echelon inventory system. *Management Science*, *45*(10), 1416–1431. https://doi.org/10.1287/mnsc.45.10.1416

Altiparmak, F., Dengiz, B., & Bulgak, A. A. (2002). Optimization of buffer sizes in assembly systems using intelligent techniques. *Proceedings of the Winter Simulation Conference*, *2*, 1157–1162 vol.2. https://doi.org/10.1109/WSC.2002.1166373

Bai, Y., Tu, J., Yang, M., Zhang, L., & Denno, P. (2020). A new aggregation algorithm for performance metric calculation in serial production lines with exponential machines: Design, accuracy and robustness. *International Journal of Production Research*, *59*(13), 4072–4089. https://doi.org/10.1080/00207543.2020.1757777

Basten, R. J. I., & Arts, J. J. (2017). Fleet readiness: Stocking spare parts and high-tech assets. *IISE Transactions*, *49*(4), 429–441. https://doi.org/10.1080/0740817X.2016.1243281

Basten, R. J. I., & Van Houtum, G.-J. (2014). System-oriented inventory models for spare parts. *Surveys in Operations Research and Management Science*, *19*(1), 34–55. https://doi.org/10.1016/j.sorms.2014.05.002

Basu, R. N. (1977). The interstage buffer storage capacity of non-powered assembly lines – A simple mathematical approach. *International Journal of Production Research*, *15*(4), 365–382. https://doi.org/10.1080/00207547708943136

Birolini, A. (2012). *Quality and reliability of technical systems: Theory, practice, management*. Springer Science & Business Media. https://doi.org/10.1007/978-3-662-02970-1

Blumenfeld, D. (2001). *Operations research calculations handbook*. CRC Press.

Bulgak, A. A., Diwan, P. D., & Inozu, B. (1995). Buffer size optimization in asynchronous assembly systems using genetic algorithms. *Computers & Industrial Engineering*, *28*(2), 309–322. https://doi.org/10.1016/0360-8352(94)00188-S

Burman, M. H. (1995). *New results in flow line analysis* (Doctoral dissertation). Massachusetts Institute of Technology.

Buzacott, J. A. (1967). Automatic transfer lines with buffer stocks. *International Journal of Production Research*, *5*(3), 183–200. https://doi.org/10.1080/00207546708929751

Buzacott, J. A., & Hanifin, L. E. (1978). Models of automatic transfer lines with inventory banks a review and comparison. *AIIE Transactions*, *10*(2), 197–207. https://doi.org/10.1080/05695557808975204

Buzacott, J. A., & Kostelski, D. (1987). Matrix-geometric and recursive algorithm solution of a two-stage unreliable flow line. *IIE Transactions*, *19*(4), 429–438. https://doi.org/10.1080/07408178708975416

Chang, S., & Gershwin, S. B. (2010). Modeling and analysis of two unreliable batch machines with a finite buffer in between. *IIE Transactions*, *42*(6), 405–421. https://doi.org/10.1080/07408170903228934

Cheng, G., Zhou, B., & Li, L. (2017). Geometric process repair model for an unreliable production system with an intermediate buffer. *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering*, *231*(4), 747–759.

Choong, Y. F., & Gershwin, S. B. (1987). A decomposition method for the approximate evaluation of capacitated transfer lines with unreliable machines and random processing times. *IIE Transactions*, *19*(2), 150–159. https://doi.org/10.1080/07408178708975381

Cohen, M. A., Zheng, Y.-S., & Agrawal, V. (1997). Service parts logistics: A benchmark analysis. *IISE Transactions*, *29*(8), 627–639. https://doi.org/10.1023/A:1018513900264

Colledani, M., Lundholm, T., Ratti, A., & Tolio, T. A. M. (2012). Analytical methods for performance evaluation of transfer lines with shared repair capacity at scania. *VTI 2012-Vehicle technology integration Conference*, 1–6.

Colledani, M., & Tolio, T. (2009). Performance evaluation of transfer lines with general repair times and multiple failure modes. *Annals of Operations Research*, *182*(1), 31–65. https://doi.org/10.1007/s10479-009-0595-3

Colledani, M., & Tolio, T. A. M. (2012). Integrated quality, production logistics and maintenance analysis of multi-stage asynchronous manufacturing systems with degrading machines. *CIRP annals*, *61*(1), 455–458. https://doi.org/10.1016/j.cirp.2012.03.072

Conway, R., Maxwell, W., McClain, J. O., & Thomas, L. J. (1988). The role of work-in-process inventory in serial production lines. *Operations Research*, *36*(2), 229–241. https://doi.org/10.1287/opre.36.2.229

Dallery, Y. (1999). An improved decomposition method for the analysis of production lines with unreliable machines and finite buffers. *International Journal of Production Research*, *37*(5), 1093–1117. https://doi.org/10.1080/002075499191427

Dallery, Y., David, R., & Xie, X.-L. (1988). An efficient algorithm for analysis of transfer lines with unreliable machines and finite buffers. *IIE Transactions*, *20*(3), 280–283. https://doi.org/10.1080/07408178808966181

Dallery, Y., David, R., & Xie, X.-L. (1989). Approximate analysis of transfer lines with unreliable machines and finite buffers. *IEEE Transactions on Automatic control*, *34*(9), 943–953. https://doi.org/10.1109/9.35807

Dallery, Y., & Gershwin, S. B. (1992). Manufacturing flow line systems: A review of models and analytical results. *Queueing Systems*, *12*(1-2), 3–94. https://doi.org/10.1007/BF01158636

De Koster, M. (1987). Estimation of line efficiency by aggregation. *International Journal of Production Research*, *25*(4), 615–625. https://doi.org/10.1080/00207548708919865

Demir, L., Tunali, S., & Eliiyi, D. T. (2014). The state of the art on buffer allocation problem: A comprehensive survey. *Journal of Intelligent Manufacturing*, *25*(3), 371–392. https://doi.org/10.1007/s10845-012-0687-9

Diamantidis, A. C., & Papadopoulos, C. T. (2006). Markovian analysis of a discrete material manufacturing system with merge operations, operation-dependent and idleness failures. *Computers & Industrial Engineering*, *50*(4), 466–487. https://doi.org/10.1016/j.cie.2005.04.004

Drent, M., & Arts, J. J. (2021). Expediting in two-echelon spare parts inventory systems. *Manufacturing & Service Operations Management*, *23*(6), 1431–1448. https://doi.org/10.1287/msom.2020.0888

Driessen, M., Arts, J. J., Van Houtum, G.-J., Rustenburg, J. W., & Huisman, B. (2015). Maintenance spare parts planning and control: A framework for control and agenda for future research. *Production Planning & Control*, *26*(5), 407–426. https://doi.org/10.1080/09537287.2014.907586

Dudick, A. L. (1979). *Fixed-cycle production systems with in-line inventory and limited repair capability* (Doctoral dissertation). Columbia University.

Enginarlar, E., Li, J., Meerkov, S. M., & Zhang, R. Q. (2002). Buffer capacity for accommodating machine downtime in serial production lines. *International*

Journal of Production Research, *40*(3), 601–624. https://doi.org/10.1080/00207540110091703

Feeney, G. J., & Sherbrooke, C. C. (1966). The (s-1, s) inventory policy under compound poisson demand. *Management Science*, *12*(5), 391–411. https://doi.org/10.1287/mnsc.12.5.391

Fitouhi, M.-C., Nourelfath, M., & Gershwin, S. B. (2017). Performance evaluation of a two-machine line with a finite buffer and condition-based maintenance. *Reliability Engineering & System Safety*, *166*, 61–72. https://doi.org/10.1016/j.ress.2017.03.034

Frandson, A., Berghede, K., & Tommelein, I. D. (2013). Takt time planning for construction of exterior cladding. *Proceedings of the 21st Annual Conference of the International Group for Lean Construction*, 527–536.

Frandson, A., & Tommelein, I. D. (2014). Development of a takt-time plan: A case study. *Construction Research Congress 2014: Construction in a Global Network*, 1646–1655. https://doi.org/10.1061/9780784413517.168

Gad, A. F. (2021). Pygad: An intuitive genetic algorithm python library. https://doi.org/10.48550/arXiv.2106.06158

Gan, S., & Shi, J. (2014). Maintenance optimization for a production system with intermediate buffer and replacement part order considered. *Eksploatacja i Niezawodność*, *16*(1), 140–149.

Gan, S., Zhang, Z., Zhou, Y., & Shi, J. (2013). Intermediate buffer analysis for a production system. *Applied Mathematical Modelling*, *37*(20), 8785–8795. https://doi.org/10.1016/j.apm.2013.03.073

Gan, S., Zhang, Z., Zhou, Y., & Shi, J. (2015). Joint optimization of maintenance, buffer, and spare parts for a production system. *Applied Mathematical Modelling*, *39*(19), 6032–6042. https://doi.org/10.1016/j.apm.2015.01.035

Gebennini, E., & Gershwin, S. B. (2013). Modeling waste production into two-machine-one-buffer transfer lines. *IIE Transactions*, *45*(6), 591–604. https://doi.org/10.1080/0740817X.2012.748994

Gebennini, E., & Grassi, A. (2015). Discrete-time model for two-machine one-buffer transfer lines with buffer bypass and two capacity levels. *IIE Transactions*, *47*, 715–727. https://doi.org/10.1080/0740817X.2014.952849

Gebennini, E., Grassi, A., Fantuzzi, C., Gershwin, S. B., & Schick, I. C. (2013). Discrete time model for two-machine one-buffer transfer lines with restart policy. *Annals of Operations Research*, *209*, 41–65. https://doi.org/10.1007/s10479-011-0868-5

Gebennini, E., Grassi, A., Fantuzzi, C., & Rimini, B. (2017). Discrete time model of a two-machine one-buffer serial system with inventory level-dependent operation. *Computers & Industrial Engineering*, *113*, 46–63. https://doi.org/10.1016/j.cie.2017.09.0070

Geman, S., & Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6), 721–741. https://doi.org/10.1109/TPAMI.1984.4767596

Gershwin, S. 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. https://doi.org/10.1287/opre.35.2.291

Gershwin, S. B. (1991). Assembly/disassembly systems: An efficient decomposition algorithm for tree-structured networks. *IIE Transactions*, *23*(4), 302–314. https://doi.org/10.1080/07408179108963865

Gershwin, S. B. (1994). *Manufacturing systems engineering*. Prentice Hall.

Gershwin, S. B. (2018). The future of manufacturing systems engineering. *International Journal of Production Research*, *56*(1-2), 224–237. https://doi.org/10.1080/00207543.2017.1395491

Gershwin, S. B., & Berman, O. (1981). Analysis of transfer lines consisting of two unreliable machines with random processing times and finite storage buffers. *AIIE Transactions*, *13*(1), 2–11. https://doi.org/10.1080/05695558108974530

Gershwin, S. B., & Schick, I. C. (1983). Modeling and analysis of three-stage transfer lines with unreliable machines and finite buffers. *Operations Research*, *31*(2), 354–380. https://doi.org/10.1287/opre.31.2.354

Gershwin, S. B., & Schor, J. E. (2000). Efficient algorithms for buffer space allocation. *Annals of Operations Research*, *93*(1/4), 117–144. https://doi.org/10.1023/A:1018988226612

Göttlich, S., Herty, M., Ringhofer, C., & Ziegler, U. (2012). Production systems with limited repair capacity. *Optimization*, *61*(8), 915–948. https://doi.org/10.1080/02331934.2011.615395

Hajek, B. (1988). Cooling schedules for optimal annealing. *Mathematics of Operations Research*, *13*(2), 311–329. https://doi.org/10.1287/moor.13.2.311

Heavey, C., Papadopoulos, H., & Browne, J. (1993). The throughput rate of multistation unreliable production lines. *European Journal of Operational Research*, *68*(1), 69–89. https://doi.org/10.1016/0377-2217(93)90077-Z

Helber, S. (1998). Decomposition of unreliable assembly/disassembly networks with limited buffer capacity and random processing times. *European Journal of Operational Research*, *109*(1), 24–42. https://doi.org/10.1016/S0377-2217(97)00166-5

Helber, S. (1999). *Performance analysis of flow lines with non-linear flow of material*. Springer Science & Business Media. https://doi.org/10.1007/978-3-642-95863-2

Helber, S. (2000). Approximate analysis of unreliable transfer lines with splits in the flow of material. *Annals of Operations Research*, *93*(1-4), 217–243. https://doi.org/10.1023/A:1018923907048

Helber, S., & Jusić, H. (2004). A new decomposition approach for non-cyclic continuous material flow lines with a merging flow of material. *Annals of Operations Research*, *125*(1-4), 117–139. https://doi.org/10.1023/B:ANOR.0000011188.11019.33

Helber, S., Schimmelpfeng, K., Stolletz, R., & Lagershausen, S. (2011). Using linear programming to analyze and optimize stochastic flow lines. *Annals of Operations Research*, *182*(1), 193–211. https://doi.org/10.1007/s10479-010-0692-3

Hillier, F. S., & Boling, R. W. (1967). Finite queues in series with exponential or erlang service times-A numerical approach. *Operations Research*, *15*(2), 286–303. https://doi.org/10.1287/opre.15.2.286

Hillier, F. S., & Boling, R. W. (1979). On the optimal allocation of work in symmetrically unbalanced production line systems with variable operation times. *Management Science*, *25*(8), 721–728. https://doi.org/10.1287/mnsc.25.8.721

Hillier, F. S., So, K. C., & Boling, R. W. (1993). Notes: Toward characterizing the optimal allocation of storage space in production line systems with variable processing times. *Management Science*, *39*(1), 126–133. https://doi.org/10.1287/mnsc.39.1.126

Hillier, M. S. (2000). Characterizing the optimal allocation of storage space in production line systems with variable processing times. *IIE Transactions*, *32*(1), 1–8. https://doi.org/10.1023/A:1007633112170

Hong, Y., & Seong, D. (1993). The analysis of an unreliable two-machine production line with random processing times. *European Journal of Operational Research*, *68*(2), 228–235. https://doi.org/10.1016/0377-2217(93)90305-7

Hu, Q., Boylan, J. E., Chen, H., & Labib, A. (2018). OR in spare parts management: A review. *European Journal of Operational Research*, *266*, 395–414. https://doi.org/10.1016/j.ejor.2017.07.058

Hudson, S., McNamara, T., & Shaaban, S. (2015). Unbalanced lines: Where are we now? *International Journal of Production Research*, *53*(6), 1895–1911. https://doi.org/10.1080/00207543.2014.965357

Inman, R. (1999). Empirical evaluation of exponential and independence assumptions in queuing models of manufacturing systems. *Production and Operations Management*, *8*(4), 409–432. https://doi.org/10.1111/j.1937-5956.1999.tb00316.x

Jacobs, D., & Meerkov, S. M. (1995a). Mathematical theory of improvability for production systems. *Mathematical Problems in Engineering*, *1*(2), 95–137. https://doi.org/10.1155/S1024123X9500010X

Jacobs, D., & Meerkov, S. M. (1995b). System-theoretic analysis of due-time performance in production systems. *Mathematical Problems in Engineering*, *1*(3), 225–243. https://doi.org/10.1155/S1024123X95000147

Jacobs, D., & Meerkov, S. M. (1995c). A system-theoretic property of serial production lines: Improvability. *International Journal of Systems Science*, *26*(4), 755–785. https://doi.org/10.1080/00207729508929067

Jafari, M. A., & Shanthikumar, J. G. (1987). An approximate model of multistage automatic transfer lines with possible scrapping of workpieces. *IIE Transactions*, *19*(3), 252–265. https://doi.org/10.1080/07408178708975394

Jafari, M. A., & Shanthikumar, J. G. (1989). Determination of optimal buffer storage capacities and optimal allocation in multistage automatic transfer lines. *IIE Transactions*, *21*(2), 130–135. https://doi.org/10.1080/07408178908966216

Kahan, T., Bukchin, Y., Menassa, R., & Ben-Gal, I. (2009). Backup strategy for robots' failures in an automotive assembly system. *International Journal of Production Economics*, *120*(2), 315–326. https://doi.org/10.1016/j.ijpe.2007.09.015

Karamatsoukis, C. C., & Kyriakidis, E. G. (2010). Optimal maintenance of two stochastically deteriorating machines with an intermediate buffer. *European Journal of Operational Research*, *207*(1), 297–308. https://doi.org/10.1016/j.ejor.2010.04.022

Kennedy, W. J., Patterson, J. W., & Fredendall, L. D. (2002). An overview of recent literature on spare parts inventories. *International Journal of Production Economics*, *76*(2), 201–215. https://doi.org/10.1016/S0925-5273(01)00174-8

Kiesmüller, G., & Sachs, F. (2020). Spare parts or buffer? How to design a transfer line with unreliable machines. *European Journal of Operational Research*, *284*(1), 121–134. https://doi.org/10.1016/j.ejor.2019.12.005

Kiesmüller, G. P., & Zimmermann, J. (2018). The influence of spare parts provisioning on buffer size in a production system. *IISE Transactions*, *50*(5), 367–380. https://doi.org/10.1080/24725854.2018.1426134

Kim, J., & Gershwin, S. B. (2005). Integrated quality and quantity modeling of a production line. *OR spectrum*, *27*(2-3), 287–314. https://doi.org/10.1007/s00291-005-0202-1

Koenigsberg, E. (1959). Production lines and internal storage—A review. *Management Science*, *5*(4), 410–433. https://doi.org/10.1287/mnsc.5.4.410

Kolb, O., & Göttlich, S. (2015). A continuous buffer allocation model using stochastic processes. *European Journal of Operational Research*, *242*(3), 865–874. https://doi.org/10.1016/j.ejor.2014.10.065

Kranenburg, A. A. (2006). *Spare parts inventory control under system availability constraints* (Doctoral dissertation). Technische Universiteit Eindhoven. https://doi.org/10.6100/IR616052

Kranenburg, A. A., & Van Houtum, G.-J. (2007). Effect of commonality on spare parts provisioning costs for capital goods. *International Journal of Production Economics*, *108*(1-2), 221–227. https://doi.org/10.1016/j.ijpe.2006.12.025

Kubat, P., & Sumita, U. (1985). Buffers and backup machines in automatic transfer lines. *International Journal of Production Research*, *23*(6), 1259–1270. https://doi.org/10.1080/00207548508904780

Kuhn, H. (2003). Analysis of automated flow line systems with repair crew interference. *Analysis and modeling of manufacturing systems* (pp. 155–179). Springer. https://doi.org/10.1007/978-1-4615-1019-2_7

Kyriakidis, E. G., & Dimitrakos, T. D. (2006). Optimal preventive maintenance of a production system with an intermediate buffer. *European Journal of Operational Research*, *168*(1), 86–99. https://doi.org/10.1016/j.ejor.2004.01.052

Lamghari-Idrissi, D., Basten, R. J. I., Dellaert, N., & Van Houtum, G.-J. (2020a). Which spare parts service measure to choose for a front-end wafer fab? *IEEE Transactions on Semiconductor Manufacturing*, *33*(4), 504–510. https://doi.org/10.1109/TSM.2020.2992731

Lamghari-Idrissi, D., Basten, R. J. I., & Van Houtum, G.-J. (2020b). Spare parts inventory control under a fixed-term contract with a long-down constraint. *International Journal of Production Economics*, *219*, 123–137. https://doi.org/10.1016/j.ijpe.2019.05.023

Lamghari-Idrissi, D., Soellaart, D., Basten, R. J. I., & Dellaert, N. (2019). Influence of spare parts service measures on the performance of front-end wafer production process. *2019 Winter Simulation Conference (WSC)*, 2269–2280. https://doi.org/10.1109/WSC40007.2019.9004919

Lamghari-Idrissi, D. P. T. (2021). *A new after-sales service measure for stable customer operations* (Doctoral dissertation). Technische Universiteit Eindhoven.

Lau, H. C., Song, H., See, C. T., & Cheng, S. Y. (2006). Evaluation of time-varying availability in multi-echelon spare parts systems with passivation. *European Journal of Operational Research*, *170*(1), 91–105. https://doi.org/10.1016/j.ejor.2004.06.022

Law, A. M. (2014). *Simulation modeling and analysis* (Vol. 5). McGraw-Hill New York.

Le Bihan, H., & Dallery, Y. (2000). A robust decomposition method for the analysis of production lines with unreliable machines and finite buffers. *Annals of Operations Research*, *93*(1-4), 265–297. https://doi.org/10.1023/A:1018996428429

Li, J., Blumenfeld, D., & Alden, J. (2006). Comparisons of two-machine line models in throughput analysis. *International Journal of Production Research*, *44*(7), 1375–1398. https://doi.org/10.1080/00207540500371980

Li, J. (2005). Overlapping decomposition: A system-theoretic method for modeling and analysis of complex manufacturing systems. *IEEE Transactions on Automation Science and Engineering*, *2*(1), 40–53. https://doi.org/10.1109/TASE.2004.835576

Li, J., Blumenfeld, D. E., Huang, N., & Alden, J. M. (2009). Throughput analysis of production systems: Recent advances and future topics. *International Journal of Production Research*, *47*(14), 3823–3851. https://doi.org/10.1080/00207540701829752

Li, K.-F. (1987). Serial production lines with unreliable machines and limited repair. *Naval Research Logistics*, *34*(1), 101–108. https://doi.org/10.1002/1520-6750(198702)34:1<101::AID-NAV3220340109>3.0.CO;2-K

Liberopoulos, G. (2018). Performance evaluation of a production line operated under an echelon buffer policy. *IISE Transactions*, *50*(3), 161–177. https://doi.org/10.1080/24725854.2017.1390800

Lim, J.-T., Meerkov, S. M., & Top, F. (1990). Homogeneous, asymptotically reliable serial production lines: Theory and a case study. *IEEE Transactions on Automatic Control*, *35*(5), 524–534. https://doi.org/10.1109/9.53518

Linck, J., & Cochran, D. S. (1999). *The importance of takt time in manufacturing system design* (tech. rep.). SAE Technical Paper. https://doi.org/10.4271/1999-01-1635

Liu, X.-G., & Buzacott, J. A. (1990). Approximate models of assembly systems with finite inventory banks. *European Journal of Operational Research*, *45*(2-3), 143–154. https://doi.org/10.1016/0377-2217(90)90181-A

Lopes, T. C., Sikora, C. G. S., Molina, R. G., Schibelbain, D., Rodrigues, L. C. A., & Magatão, L. (2017). Balancing a robotic spot welding manufacturing line: An industrial case study. *European Journal of Operational Research*, *263*(3), 1033–1048. https://doi.org/10.1016/j.ejor.2017.06.001

Mak, K. L. (1986). The allocation of interstage buffer storage capacity in production lines. *Computers & Industrial Engineering*, *10*(3), 163–169. https://doi.org/10.1016/0360-8352(86)90002-1

Martin, G. E. (1990). Optimal buffer storage capacity in unpaced lines. *Computers & Industrial Engineering*, *18*(3), 401–405. https://doi.org/10.1016/0360-8352(90)90062-Q

Mascolo, M. d., David, R., & Dallery, Y. (1991). Modeling and analysis of assembly systems with unreliable machines and finite buffers. *IIE Transactions*, *23*(4), 315–330. https://doi.org/10.1080/07408179108963866

MATLAB. (2022). *9.12.0.1975300 (r2022a) update 3*. The MathWorks Inc.

Matta, A., & Simone, F. (2016). Analysis of two-machine lines with finite buffer, operation-dependent and time-dependent failure modes. *International Journal of Production Research*, *54*(6), 1850–1862. https://doi.org/10.1080/00207543.2015.1085654

Matta, A. (2008). Simulation optimization with mathematical programming representation of discrete event systems. *2008 Winter Simulation Conference*, 1393–1400. https://doi.org/10.1109/WSC.2008.4736215

Matta, A., Pezzoni, M., & Semeraro, Q. (2012). A kriging-based algorithm to optimize production systems approximated by analytical models. *Journal of Intelligent Manufacturing*, *23*(3), 587–597. https://doi.org/10.1007/s10845-010-0397-0

Meller, R. D., & Kim, D. S. (1996). The impact of preventive maintenance on system cost and buffer size. *European Journal of Operational Research*, *95*(3), 577–591. https://doi.org/10.1016/0377-2217(95)00313-4

Mirzahosseinian, H., & Piplani, R. (2011). A study of repairable parts inventory system operating under performance-based contract. *European Journal of Operational Research*, *214*(2), 256–261. https://doi.org/10.1016/j.ejor.2011.04.035

Moinzadeh, K., & Lee, H. L. (1986). Batch size and stocking levels in multi-echelon repairable systems. *Management Science*, *32*(12), 1567–1581. https://doi.org/10.1287/mnsc.32.12.1567

Moinzadeh, K., & Schmidt, C. P. (1991). An (s-1, s) inventory system with emergency orders. *Operations Research*, *39*(2), 308–321. https://doi.org/10.1287/opre.39.2.308

Muckstadt, J. A. (2005). *Analysis and algorithms for service parts supply chains*. Springer Science & Business Media. https://doi.org/10.1007/b138879

Okamura, K., & Yamashina, H. (1977). Analysis of the effect of buffer storage capacity in transfer line systems. *AIIE Transactions*, *9*(2), 127–135. https://doi.org/10.1080/05695557708975134

Papadopoulos, C. T., a, M. E. J., & Tsadiras, A. K. (2013). A DSS for the buffer allocation of production lines based on a comparative evaluation of a set of search algorithms. *International Journal of Production Research*, *51*(14), 4175–4199. https://doi.org/10.1080/00207543.2012.752585

Papadopoulos, C. T., Li, J., & O'Kelly, M. E. (2019). A classification and review of timed markov models of manufacturing systems. *Computers & Industrial Engineering*, *128*, 219–244. https://doi.org/10.1016/j.cie.2018.12.019

Papadopoulos, C. T., O'Kelly, M. E., Vidalis, M. J., & Spinellis, D. (2009). *Analysis and design of discrete part production lines*. Springer. https://doi.org/10.1007/978-0-387-89494-2

Papadopoulos, H. T., & Heavey, C. (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. https://doi.org/10.1016/0377-2217(95)00378-9

Papadopoulos, H., Heavey, C., & O'Kelly, M. (1989). Throughput rate of multistation reliable production lines with inter station buffers. *Computers in Industry*, *13*(3), 229–244. https://doi.org/10.1016/0166-3615(89)90113-9

Park, T. (1993). A two-phase heuristic algorithm for determining buffer sizes of production lines. *International Journal of Production Research*, *31*(3), 613–631. https://doi.org/10.1080/00207549308956747

Pintelon, L. M., & Gelders, L. F. (1992). Maintenance management decision making. *European Journal of Operational Research*, *58*(3), 301–317. https://doi.org/10.1016/0377-2217(92)90062-E

Powell, S. G., & Pyke, D. F. (1996). Allocation of buffers to serial production lines with bottlenecks. *IIE Transactions*, *28*(1), 18–29. https://doi.org/10.1080/07408179608966249

Rippe, C., & Kiesmüller, G. P. (2022). The repair kit problem with imperfect advance demand information. *European Journal of Operational Research*. https://doi.org/10.1016/j.ejor.2022.04.019

Ruifeng, C., & Subramaniam, V. (2012). Increasing production rate in kanban controlled assembly lines through preventive maintenance. *International Journal of Production Research*, *50*(4), 991–1008. https://doi.org/10.1080/00207543.2011.551844

Rustenburg, J. W., Van Houtum, G.-J., & Zijm, W. H. M. (2000). Spare parts management for technical systems: Resupply of spare parts under limited budgets. *IIE Transactions*, *32*(10), 1013–1026. https://doi.org/10.1023/A:1007676901673

Sachs, F., Helber, S., & Kiesmüller, G. (2022a). Evaluation of Unreliable Flow Lines with Limited Buffer Capacities and Spare Part Provisioning. *European Journal of Operational Research*, *302*(2), 544–559. https://doi.org/10.1016/j.ejor.2022.01.005

Sachs, F., Kiesmüller, G., & Helber, S. (2022b). *Design of Unreliable Flow Lines with Limited Buffer Capacities and Spare Part Provisioning* [Working Paper].

Shanthikumar, J. G., & Tien, C. C. (1983). An algorithmic solution to two-stage transfer lines with possible scrapping of units. *Management Science*, *29*(9), 1069–1086. https://doi.org/10.1287/mnsc.29.9.1069

Sherbrooke, C. C. (1968). Metric: A multi-echelon technique for recoverable item control. *Operations Research*, *16*(1), 122–141. https://doi.org/10.1287/opre.16.1.122

Sherbrooke, C. C. (2006). *Optimal inventory modeling of systems: Multi-echelon techniques* (Vol. 72). Springer Science & Business Media. https://doi.org/10.1007/b109856

Smets, L., Van Houtum, G.-J., & Langerak, F. (2012). Design for availability: A holistic approach to create value for manufacturers and customers of capital goods. *Journal of Systems Science and Engineering*, *21*, 403–421. https://doi.org/10.1007/s11518-012-5202-5

Song, J.-S., & Zhang, Y. (2020). Stock or print? impact of 3-d printing on spare parts logistics. *Management Science*, *66*(9), 3860–3878. https://doi.org/10.1287/mnsc.2019.3409

Soyster, A. L., Schmidt, J. W., & Rohrer, M. W. (1979). Allocation of buffer capacities for a class of fixed cycle production lines. *AIIE Transactions*, *11*(2), 140–146. https://doi.org/10.1080/05695557908974454

Spinellis, D. D., & Papadopoulos, C. T. (2000a). A simulated annealing approach for buffer allocation in reliable production lines. *Annals of Operations Research*, *93*(1/4), 373–384. https://doi.org/10.1023/A:1018984125703

Spinellis, D. D., & Papadopoulos, C. T. (2000b). Stochastic algorithms for buffer allocation in reliable production lines. *Mathematical Problems in Engineering*, *5*(6), 441–458. https://doi.org/10.1155/S1024123X99001180

Su, W., Xie, X., Li, J., & Zheng, L. (2016). Improving energy efficiency in bernoulli serial lines: An integrated model. *International Journal of Production Research*, *54*(11), 3414–3428. https://doi.org/10.1080/00207543.2016.1138152

Südbeck, I., Mindlina, J., Schnabel, A., & Helber, S. (2022). *Using recurrent neural networks for the performance analysis and optimization of stochastic milkrun-supplied flow lines* [Working paper].

Syntetos, A. A., Babai, M. Z., & Altay, N. (2012). On the demand distributions of spare parts. *International Journal of Production Research*, *50*(8), 2101–2117. https://doi.org/10.1080/00207543.2011.562561

Tan, B. (1997). Variance of the throughput of an n-station production line with no intermediate buffers and time dependent failures. *European Journal of Operational Research*, *101*(3), 560–576. https://doi.org/10.1016/S0377-2217(96)00191-9

Tan, B. (1998). Effects of variability on the due-time performance of a continuous materials flow production system in series. *International Journal of Production Economics*, *54*(1), 87–100. https://doi.org/10.1016/S0925-5273(97)00132-1

Tan, B. (1999). Asymptotic variance rate of the output of a transfer line with no buffer storage and cycle-dependent failures. *Mathematical and Computer Modelling*, *29*(7), 97–112. https://doi.org/10.1016/S0895-7177(99)00065-5

Tan, B., & Gershwin, S. B. (2009). Analysis of a general markovian two-stage continuous-flow production system with a finite buffer. *International Journal of Production Economics*, *120*(2), 327–339. https://doi.org/10.1016/j.ijpe.2008.05.022

Tempelmeier, H. (2003). Practical considerations in the optimization of flow production systems. *International Journal of Production Research*, *41*, 149–170. https://doi.org/10.1080/00207540210161641

Tolio, T. A. M., Matta, A., & Gershwin, S. B. (2002). Analysis of two-machine lines with multiple failure modes. *IIE Transactions*, *34*(1), 51–62. https://doi.org/10.1080/07408170208928849

Tolio, T. A. M., & Ratti, A. (2018). Performance evaluation of two-machine lines with generalized thresholds. *International Journal of Production Research*, *56*(1-2), 926–949. https://doi.org/10.1080/00207543.2017.1420922

Van der Duyn Schouten, F., & Vanneste, S. (1995). Maintenance optimization of a production system with buffer capacity. *European Journal of Operational Research*, *82*(2), 323–338. https://doi.org/10.1016/0377-2217(94)00267-G

Van Houtum, G.-J., & Kranenburg, B. (2015). *Spare parts inventory control under system availability constraints* (Vol. 227). Springer Science & Business Media. https://doi.org/10.1007/978-1-4899-7609-3

Van Rossum, G., & Drake, F. L. (2009). *Python 3 reference manual*. CreateSpace.

Vaughan, T. S. (2005). Failure replacement and preventive maintenance spare parts ordering policy. *European Journal of Operational Research*, *161*(1), 183–190. https://doi.org/10.1016/j.ejor.2003.06.026

Vidalis, M. I., & Papadopoulos, H. T. (1999). Markovian analysis of production lines with coxian-2 service times. *International Transactions in Operational Research*, *6*(5), 495–524. https://doi.org/10.1111/j.1475-3995.1999.tb00170.x

Wang, H. (2002). A survey of maintenance policies of deteriorating systems. *European Journal of Operational Research*, *139*(3), 469–489. https://doi.org/10.1016/S0377-2217(01)00197-7

Weckenborg, C., Kieckhäfer, K., Müller, C., Grunewald, M., & Spengler, T. S. (2019). Balancing of assembly lines with collaborative robots. *Business Research*, 1–40. https://doi.org/10.1007/s40685-019-0101-y

Weiss, S., Schwarz, J. A., & Stolletz, R. (2019). The buffer allocation problem in production lines: Formulations, solution methods, and instances. *IISE Transactions*, *51*(5), 456–485. https://doi.org/10.1080/24725854.2018.1442031

Weiss, S., & Stolletz, R. (2015). Buffer allocation in stochastic flow lines via sample-based optimization with initial bounds. *OR Spectrum*, *37*(4), 869–902. https://doi.org/10.1007/s00291-015-0393-z

Wijngaard, J. (1979). The effect of interstage buffer storage on the output of two unreliable production units in series, with different production rates. *AIIE Transactions*, *11*(1), 42–47. https://doi.org/10.1080/05695557908974399

Wong, H., Van Houtum, G.-J., Cattrysse, D., & Van Oudheusden, D. (2005). Simple, efficient heuristics for multi-item multi-location spare parts systems with lateral transshipments and waiting time constraints. *Journal of the Operational Research Society*, *56*(12), 1419–1430. https://doi.org/10.1057/palgrave.jors.2601952

Wong, H., Van Houtum, G.-J., Cattrysse, D., & Van Oudheusden, D. (2006). Multi-item spare parts systems with lateral transshipments and waiting time constraints. *European Journal of Operational Research*, *171*(3), 1071–1093. https://doi.org/10.1016/j.ejor.2005.01.018

Zhang, M., Pastore, E., Alfieri, A., & Matta, A. (2022). Buffer allocation problem in production flow lines: A new Benders-decomposition-based exact solution approach. *IISE Transactions*, *54*(5), 421–434. https://doi.org/10.1080/24725854.2021.1905195

Zimmermann, J., & Kiesmüller, G. P. (2019). *The influence of processing time variability on spare parts inventory and buffer size in an unreliable flow line* [Working Paper].