A multi-objective parameter calibration approach

Master's thesis

Industrial Engineering & Management

–

D. S. Belter

Master's thesis (MSc)

Industrial Engineering & Management
University of Twente, Enschede, the Netherlands

Title

A multi-objective parameter calibration approach

Author

Daniel Sebastian Belter

Student number

0166871

E-mail

d.s.belter@student.utwente.nl

Supervisors

Dr. ir. M.R.K. Mes, University of Twente
Dr. ir. J.M.J. Schutten, University of Twente
Ing. R. van der Zee, Vanderlande Industries B.V.

Date

18 July 2013

Abstract

This master thesis presents a diagnostic study on scalarized, multi-objective automatic calibration of the baggage batching process of a Vanderlande system at Schiphol South Terminal. Several simulation and emulation steps are necessary within the design and implementation of such a system. The problem we study is how to incorporate feedback from later implementation stages into initial simulation or emulation models that are based on a high abstraction level. Our main objective is to automate the calibration methodology for large-scale simulation models that have to be matched with realistic values from follow-up project stages or the implemented on-site system itself. This automatic calibration can help to reduce overall project lead-time and enhance the validity of simulations and emulations. We test the performance of several calibration methodologies according to their convergence speed and accuracy of parameter estimation in comparison to an initial parameter estimation based on histogram frequency matching. The test calibration methods are Random Search, Latin Hypercube Sampling, a Simulated Annealing adaptation, and a combination approach that merges Simulated Annealing with a Latin Hypercube (LHS-SA). All test approaches performed significantly better than the initial, manual parameter estimation. The singular use of our Simulated Annealing modification shows the best convergence characteristics of all tested calibration algorithms regarding both speed and accuracy. However, its cooling scheme is vital for a successful calibration attempt. It was not possible to prove that synergy effects of the LHS-SA method exist, but still we suspect them to be present. This might be due to the fact that the researched problem structure did not include many local minima. However, we cannot clearly verify this by our observations. Our Simulated Annealing adaptation has shown to be an effective automatic methodology for the calibration of multi-objective problems which include several tunable parameters.

Keywords

Simulation calibration, calibration automation, multivariate calibration, scalarization, parameter estimation, Latin Hypercube Sampling, Simulated Annealing adaptation, baggage handling system, baggage batching

Summary

In the beginning of this research, we identify a calibration problem within Vanderlande Industries. Its solution may translate into significant time and financial savings in the execution of an arbitrary company project. During the transition phases of such a project, a systematic calibration approach can help to make a difference by recurrently incorporating feedback of additional knowledge about the system.

In order to investigate more about the benefits of a useful automated calibration methodology, we examine the current situation and practice on calibration within Vanderlande. We noticed that often calibration attempts at Vanderlande are executed in a trial-and-error fashion. However, scientific literature shows that there are better ways to calibrate a system. In this research, we focus on the baggage batching process at the early-baggage-buffer (i.e., BagStore) at Schiphol South Terminal for further calibration testing. In particular, we look at the process of releasing early checked-in baggage from the baggage buffer to the in-cache lines near the make-up robots that automatically stack baggage into airplane load units (i.e., containers). We revise mainly the delivery quality of luggage. Therefore, we develop our own simplified simulation model of the implemented baggage handling system to demonstrate that even a rather abstract model can be calibrated decently towards data from practice. Moreover, we show that a systematic automatic calibration method will outperform manual attempts in this respect clearly.

After we specify our working point, we conduct a thorough literature study on calibration and related issues. In that section, we explain the distinction between verification, validation and calibration and what the need of the latter is. Moreover, we show how deviations between models or between a model and reality are quantified and compared and what their reasons of existence are. Also we elucidate the benefits and trade-offs of automatic, in comparison to manual, calibration. In addition, we elaborate on single and multivariate calibration approaches and follow the latter in our research. We then clarify a suitable mathematical problem formulation, explain general calibration limitations, and show what sense sensitivity analyses make in this regard. As for automatic calibration, we examine various combinatorial optimization schemes and illustrate why Simulated Annealing is a popular sequential approach used in calibration. Also Latin Hypercube Sampling is a widespread non-sequential method that appears to have positive effects on calibration performance. Due to this, we test both of the mentioned optimization algorithms for the situation at Vanderlande and also combine them to see if there exist any synergy effects when a non-sequential and a sequential approach are merged.

Thereafter, we conduct a data analysis on various calibration input measures for our anticipated simulation model, as well as on benchmark measures from reality to assess how well the applied calibration approaches perform regarding their goodness-of-fit. We further describe the calibration model set-up, its scope and the included assumptions we make. In Chapter 6, we explain our four test calibration approaches in detail and how we implement the model set-up into the Plant Simulation software package from Tecnomatix Technologies Ltd. Additionally, we carry out a sensitivity analysis on the objective function weights and describe the experimental set-up to assess algorithm convergence behavior with the usage of power regression.

Afterwards, we explain the results that we conduct with our calibration model. This includes scattering patterns of candidate solutions per test calibration approach, regression of algorithm convergence and the averaging of those regression formulations. To assess the statistical substance of our regression, we perform a relative error evaluation, in which all test calibration methods achieve a relative error of less than 20%. Finally, we choose our adaptation of Simulated Annealing as the preferred choice for automatic calibration.

For future research we suggest the incorporation of gradient surface information of the problem structure, e.g., by the usage of the Vandermonde method for multi-nonlinear regression.

Preface

When I started this master's thesis project around eight months ago, I realized that this would be my last big challenge as a student of the University of Twente – a period of my life that I really enjoyed with all its peaks and occasional downsides. Also, it is the conclusion of my graduation internship at Vanderlande Industries B.V., which gave me the extraordinary chance to prove myself scientifically and socially in a very exciting professional environment. It was a happy, but also very hard-working, period for me in that I learned a deeper meaning of the terms prioritization, out-of-the-box-thinking, and responsibility.

First of all, I want to deeply thank my company supervisor Mr. Remko van der Zee and the head of the Major Projects division Mr. Rudi Debets for taking me on board with this graduation project and the great opportunity, which this represents to me. In particular, Mr. van der Zee was always there for me when I needed support for my project and tried to help me as much as he could with his knowledge and experience.

Of course, also very sincere thanks go to my university supervisors Dr. Martijn Mes and Dr. Marco Schutten. In my opinion, both of them were excellent academic sparring partners, whose professional opinion I personally value a lot. I was glad to work with them on this project and hope they will also keep me in good memories as an enthusiastic student.

Furthermore, I want to give thanks to my beloved wife Moon-hee, who always is an inspiration to me. Being totally alone executing this kind of a project can be quite a though experience on someone's mind. In that sense, I am endlessly happy that she always supported me in these, sometimes hard, times with all her positivity and love. In addition, I want to mention my friends who also helped me substantially with this thesis: Arturo Perez Rivera, Harm Hoeksema, Lara López and Mariana Goldak. Due to some of their creative ideas I was able to quickly escape several moments of doubts and hesitation. Also, I very deeply want to thank my parents for always supporting me, not just now during the conduction of my thesis, but throughout all my studies and far beyond.

This graduation is not the end, it is just the beginning!

Sincerely,

Daniel Belter

List of abbreviations & Glossary

Throughout the text body, words written in "*italic*" can be found in this glossary.

Abbreviation	*Brief description*
70MB project	Major project at Schiphol Airport to connect all piers and separated baggage handling areas with each other to be capable to handle 70 million passengers per year
BagStore	Baggage storage unit at Schiphol South terminal that temporarily stores early checked-in bags of passengers and thus functions as a bag buffer
BHS	Baggage Handling System
CDF	Cumulative Density Function
Flow test	A flow test is a end-of-project capacity test of a BHS to show the customer that negotiated baggage handling performance can be yielded within a certain amount of time
GOF	Goodness-of-fit: A measure that statistically describes the difference between observed and expected values
IE&M	Industrial Engineering & Management
In-cache line	Conveyor lines in front of any make-up robot to buffer arriving baggage from the BagStore
LHS	Latin Hypercube Sampling
LU	Load unit: airplane container
Make-up robot	A robot that automatically can load baggage into a container load unit
MSc	Master of Science
PD file	Project Definition file: An evolving digital record that contains crucial project information (e.g., system layout, equipment capacities, other constraints)
PDF	Probability Density function
PLC	Programmable Logic Controller: specified purpose computer designed for multiple input and output arrangements for lower level equipment control
RS	Random Search
SA	Simulated Annealing (and its adaptations)
VI	Vanderlande Industries B.V.

Table of Contents

ABSTRACT ..... **I**

KEYWORDS ..... **I**

SUMMARY ..... **II**

PREFACE ..... **III**

LIST OF ABBREVIATIONS & GLOSSARY ..... **IV**

CHAPTER 1: INTRODUCTION **1**

1.1 RESEARCH MOTIVATION 1
1.2 PROBLEM IDENTIFICATION 1
1.3 RESEARCH SCOPE 3
1.4 RESEARCH GOAL 4
1.5 PROBLEM STATEMENT 4
1.6 RESEARCH QUESTIONS 4
1.7 CONTRIBUTIONS 5

CHAPTER 2: PROBLEM ANALYSIS **6**

2.1 VANDERLANDE INDUSTRIES B.V. 6
2.2 CURRENT SITUATION 8
2.3 BAGGAGE BATCHING 10
2.3.1 Batching process implementation in practice 11
2.3.2 Abstraction deviations of the batching process in practice and simulation 14
2.4 DESIRED SITUATION 14
2.5 DISCUSSION 14

CHAPTER 3: LITERATURE REVIEW **15**

3.1 DISTINCTION BETWEEN MODEL VERIFICATION, VALIDATION AND CALIBRATION 15
3.2 DEVIATION MEASUREMENTS FOR CALIBRATION 16
3.3 CONCEPTS OF MODEL CALIBRATION 18
3.4 CALIBRATION PROBLEM FORMULATION 20
3.5 MANUAL AND SEMI-AUTOMATIC CALIBRATION 21
3.6 AUTOMATIC CALIBRATION 22
3.7 SEARCH SCHEMES USED IN AUTOMATIC CALIBRATION 24
3.8 SENSITIVITY ANALYSIS PRIOR TO CALIBRATION 26
3.9 DISCUSSION 27

CHAPTER 4: DATA ANALYSIS **28**

4.1 SIMULATION MODEL INPUT 28
4.1.1 Batching hours per day 29
4.1.2 Batch composition / batch request inter-arrival time 29
4.1.3 Batch size 31
4.1.4 Destination robot allocation 31
4.1.5 Dispatch delay of a batch after batch composition 31
4.1.6 Travel times 32
4.1.7 Equipment capacity 34

4.2 CALIBRATION MODEL BENCHMARK DATA 34
4.2.1 Inter-arrival time between batches at in-cache lines 34
4.2.2 In-system time per batch 35
4.3 DISCUSSION 36

CHAPTER 5: MODEL SET-UP 37

5.1 SCOPE & LEVEL OF DETAIL 37
5.2 TYPE OF SIMULATION MODEL 38
5.3 ASSUMPTIONS 38
5.4 PROCESS FLOW WITHIN THE MODEL 39
5.5 OBJECTIVE FUNCTION & CONSTRAINTS 40
5.6 MINIMUM NUMBER OF MODEL REPLICATIONS 41
5.7 DISCUSSION 41

CHAPTER 6: SOLUTION APPROACH & EXPERIMENTATION 42

6.1 AUTOMATIC CALIBRATION APPROACHES 42
6.2 MODEL IMPLEMENTATION 44
6.3 VERIFICATION & VALIDATION 45
6.4 SENSITIVITY ANALYSIS 47
6.5 EXPERIMENTAL SET-UP 47
6.6 DISCUSSION 48

CHAPTER 7: RESULTS & FINDINGS 49

7.1 COMPARISON OF CALIBRATION METHODOLOGIES 49
7.1.1 Computational calibration results 49
7.1.2 Scattering behavior of candidate solutions 50
7.1.3 Convergence speed & quality 53
7.2 DISCUSSION 57

CHAPTER 8: CONCLUSIONS & RECOMMENDATIONS 58

8.1 CONCLUSIONS 58
8.2 RECOMMENDATIONS 59

REFERENCES ..... A

APPENDIX ..... J

SUBJECT INDEX ..... FF

Chapter 1: Introduction

The aim of this research is to improve the transition process between conceptual planning and actual implementation of Major Projects within Vanderlande Industries. To this end, we strive to automate the calibration methodology for large-scale simulation models that have to be matched with realistic values from follow-up project stages or the implemented on-site system itself. Eventually, this can help to decrease overall project lead-time and to enhance the validity of simulations and emulations with which additional sensitivity analyses for later process optimization can be conducted. In the following sections we elaborate on the key framework aspects of this research. We divide this chapter into seven sections: research motivation (Section 1.1), problem identification (Section 1.2), research scope (Section 1.3), research goal (Section 1.4), problem statement (Section 1.5), research questions (Section 1.6) and the contributions that we expect to deliver (Section 1.7).

1.1 Research motivation

This study is initiated by the "integration group" within the Major Projects engineering department of the Baggage Handling division[*] at Vanderlande Industries[†] and is a graduation assignment for the Master of Science program of "Industrial Engineering & Management" at the University of Twente. We were approached by the integration group – which is the final problem owner[‡] – to decrease cost and time efforts spent during the transition of project phases that occur during the implementation of a mid- or large-sized baggage handling system. In Section 1.2, we identify the main problem and brake it down into various underlying causes to find the most relevant problems that are connected to the observed key issue.

1.2 Problem identification

We determine the main problem that we will solve as:

> **The necessity for a decrease in lead-time throughout the final implementation stages of major projects, especially focusing on the on-site realization of large-scale baggage handling systems at airports, e.g., at Amsterdam Schiphol Airport.**

Figure 1 elucidates the classification and "localization" of the core problems. Essentially, this graph intends to establish a cause-and-effect relationship among issues that are relevant for the main problem, which the problem owner experiences. As stated, the reduction of lead-time of a generic major project is the main issue we focus on. We leave higher level implications of this problem out of consideration, since they fall outside of the scope of this research, which we explain in the next section. Nevertheless, we can retrieve those implications from Appendix A4.

In order to be able to focus on lower level implications of the above-mentioned main problem, we look into the organization starting from the scope of a general project (see Appendix A2). First of all, a good coordination of the realization on-site will eventually decrease financial and temporal efforts spent. In case of the practical realization, this is mainly due to a low number of *flow or site-acceptance tests* conducted at the end of a the implementation phase – these are stress tests of the built system to show that customer requirements can be met for extraordinary high volume instances. Fast and fault free hardware, as well as, software implementation on-site mainly realizes this. The latter of both we follow further. Again, for a lean software implementation, a fast estimation of important parameter settings is necessary - in particular during the transition period

[*] See Appendix A1 for more details about company and departmental specifics.
[†] We provide an outline of the host company in the next chapter.
[‡] Appendix A3 shows the definition of a "problem owner" as well as an outline of the research methodology.

among several project phases. This incurs that parameter settings can be adjusted correctly if an organized information feedback loop for achieved and planned system outcomes exits.

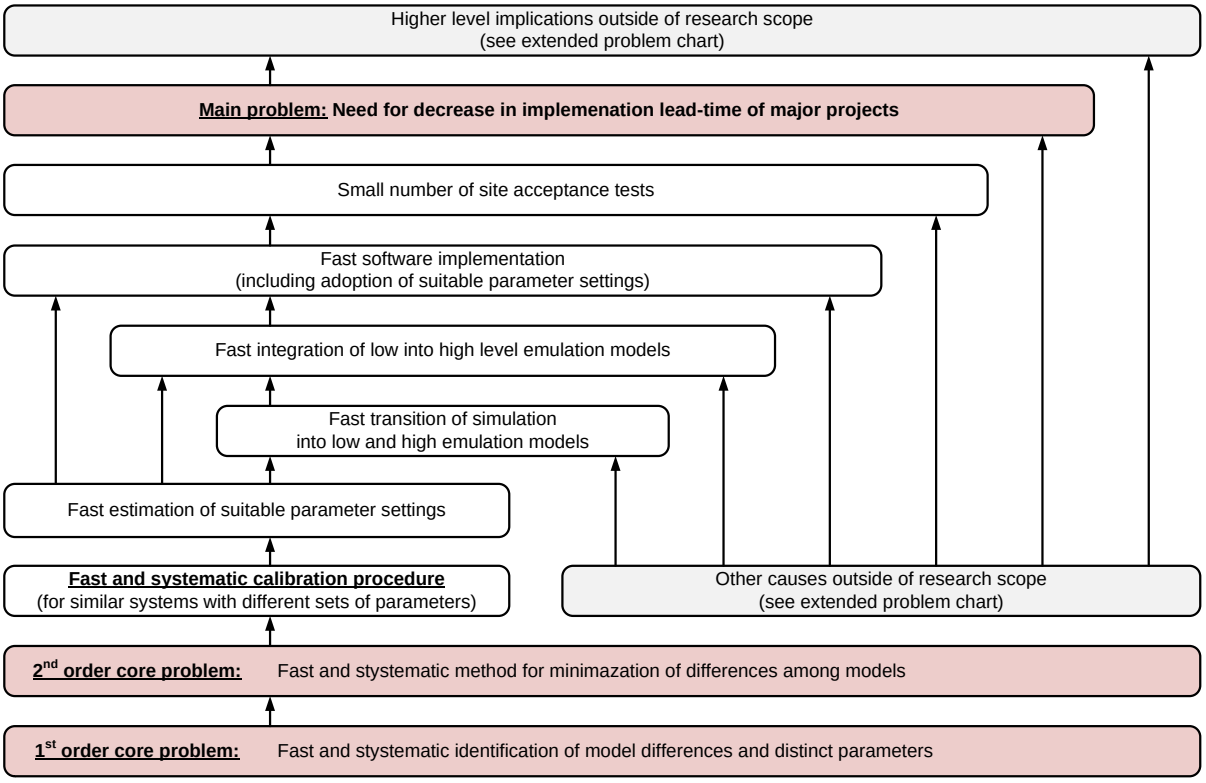


Figure 1: Problem chart

At Vanderlande baggage handling systems, performance tests are executed a-priori in two different ways: simulation and emulation. With simulation, they assess the achievement of promised capacity requirements the fictional system setup. In contrary, emulation tests actual system behavior with implemented Java code and the Programmable Logic Controller units (PLCs) that will be placed on-site later on. Further, emulation is split up into high and low level test scenarios. In the high level version, the logistical logic of the baggage handling is reviewed, while the low level emulation solely focuses on the testing of task instruction and report transfer of PLCs.

Taking this into account, an adequate software realization additionally depends on a fast and successful transition from system simulation and emulation test models[§] into the real-life system. However, this is tributary on a good integration of low level emulation into high level emulation, as well as an appropriate transition of the simulation model into both emulation model types. Thus conclusively, there is a need for a systematic calibration methodology which can quickly identify and decrease deviations among different types of models or with the actual system.

Also other factors exist that can have significant influence on the software implementation or on its underlying aspects. However, we exclude these from this research, since they are not feasible within the timeframe given, or there exist deeper layers of acting problems whose exploration is too complex. For instance, extensive validation procedures, component development and realistic requirement assessment are part of those excluded factors (see Appendix A4).

[§] Emulation = simulation of test instances with parts of the already realized system or with its individual components in-house at Vanderlande Industries.

Nevertheless, we believe that we can influence the main problem of the integration group significantly by a fast and systematic calibration procedure that initially recognizes inconsistencies in system performance of distinct models or the implemented system and then further minimize those deviations. Due to this, we chose to treat the two core problems visible in Figure 1 in-depth.

1.3 Research scope

The main constraining factors for this study are the temporal restrictions regarding the project horizon. This significantly influences the degree of realization, respectively implementation, our research outcomes. Moreover, the main focus of this research is the automation of the calibration method of the baggage batching process which is implemented at Schiphol South, since this endeavor is generally considered non-trivial to solve. Due to the fact that suitable tuning of the parameters related to the batching algorithm is difficult to execute in the currently applied trial-and-error method, we strive to automate this process as much as possible. In the next paragraphs, we provide a brief outline of the basic idea behind the anticipated calibration methodology.

The baggage batching process and its calibration

We develop a suitable automatic calibration method for the baggage handling system of which we can retrieve the layout in Figure 5, p. 10. Especially, the baggage-delivery-performance-matching on the routes from the early-baggage-storage-unit, also called *"BagStore"*, and the baggage make-up robots[**] is of interest for Vanderlande Industries and, in particular, the integration group. An in-depth description of the batching process and its implementation in practice and in simulation is given in the next chapter.

A challenging issue regarding the calibration of this process is that the algorithms implemented in simulation and reality use noticeably distinct process steps and different parameter sets to imitate the same anticipated batching behavior. Thus, there exists a gap among the level of abstraction of the simulation and the real process. This is the reason why so far it is unclear to Vanderlande how to systematically match related system outcomes that are supposed to be nearly equal to each other.

The basic thought behind the anticipated calibration method is to generate similar process output from identical input scenarios in case of systems with differentiated parameter sets. "Similar output" does not mean identical though. Nevertheless, simulation results should meet a certain range of outcome in comparison to later-stage emulation model counterparts or practice. An acceptable range, or Goodness-of-fit (*GOF*) we specify later on. It is clear that one particular parameter setting cannot always generate similar output in comparison to other models or the actual system under the condition of identical input. Since, both simulation and reality have variability which cannot be exactly matched, there will always be certain uncertainty in comparison. In between computational models, a variance reduction technique called "Common Random Numbers" (Law, 2007) can be applied to decrease this gap, since the "same" randomness is used in the models that are compared. This implies that experiments can be carried out under the same "environmental conditions" such that other factors will be accountable for differences. However, between any type of model and reality, this can only be approached to a certain degree. It is our aim to find a calibration method that generates an acceptable Goodness-of-fit under the assumption that representative samples of observations have been taken from reality for calibration matching.

[**] A make-up robot is a device that is capable of automatically stacking pieces of luggage that sequentially arrive from a conveyor belt into a designated aircraft container, which is also called Load Unit (LU)

1.4 Research goal

Based on the problem chart (see Figure 1), we formulate the following research goal:

> **Reducing temporal and financial efforts of Vanderlande Industries with regards to the currently applied calibration procedure of major projects, focusing upon high level system behavior through identifying system performance deviations and their calibration optimization**

There is a need for a systematic approach that reduces efforts spent during the "realize on-site" phase (Vanderlande Industries B.V., 2012f) – which includes lead-time reduction. The main focus is put on "high-level" system behavior, which refers to in the logistical process layer of a system, excluding the embedded controls and hardware components. Those are part of the low level system management.

1.5 Problem statement

From the prior-mentioned research goal, we derive the following problem statement:

> **How can a structured methodology be applied to the calibration process within Vanderlande Industries so that it is capable of identifying, measuring, and minimizing performance differences among subsequent system models and practice automatically in a computationally inexpensive manner?**

Especially, we focus on the incorporation of feedback from later project phases and its integration into system models of earlier process stages. The identification and optimization of deviations between the simulated and the implemented system stands central.

1.6 Research questions

To answer the problem statement, we have to formulate several research questions. The chapter structure of this thesis follows the order of the research questions below, since, overall, the answer of a prior question delivers the input that is necessary to answer its successor.

Q1: What is the current calibration process for baggage batching at Vanderlande Industries?

- **a. What are the Key Performance Indicators (KPIs) that are relevant for calibration?**
- **b. How is parameter calibration currently executed?**

First, we assess the current situation regarding calibration of the baggage batching process. The goal of answering this question is to determine the gap between the current and the desired situation. Also, we try to discover important key performance indicators for later calibration.

Q2: What is currently known in the literature about automated model calibration?

- **a. What are suitable practices to identify relevant deviations between simulated and real systems?**
- **b. What optimization methods are appropriate for minimizing output deviations between simulated and real systems incurring distinct sets of tunable parameters under the condition of identical input?**

We have to clarify what is known in the literature about the analysis of distinct simulated and real system output and what the most suitable methods are for automated calibration optimization and output matching. Based on this, we can create a method fitting to the context of Vanderlande Industries.

Q3: What suitable calibration methodologies can be developed from the practices known in the literature to fit the problem at hand?

After the revision of all relevant literature and proven techniques in the field of automated model calibration, we develop a calibration approach suitable for the baggage batching procedure that is implemented at Schiphol South terminal.

Q4: How well does the developed calibration method perform in a representative test case?

We test the proposed calibration methodology suitable test cases. Moreover, we assess whether and how the anticipated calibration procedure achieves its indented purpose of adequately matching model and system performance.

1.7 Contributions

We strive to discover generic coherences that help to minimize calibration efforts spent in realizing large multi-stage projects in practice. We use a case study from which we draw conclusions for more general cases. Thus, "generalizability" receives a significant value within this research – also with considerations for a possible scientific publication of this study. In addition, we elaborate on specific recommendations for Vanderlande Industries. Scientifically, this research contributes to the available literature about automated calibration optimization for processes that are heavily dependent on evolving simulations in their design stage and which are supposed to incorporate feedback from later project phases. Also, we provide in-depth insights on convergence characteristics of the calibration algorithms that we test, which can be used for further researches in the field of calibration automation.

Chapter 2: Problem analysis

We outline the problem context in this chapter. This includes that we describe the organization Vanderlande Industries B.V. briefly (Section 2.1), as well as the baggage handling integration group, which is the research host. Subsequently, we elaborate on the current problem situation (Section 2.2) and the baggage batching process (Section 2.3) in detail. Thereafter, we present our view on the desired situation (Section 2.4) and conclude essential findings of this chapter in a brief discussion (Section 2.5).

2.1 Vanderlande Industries B.V.

In the next paragraphs, we introduce Vanderlande Industries, headquartered in Veghel, The Netherlands. Afterwards, we give some insights about the Major Projects Baggage Handling Systems division and relevant processes.

Public perception

Vanderlande Industries B.V. is a large mechanical engineering corporation that is specialized in manufacturing automation solutions for warehouse, parcel and postal automation, as well as sophisticated baggage handling systems for small and large airports. The contemporary departmental structure of the company can be found in Appendix A1. Currently, Vanderlande employs 2,300 people worldwide (Vanderlande Industries B.V., 2012a). Even though the company is headquartered in the Netherlands, it has stationed customer centers around the globe to provide fast hands-on service.

Project management

The work that is done within Vanderlande Industries is mainly based on unique projects, which urges the organization to be highly flexible with respect to its resources. Figure 2 depicts the main process steps of an arbitrary project within Vanderlande Industries.

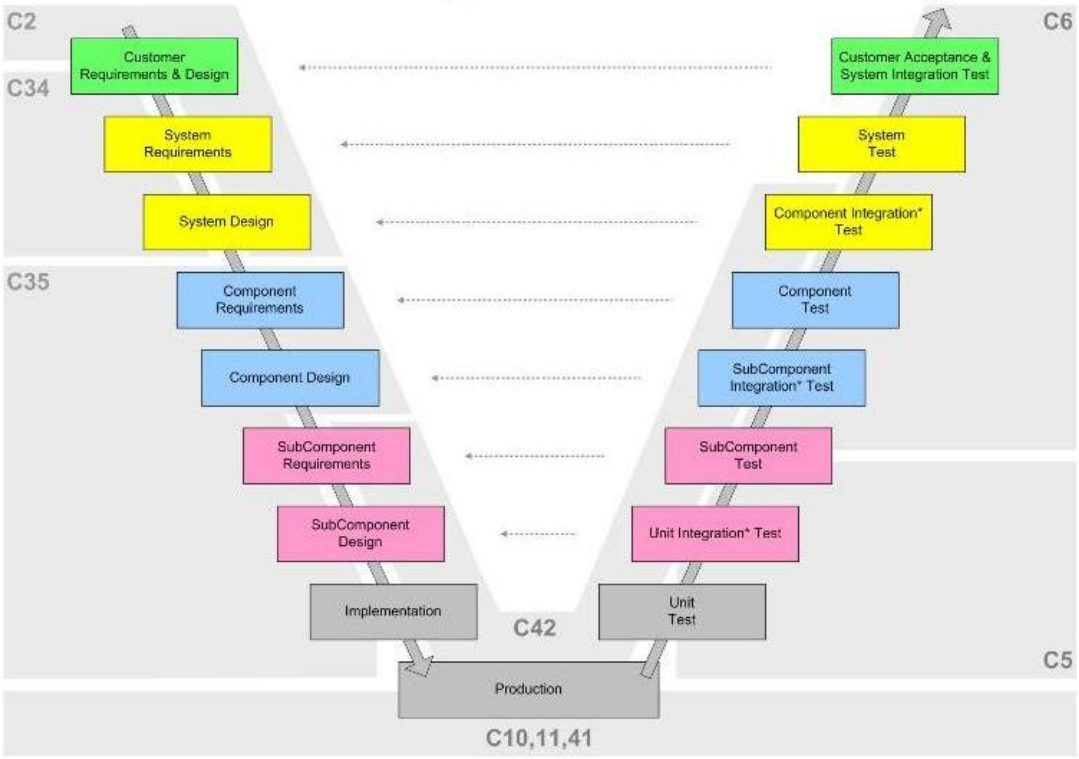


Figure 2: Phases of project management within Vanderlande Industries

After a successful sales phase, project management at Vanderlande occurs in several phases. This phase division is derived from the "engineering V-model" (see Figure 2) that is often used in iterative technical processes. The V-model thus actually implies the existence and usage of feedback loops throughout the entire project process – however this is not self-evident for all involved processes during the engineering and implementation. In general, a divisional structure works well if transition of knowledge is done systematically and standardized – however that is not always the case in practice. Unintentionally, mistakes often happen in this kind of transition. Here the main issue seems to be that distinct functional groups with their own internal goals and mindsets are working on an evolving product, which impedes the avoidance of "hand-over errors" and the gain of knowledge throughout the course of time – in this case we consider the evolution from simulation models to low and high level emulations towards the actual system integration on-site.

Research-relevant processes & Definitions

In order to clarify what knowledge transitions comprise, **Figure 3** visualizes the detailed test process flow in the development and implementation phase of a baggage handling system.

After knowing the basic requirements of a baggage handling system, the simulation group can start to build a simulation model, which considers the rough system layout and the adherence of minimum capacities (see the first two left-hand blocks in Figure 2). The basic requirements, as well as the simulation model and outcomes, are saved in a Project Definition file (PD file).

When lower level emulations are finished successfully, the tested components are again emulated, respectively "imitated", at the overall system level (see the blocks "Component Test", "Component Integration Test" and "System Test" in Figure 2). This is done for the testing of the logical layer of the baggage handling system, as e.g., the process step controller or the routing controller.

This logical layer has two inputs: the imitated technical component behavior, which is also referred as "PLC stubs", and the rough-cut model from the simulation, which is extracted from the Project Definition file. This assessment level is critical, since here the main preparations regarding the on-site flow tests are considered and finally defined.

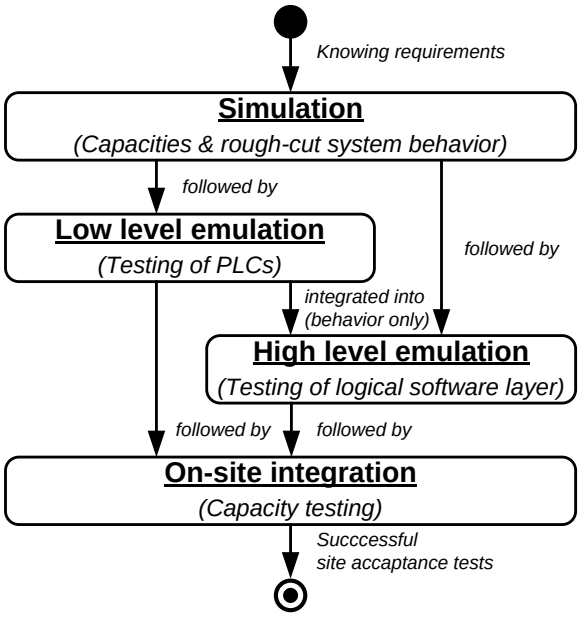


Figure 3: Engineering and implementation phases of major projects

While the logical layer is tested in-house, the hardware components are put into place on-site. Then, software and related updates are installed. Now the most difficult part of the implementation phase starts, the actual configuration of the overall system behavior. This normally consumes a lot of time and financial efforts – partly due to the fact that the installment of the final system is performed mostly during running airport operations. This issue makes a trial-and-error testing approach uncomfortable. Thus, towards the end of a project, many testing activities are re-located from in-house to on-site setups. The main activities in that respect are baggage flow tests, where the achievement of certain capacities and other objectives is evaluated in a pass-or-fail fashion. If these tests are conducted successfully, the project development and implementation phase finishes and the project is delivered to the customer. Thus, there exists an issue on how to improve the described multi-stage process transition, so that the efforts eventually spent on-site are minimized.

2.2 Current situation

Now we elaborate on the current situation regarding the calibration of the baggage batching process at Schiphol South terminal. By doing this, we intend primarily to answer research question Q1.

Current calibration practice

We notice from earlier company internal studies (Lith *et al.*, 2012, Thoonen *et al.*, 2012) and from several interviews with personnel involved in system implementation that, e.g., the *70MB*[††] and similar projects struggle with deadline compliance due to a lack of fast and successful calibration attempts. This issue occurs often in the transition from simulation towards high level emulation and later on again in the transition from high level emulation to on-site integration. Therefore, the actual implementation phase is lengthened unnecessarily. We suspect that this fact is caused by the lack of information feedback loops after the design phase of an arbitrary project (see Figure 4), as well as by difficulties in the comparison of system models on a multi-dimensional level which includes distinct sets of tunable parameters. This is due to issues in the identification and assessment of deviations in system performances, which normally accumulate throughout the development phases. Thus, a methodology that can clarify these differences between anticipated and realized system behavior, and that furthermore can minimize those deviations is necessary.

As outlined before, Vanderlande Industries follows specific project development and implementation guidelines. The central project flow is surrounded by existing, as well as non-existing feedback loops. However, a substantial amount of "translation errors" is likely to occur during the process of information transfer. Since this transfer happens many times throughout an arbitrary project, the amount of errors accumulate "downstream" in identification of high-level system performance deviations and their calibration optimization time – if no intervention is performed. Initially, this is the reason why feedback loops are installed within an implementation procedure so that useful intermediate outcomes of earlier project phases are adapted with the knowledge gained in later stages. This should prevent that mistakes are passed on to later project phases where they become increasingly costly to be fixed.

Vanderlande Industries already tries to establish feedback loops between subsequent project phases. However these endeavors partially remained uncompleted. Most systematic feedback loops exist currently between simulation modeling and the required fit of the system buyers and, additionally, for the overall project evaluation. However, the feedback loops between simulation and low / high level emulation, between both types of emulation models, between emulation and on-site integration and, finally, between simulation and on-site integration are still not a practiced standard (see Figure 4).

[††] Major project at Schiphol Airport to connect all piers and separated baggage handling areas with each other

We argue that the inclusion of such additional information loops by the means of a smart calibration methodology will decrease the system deviations, which accumulate throughout the delivery of several model versions. This can have added value for a reduction of lead-time – from which mainly the integration group would benefit, since on-site implementation and testing is one of its core responsibilities. In general, the project principal has a kind of calibration method in mind that functions as a tool that can detect inconsistencies of algorithm behavior and adapts these to a minimum. Thus, it is important to not just recognize where deviations occur, but also, in particular, how to minimize those deviations effectively.

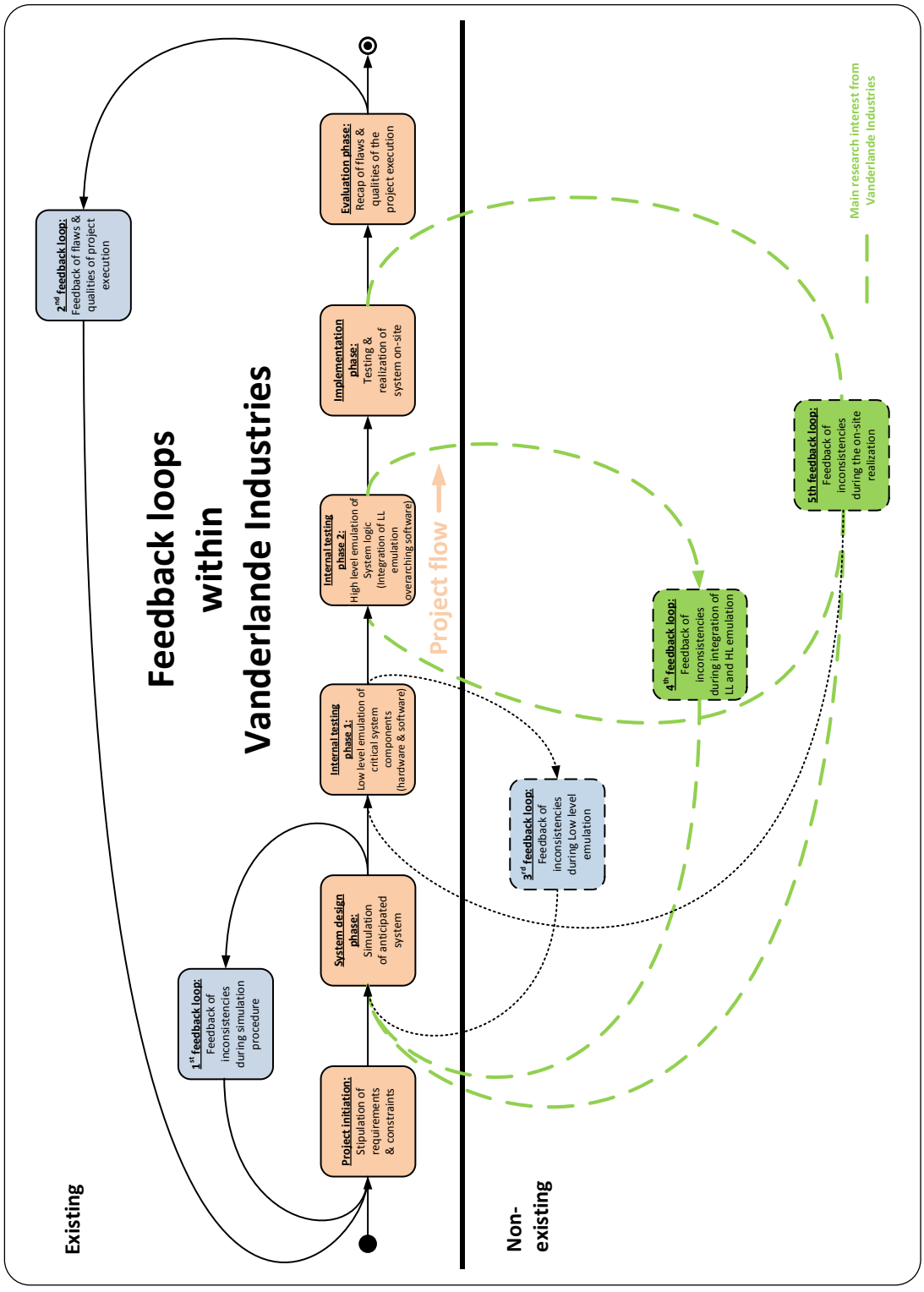


Figure 4: Feedback loops within Vanderlande

2.3 Baggage batching

We focus on the calibration feedback loop between implemented system and a high abstraction simulation model of the baggage batching process at the Schiphol South terminal. Figure 5 shows the functional layout of the batching environment (Vanderlande Industries B.V., 2012d). Explanations follow in Section 2.3.1.

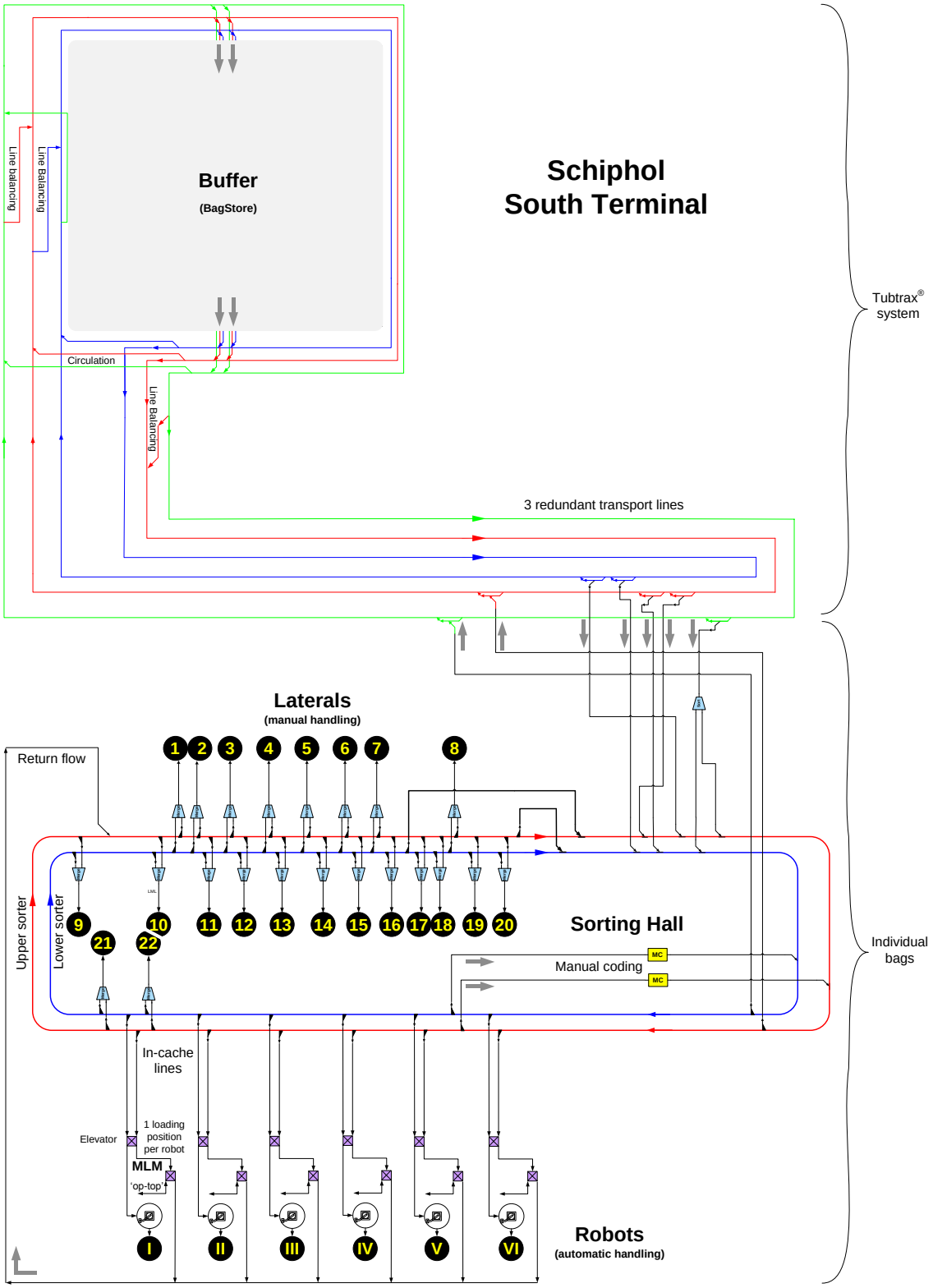


Figure 5: Operational layout of the baggage batching process

2.3.1 Batching process implementation in practice

Prior to the batching process, early check-in of pieces of luggage are stored in the BagStore, which can be understood as a large baggage buffer. In case a flight schedule demands that bags are batched from this buffer, the designated make-up robots need to have a load unit available (normally an airplane container) to finally comply to the flight schedule request for initiating a batch. However, also a couple of other system components are involved in this process. If all of those sub systems comply to the batch request, the actual batching and transport process can start. The real baggage batching process strongly relates to the simulated process. However, due to the adaptations throughout the course of project implementation a couple of differences are incorporated in the algorithms used. We outline the practice of the on-site baggage batching in the following paragraphs. As a visualization help on the system components, we refer to Figure 6.

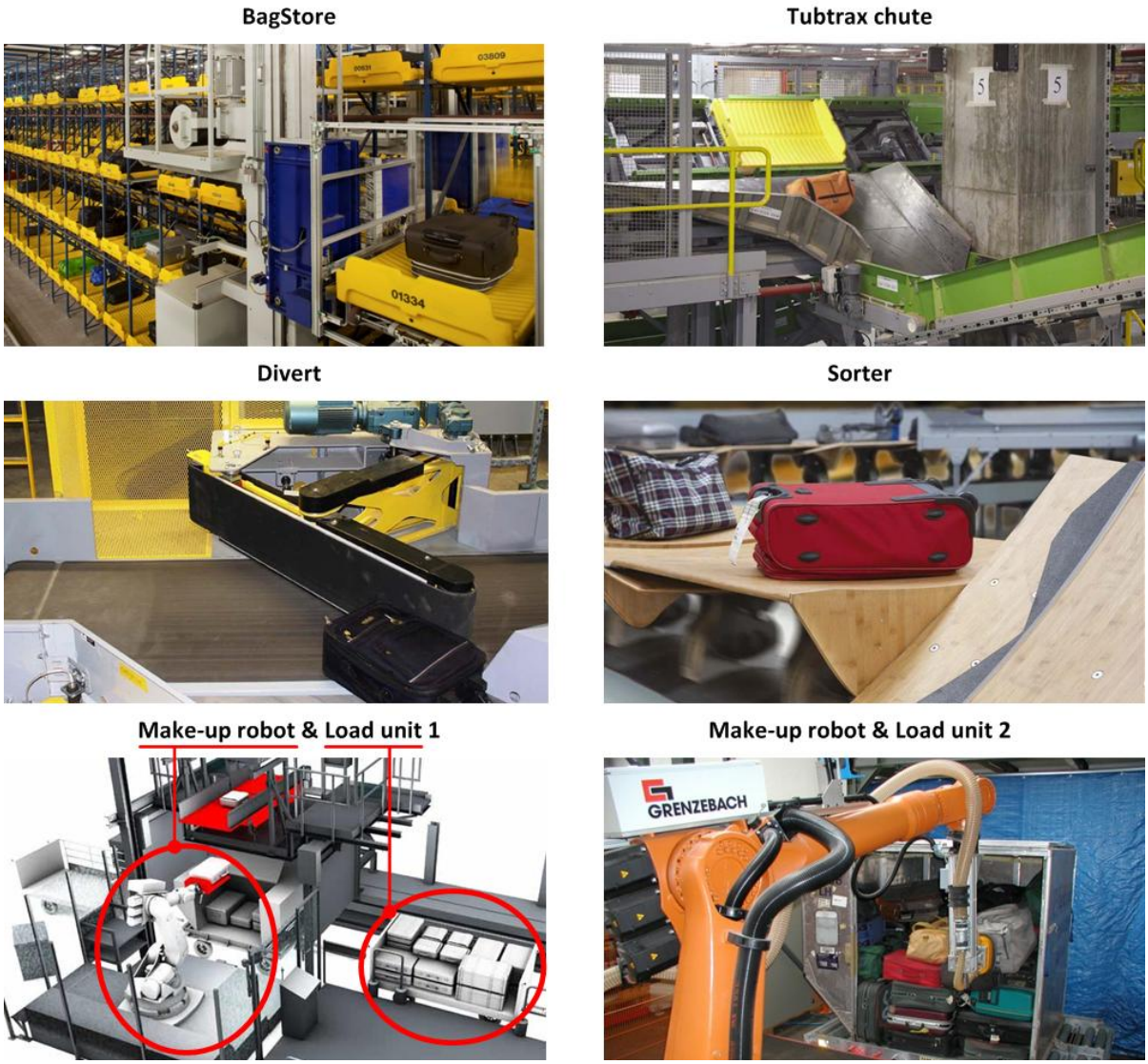


Figure 6: Illustration of several components of the baggage handling system at Schiphol South Terminal

Process description

Van der Meulen (2010) describes the implemented baggage batching process in detail. For more information on the actual batching process the reader is referred to this document and the references therein. Overall, the implemented baggage batching process is controlled by five different sub process managers, i.e., software applications, which eventually are bundled in the Batch Composition Manager (BCM, see Figure 7).

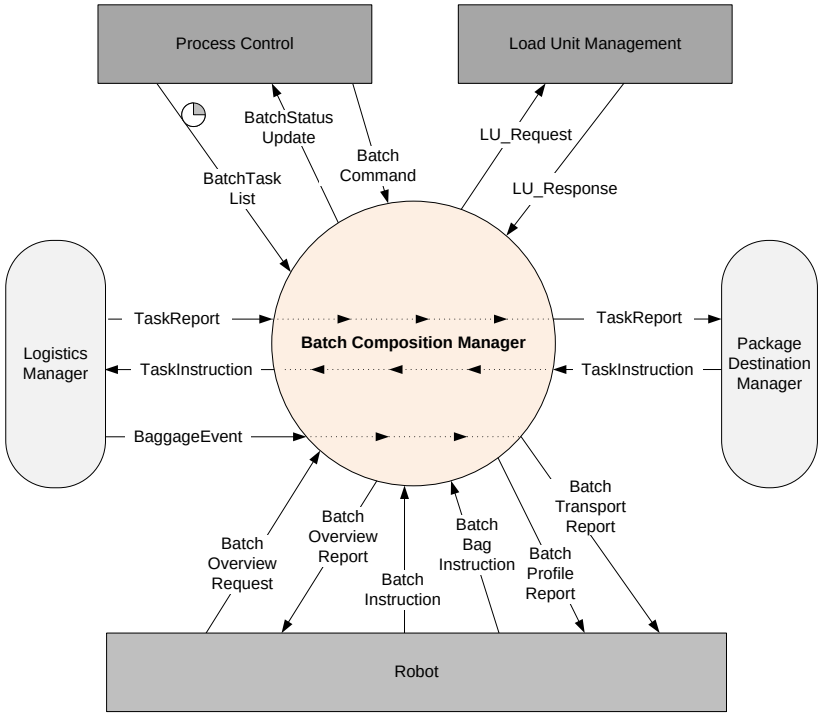


Figure 7: Software application environment of baggage batching by Van der Meulen (2010)

As we can see above, the decision making process on baggage batching is rather complex. The package destination manager is the first entity that requests the initiation of batches, since it is directly linked to the information from the flight schedule. If this request has entered the BCM, the Robot has to be verified as ready for batching – also a load unit has to be connected to it. Then, a batch can be released and the Logistic Manager takes over to guide the bags within a batch throughout the inter-connected transport lines at the Schiphol South terminal. Figure 8 roughly shows how the baggage batching process is executed in practice.

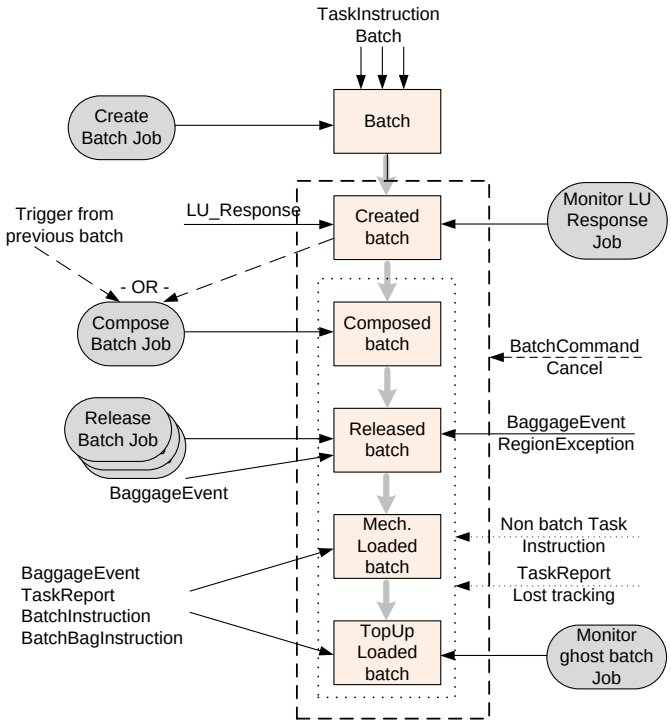


Figure 8: Implemented baggage batching process, taken from Van der Meulen (2010)

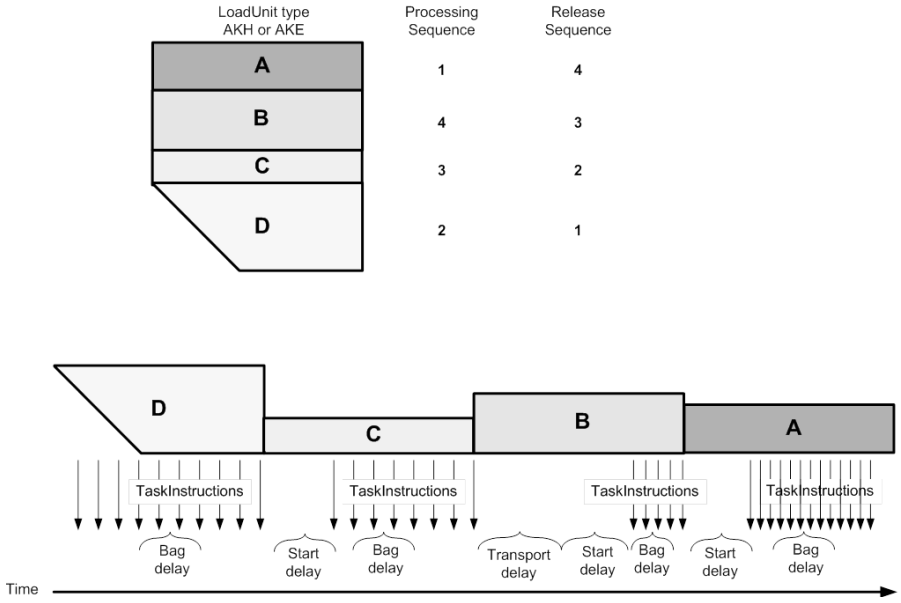


Figure 9: Batch release by subgroups, taken from Van der Meulen (2010)

An important aspect of the implemented batching process is the division and release of sub-groups of a batch. In Figure 9 we illustrate the splitting of a load unit (LU), which is normally an airplane container (upper trapezoid-like shape). The LU is filled with up to four different types of luggage. This subgroup division is done by the criteria baggage weight and volume. The bag selection algorithm applied to determine the right composition of bags within each subgroups is based on the idea of the Simulated Annealing algorithm (Kirkpatrick *et al.*, 1983). The main features per subgroup are shown in Table 1.

Table 1: Subgroup division and planned start and release delays, taken from van der Meulen (2010)

Subgroup	Description	Aim	Processing sequence	Release sequence
A	Most light, ordered by volume (descending)	The most light bags at the top. These bags can be loaded manually.	1	4
B	Remaining ordered by volume (descending)	The remaining bags. The bags which are not selected by filter A, D or C.	4	3
C	Most heavy, ordered by weight (descending)	The most heavy bags on top of a smooth layer	3	2
D	Average height, ordered by weight (descending)	A smooth layer of bags of average height on the bottom, the most heavy bags at the bottom	2	1

Initially, the total weight and volume of all eligible bags for one batch is determined. This is necessary to fit the batch base to an approximate bell-shaped curve of the normal distribution. The subgroup division algorithm then assigns those potential bags into a suitable subgroup category. As can be seen in Table 1, average bags according their volume and size are released first, followed by the heaviest bags in the batch. Thereafter, residual bags, thus those without a special volume or weight feature, are cleared from the BagStore, while the lightest bags finish the charge of a load unit.

For further reference, the detailed control and component environment the batching process is embedded into can be found in Appendix A4. In addition, the detailed physical progression of the on-site implemented batching process can be viewed in Appendix A5.

2.3.2 Abstraction deviations of the batching process in practice and simulation

One of the reasons why calibration is a non-trivial endeavor, is the difference in abstraction that is incorporated in later process stages in comparison to the initial simulation setup. We outline these deviations briefly.

The detailed simulation setup of the BagStore and the connected baggage batching process at Schiphol South are specified in McMenamin and de Jongh (2006) and related documents, as e.g., in Thoonen *et al.* (2012).

Process description of baggage batching in simulation

Baggage batching starts after the storage of early checked-in bags based on available flight load files provided by the system buyer. Checked-in bags are divided into various categories from which only "Buffer bags" and "Batch bags" are eligible for batching. The difference among these two types is that the "Batch bags" are designated for the batching by make-up robots, while "Buffer bags" are batch-wise sent to laterals where they are loaded manually. Other types of bags, such as oversized bags or those lost in tracking, are also considered in the simulation, but they do not play a significant role for the baggage batching procedure itself.

All in all, the simulated batching process appears to be similar to the implemented one, but, for instance, the subgroup division of batches is not considered. Furthermore, only automatic batching is possible, while in the practice application also operators can chose to handle batches manually or even in a semi-automatic manner. Measurements of volume and weight are also not taken into account.

Summary key differences in abstraction level

The main differences basically are that in the simulated system no subgroup processing is taken into account and that batch releases are only steered automatically. Also bag attributes such as dimensions and weight are not considered for the performance in the simulation model. For more details about this aspect, a summarizing graph of these deviations can be found in Appendix A8.

2.4 Desired situation

We strive to develop a calibration method that enables us to incorporate feedback information from the realized system quickly into a simplified simulation model that is based on a rather high abstraction level. Thus a simulation does not have all the options included that are available in later stages. Nevertheless, we believe that such a simulation can be tuned towards realistic behavior with benchmark information from the implemented system. Of course, often surrogates or combination parameters have to be used in that kind of simulation to approach the anticipated behavior as close as possible.

2.5 Discussion

We elaborated on the current situation at Vanderlande Industries regarding the baggage batching process at Schiphol South terminal. Simulations and the actually implemented processes incur a distinct level of abstraction which causes their final system behavior to be different from each other. The main distinctions in abstraction were outlined and the implemented process was described. With this knowledge, we can start to create our own simulation model in which we can test various calibration approaches systematically.

Chapter 3: Literature review

In this chapter, we discuss relevant literature for our research project and related works in the field of automatic calibration. In Section 3.1 we make a distinction between the definitions of verification, validation and calibration. Later on we explain how model deviations in comparison to benchmark data are in fact measured (Section 3.2) and which solution concepts exist for calibration (Section 3.3). Then we elucidate common mathematical problem formulations for calibration (Section 3.4) and, later, show distinguishing features of manual, semi-automatic (Section 3.5) and automatic calibration approaches (Section 3.6). We further explain optimization schemes that exist to decrease deviations between models and benchmark data (Section 3.7). We then elaborate on the importance of sensitivity analysis for calibration modeling (Section 3.8) and conclude this chapter with a short discussion (Section 3.9) on essential findings in the literature that we incorporate into our research.

3.1 Distinction between model verification, validation and calibration

To create a realistic (simulation) model, Law (2007) suggests the use of the concepts "verification" and "validation", which he put into a linear order. By executing such a sequence correctly, the model can gain "credibility". Mazzotti and Vinci (2007) points out that validation and calibration are essential processes for the creation of reliable models. Li *et al.* (2008) and Madsen (2003) state that validation and calibration are recurrent and thus iterative problems, which are seldom solved in a linear way. Since there exist different views on how to create realistic models, we first review the basic ideas behind verification, validation and calibration and how these approaches are related with each other. There exist many works which can be used as an in-depth reference about issues concerning verification and validation.[‡‡]

Verification is the process of ensuring that the computer program of the planned model and its implementation are done correctly (Sargent, 2008, Schelsinger *et al.*, 1979) or, according to Law (2007), that the assumptions document of the concept model is correctly translated into a computer program – which related to debugging the simulation program. Thus with verification, mainly the technical realization of the model is put central. This does not include any attempt yet to assess whether the implemented process matches with the one from practice.

Validation, on the other hand, is defined by Law (2007) and Fishman and Kiviat (1968) as the process of determining whether a simulation model is an accurate representation of the real system, to achieve particular objectives of a study. The difference between validation and calibration is that validation has the objective to confirm whether the computer model precisely represents the real process for a couple of critical instances, while calibration aims to adjust the unknown input parameters by comparing the computer model output with the real observed data (Yuan and Szu Hui, 2013). Additional information about the relationship between validation and calibration can be found in Oberkampf and Roy (2010) and the references therein.

Regarding model calibration, several definitions exist in the literature. Li *et al.* (2010) and Van Griensven and Bauwens (2005) define it as the process of adjusting parameters until model outputs are sufficiently similar to observed values. Rykiel (1996) states similarly that calibration is the adjustment of parameters and constants to improve a model towards reality. In this respect, it is important to mention that the input parameters that are used for calibration are more conceptual. Thus often they cannot be determined directly from data (Bekele and Nicklow, 2007, Madsen, 2003). Generally, one looks at the tuning of "unobservable parameters" that are

[‡‡] For example, Balci (1998), Banks *et al.* (2005), Carson (1986, 2002), Feltner and Weiner (1985), Law (2005, 2007), Naylor and Finger (1967),Oberkampf and Roy (2010) , Sargent (2004), Shannon (1975) and Van Horn (1971)

critical for the matching for the model performance to empirical data (Vanni *et al.*, 2010, Weinstein, 2006). Also the calibration of assumptions is suitable for calibration (Vanni *et al.*, 2010). The initial goal of calibration is to find an optimal, and thus an unique setting, of the parameters that are calibrated, which maximizes the fit between the model and the actual system (Moore and Doherty, 2006). However, this goal of a unique optimal setting for parameter values comes with an important trade-off, which we discus later in this chapter.

3.2 Deviation measurements for calibration

There are various reasons known for the deviation of model output in comparison to actual data measurements. Madsen (2000) mentions the most prominent of them:

(1) Errors in input data.
(2) Errors in recorded observations: Experimental results could be wrong due to set-up, process, or measurement errors (Byers *et al.*, 2002)
(3) Errors and simplifications inherent in the model structure, that do not adequately describe physical realities (Byers *et al.*, 2002)
(4) Errors due to the use of non-optimal parameter values.

In model calibration, only error source (4) should be minimized. However, the calibration of model parameters in general can compensate for the other error sources as well (Madsen, 2000). Beck (1991) concludes, that a calibrated model incurs collected knowledge about the system studied. Therefore, calibration does not only aim to find parameter settings that minimizes a given objective function. This is not an easy task since the models are frequently nonlinear (Kuczera (1997). Calibration also aims for reduction of uncertainties of parameter values (Gaume *et al.*, 1998). The need and the importance of calibration is recognized in many practical models, e.g., nuclear radiation release (Kennedy and O'Hagan, 2001), hydrologic (Kanso *et al.*, 2006), and biological models (Henderson *et al.*, 2009), as is stated by Yuan and Szu Hui (2013). Similarly to Vanderlande Industries, the "predictive accuracy" (Campbell, 2006) of the models is essential. Furthermore, the use of calibration is valuable in other ways as well, e.g., the reusability of large-scale simulations for changing simulated environments (Huang *et al.*, 2010). In that case calibration can be used as a sort of "feedback loop" (Campbell, 2006) that adapts simulation parameters to the new situation. As a benefit, temporal and financial efforts for simulations remain limited in comparison the initiation of entirely new modeling projects. Vanni *et al.* (2010) give another advantage of using model calibration to estimate the parameters of the model, the estimation process will induce correlation between the parameter estimates – this can be particularly useful if the correlation among parameters is not known yet and if this is up for identification. This can be interesting for situations where non-linearity among parameters plays a significant role. However, other researches argue that correlation should be assessed in advance to any calibration attempt as much as possible by a thorough sensitivity analysis (Skahill and Doherty, 2006). Systematic methods on the conduction of such a sensitivity analysis are described later in this chapter.

How to compare outcome deviations of models and the real system

To determine the performance deviations among the measures of the Key Performance Indicators (KPIs) of the model and the comparing system, the concept of the "goodness-of-fit" is applied. Next, the most common deviation measures are introduced briefly. The work of Janssen and Heuberger (1995) can be used for a more detailed elaboration on performance measures for comparing model predictions and observations using different goodness-of-fit objective functions. Commonly, in calibration or output matching, identical twin experiments are used to evaluate the goodness-of-fit, e.g., in Harmon and Challenor (1997). This type of experimental set-up is comparable to the "correlated inspection approach" described by Law (2007). Hereby the

model is exposed to exactly the same input as the real system, such that output variance that is caused by randomness incurred in the data input is minimized as much as possible.

Several techniques to assess the goodness-of-fit between two sample data series exist. A subjective, but fast, manner of assessing the goodness-of-fit between a model and the actual system is the use of a Turing test (Kleijnen, 1995, Sargent, 1996). Here, a model receives "face validity" if experts perceive its results congruent with the actual system behavior (Law, 2007).

Graphical techniques to compare data samples are also useful and comprehensive to visually explore the obtained and comparison data and to search for differences among it (Balci, 1998, Montgomery and Runger, 2002). Furthermore, statistical methods and derived aggregation methods are used in practice. Graphical methods which are commonly known are histograms, density plots, and divers frequency comparisons (Sargent, 1996). Furthermore, also graphical comparison methods are used that rely on the visualization of cumulative distribution functions instead of the density distribution function, such as the Distribution-Function-Differences plot (Law, 2007). The main disadvantage of those techniques is, however, that they are not reliable as indicators for the distribution observed, unless the sample size that is used as a comparison base is large enough (Law, 2007). Moreover, a couple of other plots are available from the literature that amplify the differences among distinct distributions, e.g., the Q-Q plot and the P-P plot (Gibbons and Chakraborti, 2003, Law, 2007).

Next to graphical methods, there are also numerous statistical data comparison methods known in the literature, which, to a vast extent, we can find in the works of Balci (1998), Janssen and Heuberger (1995), Law (2007). Comparing the outcome means of two samples can be done with a two sample t-test, as for instance in Kong *et al.* (2009). Such a t-test can be divided into an independent test or a paired test (Larsen and Marx, 2006, Law, 2007). According to Larsen and Marx (2006), it can be more suitable to use the paired test over the independent test for parameter estimation. In the case of calibration, where the same input is used in both the real system and the model, observations can be paired per input instance. One of the most common and general statistical tests used to compare observed and expected data points is Pearson's chi-square test (Pearson, 1900). The chi-square test compares frequency counts of histogram intervals. Law (2007) states that this test is useful for small and mid-scale sample sizes. However, when the number of observations becomes large, the chi-square test almost always rejects H_0 (Gibbons, 1985). This is due to the fact that normally variance is decreased when the number of experimental trials increase, thus the confidence intervals of the model and the system output shrink as well. As a result, model and expected output "intersects" less, and the chi-square statistic rejects the null-hypothesis. The chi-square test, similar to the Anderson-Darling test, mainly focuses on the tail differences of the compared distributions (Law, 2007).

Another approach to assess the matching potential of data series is the Kolmogorov–Smirnov test ("K-S test") developed by Kolmogorov (1933) and Smirnov (1948). The K-S test does not compare histograms such as the chi-square test, but the maximum distance between the cumulative distribution functions of the empirical and the hypothesized observations (Law, 2007). According to Stephens (1974), the K-S test is more powerful than the chi-square test, since the significance of the test statistic is independent of the sample size. Also the K-S test focuses more on the deviations in the middle of the distributions and less on those detected in the tails (Law, 2007).

For more insights into other statistical tests that can be used to compare observed and expected values, respectively the means of two samples, the reader is referred to Larsen and Marx (2006), Law (2007) or Balci (1998) and the references therein.

There are some difficulties connected to statistical, respectively time series, techniques. Normally, it must be assumed that the assessment data is stationary. In a stationary process, the mean, the variance and the related auto-correlation is stable over time. As a consequence, conclusions drawn from these test can be invalid, if the conditions connected to auto-correlation and stationarity are not satisfied. However, if this assumption is categorically disqualified, only the graphical methods can be used to judge the fit of the observed and the expected values (Law, 2007).

Other test statistics that are common as objective functions in calibration studies are the Mean Absolute Error (MAE), the Mean Squared Error (MSE), the Root Mean Square Error (RMSE), the Mean Percentage Error (MPE), deviation coefficients, such as the Nash–Sutcliffe efficiency coefficient (NSE) or the log-transformation error (Yu and Yang, 2000). The formula of the RSME is depicted below:

$$Root\ Mean\ Squared\ Error\ (RMSE) = \sqrt{\sum_{i=1}^{n}(Observed_i - Expected_i)^2/n}\ , \tag{1}$$

where n is the total amount of observations and i is an individual observation. "$Observed_i$" refers to the output retrieved from the simulation model, while "$Expected_i$" stands for the benchmark measure taken from a later model version or, even better, from practice.

The MSE, as well as the RMSE and the MPE, are linear scores that incur that all measured differences are equally weighted based on the total number of measurements (Hyndman and Koehler, 2006). Frequently applied comparison functions in the field of automatic calibration are the MSE[§§] and the RSME[***] (Yu and Yang, 2000).

All of the above-mentioned objective functions are a good measure of accuracy, but only to compare forecasting errors of different models for a particular variable and not between variables, as it is scale-dependent (Hyndman and Koehler, 2006). Furthermore, they are error aggregations that give the errors in prediction a single measurement of predictive power. The RSME and MSE in comparison to the MAE attach a relatively big weight to errors of large magnitude. This is due to the fact that the error term is squared before the average is taken. In conclusion, MSE and RMSE are powerful if large errors are avoided in a model. This is probably one of the main reasons why RSME and MSE are often chosen for in the field of automatic calibration. In particular the RSME is useful, since it presents the statistical error in the same unit as the initial Key Performance Indicator (KPI) and therefore, can be interpreted as the extent of a "typical" error made by the model (Hyndman and Koehler, 2006).

3.3 Concepts of model calibration

Model calibration in general can be divided single and multivariate calibration, which considers the handling of one or several matching objectives. Furthermore a division can be made into three different calibration approaches: manual, semi-automatic and automatic. At first the typology based on the amount of matching objectives is elucidated; thereafter the implications due to the degree of automation are explained.

[§§] For example, Chiu *et al.* (2010), Gupta *et al.* (2009), Ito *et al.* (2010), Kong *et al.* (2009), Moussu *et al.* (2011), Pokhrel *et al.* (2012), Rode *et al.* (2007), Yum and Lee (1991)

[***] For example, Bekele and Nicklow (2007), Graefe *et al.* (2005), Liu and Liu (2011), Liu and Sun (2010), Madsen (2000, 2003), Wagener and Wheater (2006a)

Single- and multivariate calibration

In model calibration, one can try to match the value of a single goodness-of-fit objective function, which is called single-variate or single-objective calibration. Alternatively, it can be aimed to match several objective functions to comparison data, which is then called multivariate calibration (Gupta *et al.*, 1998).

In the literature, various examples of single-objective calibration can be found, e.g., in Bendall and Skinner (1998), Jagner *et al.* (1993) and Nash and Sutcliffe (1970). The advantages of performing single objective calibration are obvious: it is relatively easy to implement a single measure comparison to an existing model and it is quite explicit about the found near-optimal value, since there are no trade-offs involved that are due to conflicting goodness-of-fit objective functions (Madsen, 2003). On the contrary, single-objective calibration incurs explicit drawbacks: Li *et al.* (2010), Liu and Sun (2010), Madsen (2000), Yapo *et al.* (1998) conclude that any single-objective function, no matter how carefully chosen, may not adequately measure the ways in which a model fails to match the important characteristics of the observed data. This ultimately led to the recent focus on multivariate calibration approaches. Also, there are doubts about the uniqueness of optimal parameter settings that can be found by a calibration procedure as insinuated earlier in this chapter. Moore and Doherty (2006) already stated that no matter which regularization methodology is employed, the inevitable consequence of its use is a loss of detail in the calibrated field. It is likely that no unique set of parameter values exists that generates a significantly good fit of the model and the system for every possible system scenario. To avoid this, the optimization problem can be formulated as a multi-objective calibration problem that tries to fit a near optimal Pareto front of solutions (Moore and Doherty, 2006).

With regards to multivariate calibration, many example studies[†††] are available as well. The works of Gupta *et al.* (1998) and Yapo *et al.* (1998) and their consideration for multi-objective model matching shifted the focus for calibration in the hydrologist field of science, where a vast amount of automatic calibration approaches had been studied so far. Multivariate calibration has a number of advantages above single-objective calibration that are discussed in several publications:

- Multiple criteria can be accommodated in a representative way (Geng *et al.*, 2011).
- Instead of giving a single 'optimum' design, which may not provide a good balance between multiple criteria, multi-objective methods lead to a set of designs that are diverse and non-dominated to each other. Thus, a set of optimum designs in the Pareto sense can be generated (Geng *et al.*, 2011).
- It allows for simultaneous optimization of even conflicting goodness-of-fit objectives (Madsen, 2003).
- In most cases the simultaneous use of more than one model output variable can improve 'parameter identifiability'" (Gupta *et al.*, 1998).

Nevertheless, multivariate calibration also has several shortcomings:

- Up to now, multiple objective heuristics are hard to solve in a reasonable amount of time due to the enormous dimensions of solution space. Therefore, most of the studies that consider multivariate calibration approaches only include up to two goodness-of-fit objectives (Geng *et al.*, 2011).
- Higher dimensional objective calibration is hard to display comprehensively in a graphically way (Geng *et al.*, 2011).
- It is more complex to implement than single-objective heuristics (Abraham and Goldberg, 2005). Aggregation approaches to combine multiple objectives into a single-variate heuristics, such as in Kong *et al.* (2009), or pure single-objective calibration are a less complex alternative (Geng *et al.*, 2011).

[†††] Bekele and Nicklow (2007), Li *et al.* (2010), Liu (2009), Madsen (2000, 2003), Moussu *et al.* (2011), Paulo *et al.* (2012), Rode *et al.* (2007), Shrestha and Rode (2008), Yan and Haan (1991), Yu and Yang (2000)

Two different approaches exist in multivariate calibration on how to handle various goodness-of-fit objectives: objective aggregation (Madsen, 2000, 2003, Van Griensven and Bauwens, 2003, 2005) and Pareto solution fronts[‡‡‡]. Also attempts are made to combine both manners into a fuzzy approach, where aggregation is performed after the full multi-objective optimization run to maintain the diversity of the non-dominated solution of the Pareto front (Shrestha and Rode, 2008). However, this is only partially successful.

Regarding the aggregation of several objectives, "scalarization" is a popular concept used. Hereby, a multi-objective calibration problem is transformed into a single objective problem by applying an aggregate weighting scheme to the distinct objective measurements (Madsen, 2000). This is done to avoid the selection complexity present in a Pareto front. The mathematical formulation of such as scheme is shown in Section 3.4. The major drawback of scalarization is that significant information about tradeoff characteristics might be lost and that the search space is not fully evaluated (Singh *et al.*, 2004). Also the relative importance of various objectives might not be fully considered in the calibration process (Yuan and Szu Hui, 2013).

The Pareto front approach is based on the concept of "Equifinality" (Beven and Freer, 2001), which incurs that a set of decision variables that are non-dominated within the search space generate the Pareto optimal set (Duan *et al.*, 1992, Goldberg, 1989, Zitzler and Thiele, 1998). This typically causes a long valley of optimal solutions (Skahill and Doherty, 2006). As a consequence, a typical multi-objective optimization problem produces a set of solutions which are superior to the rest of the solutions, but are inferior to other solutions in one or more objectives (Liu, 2009). Reasons for the emergence of the Pareto front are the non-linearity of objective functions with several local extremes (Skahill and Doherty, 2006). This is amplified through the inclusion of model and data errors that distort realistic behavior of the modeled system (Pokhrel *et al.*, 2012). Moreover, the Pareto set of optimal solutions from a multi-objective calibration increases as more objective functions are included in calibration (Shrestha and Rode, 2008). Ultimately, this leads to a decision making problem that modelers have to face to select a preferred solution from numerous Pareto optimal sets (Khu and Madsen, 2005). This is the case, since it is basically never clear whether the global optimum is the best fit or not (Skahill and Doherty, 2006). Khu *et al.* (2006) present a preference ordering approach for the generation of a limited number of optimal solutions, which are Pareto-efficient in the individual subsets of objectives. However, in a multi-objective problem with conflicting objectives, it may not always be possible to find a solution, which is Pareto-efficient for all individual objectives (Shrestha and Rode, 2008). Another issue is that in practice, the complete Pareto set may be computationally too expensive to calculate, and only parts of the Pareto optimal solutions might be interesting (Madsen, 2003). Thus, no optimal approach has be found yet to select the best solution parameter vector out of a non-dominated Pareto front that includes contradicting objective functions. Nevertheless, if a near-optimal Pareto front is obtained, a consecutive manual calibration step can be simplified to a large extent (Pokhrel *et al.*, 2012).

3.4 Calibration problem formulation

If the calibration problem is treated with step-wise parameters that are individually constrained by lower and upper bounds, we consider it as a combinatorial optimization problem (*COP*). In this case, all the possible combinations of parameter settings are limited in total. Two well-known COPs are the Travelling Salesman Problem (TSP) and the Knapsack Problem. The main issue with COPs is that the number of options to evaluate the solution space is increasing exponentially with the number of adjustable variables that are introduced (Schrijver, 2003). If the parameter values vary in non-restricted continuous solution space, the conditions of the COP are not met anymore, since the possible evaluation options grow to infinity.

[‡‡‡] For example, Bekele and Nicklow (2007), Beven (1993), Beven and Binley (1992), Geng *et al.* (2011), Li *et al.* (2010), Moussu *et al.* (2011)

In general, as for single-objective calibration, the mathematical problem formulation is written in Formula (2) (Yu and Yang, 2000, Yum and Lee, 1991).

$$\min_{\theta \in \Theta} \{F_{GOF}(\theta)\}, \tag{2}$$

where $F_{GOF}(\theta)$ is the objective function (e.g., RSME). θ stands for the parameter settings that are variable; however, these settings come from the set Θ, which is restrained by the feasible parameter ranges. The parameter space is normally defined as a hypercube with upper and lower limits for each tunable parameter that is considered for calibration. Madsen (2003) states that these limits are chosen according to physical and mathematical constraints, information about the physical characteristics of the system and from modeling experience. The multivariate calibration problem is formulated slightly different, which we can see from Equation (3) (Gupta et al., 1998).

$$\min_{\theta \in \Theta} \{F_{GOF,i}(\theta)\}, \tag{3}$$

where, $F_{GOF,i}(\theta)$ are the different objective functions for the each key performance indication i (i.e. KPI_i). Again θ are the variable parameter settings which are constrained to the feasible parameter solution space Θ (Gupta *et al.*, 1998, Li *et al.*, 2010).

The solution of Equation (3) in general, will not be a single unique set of parameters but consists of the Pareto set of solutions (Madsen, 2003). As explained earlier, this is due to the various trade-offs between the different objectives (Gupta *et al.*, 1998, Madsen, 2000). If objective aggregation, such as scalarization, is incorporated into the multivariate calibration, the problem formulation is adapted as follows (Kong *et al.*, 2009, Madsen, 2003):

$$\min_{\theta \in \Theta} \{\sum_i w_i * F_{GOF,i}(\theta)\}, \tag{4}$$

where, $F_{GOF,i}(\theta)$, θ and Θ is similarly defined as in Equations (2) and (3). However, in this formulation, the deviation measures are summed up to a grand total to merge the all optimization measures towards a single point of reference. Additionally, the weight w_i is introduced. It represents the weight or importance scale that is attached to each goodness-of-fit objective function. These weights should reflect the measurement uncertainties, and the correlation between the measurements. Therefore, smaller weights should be given to more uncertain measurements and to clusters of measurement points. Objective aggregation is decreasing the implementation complexity of the calibration. This, however can cause similar disadvantages as in single-objective calibration. Consequently, if aggregation of objective functions is done, it ought to be performed carefully, so that important KPIs are not under nor overrepresented in the applied calibration method.

In the next section, different manners of calibration are discussed to solve the above-mentioned problem formulations. Basically, calibration can be executed manually, automatically, but of course also semi-automatically.

3.5 Manual and semi-automatic calibration

Manual calibration is generally considered as trial-and error parameter adjustments, where the goodness-of-fit of the calibrated model is often based on a visual judgment by comparing the simulated and the observed measurements (Madsen, 2000, White, 1995) or by regression and statistical techniques (Wagener and Wheater, 2006b). Thus, it is tried to match the model response to historical input–output data in an rather arbitrary manner (Liu and Sun, 2010). Various examples of manual calibration studies can be found in the liter-

ature.[§§§] Manual calibration of large simulation models became increasingly unpopular in the last decades, since it incurs numerous drawbacks. For instance, many researches showed that this type of calibration is:

- Time-consuming if the searchable parameter space is large[****]
- Subjective, since it is done in general quite opportunistic rather than systematic (Gilson *et al.*, 2011, Jie *et al.*, 2012, Ndiritu, 2009, Straatman *et al.*, 2004)
- Needs extensive training to be workable (Boyle *et al.*, 2000)
- Not easy to transfer to another modeler or another model (Boyle *et al.*, 2000)
- Likely to ignore specific data points based on first impressions (Ndiritu, 2009)
- More prone to generate suboptimal parameter sets than automatic methods (Ndiritu, 2009)
- A larger number of interacting parameters can have unpredictable effects when multiple parameters are adjusted manually (Bekele and Nicklow, 2007, Gupta *et al.*, 1999, Vanni *et al.*, 2010)

On the other hand, the main advantage of manual calibration is that more natural values for parameters are chosen, since the modeler has to have an in-depth understanding of the model at hand, which can be different with automated calibration, since its optimization algorithm solely obeys to the predefined objective function. This might lead to parameter settings at the edge of their range feasibility (Ndiritu, 2009). Semi-automatic calibration studies, e.g., Pokhrel *et al.* (2012) or Spear (1997), try to combine this strong point of manual calibration with automatic procedures to match objectivity with the in-depth calibration insights (Ndiritu, 2009). However, most of the recent calibration studies aim for a fully automated approach.

3.6 Automatic calibration

From the early 1970s, recommendations (Ayres and Stamper, 1995) are made to calibrate models based on measured data. Automatic calibration (*AC*) has been researched in various scientific fields, e.g., in traffic engineering (Huang *et al.*, 2010), hydrology[††††], civil engineering (Liu and Liu, 2011), lithography (Byers *et al.*, 2002), health care (Kong *et al.*, 2009), biology (Ito *et al.*, 2010, Rose *et al.*, 2007, Straatman *et al.*, 2004), sensor technology (Geng *et al.*, 2011) and in mathematics and statistics (Agyei and Hatfield, 2006, Kanungo and Zheng, 2004, Li *et al.*, 2008, Yuan and Szu Hui, 2013).

Liu (2009), Liu *et al.* (2004) and Madsen (2000, 2003) summarize that in automatic calibration, parameters are adjusted automatically according to a specific search scheme for optimization of certain calibration criteria (objective functions, respectively numerical measures of the goodness-of-fit). This process is repeated until a specified stop criterion is satisfied, e.g., maximum number of model evaluations, convergence of the objective functions, or convergence of the parameter set. There are various advantages of automatic calibration:

- Elimination of one source of arbitrariness in modeler decisions by standardization (Rose *et al.*, 2007)
- Estimation of parameters within an acceptable time (Vanni *et al.*, 2010), thus faster parameter search convergence (Bekele and Nicklow, 2007, Liu, 2009, Madsen, 2000)
- Reduction of human bias in the parameter search, thus higher objectivity (Bekele and Nicklow, 2007, Liu, 2009, Vanni *et al.*, 2010)

[§§§] For example, Byers *et al.* (2002), Campbell (2006), Hughes (2004), Hughes *et al.* (2007), Ito *et al.* (2010), Jie *et al.* (2012), Liu and Liu (2011), Mwelwa (2004)

[****] Gilson *et al.* (2011), Madsen (2000), Rode *et al.* (2007), Straatman *et al.* (2004), White (1995)

[††††] For example, Bekele and Nicklow (2007), Jiang *et al.* (2013), Li *et al.* (2010), Liu (2009), Liu and Sun (2010), Madsen (2000, 2003), Moussu *et al.* (2011), Ndiritu (2009), Rode *et al.* (2007), Shrestha and Rode (2008), Skahill and Doherty (2006), Yu and Yang (2000)

- Better predictive performance as AC fits better to the real process with more accurate unknown calibration parameter values (Yuan and Szu Hui, 2013)
- Relatively easy to implement (Jiang *et al.*, 2013, Liu, 2009)
- Hidden combinations of parameter settings might be explored, since more solution space is searched than with manual calibration (Ndiritu, 2009)
- Reproducibility by other modelers (Straatman *et al.*, 2004)

On the other hand, there are several disadvantages and trade-offs known regarding automatic calibration:

- Often chosen parameter values are near the extreme of reasonable values, since automated methods try to 'squeeze' the right outcome into the simulation model. This is the reason why ranges of parameters need to exist. So far no approaches have been developed on how to incorporate such qualitative information into an AC method (Ito *et al.*, 2010, Kuroda and Kishi, 2003, Rose *et al.*, 2007). Concerns are often cited that automatic calibration fails to attach physical reality to the parameters and the resulting modeling may therefore not make sense (Ndiritu, 2009). Thus Key Performance Indicators (KPIs) and influencing parameters and their ranges have to be selected with great care. (Rose *et al.*, 2007)
- As for multivariate automatic calibration, aggregation of objective functions, e.g., by scalarization, introduces yet another manner of arbitrary calibration which was to be avoided initially (Rose *et al.*, 2007). But also choosing a preferred optimal parameter setting from the near-optimal, multi-dimensional Pareto front is rather subjective (Cooper *et al.*, 1997, Liu and Sun, 2010, Madsen, 2000, Yapo *et al.*, 1998).
- Models always simplify some aspects of the system. Minimization of these confounding effects requires careful use of automatic calibration and of available field data to constrain model parameters. (Ito *et al.*, 2010)
- A danger of AC is to have less in-depth understanding of the model and its antiquated goal and therefore to choose incorrect parameter settings (Ndiritu, 2009)
- Parameter ranges are to somewhat constrained by prior knowledge, which simplify the calibration problem (Straatman *et al.*, 2004). The modeler actually just researches what is already suspected, but it might be necessary to find the right setting for certain parameters outside of the predefined ranges.
- If AC uses an iterative search scheme, it becomes path dependent which means that a second calibration run (with different random choices in its cause) can easily result in a different "optimal" parameter set. However this normally still leads to acceptable solutions (Straatman *et al.*, 2004).
- Even if a model is calibrated perfectly regarding the goodness-of-fit, it is still not guaranteed that this accounts for all the data instances that one does not knows about. In reality there is often hidden arbitrariness that cannot be incorporated into the model. Since the fitting error will never reach zero, some stopping threshold larger than zero can to be applied (Straatman *et al.*, 2004). Decision making upon this issue introduces additional subjectivity into the calibration procedure.

General limitations of any type of calibration

Every type of calibration, manual or automatic, single or multivariate, may suffer an obvious shortcoming: data availability regarding the right quantity and quality is influential on the success of the calibration optimization (Ndiritu, 2009). However, the question arises on how much data we need to be sure that the calibrated model can deal reliably and with reasonable accuracy with situations that fall outside the known data set. This question still remains to be dealt with in the scientific field of calibration methods (Straatman *et al.*, 2004).

3.7 Search schemes used in automatic calibration

According to Li *et al.* (2010), the arbitrary nature of manual model calibration has motivated the development of automatic calibration techniques, including gradient-based methods, such as the Gauss–Levenberg–Marquardt method (Doherty and Johnston, 2003), population-evolution-based algorithms, e.g., Shuffled Complex Evolution method (Qingyun *et al.*, 1992), and regionalization or spatial generalization (Lamb and Kay, 2004). In addition, recently also a lot of multi-agent-based algorithms are implemented to search the parameter space for calibration (Jiang *et al.*, 2013).

Based on the works of Schrijver (2003), Agyei and Hatfield (2006), and Youssef *et al.* (2001) a categorization of search schemes for combinatorial optimization problems is made, which is shown Figure 10. In general, these search schemes can be divided into two major categories: exact methods and in-exact methods that are vastly using probabilistic approximation approaches or meta-heuristics for detecting optima in the solution space. When the search space in which the optimal solution is embedded turns rather large and is further non-linear, the type of search schemes that uses exact algorithms becomes impractical to handle. This is due to the fact that the number of options that have to be evaluated often grows exponentially. As a result, the growing search space frequently cannot be evaluated in an acceptable amount of time.

Since the vast majority of models incorporates a large number of parameters that are eligible for calibration (Bekele and Nicklow, 2007), exact algorithms are not practical to use for complete parameter space evaluation. Instead, approximation methods have been developed to assess parts of the available search space with some randomness. Also, the more these methods evolve, more and more “search intelligence” is added to them, which relies on various learning algorithms. These methods can be further divided into sequential methods and non-sequential ones. The latter are often called space-filling or sampling techniques, since they, a-priori any calibration attempt, fill in the parameter options that will be assessed by subsequent model runs. Within the possible parameter ranges, this can be done in a totally uniform manner, e.g., with random sampling, or in a bit more sophisticated way by dividing the ranges into smaller sub-ranges, e.g., by Latin Hypercube Sampling (LHS) or orthogonal sampling (Bekele and Nicklow, 2007, Li *et al.*, 2010, Yuan and Szu Hui, 2013). For a full discussion of common space-filling designs, we refer to McKay *et al.* (1979). Often calibration studies use space-filling designs to generate “good”, and thus well spread, starting points for sequential search procedures (Bekele and Nicklow, 2007).

Sequential search procedures differ from non-sequential approaches due to their ability to use current information on prior solutions for evaluation of the search direction in successive search iterations. Moreover, they can generate new independent candidate solutions, instead of following a predefined array of evaluation solutions, as the non-sequential methods do. One can split sequential approaches up into local and global search methods (Schrijver, 2003, Skahill and Doherty, 2006). Local methods incur a significant chance to get stuck in local extremes, if non-linear search space is evaluated. They are searching solution space in a point-to-point manner based on certain hill climbing techniques. Examples of local search approaches are random search, such as grid search, and gradient-based search methods, as e.g., steepest descent or the (Quasi-) Newton method. These methods are more sophisticated than pure random search, however, the downside to a strong gradient focus is that these algorithms cannot easily search other areas in the solution space after it is decided to “climb” a particular “hill” in the data (Schrijver, 2003). This problem caused the development of global search schemes that rely on a randomized evaluation structure.

Global search methods are divided into two major categories: point-to-point search and parallel or simultaneous search. Popular examples of the first category are Simulated Annealing (Kirkpatrick *et al.*, 1983) and Tabu search (Glover, 1989). The strength of a randomized search design is to occasionally accept worse neighborhood solutions throughout the search process to escape a local extreme. For point-to-point search methods

this occurs in an individual manner based on the currently best solution. Parallel search on the contrary, tries to spread these random scatters to a larger extent with the intention to cover more initial solution space than point-to-point methods. Prominent approaches for simultaneous searches are population-based evolutionary algorithms and multi-agent-based algorithms. An important advantage of evolutionary algorithms, as well as point-to-point global approaches, is that they are not restricted to assumptions of stationarity, and they can integrate data from a variety of sources (Sen *et al.*, 1995). Popular examples of evolutionary search algorithms are the Shuffled Complex Evolution (Duan *et al.*, 1992) and the Genetic Algorithm of Holland (1975) and Wang (1991). We can find a comprehensive explanatory comparison of the Genetic Algorithm and Simulated Annealing Kong *et al.* (2009). Regarding multi-agent algorithms, typical examples are Particle Swarm Optimization (Kennedy and Eberhart, 1995) and Ant Colony Optimization (Dorigo, 1992).

Skahill and Doherty (2006) state that gradient-based algorithms have been the traditional choice for automated calibration because they are easy to implement and computationally efficient. However, since these methods are local search approaches, their biggest drawback is that they only have limited capability to find the global optimum in the tunable parameter vector. This has been noticed by numerous researches[‡‡‡‡]. Based on that, more and more evolutionary and multi-agent-based approaches are used as search scheme to find near-optimal parameter vectors – for both aggregated objective functions and Pareto fronts. Rose *et al.* (2007) states that these calibration methods share the general approach (Freedman *et al.*, 1998, Vallino, 2000) of specifying an objective function based on data-model goodness-of-fit, and using accumulated information from previous model runs to determine how to change parameter values for subsequent runs. This is repeated until a minimum for the goodness-of-fit function is found. Often checks, using new starting parameter values or applying perturbations to parameter values, are then made to ensure the minimum is a global minimum of the objective function.

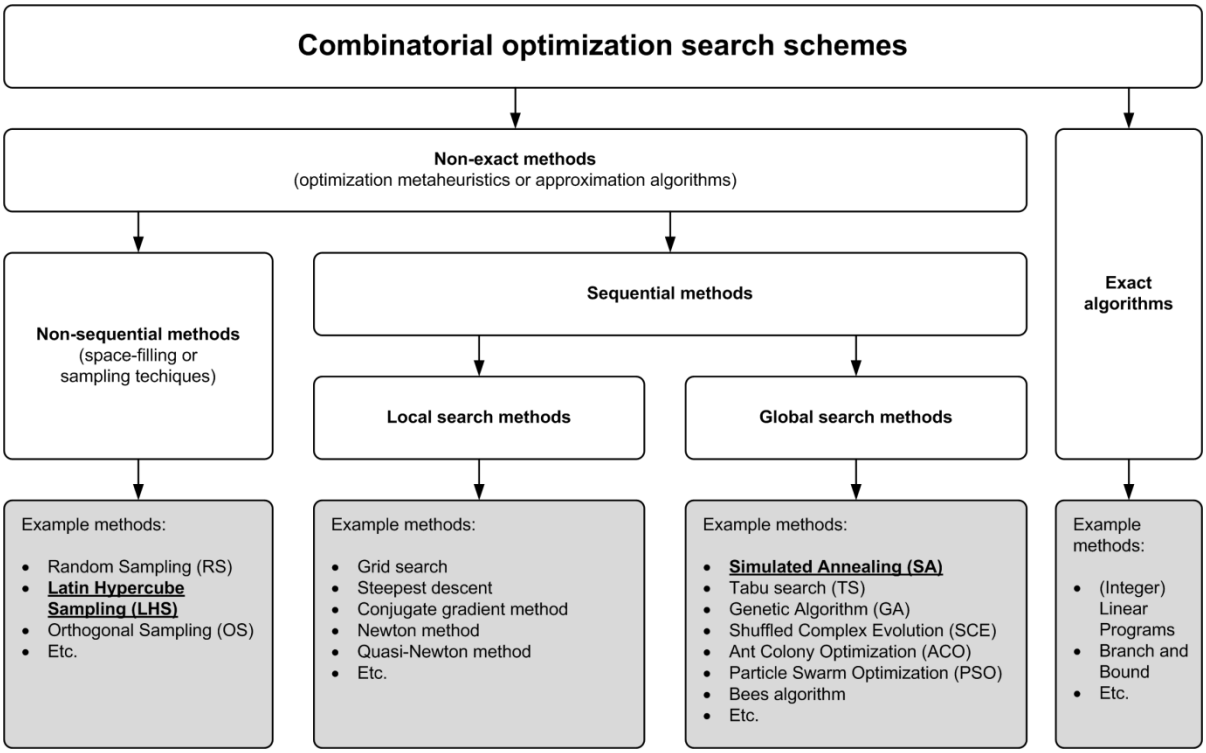


Figure 10: Categorization of combinatorial optimization search schemes

[‡‡‡‡] Agyei and Hatfield (2006), Chiu *et al.* (2010), Ito *et al.* (2010), Rode *et al.* (2007), Skahill and Doherty (2006)

Global search algorithms that are commonly used for automatic calibration applications are Simulated Annealing (Matear, 1995), the Genetic Algorithm (Ward *et al.*, 2010), but also the Shuffled Complex Evolution (Agyei and Hatfield, 2006, Duan *et al.*, 1992, Kuczera, 1997, Moussu *et al.*, 2011) and Particle Swam Optimization (Gill *et al.*, 2006, Jiang *et al.*, 2013, Liu, 2009). So far it is difficult to tell which of the global search schemes operates the best. Currently, it only has been shown that all of the global methods perform at least as good as or better than local gradient-based search algorithms (Agyei and Hatfield, 2006, Rode *et al.*, 2007). Kong *et al.* (2009) compared the performance of Simulated Annealing (SA) and the Genetic Algorithm (GA). They concluded that SA outperformed GA on the basis of their disease model. SA showed significantly better performance with regards to the necessary computational time to reach an appropriate goodness-of-fit. Sen *et al.* (1995) noticed the same phenomenon in their study. Due to this fact, it can be stated in general that SA is a suitable search algorithm for automatic calibration if a lot of parameters need to be incorporated. For a more detailed discussion on optimization meta-heuristics in engineering, we refer to Kirkpatrick *et al.* (1983), Wong *et al.* (1988), Goldberg (1989), Holland (1975) and Statnikov and Matusov (1995).

Despite the fact that global search algorithms have been developed that are capable of handling a large number of tunable parameters and the related search space, it is advised by several studies, as e.g., Liu and Sun (2010), Li *et al.* (2008), Linden *et al.* (2005) or Madsen (2003), to conduct a thorough sensitivity analysis before calibration runs are carried out. This ought to be done to enhance computational efficiency by decreasing the size of the possible solution space. We elucidate the reasoning for this advice in the next section.

3.8 Sensitivity analysis prior to calibration

A sensitivity analysis is an approach that is used to determine what model input parameters have significantly much impact on the observed key performance measures and therefore need to be modeled with care (Law, 2007). It further ensures that the experimental design for calibration is non-collapsing. Husslage *et al.* (2006) concluded that when one of the design parameters has almost no influence on the function value, two design points that differ only in this parameter will collapse, i.e., they can be considered as the same point that is evaluated twice. Since it is the goal to "tune" the model outcomes time-efficiently, a reduction of parameter space is recommended (Vanni *et al.*, 2010). Thus, when there are multiple parameters that have to be calibrated, it is highly advisable to carry out a sensitivity analysis prior to calibration to identify those parameters that significantly change the objective function.

Various systematic approaches to conduct a sensitivity analysis are known in the literature, e.g., in Law (2007), Balci (1998), Saltelli *et al.* (2000) or Montgomery and Runger (2002). The traditional technique to evaluate changes in output measurements is called "Morris method" (Morris, 1991) or "One-Factor-At-a-Time" approach (George, 2002, Law, 2007). In this technique, factors, respectively parameters, are varied once at a time in consecutive model runs and the difference in outcome is observed. However, if two or more factors exist, this method ignores interaction effects (Law, 2007). Kleijnen (1992) and Montgomery (1991) state that other analysis methods are more effective and accurate than the Morris Method. One of those methods is called "Design of Experiments" (DOE) (George, 2002, Kleijnen *et al.*, 2005). In this method, multiple factors are varied simultaneously. The original DOE-setup focuses on a full factorial design, in which all levels of all factors are varied to investigate the involved main and interaction effects (Montgomery, 1991). The aspect that interaction effects can be quantified is the strongest point of the DOE approach. A drawback of this technique, however, is that if the number of factors is growing, the evaluation of all the interaction effects becomes too time-consuming (Law, 2007). Yet the significance of these higher-level interaction effects (starting from three-factor-interaction and higher) are often negligible according to Law (2007), Montgomery (1991) and Montgomery and Runger (2002). Due to this, fractional factorial designs have been developed. We can find various "resolutions" of fractional factorial designs, such as $2^{k-p}_{III,IV,V}$ designs, from Law (2007). Nevertheless, so

far the Morris method together with the Pareto ranking method (Goldberg, 1989), is commonly used in calibration studies to identify significantly important parameters. This is mainly due to their implementation simplicity (Liu and Sun, 2010).

3.9 Discussion

In this literature review, we clarified the distinct concepts and problem formulations of parameter calibration in comparison to verification and validation of statistical models. We further elucidated various benefits and drawbacks regarding single and multivariate, and of manual and automatic calibration. We explained which objective functions and aggregation methods of these are commonly used and which combinatorial search schemes are frequently applied for automatic calibration, and which assumptions, advantages and disadvantages they include. Also the need for an a-priori sensitivity analysis was outlined. In general, we conclude that any sort of calibration can be useful for models that suffer from a low amount of input validation data and a large base of assumption parameters.

We chose to use the Root-Mean-Square-Error (*RMSE*) formulation as objective function for our study, since it amplifies large differences among the observed and expected data, which we intend to minimize as much as possible. Furthermore, our study is a contribution to automatic aggregated, i.e., scalarized, multi-objective calibration set-ups. The matching of several KPIs will improve the level of realism of our simulation model, but due to the objective aggregation, the implementation complexity will remain lucid (Bekele and Nicklow, 2007). Regarding the choice of the combinatorial optimization search scheme, we decide to start our calibration method with Latin Hypercube Sampling (LHS), since it is simple to implement into existing models and still has the capability to systematically scatter the initial neighbor solutions to a great extent (Yuan and Szu Hui, 2013). Thereafter, we combine the a-priori LHS method with a sequential approach. We prefer Simulated Annealing for this, based on the convincingness of the research of Kong *et al.* (2009) in which the Genetic Algorithm was outperformed by Simulated Annealing regarding computation efficiency, and also due to its point-to-point search pattern that makes it simpler to implement than a parallel search scheme. For the a-priori sensitivity analysis we decide for the Morris Method based on the statements made by Liu and Sun (2010) according its ease of use. However, since we only look at one calibration parameter from which we know that it has significant impact on the system performance, we not conduct an in-depth sensitivity analysis on other calibration parameters. Nevertheless, we look at the sensitivity of the weights that we use in our objective function.

All in all, the added value of this research to the scientific literature on calibration is the evaluation and testing of a hybrid automatic calibration approach that combines the strength of systematic a-priori parameter setting scattering over the entire solution space (i.e., feasible bounds of individual parameters) with the flexibility of a sequential point-to-point search scheme. To this end, we perform several algorithm comparisons.

Chapter 4: Data analysis

In this chapter, we elaborate on the selection and justification of data input and calibration benchmark data for our simulation model. The model input (Section 4.1) is sequenced as its physical occurrence in practice. It starts with the batch composition, followed by the batch size, the release delay between batch composition and first bag release, and all relevant travel times from the BagStore exit towards the in-cache lines near the make-up robots. Then, we discuss the main benchmark measures for the calibration model (Section 4.2). At the end of this chapter, we justify our calibration parameter choice and finish with a brief discussion on our data-related findings (Section 4.3).

4.1 Simulation model input

The basis for our data analysis, is a data record of eight consecutive working days of the baggage batching process at Schiphol South Terminal. We retrieved it from Vanderlande Industries at noon of 8 March 2013. The first bag observation dates back to 02:17:53.170, 1 March 2013, and the last one to 10:01:22.016 on 8 March 2013. We recognize a process flow in the data (Figure 11), which determines the sequence of the data analysis:

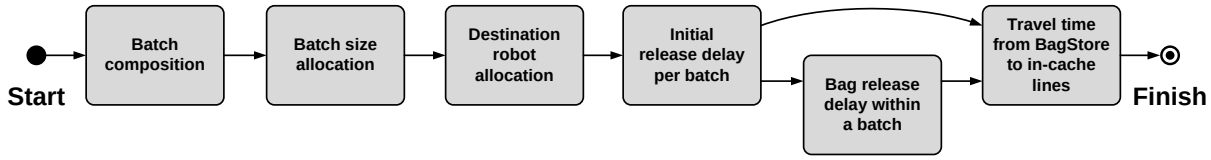


Figure 11: Logic behind data analysis sequence based on physical occurrences

Based on this process scheme, we research the statistical distributions of:

- Inter-arrival time of batch composition requests
- The number of bags assigned to an arbitrary batch
- The utilization of each make-up robot and its assignment of batches
- Initial release delay for the first bag within an arbitrary batch from the BagStore exit
- Bag release delay among bags within an arbitrary batch
- Travel times of luggage for various tracks from BagStore exit and in-cache entrances

We use frequency matching techniques to evaluate properties of the particular data involved in baggage batching. We analyzed all data statistically in an interval-based manner, except for the “working day length” and the “destination robot allocation”. All interval comparisons are determined by the Square root rule (Law, 2007). We eliminate outliers, according to the opinion of system experts, if data points show “unnatural” behavior. Additionally, data points are also considered as outliers if they are 3 or more standard deviations away from the initially measured mean of the data set they belong to.

The remaining data points we allocate into intervals and test them with the three statistical tests shown in Table 2 with an alpha value of 0.05. Next to statistical testing, we also apply graphical methods, such as the Q-Q, P-P and Difference plot, to assess the quality of hypothesized distributions of data. As we already stated in the literature review, the purpose of the Q-Q plot is to zoom in on the tail matching performance of the expected and the observed empirical distributions, while the P-P plot, as well as the Difference plot, are more effective to assess the match of mid-sized and large values of the expected and observed distributions. The difference between the latter two mentioned graphical comparison methods is that from the difference plot one can directly see to what quantitative extent the expected distribution is over- or underestimating the practice occurrences. The P-P shows this feature in a more global manner.

Table 2: Applied statistical tests in data analysis

Statistical test	Null hypothesis H_0	Alternative hypothesis H_1
Chi-square test	$H_0: f_{expected}(x) = f_{observed}(x)$	$H_1: f_{expected}(x) \neq f_{observed}(x)$
K-S test (Kolmogorov-Smirnov test)	$H_0: f_{expected}(x) = f_{observed}(x)$	$H_1: f_{expected}(x) \neq f_{observed}(x)$
Student t test (two tails, equal variance)	$H_0: \mu_{Expected} = \mu_{Observed}$	$H_1: \mu_{Expected} \neq \mu_{Observed}$

In the following, we identify four types of distributions that describe the empirical input data at hand:

Table 3: Identified theoretical distribution functions

Mathematical distribution	Arithmetic mean (first moment)	Variance (second moment)
Normal: $N(\mu, \sigma^2)$	μ	σ^2
Exponential: $Exp(\beta) = \Gamma(1, \beta)$	β	β^2
Gamma: $\Gamma(\alpha, \beta)$	$\alpha * \beta$	$\alpha * \beta^2$
Lognormal: $LN(\mu, \sigma^2)$	$e^{\mu+\sigma^2/2}$	$e^{2*\mu+\sigma^2} * (e^{\sigma^2} - 1)$

In order to determine the Key Performance Indicators (KPIs) for calibration, we need to know the primary and secondary moment relationship of the identified distributions. We intend to measure and compare these two moments from the calibration model against benchmark data taken from practice.

4.1.1 Batching hours per day

The batching process at Schiphol South normally takes place between 2 a.m. and 7 p.m. on a regular working day (which includes weekends). In Figure 12, we can retrieve the working day length during the seven complete working days from the assessment data files. The information from day eight is not included, since it is only a partial day. Based on the data at hand, we approximate the length of a working day for the BagStore with 17 hours.

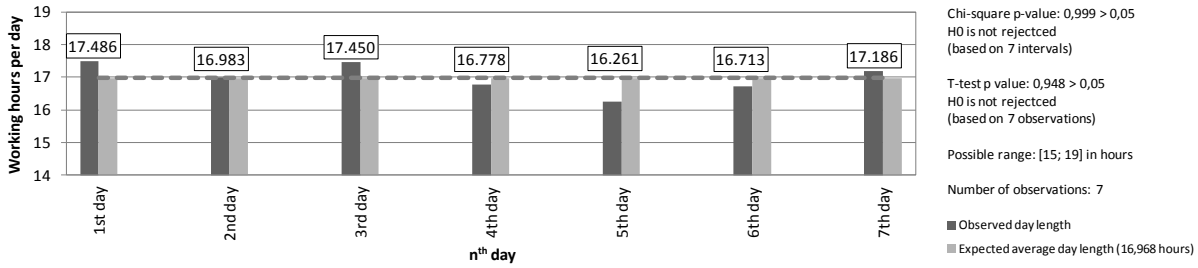


Figure 12: Batching hours per day

4.1.2 Batch composition / batch request inter-arrival time

After knowing in which interval batches are requested, we examine the properties related to their arrival. Since we suspect that these batch requests are independent starting points of the planned simulation model, we need to assess their extent of auto-correlation, which determines whether they can be considered as Independently Identically Distributed (IID) or not.

Independence and auto-correlation of batch composition requests

To assess the correlation relationship among the arrival data, we create both a scatter diagram of consecutive observations and also a specific correlation plot (see Appendix A9). The scatter diagram in Figure 13 shows request arrival n with request arrival $n+1$ in relation to their arrival time t. Through this analysis technique, we notice that consecutive request arrivals appear quite widely scattered. Thus, it seems less likely that the arrival time of $n+1$ is not predictable by the arrival time of n.

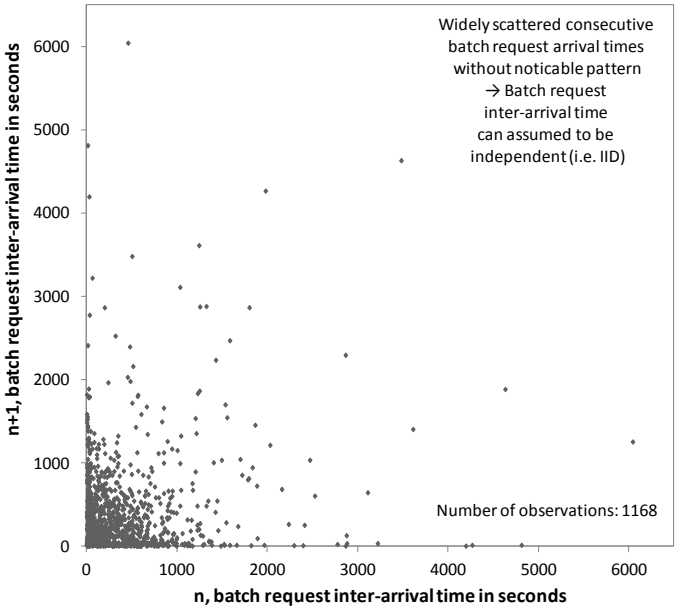


Figure 13: Test on auto-correlation of batch requests - Scatter diagram

Frequency matching

In a second step, we try to assess the theoretical distribution function of the inter-arrival time of batching requests with the earlier-described interval matching method. Figure 14 shows the observed and expected (i.e., theoretical) probability density function (*PDF*) of the batch request arrivals.

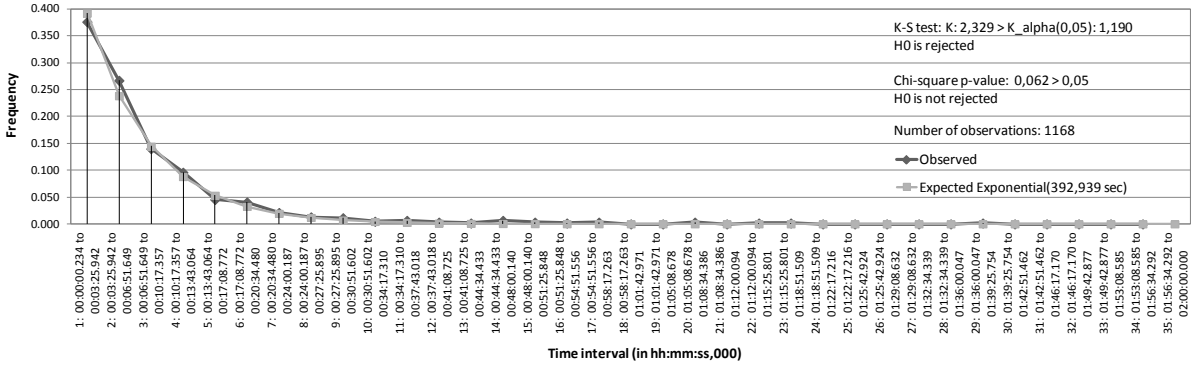


Figure 14: Probability density function of batch request arrivals

We expect an $Exp(392.939\ sec)$ distribution to match the observed data from practice. The Kolmogorov-Smirnov test rejects the null hypothesis that the expected distribution fits the observed one. However, it does not reject it strongly. The chi-square test on the other hand accepts the null hypothesis – however with a significance near the critical p-value of 0.05. The Q-Q plot, the P-P plot and the Difference plot (see Appendix A9) confirm our finding that the hypothesized distribution matches the observed one to a vast extent. Thus, the

theoretical distribution appears to have an acceptable fit. At maximum, the expected distribution underestimates the observed distribution by around 19%, the over estimation we can neglect.

4.1.3 Batch size

After batches are initiated in the system, a particular batch size is assigned per batch. We can describe this batch size with a normal distribution. Figure 15 shows the PDF of the expected $N(24.683\ bags, 7.048^2\ bags)$ distribution in comparison to the actual batch size observations.

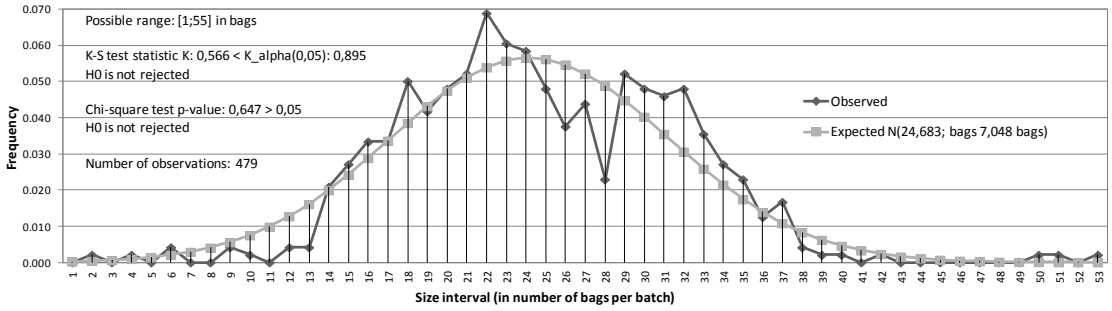


Figure 15: Probability density function of batch size

The K-S test, as well as the chi-square test both accept the null hypothesis. As for the cumulative density function (*CDF*) of the realized batch size (see Appendix A8), we notice that 99% of the requested batches have a batch size of 40 or smaller.

With regards to the Q-Q plot, the tails of the empirical distributions drift a bit off the predictions. According the P-P plot, the expected and observed distributions show a nice fit. Also by looking at the difference plot, the deviations between expectations and observations is fluctuating around a value of zero with a maximum amplitude of 2%. We conclude that there is enough reason to believe that the assigned batch size can be described with a $N(24.683\ bags, 7.048^2 bags)$ distribution.

4.1.4 Destination robot allocation

In addition to the batch size also the destination robot is assigned to a batch request when it arrives at the system (see Appendix A9). We assume that this assignment results in a uniform distribution, which evenly divides $1/6$ of all batches to each robot. According to the two tailed t-test, assuming equal variances, with a significance of 0.05 this assumption is not rejected. The p-value of the t-test gives a value of 0.24, which is clearly larger than 0.05. However, the chi-square test rejects the null hypothesis that the robot allocation is evenly spread, since the tails of both comparison distributions do not match ell. Nevertheless, we believe that the assignment of the destination robot to a batch follows a $\lceil U(0,6) \rceil$ distribution, since there exist six mark-up robots in the South Terminal with have an even chance to be assigned to a batch.

4.1.5 Dispatch delay of a batch after batch composition

After a batch is requested, an approximately normally distributed delay occurs that prevents the initiated batch from exiting the BagStore immediately. We estimate this delay with $N(15.910\ sec, 3.457^2\ sec)$. Both the K-S test and the chi-square test do not reject this null hypothesis (see Figure 16). Although, we have to take into consideration that the number of observations is rather small for these tests – which allows for more deviation in the data to be "acceptable" eventually.

Both the Q-Q and P-P plot (see Appendix A8) of this initial delay appear relatively straight-lined with respect to the fluctuation of the empirical data against its expectations. Also regarding the Difference plot, only small deviations are visible and we do not notice any trend of either under or overestimation of the prediction.

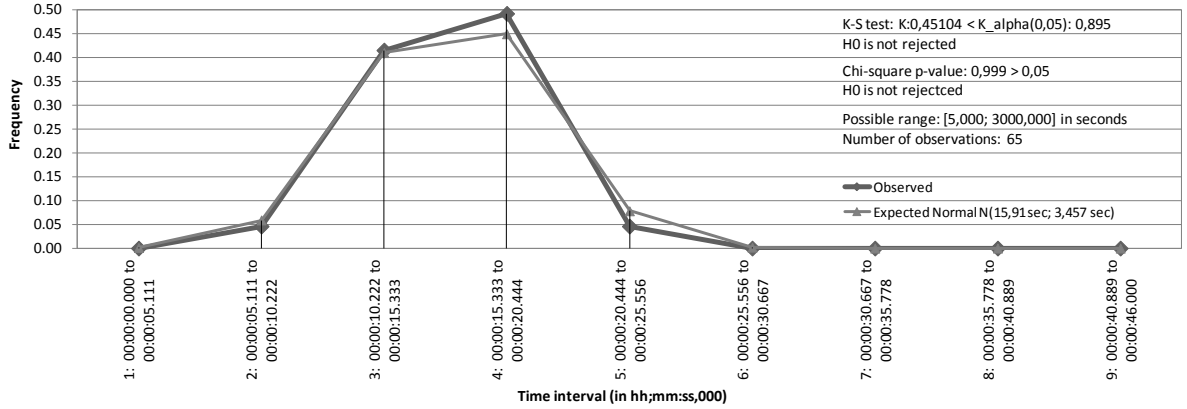


Figure 16: Probability density function of release delay between batch composition and first bag release per batch

4.1.6 Travel times

When a batch is released from the buffer, each of its bags travels an individual travel time from the BagStore to the in-cache line of its designated robot.

BagStore exit to in-cache line entrance

If we consider the total travel time from the BagStore to the in-cache lines, this duration seems normally distributed with $N(240.177\ sec, 17.688^2\ sec)$ (see Appendix A8). However, this assumption cannot be statistically verified, since H_0 is rejected by both the chi square and the K-S test. Luckily, more data on travel times is available, which we examine in the following. We notice that approximately 99% of all batches travel up to maximal 276 seconds on the conveyors from the buffer to the in-cache lines.

BagStore exit to sorter entrance

In Figure 17, we show the PDF-analysis on BagStore-To-Sorter travel times. Even though the sample size is rather big (i.e., 1805 observations), the chi-square test does not reject the null hypothesis. The K-S test on the other hand does reject the H_0, with just 8,27% above the allowed critical value of 0.895. Despite this fact, the $N(191.921\ sec, 16.022^2\ sec)$ distribution seems well fitting to the empirical observations. All three graphical comparison plots (see Appendix A8), indicate that the Buffer-to-Sorter travel time can be predicted well.

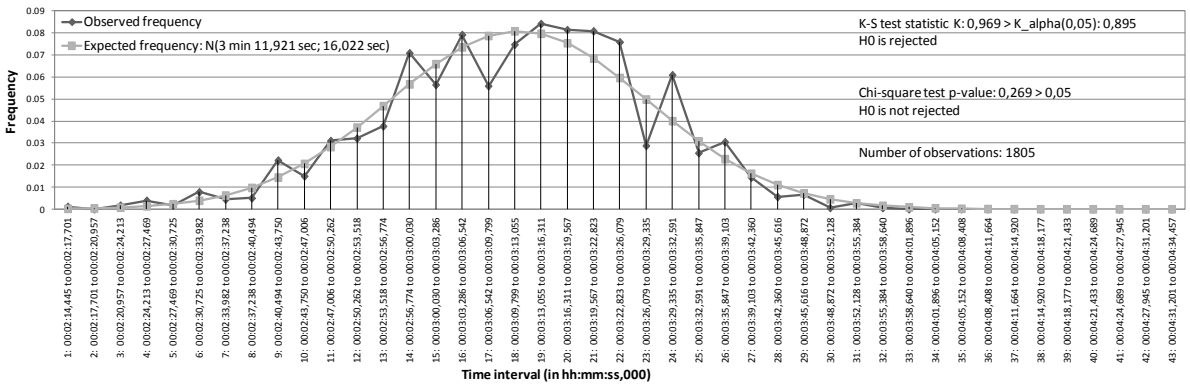


Figure 17: Probability density function of Buffer to Sorter travel time

Sorter entrance to In-cache entrance

The analysis of the inner-sorter travel times (from sorter entrance to in-cache lines) gives a rather strange initial impression (see Figure 18).

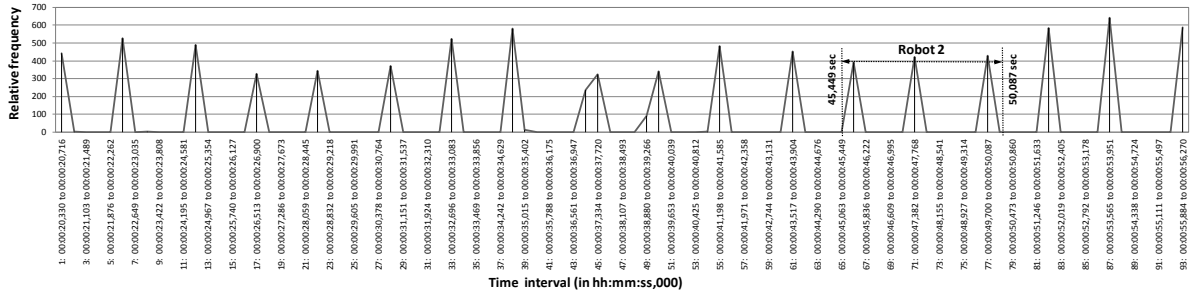


Figure 18: Unfiltered density of inner-sorter travel time, based on data of all robots

When we take a closer look at the observed multi-peak behavior of the travel times, we discover that there are three peaks noticeable per robot destination. Each peak appears to be normally distributed (see Figure 19). However, we cannot verify this yet with neither the K-S test nor the Chi-square test. When we zoom into the individual peaks per make-up robot destination, and see the following:

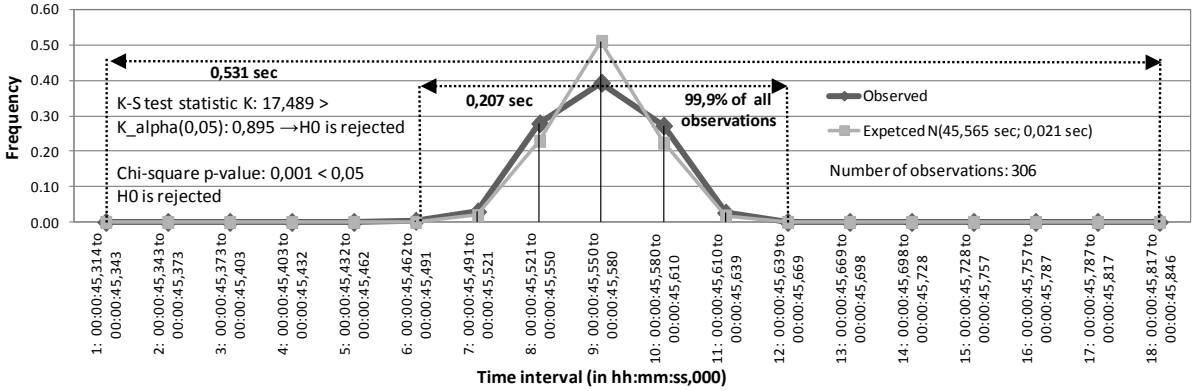


Figure 19: Zoomed-in inner-sorter travel time with focus on first "data peak" of Robot 2 data

The prior-described peak behavior is a result of the sorter entrance choice and is $N(45.565\,sec, 0.021^2\,sec)$ distributed within the range of 45.3 to 45.8 seconds e.g., for the second make-up robot . There are three sorter entrance: one close to the in-cache lines, one moderately distant and one far away. Based on the peak value observations, Figure 20 summarizes the expected travel times on the sorter.

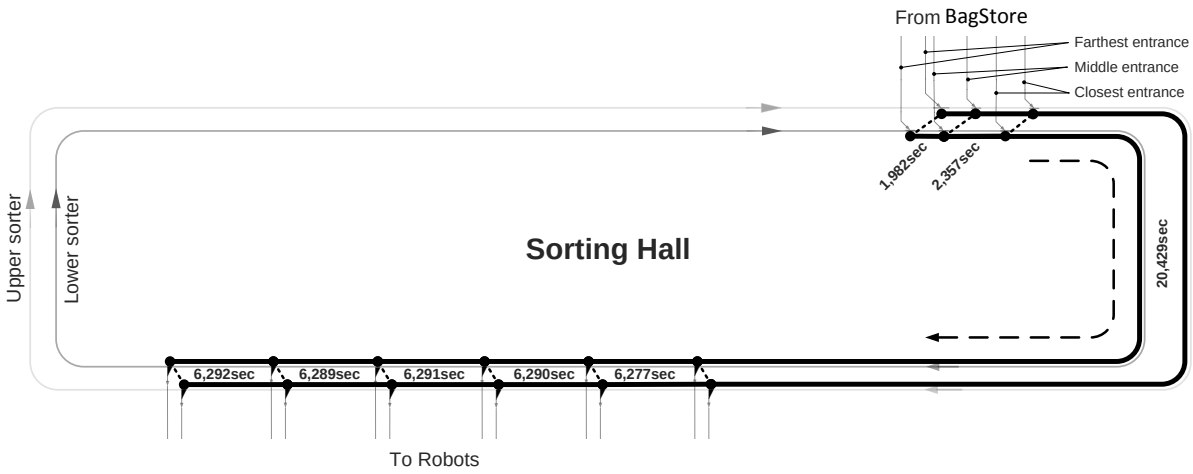


Figure 20: Inner-sorter travel times

4.1.7 Equipment capacity

Another restraining factor for the baggage batching process is the capacity of the individual system components. From the Material Flow Diagram (MFD) of the Schiphol South Terminal (Vanderlande Industries B.V., 2012d), the following equipment capacities could be retrieved:

Component	Maximum capacity in bags per hour (bph)
Buffer exit	1250
Tubtrax transport line	2400
Tubtrax divert	1300
Connector line to sorter	1500
Sorter	2550
In-cache line	1500

4.2 Calibration model benchmark data

Since automatic parameter calibration is the main focus of this research, we need benchmark measurements to assess the goodness-of-fit quality of the calibration methods. Here for we select two data measures: the inter-arrival time of batches at the in-cache lines and the total in-system time per batch.

4.2.1 Inter-arrival time between batches at in-cache lines

Figure 21 shows the PDF of inter-arrival time of batches at the in-cache lines near the make-up robots. Based on 833 observations, we expect a lognormal-like distribution for the inter-batch arrivals, that is distributed with $LN(7.076\ sec, 0.977\ sec)$. However, this assumption cannot be verified, since H_0 is rejected by any of the statistical tests applied. Nevertheless, the Q-Q plot as well as the P-P plot appear to feed the suspicion that the inter-batch arrivals at the in-cache lines are log-normally distributed with the mentioned parameters.

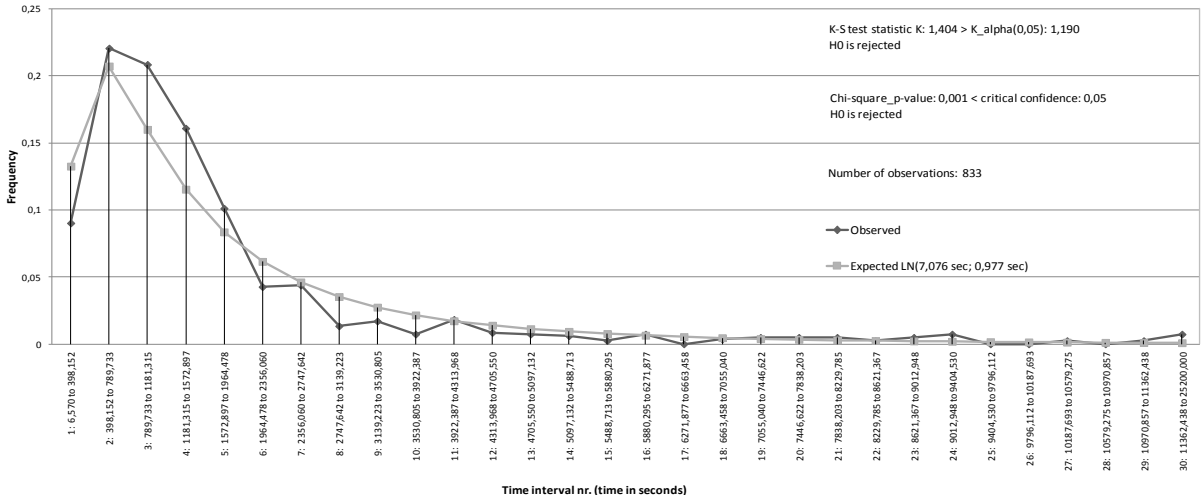


Figure 21: Probability density function of inter-arrival time between batches at in-cache lines

4.2.2 In-system time per batch

Another benchmark measure for our calibration model is the overall in-system time of a batch. We define this time as the time difference between the batch request arrival and the arrival of the last bag of that particular batch at the in-cache line near the make-up robot. Figure 22 shows the PDF of this performance measure.

Only with the K-S test we could verify that the hypothesized $N(356.767\ sec, 25.912^2\ sec)$ distribution represents the empirical data set with statistical significance. The Chi-square test is rejecting H_0 by far, since the right tail of the observed and theoretical distributions show large deviations from each other. This fact is stressed by the graphs of the cumulative density function, as well as the Q-Q plot. With respect to the predictive performance of the expected normal distribution only 85.125% of the empirical observations can be explained with the above-mentioned normal distribution. The edges of the right tail are drifting off noticeably from the center line and do not even return towards it, but remain drifting away in an increasing manner (see Appendix A9).

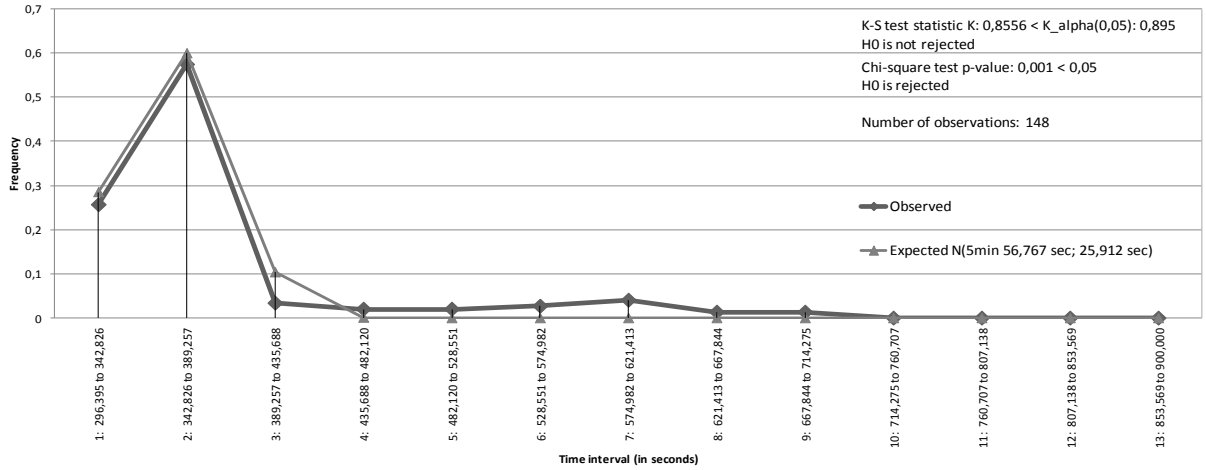


Figure 22: Probability density function of in-system time per batch

Both the PDF, as well the CDF (see Appendix A9) show that the data for the in-system time per batch has a minimum threshold of 296.395 seconds. Regarding the analysis from the P-P and Difference plot, the expected distribution seems to steadily underestimate the empirical distribution – without, however, drifting off a lot from the center line. Also, the main uncertainty comes from the right tail performance, where a strong underestimation of $N(356.767\ sec, 25.912^2\ sec)$ is visible – with a maximum of around 15% in deviation. Nevertheless, we assume that the in-system time per batch is normally distributed with the mentioned parameters.

Release delay per bag within a batch from Buffer exit

One instance measurement has been left out in the data analysis so far – the inner-batch release time. This measure appears difficult to quantify and therefore represents a good candidate for the tunable calibration parameter in our calibration model. We define the inner-batch release delay as the time at the BagStore exit that is delaying the final release of bags within a certain batch. Thus, if a bag is not the first bag of a batch, this delay applies to it.

Figure 23 shows the expected and the empirical PDF of the inner-batch release delay. One can already see that the proposed $Exp(3.698\ sec)$ distribution does not fit well with regards to the actual observations. Initially, the hypothesized distribution overestimates the low value release times, while in the following strongly underestimating mediocre release times and then again overestimating large release times. Due to this fact, all applied statistical tests could not verify that the expected distribution matches the observed one well. Thus, the Chi-square test, as well as the K-S test reject the null hypothesis.

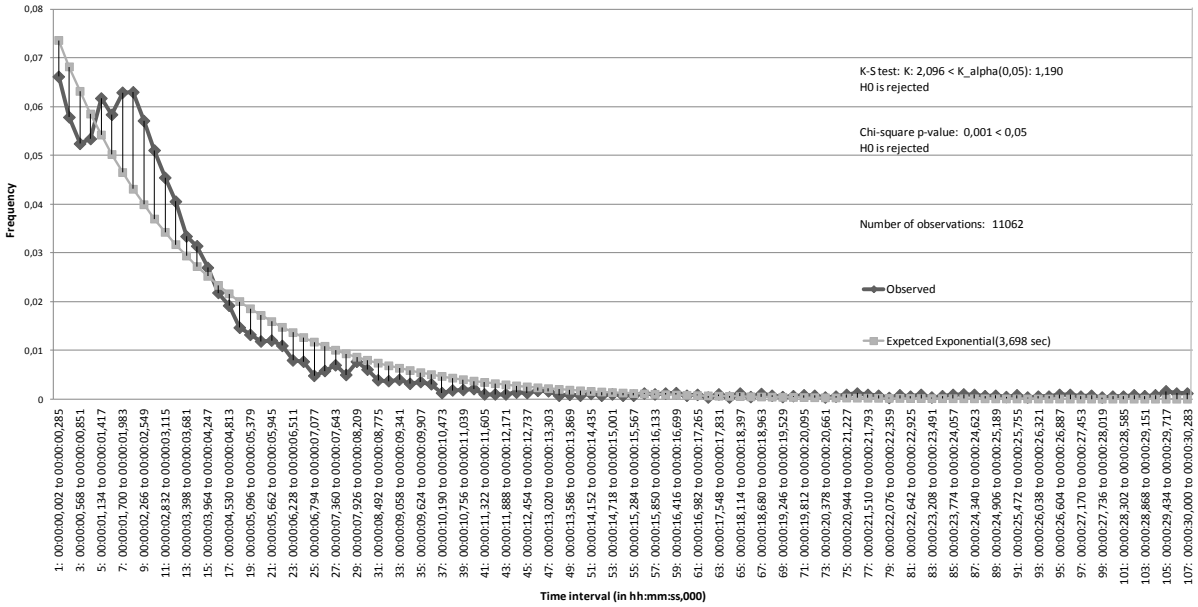


Figure 23: Probability density function of inner-batch release delay – large deviations visible

The inner-batch delay time appears rather unpredictable to us, which is why we choose it to be the main calibration parameter in our study.

4.3 Discussion

In this chapter, we carried out a data analysis on calibration model input and benchmark measures based on empirical observation of the implemented baggage handling system at Schiphol South Terminal. We found an independent starting point for our calibration model, since the batch request arrival times can be assumed to be IID and exponentially distributed. This implies that the batch requests follow a Poisson arrival process. Moreover, we could verify the fit of most of the input distributions for the calibration model with statistical tests and graphical methods such as Q-Q, P-P, and difference plots. Also we identified benchmark measures for a later goodness-of-fit assessment of the calibration model, and we justified the final selection of the main calibration parameter that we are researching.

We finally decided to study the inner-batch release delay in our calibration model, since this parameter is the hardest to quantify with a decent level of accuracy by histogram frequency matching. No statistical test could give a clear indication about the underlying statistical distribution and including distribution parameters that the inner-batch release time is based on. Thus, we make it our task to find the right match for this tripartite parameter – i.e., the right distribution form, its mean and its skew. Due to this fact and also since we know that this inner-batch release delay is simplifying the subgroup release of baggage in reality, it appears suitable to us as a calibration parameter to test several automatic parameter estimation methodologies.

Chapter 5: Model set-up

In this chapter, we elaborate on the implemented calibration model. This includes a detailed model description about level of detail and scope incurred in the model (Section 5.1), the type of the used simulation model (Section 5.2) and surrounding assumptions (Section 5.3). Furthermore, we elucidate the process flow of the calibration model (Section 5.4). In Section 5.5, we show the applied formulation of the calibration objective function and the related constraints. Section 5.6 examines the determination of the minimum number of model replications. We complete this chapter with a brief discussion on the key aspects of the calibration model set-up (Section 5.7).

5.1 Scope & Level of detail

Regarding the modeling scope, we intend to describe reality as close as possible, given the available data. The model considers the initiation of batching requests in the early baggage buffer (BagStore) until the arrival of those batches at the in-cache lines near the make-up robots. This includes the batch size assignment per batch, the allocation of relevant travel times and the release delay moments of batches and bags. In the list below we explain the aspects of the real system that we left out of consideration for the simulation model.

- *Breakdown and repair behavior.* This aspect is indirectly taken into account in the travel and delay times that we observed from practice. For instance, if a breakdown happened that decreased the transportation speed of luggage, the recorded travel time on a certain equipment segment is larger than normal. If this occurs frequently, we can expect that large travel times are no outliers but include breakdown and repair actions which delay the luggage handling on the tracks.
- *Subgroup division of batches.* Since this feature was also neglected in the original simulation model of Vanderlande Industries, we also choose not to model it.
- *Bag tracking performance.* Similarly to the breakdown behavior, this we expect to be incorporated in the observed travel times from practice and the related bag and batch release delay times.
- *The baggage return flow.* The main baggage stream and the return stream only share small intersection possibilities on their travel tracks. Thus, we think that this aspect is of minor influence on the performance measures we collect.
- *Re-circulation.* Likewise to the return flow, recirculation of baggage is not likely to lead to notable deviations in the performance measures.
- *Separate Tubtrax bag carrier system.* Since only individual bags are put into tubs, the efforts to model this level of detail is not necessary to assess the required system performance.
- *Conveyor belts and transportation equipment specifics.* Partially, we model this superficially for visibility reasons, but the actual data assignment occurs at few assignment points throughout the modeled system. These points are located at the buffer exit, the sorter entrance and the in-cache entrances. Thus, e.g., travel times are assigned at the terminal station of a certain track and not during the actual transportation. This simplification does, however, not influence the quality of the measured output.
- *Batch request arrivals*. We consider these arrivals as independent starting points of the calibration model. In practice, this request is triggered by the make-up robots, based on LU availability and a particular flight schedule. Due to limitations of input data, we do not include those triggers.
- *The bag storage position in the BagStore:* Again we believe that this is already represented in the travel and delay times measured in practice.
- *Line balancing and specific merge and divert rules.* We believe that these aspects only have marginal effect on the performance measurements. Also most of those effects are likely to be included in the input travel and delay times.

5.2 Type of simulation model

The calibration model at hand uses a terminating simulation model. Since a batching day is stopped by a natural event, we do not to use a steady state simulation. This natural event is the end of a working day that is caused by the night break at Schiphol Airport. Due to this fact, it is not strictly necessary to determine a warm-up period for the simulation. Nevertheless, since the output performance is dependent on initial system conditions, it is a challenge to determine the right sample size of simulation replications to give an appropriately accurate statement about the achieved system performance. Due to the use of this type of simulation and also the scope of the calibration model, we have to make various assumptions that are described in Section 5.3.

5.3 Assumptions

Obviously, reality cannot be modeled precisely to the last detail. Therefore, we have to make some important assumptions that we need to keep in mind while interpreting the outcomes of the final calibration model.

First of all, an influential assumption is the existence of stationary model input. This implies that input distributions from the data analysis are not expected to change over time nor due to any initial system condition.

Next, we believe that always enough bags are available in the BagStore to fulfill a bag size allocation based on the data from practice. Since in reality only a batch is requested if a threshold of a minimum amount of bags for a certain flight is surpassed, we think it is justifiable to assume that a run-out of bags will not happen if a batch is initiated based on the known input data.

Another assumption that we make is that batches are never split up, even if the working hours are finished. Thus, if a batch is initiated just slightly before the closing of a working day, this batch will not be pre-emptied. Instead, it will finish its travel to the in-cache lines of the destination robot and will also count in the data of that particular working day. After the batch composition is stopped, the system receives enough time to run empty (i.e., until the next morning). Since, the average in-system time per bag is less than 6 minutes and around 99% of the batches do not take more than 11 minutes 8 seconds to be complete after their initiation, we think that this emption time is far more than needed to clear out the system from baggage. Thus, we expect all batches that have been requested during a normal working day to be vanished after a significantly shorter time than those idle hours during the night break.

As elaborated on in the previous chapter, we consider the independent, exponentially distributed batch requests as the starting point of our simulation model, which results in a Poisson arrival process.

As for the destination robot allocation, as well as the sorter entrance allocation, we assume a uniform allocation manner, where a succession batch, respectively bag, is assigned to next following robot or sorter entrance with the chance of $p = 1/6$.

Furthermore, only one discrete event trigger will be incorporated into the calibration model that initiates batch releases. This trigger, we call “last bag constraint”, which is responsible of releasing the next allowable batch for the same destination robot, if the last bag of the previous batch has arrived in the in-cache lines of this particular robot. Of course, in reality the release action is somewhat more complicated. However, we believe that this is not influencing our outcomes in a negative way. Overall, we think that we take sufficient and, particular representative, data points into account to determine the input of the simulation model.

5.4 Process flow within the model

The simulation model follows the process flow as depicted in Figure 24. It runs a predefined number of experiments and simulation runs. The decision on which experiments to run is determined by the calibration model (i.e., first dark square from the top). We explain the incurred methodologies in Chapter 6.

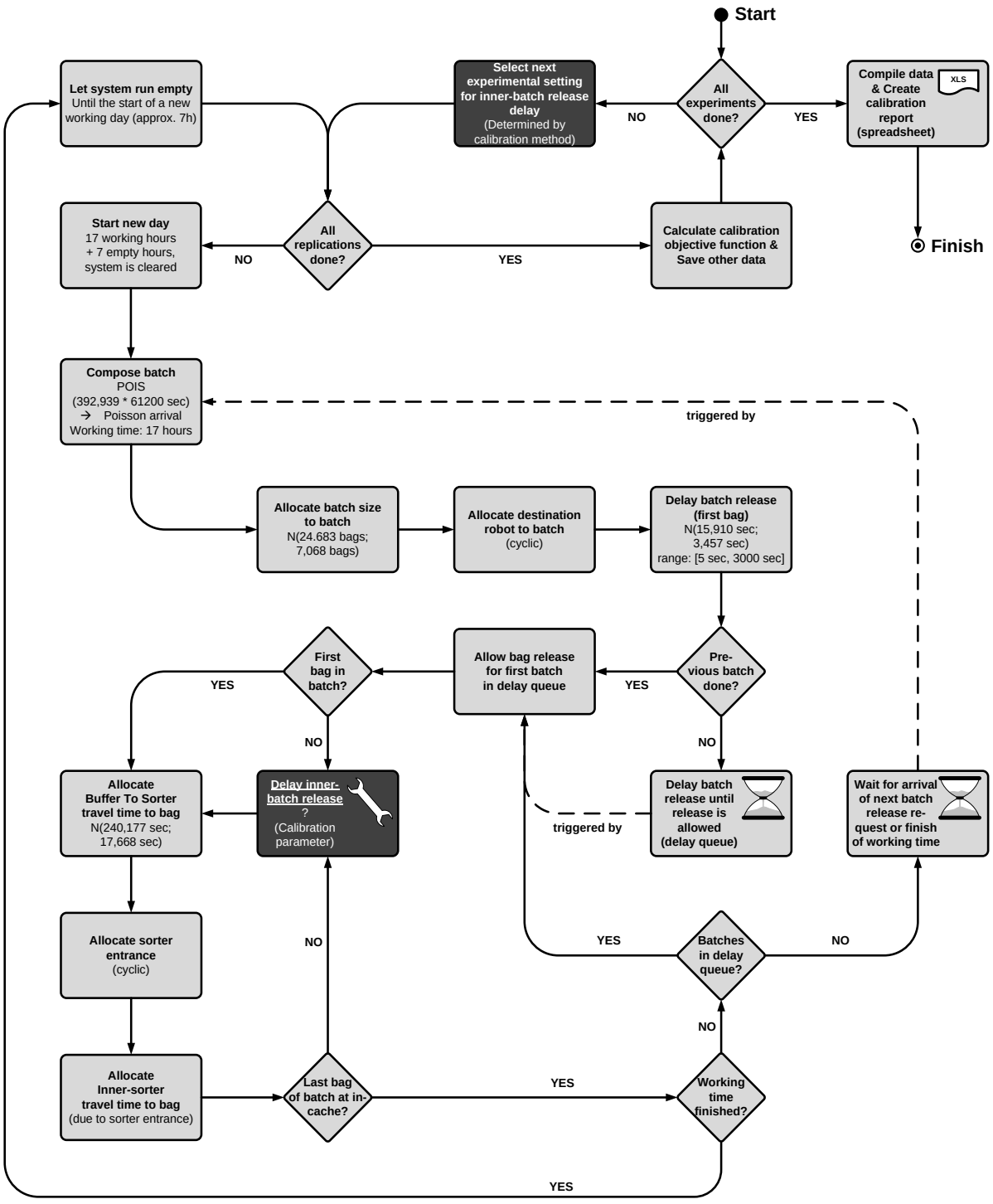


Figure 24: Simulation model process flow chart

A single simulation run starts with the initiation of a new working day, which lasts 17 hours in which batch arrivals are allowed. After that, the system is emptied until the next morning. After a batch is requested, the related batch size, its designated robot and the first-bag-release-delay are assigned to a batch ID. When this last mentioned delay passed, the batch can leave the BagStore exit. However, a batch is only allowable for release, if the last bag of the previous batch has arrived at the in-cache line of the same robot the current batch is directed to. If this is the case, the first bag of that batch is released and relevant travel times from BagStore exit to in-cache entrance are allocated to this bag. According to the travel time allocation, the situation is similar, if a bag in a batch is not the first one. However then it has to wait a certain time after it is eligible for release in relation to its predecessor. This "inner-batch-release-day" is the main focus of our calibration, since we determined it to be the only parameter to be calibrated. The release of bags within a batch is continuing until the batch size is reached. The last bag of a particular batch has the function to trigger a kind of traffic light at the BagStore exit. This traffic light is put on "go" when the last bag per batch has passed the entrance of the in-cache line. If no batch has been requested and prepared for travel yet, the "traffic light" will remain on "go" until a new ready-to-travel batch arrives.

5.5 Objective function & Constraints

The objective function of our calibration model is a scalarized root mean square error and thus represents a combination of Equations (1) and (4), pp. 18-21:

$$\min_{\theta\in\Theta}\left\{\sum_{i=1}^{I} w_i * \sqrt{\sum_{n=1}^{N}\left(Observed\ \emptyset_{i,n}(\theta) - Expected\ \emptyset_i\right)^2/N}\right\}, \tag{5}$$

where $Expected\ \emptyset_i$ is the a-priori determined mean value of KPI_i and $Observed\ \emptyset_{i,n}(\theta)$ is the measured mean value of KPI_i in the calibration model run n. Hereby θ represents the applied parameter setting that is constrained by the feasible parameter solution space Θ. The weight w_i is a parameter given to each individual KPI objective function. Further, N equals the total number of model replications.

Based on the recommendation of Madsen (2003) to attach weights to a scalarized goodness-of-fit objective function based on the predictive uncertainty of the involved KPIs, the weights w_i are determined as shown in Table 4. A sensitivity analysis upon these weights is carried out in the next chapter.

Table 4: Key Performance Indicators and their weights in the calibration objective function

Key Performance Indicator i	i	Weight w_i	Predictive certainty	Rationale
Mean inter-arrival time between batches at in-cache lines	1	$3/10$	Moderate	In Section 4.2.1, we could not verify this KPI with both the X^2-test nor with the K-S test well. We believe it has just moderate predictability.
Mean in-system time per batch	2	$3/10$	Moderate	In Section 4.2.2, the distribution of this KPI is accepted by the K-S test. However, the X^2-test rejected it strongly. Also the CDF shows, that only up to 85% of the observations are explicable.
First bag start release delay per batch due to last bag constraint	3	$4/10$	System immanent, thus high	This KPI is system-immanent in practice. A new batch can only be requested if the last bag of a previous batch arrived at the destination in-cache line or if the previous batch is cancelled.

Table 5: Expected Key Performance Indicator values from practice

Key Performance Indicator i	i	Expected $\emptyset_i$ value	Rationale
Mean inter-arrival time between batches at in-cache lines	1	$1954{,}100\ seconds$	see Section 4.2.1
Mean in-system time per batch	2	$356{,}767\ seconds$	see Section 4.2.2
First bag release delay per batch due to last bag constraint	3	$0{,}000\ seconds$	see Section 5.3

The $Expected\ \emptyset_i$ from Chapter 4 are summarized in Table 5. A remaining issue is the determination of the minimum number of model replications N_{min} with an appropriate certainty, which we explain in Section 5.6.

5.6 Minimum number of model replications

According to Law (2007) there exist two popular approaches determining N_{min} to achieve a good predictive certainty of the measured KPIs: the confidence interval procedure and the sequential procedure. We choose for the first due to its implementation simplicity. Both approaches require the a-priori determination of a relative error γ of the input KPI measures. In practice, γ often receives a value of 0.1, which means that for all measured confidence intervals we only accept a maximum of 10% of those intervals to be bigger than γ. However, γ has to be adjusted towards $\gamma' = \gamma/(1+\gamma)$, since it should represent the "actual" relative error of γ (Law, 2007). With the confidence interval procedure, N_{min} is calculated the following way:

$$N_{\min}(\gamma) = \max_{KPI_i \in \forall\, KPIs} \left\{ \min \left\{ j \geq n\colon \left[t_{j-1,1-\alpha/2} \sqrt{S^2{}_i(n)/j} \right] / |\bar{X}_i(n)| \leq \frac{\gamma}{1+\gamma} \right\} \right\}, \tag{6}$$

where $\bar{X}_i$ is the observed mean of KPI_i, $S^2{}_i$ is the observed variance of KPI_i and j is the first integer value that satisfies $j \geq S^2{}_i(n) * \left[t_{j-1,1-\alpha/2} \,/\, (\gamma' * \bar{X}_i(n)) \right]^2$. The term $t_{j-1,1-\alpha/2}$ is a parameter based on the Student-t distribution, where α is the desired confidence and $j-1$ are the degrees of freedom. The number of initial test replications n should be rather large to ensure that the precision of $\bar{X}_i$ and $S^2{}_i$ is precise enough. However, if n is taken too large computer resources might be wasted. If $n > 100$, $t_{j-1,1-\alpha/2}$ is approximated with 1.96, otherwise the Student-t distribution has to be referred (Larsen and Marx, 2006, Law, 2007).

5.7 Discussion

In this chapter, we elaborated on the calibration model set-up in detail. We explained several assumptions, which we made to realize the desired scope of the model and further discussed the process flow within an individual simulation run. Moreover, we illuminated the chosen calibration objective function, as well as the determination of the necessary minimum number of model replications for the intended simulation set-up. In chapter 6, we explain the test calibration methodologies together with the actual model implantation in Plant Simulation.

Chapter 6: Solution Approach & Experimentation

In this chapter, we discuss the calibration approaches for parameter estimation (Section 6.1), the realization of the calibration model (Section 6.2), as well as verification and validation of our simulation model (Section 6.3). In addition, we carry out a brief sensitivity analysis upon the chosen objective functions weights (Section 6.4) and elaborate on our experimental set-up (Section 6.5). We finish with a short discussion on the key issues we examined in this chapter (Section 6.6).

6.1 Automatic calibration approaches

Here we describe four distinct automatic calibration approaches that we evaluate. As explained in the literature chapter, we are interested in Latin Hypercube Sampling (*LHS*), further we adapted Simulated Annealing (*SA*) and create a combination approach of LHS and SA for calibration testing. Figure 25 illustrates the LHS-SA combination approach – again we hint that the entire figure below represents the detailed version of the sub-process in the flow-chart block "inner-batch release delay selection" of Figure 24, p. 39.

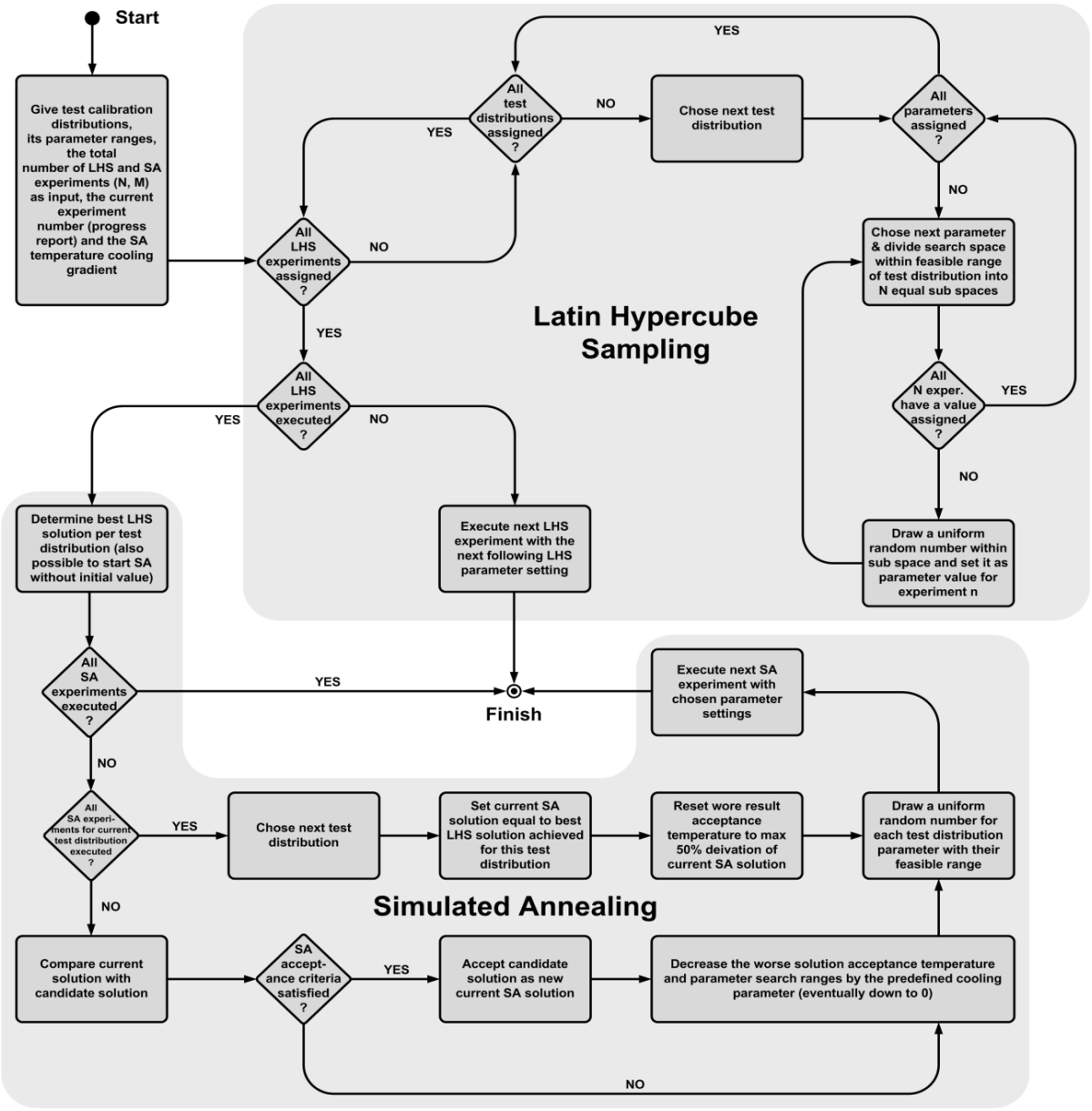


Figure 25: Calibration methodologies – combination of a Latin Hypercube with Simulated Annealing

Both LHS and SA require detailed input upon the hypothesized statistical distributions that they are tested with. This input includes: the type of distribution, the number of distributions parameters, feasible ranges per distribution parameter, and, of course, the total number of simulation runs we want to execute per distribution. As for SA, additional input is the start value of the acceptance temperature, which determines to what degree a worse candidate solution is accepted as a current solution, as well as the related cooling gradient that decreases this acceptance rate in the course of the series of simulation runs – and with it the eligible search space per distribution parameter.

LHS works rather simple: it scatters the randomly drawn experimental parameter settings quite evenly over the entire feasible search space. Regarding this, the total feasible search space is divided into M sub ranges, where M is the ceiled square root of the total number of LHS iterations allowed. Because of this divisional structure, a hypercube is generated for each distribution parameter. If a distribution has, e.g., two parameters, such as many do, each sub range of the first parameter is randomly combined M times with a sub range of the second parameter. Within these sub ranges a uniform draw is executed during every simulation run, which determines the final experimental setting for a certain parameter combination. This process is repeated until the end of all specified LHS iterations. The left part of Figure 26 illustrates how LHS scatters candidate parameter settings over the entire feasible search space. Additionally, for better understanding of the implementation of the LHS method in the calibration model, we show the distribution of the candidate scatter over a predefined number of simulation runs (here 9) in a 3D graph (right side of Figure 26).

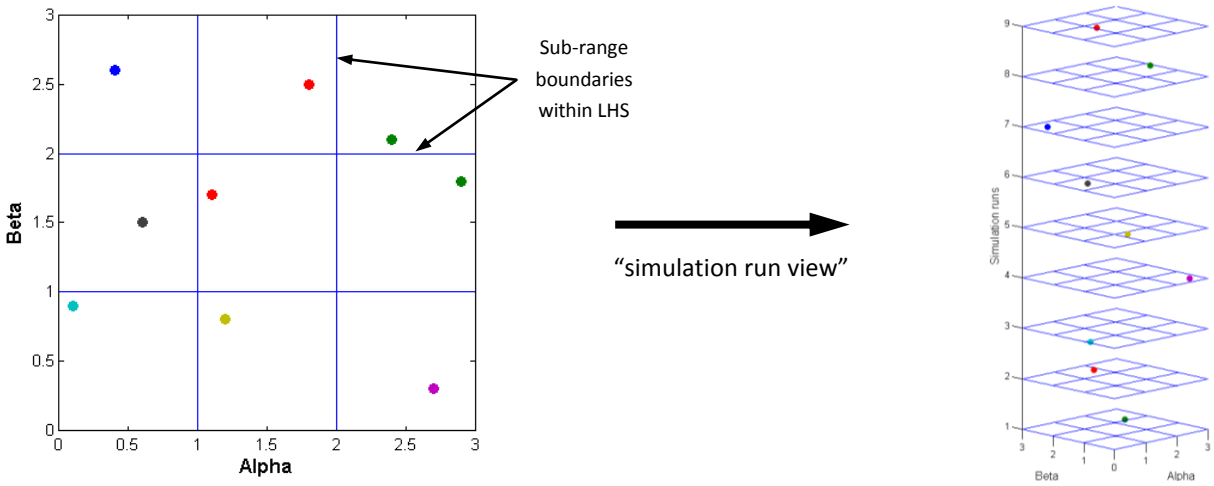


Figure 26: Scatter of candidate solutions throughout simulation runs with Latin Hypercube Sampling

If SA and LHS are used in a combined manner, SA starts its optimization from the best LHS setting found per test distribution and continues from there. We decide to divide the total simulation run length into half in this case – which means that LHS receives half of the runs to find a good solution and thereafter SA tries to do the same in the second half of the simulation run-length. If SA is used as a stand-alone algorithm, it executes its first distribution parameter draws over the entire feasible parameter ranges. The cooling parameter of SA is determined by Formula (9). Originally, the SA version of Kirkpatrick (1983) accepts a worse candidate solution with a negative-exponential PDF, as shown in Formula (7).

$$P_{acceptance\ worse\ solution} = e^{\frac{current\ solution - candidate\ solution}{cooling\ parameter}}, \text{ if candidate solution > current solution} \quad (7)$$

However, we chose for a linear decline of the acceptance temperature and with that also a linear decline of the upper bound of the worse accepted candidate solution. We do this, since we want to handle a hard acceptance threshold, while in the original version there is still a chance to accept obviously bad candidates. We

choose this SA implementation, since we prefer that a linear cooling scheme explores a longer amount of time larger parts of the feasible solution space than the original approach. The original scheme focuses on searching more candidates near the local optimum at the end of an iteration set. We, however, want to include a focus on the wide-spread search in the beginning of a simulation run set, since we believe that SA might otherwise converge to a local minimum too fast. Next, we describe our implementation of SA.

$$Initial\ acceptance\ temperature = 100\% \tag{8}$$

$$Cooling\ gradient\ per\ simulation\ run = \\ initial\ acceptance\ temperature / total\ number\ of\ allowed\ simulation\ runs \tag{9}$$

The acceptance temperature of Formula (8) will always be close to the value zero at the end of all SA runs per test distribution (i.e., iteration set), since only true improvements of the objective function are accepted eventually. In our SA version, we define the upper bound of worse solutions in Formula (10), which decreases over time down to the "best solution so far". Another adaptation of the original SA is that not only the acceptance of worse solutions decreases, but with it the eligible parameter search space per distribution parameter. This occurs when a candidate solution is accepted as current solution (see Figure 27). Formulas (11) and (12) illustrate this. We choose the values "2" and "1/3" in the formulas below to keep SA flexible enough to escape local minima while simultaneously tightening up the eligible search space per distribution parameter to achieve convergence.

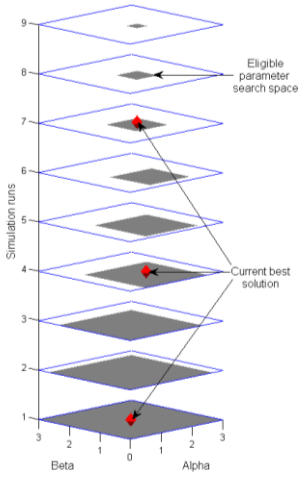


Figure 27: Narrowing solution space with SA

$$Upper\ bound_{worse\ acceptable\ candidate\ solution} = \\ 2 * Initial\ parameter\ guess * Acceptance\ Temperature + Best\ solution\ so\ far \tag{10}$$

$$Upper\ bound\ new_{parameter_i} = \\ Value\ parameter_{i\ current\ solution} + |Upper\ bound_{initial} - Lower\ bound_{initial}| * \frac{1}{3} * Acceptance\ temperature, \forall i \tag{11}$$

$$Lower\ bound\ new_{parameter_i} = \\ Value\ parameter_{i\ current\ solution} - |Upper\ bound_{initial} - Lower\ bound_{initial}| * \frac{1}{3} * Acceptance\ temperature, \forall i \tag{12}$$

If the value of the new parameter bounds is surpassing their initial parameter limits, the initial bounds are applicable again. Also the upper bound is not allowed to become smaller than the lower bound and vice versa, since the determination of the new experimental setting is a uniform draw as in Formula (13).

$$New\ candidate\ setting_{parameter_i} = U(Lower\ bound\ new_{parameter_i}, Upper\ bound\ new_{parameter_i}), \forall i \tag{13}$$

Additionally, we run random search for comparability reasons. This scheme follows always Formula (14) without any adaption throughout its predefined number of search iterations.

$$New\ candidate\ setting_{parameter_i} = U(Lower\ bound\ initial_{parameter_i}, Upper\ bound\ initial_{parameter_i}), \forall i \tag{14}$$

6.2 Model implementation

We incorporate the functional layout from the Material Flow Diagram (*MFD*) of the baggage basement at Schiphol South into the calibration model (see Figure 28 and Figure 29), which we build in the Plant Simulation software of Tecnomatix Technologies Ltd.

Basically, three areas in the simulation will incur essential process activities, as Figure 28 shows. These areas are the BagStore, where batches are initiated and various data allocation takes place, the transition points of the "BufferToSorter" conveyors and the entrances of the in-cache lines. These points will be used for processing the in the Buffer allocated travel times and the in-cache lines will function as a trigger for the release of consecutive batches to the same destination robot. The simulation implementation is similar to the figure below, in which the buffer is programmed as a separate modeling frame (see Figure 29).

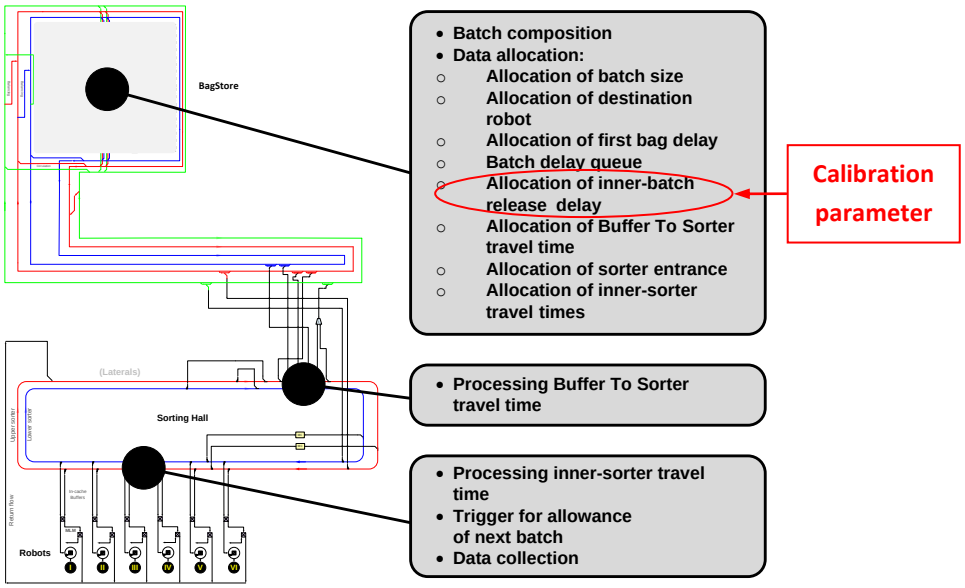


Figure 28: Layout of simulation model implementation

In the figure above we can retrieve all the essential model input that we elaborated on in Chapter 4. The calibration parameter, the inner-batch release delay, is "tuned" in the second-level Buffer frame.

6.3 Verification & Validation

As for verification of the simulation model, we executed short experimental runs to revise unexpected model behavior. First, we focused on the correct flow of baggage from the BagStore exit up to the in-cache lines. We also followed the batch creation and the initial data allocation process closely until we were sure that the model was reacting as we have been told from the internal system experts. After a couple of debugging and adaption actions we were convinced to have reached this goal.

Later on, we conducted large over-night runs to cover a vast amount of exceptional scenario cases and extreme values. Furthermore, we checked the outcome with these runs with the data from practice that we got and let system experts take a closer look into them as well. This way, we could uncover and fix inexplicable observations.

Model validation is a difficult endeavor, since there already exists a poor fit between model performance and reality. This aspect was one of the main initial motivators to carry out a parameter calibration study. Thus, we already know that any validation attempt will likely result in a poor match with data from reality. Also, the benchmark data only includes one batch request arrival rate over 3.5 working days. Since, the performance of the baggage handling system at Schiphol South Terminal is rather sensitive to the frequency of these arrivals – as we could understand from system experts – for a very detailed validation, we should conduct several other experiments with different arrival rates as well as benchmark these against comparable practice data. This, however, falls outside the scope of our research.

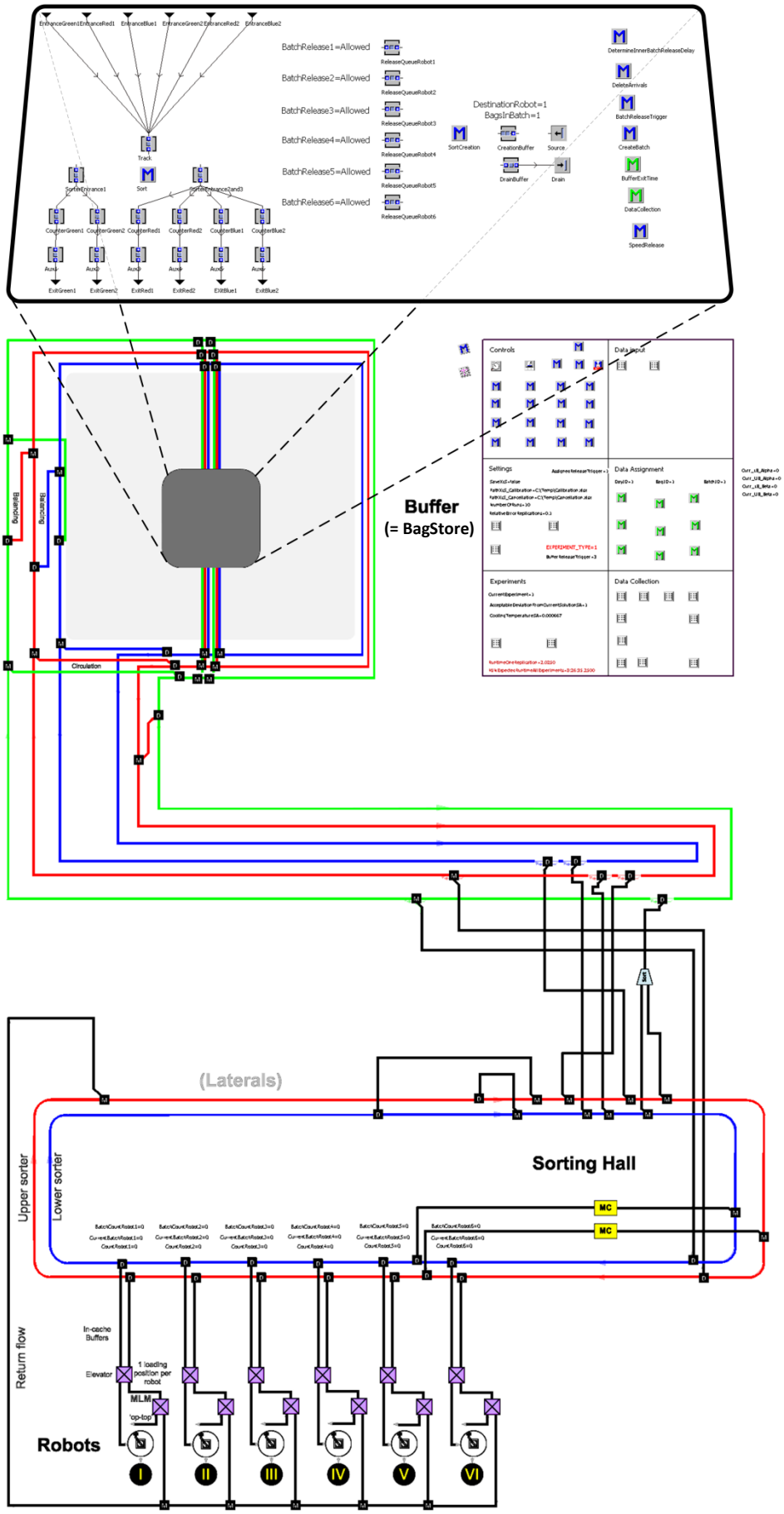


Figure 29: Combined implementation screenshot of the simulation and calibration model in Plant Simulation

6.4 Sensitivity analysis

In order to verify the impact of the chosen weights within the objective function (see Table 4, p. 40), we carried out a sensitivity analysis that compares different weight values and combinations with each other.

Influence of KPI weights on objective function

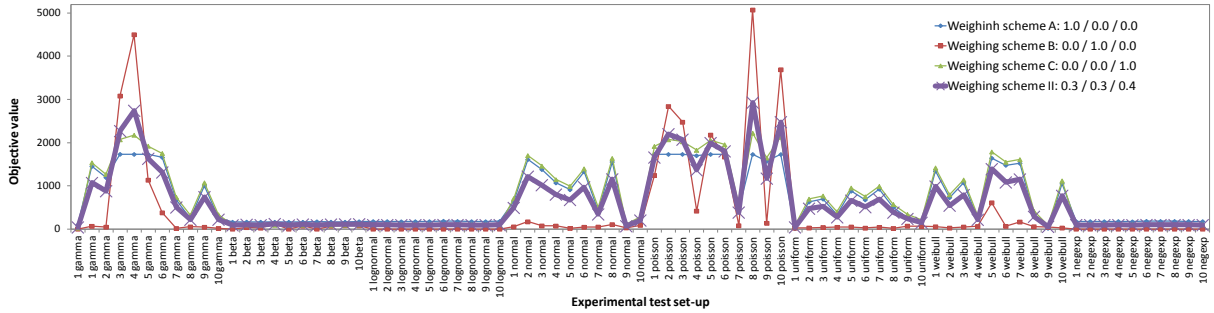


Figure 30: Individual influence per weight on the objective function and chosen weighing scheme

We vary the weight sensitivity based on the Morris Method, where we attach the most extreme value possible to a weight and compare the objective outcomes one by one. Figure 30 shows that weighing scheme A and C appear to have somewhat correlation with each other, which however varies per test distribution. Weighting scheme B reacts quite independently and functions as a strong amplifier of both negative as well as positive values within the overall objective function.

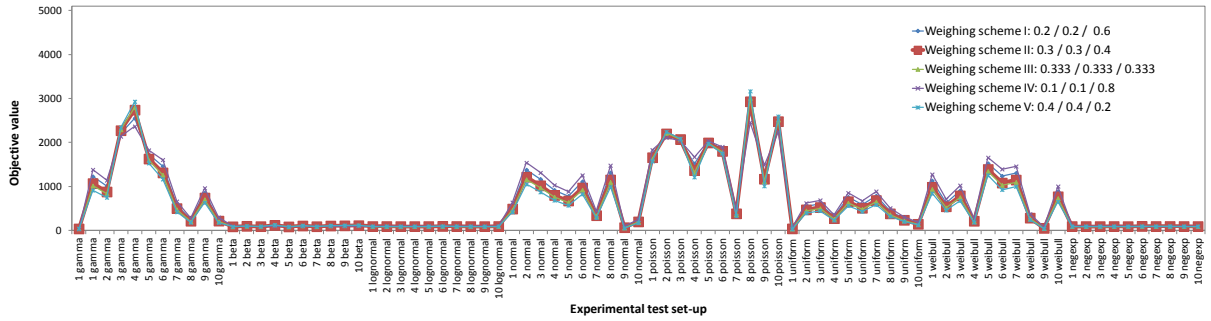


Figure 31: Different weighing schemes and their influence on the objective function

As we see in Figure 31, our chosen weighing scheme seems to provide a good trade-off between the stress of low and high values in the overall objective function. Thus, we continue to use weighing scheme II (i.e., 0.3/0.3/0.4).

6.5 Experimental set-up

The goal of our experiments is to verify whether the inner-batch release delay (see Section 5.4) is actually calibrateable and in what way the individual four automatic calibration approaches converge regarding the speed and quality of their best results. In particular, we are curious whether the combination of the a-priori Latin Hypercube Sampling (*LHS*) with the sequential Simulated Annealing (*SA*) is beneficial. We try to calibrate the inner-batch release delay with eight distinct statistical distributions (see Table 6). Thus, we execute in total $8 * 4 = 32$ large-sized experiments. The statistical distributions are restricted by upper and lower bounds for each of their distribution parameters (normally two parameters, except for the Poisson distribution). We choose these boundaries based on expert opinions and theoretical requirements for a particular distribution. For instance, it is a necessary that for the Gamma distribution that each of its parameters is bigger than zero. Therefore, negative distribution parameter settings are excluded – in addition to the logical fact that a negative delay does not make much sense in reality.

To create a better benchmarking comparison than with just the initial parameter estimation (see Chapter 4), we also execute random search. Since it is commonly known that random search is a poor-performing automatic search algorithm (Schrijver, 2003), it functions as a lower bound for algorithm convergence. If any of the other calibration algorithms achieves worse results than random search, we know that an error must have happened in our coding. Next to the benchmark algorithm, we test LHS and SA alone as well, so that we can get an indication whether the combination LHS/SA performs better in comparison to their single usage.

We conducted a couple of initial test scenario runs and realized that many of the distribution settings require at least 10 consecutive replications, i.e., 10 working days to count as one valid simulation run. We determined this through the application of Formula (6), p. 41. Since these replications are time-consuming to run, we decide to execute our experiments in a two-leveled way. We evaluate all four calibration schemes with 1500 runs for each statistical test distribution (see Table 6) and later, we repeat the same with 300 runs. We do this to assess the influence of the run-length regarding the convergence speed and quality of a calibration scheme. The reason we choose 1500 simulation runs is that this number of runs is still conductible in an acceptable amount of time. Since 300 runs are significantly less than 1500, we should be able to see any changing behavioral algorithm pattern due to the simulation runs length (i.e., allowed iterations). We conduct 3 replications of these 1500/300 run lengths to approximate algorithm convergence with an averaged regression formula. Since already one replication, of e.g., 1500 runs, is rather long, we chose to conduct only these three replications. Of course, since this is a regression approximation, randomness is incurred. The more replications are conducted the better the quality of the approximation finally is. However, computational time is a restricting issue in our research. Thus, in total, we execute $(3 * 1500 * 10 * 8 * 4) + (3 * 300 * 10 * 8 * 4) = 2.16 * 10^6$ simulation runs at 2.5 seconds per run. This adds up to an uninterrupted computation time of about 25 days.

Table 6: Experimental distributions & feasibility ranges

Statistical distribution (Step size: 0.001)	**Alpha**		**Beta**	
	Lower bound	**Upper bound**	**Lower bound**	**Upper bound**
1) Gamma	0.100	5.000	0.100	100.000
2) Beta	0.100	5.000	0.100	100.000
3) Log-normal	0.100	100.000	0.100	10.000
4) Normal *(Alpha = mean, Beta = standard dev.)*	0.100	100.000	0.100	10.000
5) Poisson	-	-	0.100	500.000
6) Uniform *(Alpha = start, Beta = stop)*	0.100	10.000	10.000	100.000
7) Weibull	0.100	5.000	0.100	100.000
8) Negative exponential	-	-	0.100	100.000

In order to assess the quality of our final averaged regression function per calibration scheme we calculate the observed relative error with the confidence interval half-width of the three replications made according to Formula (15). In general, if the observed relative error is less than 10%, we can assume that enough replications have been conducted for a statistically justified conclusion.

$$Observed\ relative\ error = \left|\left(\bar{X}(n) + t_{n-1,1-\alpha/2}\sqrt{S^2(n)/n}\right) - \left(\bar{X}(n) - t_{n-1,1-\alpha/2}\sqrt{S^2(n)/n}\right)\right| / [2 * |\bar{X}(n)|] \quad (15)$$

6.6 Discussion

In this chapter, we explained which calibration methodologies are tested and how they function. We also showed the computational implementation of the calibration model and attempts to verify and validate it. The sensitivity analysis of the weighing scheme that we incorporated into our earlier chosen objective function appeared to be a good trade off amongst the available weighing options. Furthermore, we elucidated the experimental set-up of the calibration methods programmed to assess their convergence behavior.

Chapter 7: Results & Findings

In this chapter, we summarize the results that we found with the prior explained experimental set-up (Section 7.1). First, we show the best computational outcomes of all calibration runs and further elaborate on the differences in candidate solution scattering and convergence behavior of each tested calibration method. Subsequently, we choose and justify our final calibration method choice. We conclude this chapter in Section 7.2 with a summary and brief discussion on our results.

7.1 Comparison of calibration methodologies

In the following paragraphs we show the best computational calibration results that we could achieve within our calibration model. Later on, we look deeper into the behavioral structures of our calibration methods.

7.1.1 Computational calibration results

From Table 7, we observe that the best results for the calibration of the inner-batch release delay could be achieved with the Gamma distribution in 1500 run-length series for all tested methodologies. The maximum improvement in comparison to the initial estimation of inner-batch release delay (see Chapter 4) is almost 57%. This value resulted from the parameter setting $\Gamma(0.288,15.520)$, which equals a mean of 4.790 sec. We discovered this setting with 1500 simulation runs of the Gamma distribution.

Table 7: Best results achieved per statistical test distribution

Search scheme	Statistical distribution	Obj. value	Alpha value	Beta value	Improvement
Initial guess	Gamma	44.991	1.000	3.698	**Benchmark**
Random Search (RS)	Gamma	25.949	1.037	5.385	42.324%
	Beta	77.360	4.959	1.107	<0%
	Log-normal	90.893	68.201	1.007	<0%
	Normal ($\alpha = \mu,\ \beta = \sigma$)	26.185	6.095	2.699	41.799%
	Poisson	26.144	-	6.655	41.891%
	Uniform ($\alpha = start,\ \beta = stop$)	26.470	1.272	11.369	41.112%
	Weibull	27.475	2.737	5.844	38.932%
	Negative exponential	89.946	-	4.716	<0%
Latin Hypercube Sampling (LHS)	Gamma	21.463	0.449	1.099	52.295%
	Beta	82.480	3.992	0.197	<0%
	Log-normal	91.458	64.857	2.804	<0%
	Normal ($\alpha = \mu,\ \beta = \sigma$)	23.479	5.215	4.539	47.814%
	Poisson	21.659	-	5.712	51.859%
	Uniform ($\alpha = start,\ \beta = stop$)	22.130	0.803	10.973	50.812%
	Weibull	21.597	0.859	4.941	51.997%
	Negative exponential	91.841	-	0.121	<0%
Simulated Annealing (SA)	**Gamma**	**19.367**	**0.288**	**15.520**	**56.954%**
	Beta	72.965	0.531	0.146	<0%
	Log-normal	89.091	18.799	9.159	<0%
	Normal ($\alpha = \mu,\ \beta = \sigma$)	20.916	2.541	5.511	53.511%
	Poisson	20.529	-	5.249	54.371%
	Uniform ($\alpha = start,\ \beta = stop$)	20.502	1.726	10.269	54.431%
	Weibull	20.928	1.939	5.578	53.484%
	Negative exponential	86.620	-	68.157	<0%
LHS-SA combination (L&S)	Gamma	20.059	0.449	10.990	55.416%
	Beta	76.385	1.137	1.246	<0%
	Log-normal	86.196	84.402	5.757	<0%
	Normal ($\alpha = \mu,\ \beta = \sigma$)	24.863	4.448	4.600	44.738%
	Poisson	23.318	-	5.637	48.172%
	Uniform ($\alpha = start,\ \beta = stop$)	21.921	1.328	10.895	51.277%
	Weibull	20.861	1.774	5.669	53.633%
	Negative exponential	88.660	-	60.299	<0%

The value zero represents a lower bound for the best possible fit that a calibration model can have. In comparison to the initial parameter guess that we made in Chapter 4, the best found parameter setting is almost 53% closer to this best possible fit than the initial guess.

Due to the fact that only the Gamma distribution is able to produce the best quality of calibration results for each calibration algorithm, in the next sections we only elaborate on the observed behavior for this statistical distribution. All the other distributions show similar convergence behavior, but are not further depicted.

7.1.2 Scattering behavior of candidate solutions

The exemplary scattering of candidate solutions from one representative replication of the 1500 simulation runs series is shown in Figure 32 to Figure 39. Figure 40 to Figure 47 illustrate the same for one typical replication of the 300 runs-series.

Random Search (RS)

Regarding the candidate solution scattering of Random Search throughout the allowed simulation run length, we see that the candidate settings are indeed scattered randomly. However this can result in situations where areas that already have been searched previously are searched again. We can see this in the accumulation of scatter points in Figure 33 in some areas, while other regions are left out of the search. This is visible in the same figure in the region between the α-values 0.1 to 0.7 and the β-values 35 to 55.

Latin Hypercube Sampling (LHS)

Latin Hypercube Sampling tries to avoid the scattering weakness of repeating search attempts in the same area as RS does. We can see from Figure 35 and Figure 43 that LHS performs better in scattering candidate solution settings. Both figures seem denser while the same amount of runs has been executed as for RS. Due to this fact, LHS is apparently, more capable of finding better quality solutions than RS.

Simulated Annealing (SA)

The scattering behavior of Simulated Annealing is interesting, since the sequential search evolution is clearly visible. SA starts its search widely spread at the beginning stage of the simulation period (see bottom of Figure 36) and then converges with increasing speed towards a local minimum found. Interestingly, the best solution was not found at the end of the simulation period. This illustrates the property of SA to accept worse solutions, if the acceptance temperature allows it. If the overall end of a simulation is near, SA cannot escape a certain local minimum anymore. Moreover, the comparison of Figure 36 and Figure 44 incurs some conclusions. We see a fast convergence in the 300 run-series to a sub-optimal local minimum. This stresses the importance of the right choice for the SA cooling scheme – the slower SA cools down the acceptance temperature, the higher the probability is to final a global minimum instead of a local one.

Combination LHS & SA

As explained earlier earlier for the individual SA and LHS candidate scatter, the combination approach, spreads the possible settings well over the feasible search space and subsequently converges towards a local minimum found (see Figure 38 and Figure 47). The difference with SA is that the entire feasible search space is "scanned" longer for a global minimum. Again, as for SA, the degree of the temperature cooling speed is essential for possible objective improvements. As visible in Figure 46, the SA part is cooled down to fast, which resulted in the fact that the best solution found by LHS could not be enhanced subsequently.

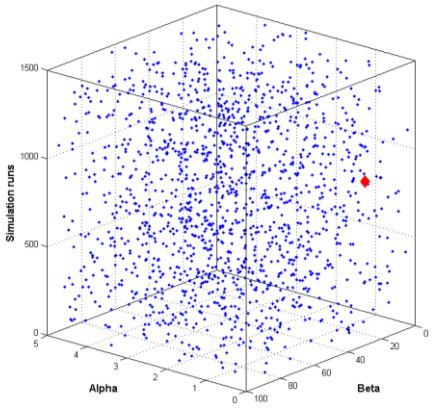


Figure 32: Random Search (Γ, 1500 runs)

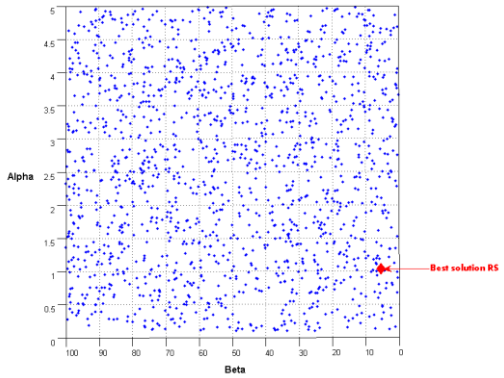


Figure 33: RS (Γ, 1500 runs) → top view

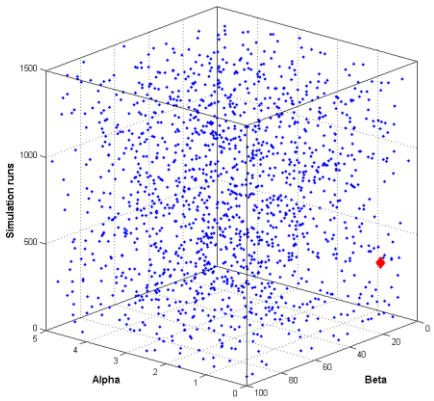


Figure 34: Latin Hypercube Sampling (Γ, 1500 runs)

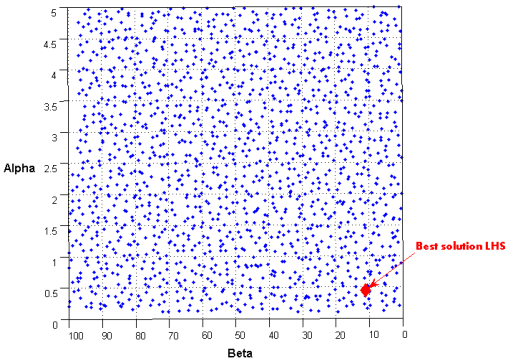


Figure 35: LHS (Γ, 1500 runs) → top view

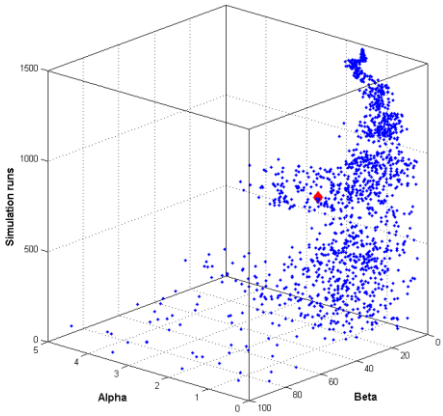


Figure 36: Simulated Annealing (Γ, 1500 runs)

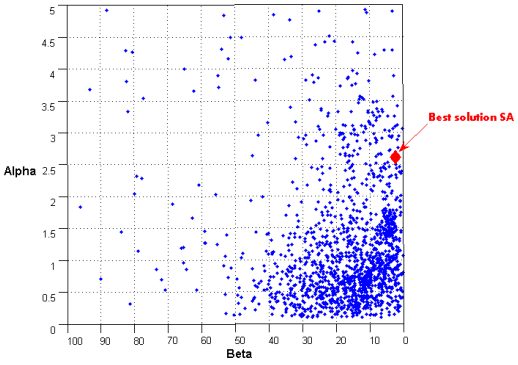


Figure 37: SA (Γ, 1500 runs) → top view

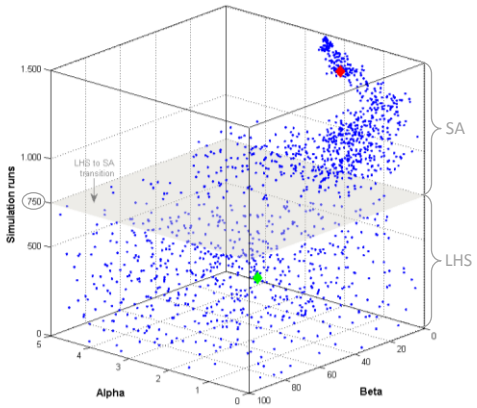


Figure 38: LHS & SA (Γ, 1500 runs)

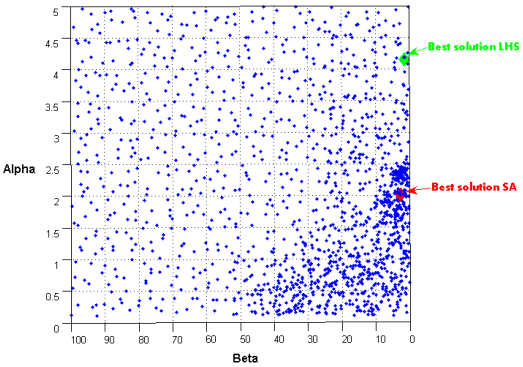


Figure 39: LHS & SA (Γ, 1500 runs) → top view

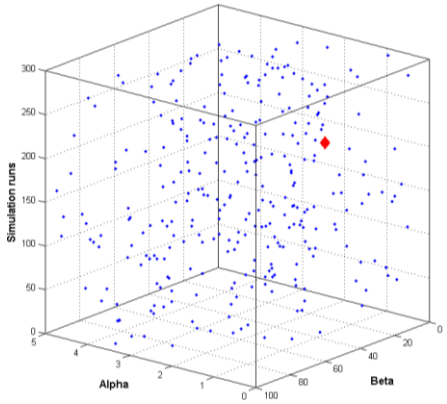


Figure 40: Random Search (Γ, 300 runs)

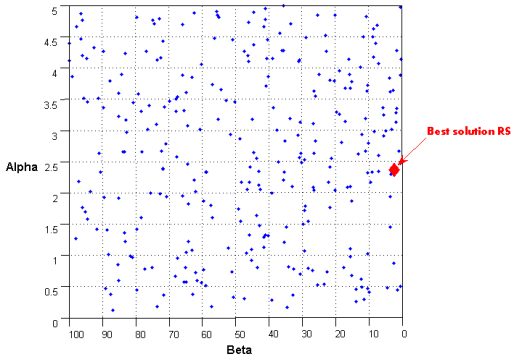


Figure 41: RS (Γ, 1500 runs) → top view

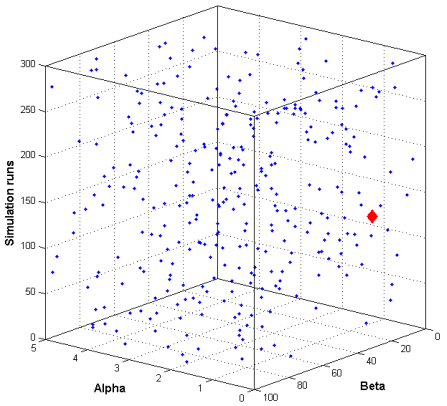


Figure 42: Latin Hypercube Sampling (Γ, 300 runs)

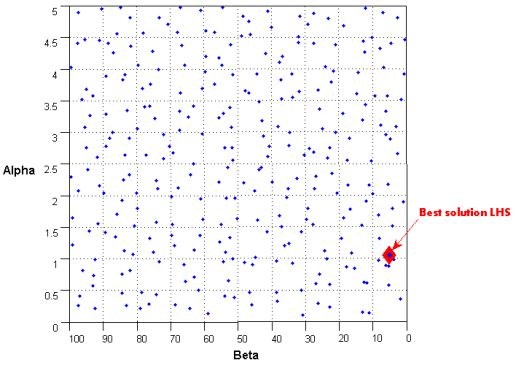


Figure 43: LHS (Γ, 300 runs) → top view

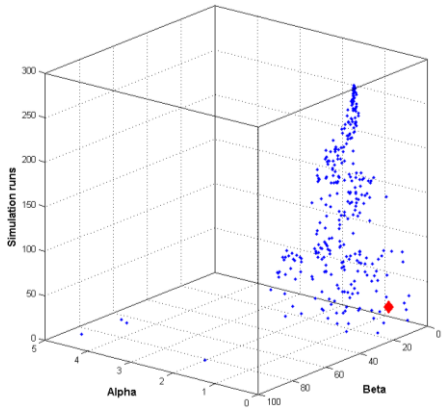


Figure 44: Simulated Annealing (Γ, 300 runs)

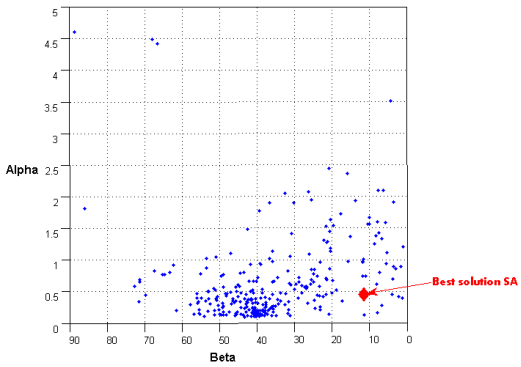


Figure 45: SA (Γ, 300 runs) → top view

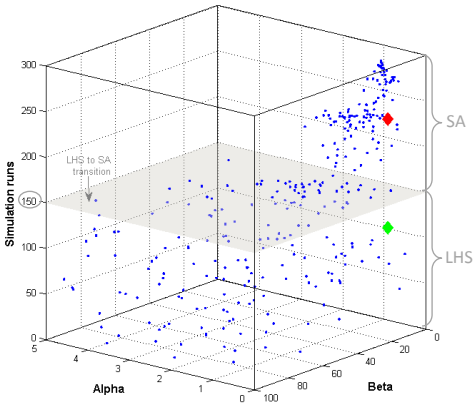


Figure 46: LHS & SA (Γ, 300 runs)

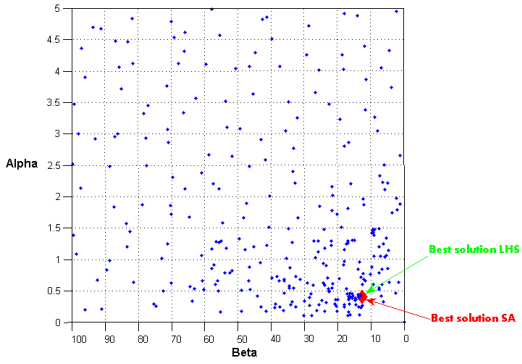


Figure 47: LHS & SA (Γ, 300 runs) → top view

7.1.3 Convergence speed & quality

Now that we know how the calibration methodologies scatter possible candidate settings, we want to quantify the convergence speed and quality of the individual approaches. We consider an algorithm as fast, if it reaches low objective values in a small amount of simulation runs. The convergence quality on the other hand, we define as the lowest possible objective value found throughout all iterations. To assess these two algorithm aspects, we make use of regression analysis. We chose the regression approach due to better comparability between the calibration methods, since else convergence is hardly expressible in continuous formula.

Regression method

Within MS Excel there exist various types of in-built regression options. Since algorithm convergence normally follows a negative exponential form (Schrijver, 2003), we use power regression to describe the observed algorithm behavior. Figure 48 shows a power regression example for three replications of Simulated Annealing experiments with each a simulation run length of 1500 iterations.

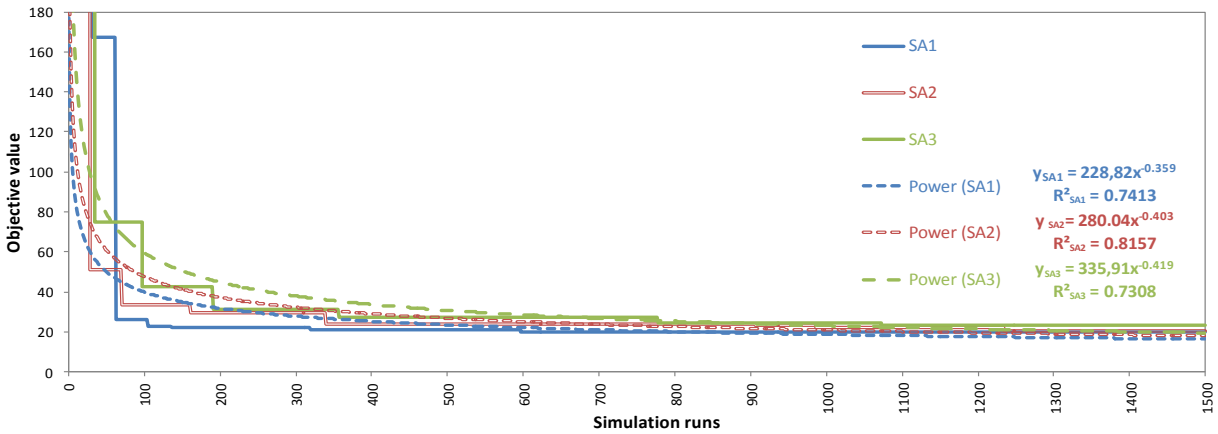


Figure 48: Regression example for Simulated Annealing objective convergence in MS Excel

As we can see for this example, the R^2 for each regression attempt is larger than 73%, which we see as a good fit of the regression model and the observed values. Commonly, if R^2 explains more than 50% of the incurred model variance, one can say that it is a decent regression approximation.

In the following, we averaged all regression functions of the conducted replications for the 1500/300 simulation runs (see Figure 49). We elaborate on the outcomes of this averaging in the next sub-section. To determine whether enough replications have been executed, we conduct a relative error evaluation for both the 1500/300 simulation run length for the regression value at their last iteration.

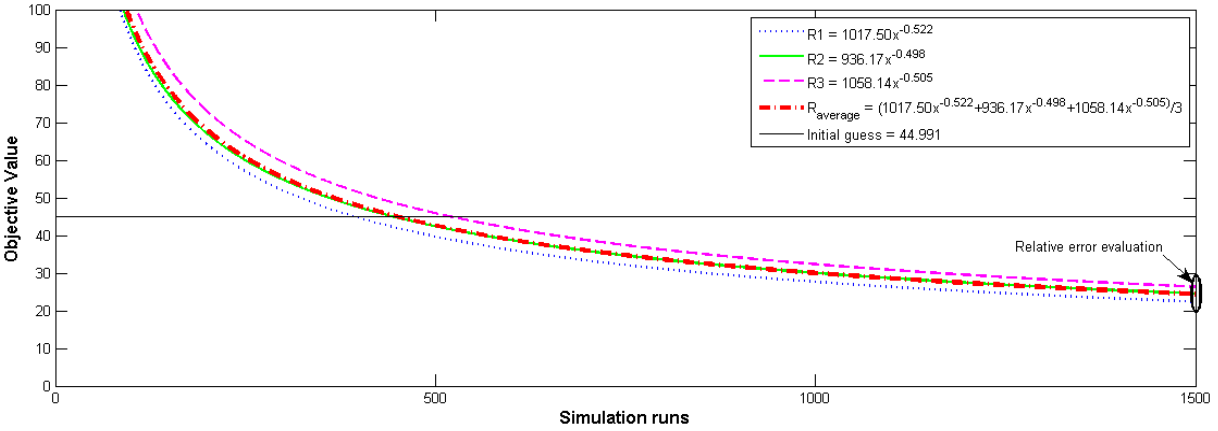


Figure 49: Example for averaging sample power regressions of Random Search

Regression averaging formulations

In the two tables below the power regression results are summarized for both the 1500 and 300 simulation run length. We only execute the regression analysis for the Gamma distribution, since it appeared to be the overall best statistical distribution regarding the achieved objective function values.

Table 8: Algorithm convergence speed approximation for 1500 simulation runs

Search scheme	Convergence function f_n	$R^2_{f_n}$	Final convergence function f_{final}
Random Search (RS)	$f_{RS_{1500_1}}(x) = 1017.50x^{-0.522}$	0.7774	$\boldsymbol{f_{RS_{1500}}(x) = \frac{1017.50x^{-0.522}+936.17x^{-0.498}+1058.14x^{-0.505}}{3}}$
	$f_{RS_{1500_2}}(x) = 936.17x^{-0.498}$	0.7532	
	$f_{RS_{1500_3}}(x) = 1058.14x^{-0.505}$	0.6994	
Latin Hypercube Sampling (LHS)	$f_{LHS_{1500_1}}(x) = 720.63x^{-0.498}$	0.7259	$\boldsymbol{f_{LHS_{1500}}(x) = \frac{720.63x^{-0.498}+542.47x^{-0.487}+803.56x^{-0.516}}{3}}$
	$f_{LHS_{1500_2}}(x) = 542.47x^{-0.487}$	0.6972	
	$f_{LHS_{1500_3}}(x) = 803.56x^{-0.516}$	0.7057	
Simulated Annealing (SA)	$f_{SA_{1500_1}}(x) = 228.82x^{-0.359}$	0.7413	$\boldsymbol{f_{SA_{1500}}(x) = \frac{228.82x^{-0.359}+280.04x^{-0.403}+335.91x^{-0.419}}{3}}$
	$f_{SA_{1500_2}}(x) = 280.04x^{-0.403}$	0.8157	
	$f_{SA_{1500_3}}(x) = 335.91x^{-0.419}$	0.7308	
LHS-SA combination (L&S)	$f_{L\&S_{1500_1}}(x) = 407.12x^{-0.438}$	0.6743	$\boldsymbol{f_{L\&S_{1500}}(x) = \frac{407.12x^{-0.438}+521.45x^{-0.461}+356.63x^{-0.402}}{3}}$
	$f_{L\&S_{1500_2}}(x) = 521.45x^{-0.461}$	0.6319	
	$f_{L\&S_{1500_3}}(x) = 356.63x^{-0.402}$	0.7011	

Table 9: Algorithm convergence speed approximation for 300 simulation runs

Search scheme	Convergence function f_n	$R^2_{f_n}$	Final average convergence function f_{final}
Random Search (RS)	$f_{RS_{1500_1}}(x) = 1063.99x^{-0.691}$	0.7382	$\boldsymbol{f_{RS_{300}}(x) = \frac{1063.99x^{-0.691}+1105.45x^{-0.688}+1241.78x^{-0.732}}{3}}$
	$f_{RS_{1500_2}}(x) = 1105.45x^{-0.688}$	0.7145	
	$f_{RS_{1500_3}}(x) = 1241.78x^{-0.732}$	0.7071	
Latin Hypercube Sampling (LHS)	$f_{LHS_{1500_1}}(x) = 937.38x^{-0.698}$	0.7189	$\boldsymbol{f_{LHS_{300}}(x) = \frac{937.38x^{-0.698}+889.52x^{-0.707}+902.36x^{-0.689}}{3}}$
	$f_{LHS_{1500_2}}(x) = 889.52x^{-0.707}$	0.6953	
	$f_{LHS_{1500_3}}(x) = 902.36x^{-0.689}$	0.6817	
Simulated Annealing (SA)	$f_{SA_{1500_1}}(x) = 540.63x^{-0.634}$	0.6049	$\boldsymbol{f_{SA_{300}}(x) = \frac{540.63x^{-0.634}+680.75x^{-0.657}+615.87x^{-0.664}}{3}}$
	$f_{SA_{1500_2}}(x) = 680.75x^{-0.657}$	0.5887	
	$f_{SA_{1500_3}}(x) = 615.87x^{-0.664}$	0.6231	
LHS-SA combination (L&S)	$f_{L\&S_{1500_1}}(x) = 1012.07x^{-0.734}$	0.816	$\boldsymbol{f_{L\&S_{300}}(x) = \frac{1012.07x^{-0.734}+856.78x^{-0.691}+851.61x^{-0.709}}{3}}$
	$f_{L\&S_{1500_2}}(x) = 856.78x^{-0.691}$	0.772	
	$f_{L\&S_{1500_3}}(x) = 851.61x^{-0.709}$	0.781	

We notice that all of the generated power regression models show a R^2-value of at least 58%. Thus, we believe that each of the individually found regression formulations has enough statistical substance to use them as a basis for further conclusions. In the next paragraphs, we look at the quality of the regression averaging with regards to the incurred relative error observed.

Regression averaging quality

Table 10 and Table 11 show the relative error assessment for both the 1500 and 300 simulation run length.

Table 10: Observed relative error for power regression of 1500 simulation runs (evaluation at last iteration)

Search scheme	Objective value (regression) $X_{1500_{1\dots3}}$	Mean $\bar{X}_{1500}(n)$	Standard deviation $S_{1500}(n)$	Relative error Z_{1500}
Random Search (RS)	20.667	20.532	1.381	**0.124**
	21.841			
	19.089			
Latin Hypercube Sampling (LHS)	17.494	16.997	1.069	**0.116**
	15.770			
	17.727			
Simulated Annealing (SA)	14.535	14.846	1.083	**0.133**
	16.052			
	13.953			
LHS-SA combination (L&S)	15.382	15.649	0.889	**0.104**
	16.641			
	14.927			

Table 11: Observed relative error for power regression of 300 simulation runs (evaluation at last iteration)

Search scheme	Objective value (regression) $X_{300_{1\dots3}}$	Mean $\bar{X}_{300}(n)$	Standard deviation $S_{300}(n)$	Relative error Z_{300}
Random Search (RS)	22.367	24.412	1.989	**0.149**
	24.528			
	26.340			
Latin Hypercube Sampling (LHS)	18.881	17.580	1.897	**0.198**
	15.403			
	18.457			
Simulated Annealing (SA)	16.568	15.649	0.935	**0.110**
	14.698			
	15.683			
LHS-SA combination (L&S)	16.542	17.768	1.163	**0.121**
	17.907			
	18.855			

We see that none of the averaged power regression formulations surpassed a relative error below 10%, which is often considered the standard. However, the achieved errors are not that far off that threshold. All of the averaged regression models vary between 10.4% and 19.8%. Nevertheless, we are convinced that we still can make a valid judgment about the convergence behavior of our calibration algorithms, since the maximum relative error does not exceed a value of 20% compared to the measured average of each calibration method. Table 12 shows the approximated number of regression replications that are needed to approach a relative error of less than 10% with a statistical confidence of 95% - which is based Formula (6). In total, this would add up to $\big((11-3)+(9-3)+(12-3)+(8-3)\big)*(1500*10*1*4)+\big((15-3)+(14-3)+(8-3)(10-3)\big)*(300*10*1*4)=2.1*10^6$ additional simulation runs ($\approx$ 25 extra computation days).

Table 12: Approximation of necessary replications for 1500/300 simulation runs to achieve $\gamma = 0.1$ with $1-\alpha = 0.95$

Search scheme	1500 simulation runs	300 simulation runs
Random Search (RS)	$\lceil 10.134 \rceil = 11$	$\lceil 14.870 \rceil = 15$
Latin Hypercube Sampling (LHS)	$\lceil 8.861 \rceil = 9$	$\lceil 13.750 \rceil = 14$
Simulated Annealing (SA)	$\lceil 11.921 \rceil = 12$	$\lceil 7.997 \rceil = 8$
LHS-SA combination (L&S)	$\lceil 7.229 \rceil = 8$	$\lceil 9.597 \rceil = 10$

Final algorithm convergence comparison

With the averaged power regression formulations per calibration search scheme, we are able to generate Figure 50 and Figure 51 below.

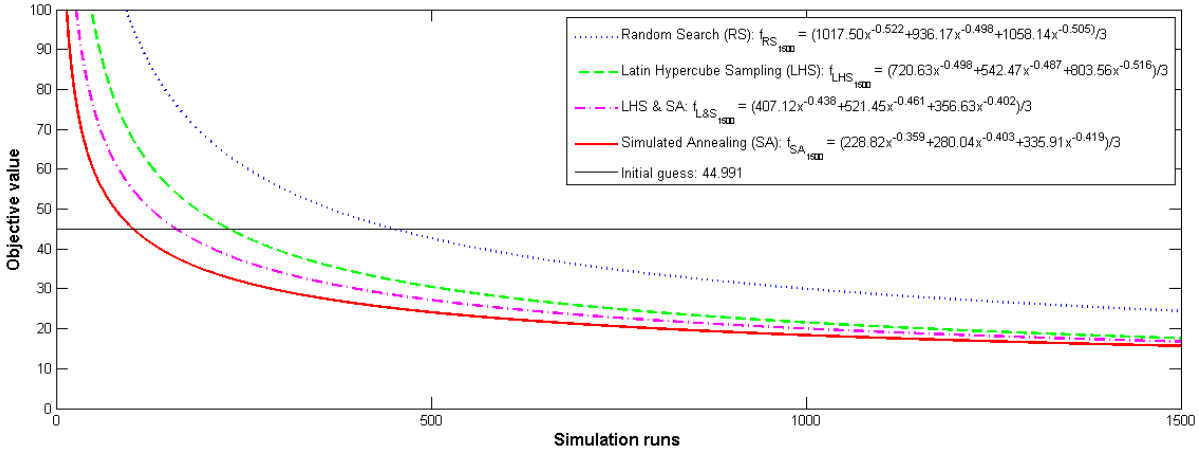


Figure 50: Final comparison of search scheme convergence for 1500 simulation runs

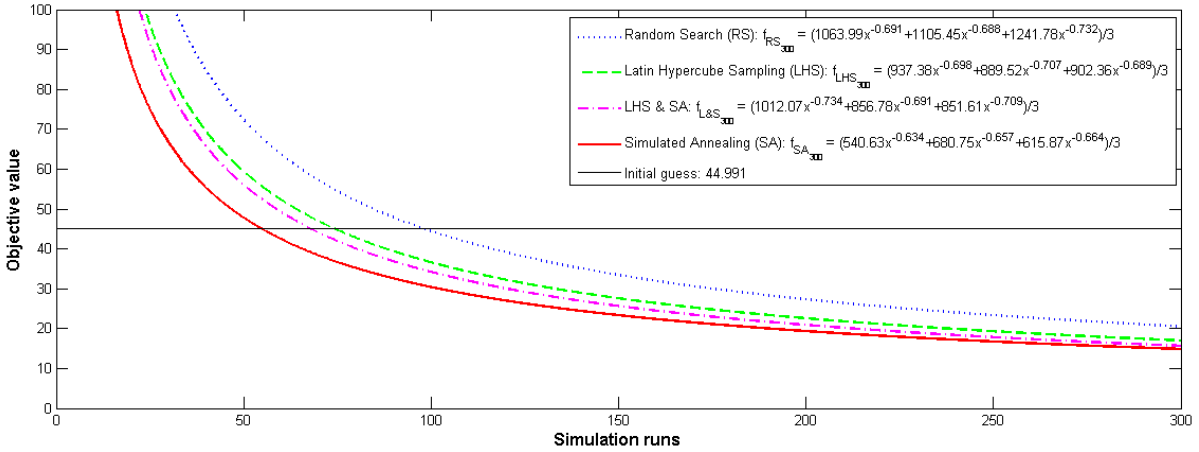


Figure 51: Final comparison of search scheme convergence for 300 simulation runs

Both figures above show clearly that the Simulated Annealing adaptation is the fasted converging calibration algorithm, which also finds the lowest minimum objective values of all the methodologies that we tested. As expected, Random Search is the worst automatic calibration method, which is surpassed in speed and convergence quality by Latin Hypercube Sampling. Unfortunately, the combination of Latin hypercube Sampling and Simulated Annealing did not show better convergence performance than any single algorithm performance. Thus, the LHS-SA combination just reached the second best place in our test set-up. A reason for this might be that the problem structure that we researched did not have many local minima. Figure 52 illustrates the problem structure with the Gamma distribution together with the candidate solution scattering of Simulated Annealing. If there would have been more minima, a longer scan of the entire search space might have had a positive effect on the search convergence – mainly regarding the final solution quality.

Nevertheless, due to the above-mentioned results, we believe that our adaptation of Simulated Annealing (see Figure 52) has the highest potential to be a useful automatic calibration algorithm in comparison to Random Search, Latin-Hypercube Sampling or the LHS-SA combination method. Sequential calibration definitely outperforms non-sequential approaches. In case of various local minima in the objective value surface, one could also decide to cool down SA even initially, which can have a similar effect such as the combination of LHS and SA to scan the objective surface longer for additional optima.

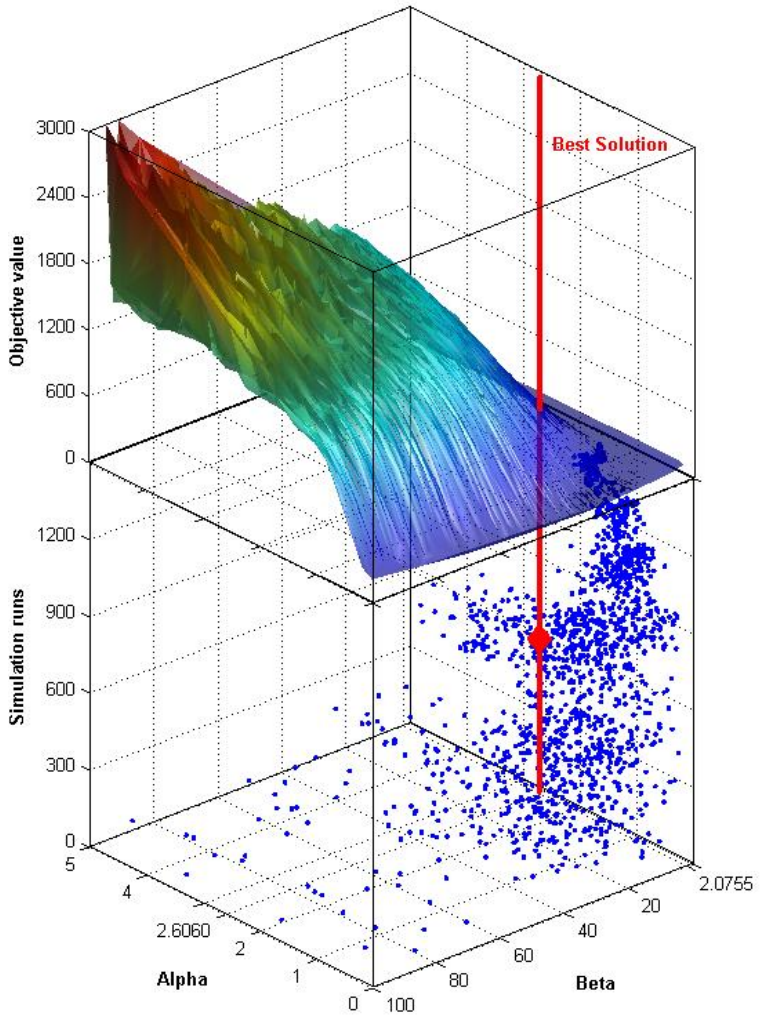


Figure 52: Simulated Annealing adaptation – Illustration of candidate solution scatter in the problem structure

7.2 Discussion

In this chapter, we presented our calibration results. Also we examined the scatter structure of candidate solutions generated by our four test calibration schemes: Random Search, Latin Hypercube Sampling, Simulated Annealing and the combination of Simulated Annealing and a Latin Hypercube. Furthermore, we conducted a regression analysis to draw conclusions about the convergence behavior per test algorithm.

In general, we see that all the automatic calibration methodologies performed better than the initial parameter estimation that we determined in Chapter 4. Ranked from best to worst, the first place takes our version of Simulated Annealing, which outperformed all the other calibration approaches clearly, followed by the combination algorithm, Latin Hypercube Sampling and, finally, Random Search. This ranking counts for both the algorithm convergence speed, as well as the convergence solution quality in the 1500/300 run length series.

Chapter 8: Conclusions & Recommendations

We evaluate our research process and conclude our results (Section 8.1) in this last chapter. Finally, we provide general recommendations (Section 8.2) on how to tackle calibration attempts –if they become necessary within Vanderlande Industries. This last section also includes further research suggestions for improvements of the introduced calibration methodologies.

8.1 Conclusions

We concluded that the inner-batch release delay in our high-level abstraction simulation model can be calibrated satisfactorily – the initial parameter estimation that we determined a-priori with histogram frequency matching could be improved by around 53% towards the lower bound of a "perfect" match regarding the benchmark data from practice. All of the tested automatic calibration approaches outperformed this initial parameter estimation.

We found that our Simulated Annealing adaptation (*SA*) is performing better than any other automatic calibration approach with regards to convergence speed as well as the overall solution quality. SA learns quickly where to look for in the feasible search space; however its cooling scheme is vital for its eventual success. This means that if the simulation run length is too short, the cooling occurs much too fast, so that the initially available search space is not adequately searched and SA might get stuck in a local minimum too quickly. In general, it appears that the longer the cooling period takes the better the calibration results become.

The combination of a Latin Hypercube Sampling design (*LHS*) and SA, however, turned out not to be beneficial in comparison to the singular use of SA in our test circumstances. This might be due to the structure of our problem, as explained in the previous chapter, since no large amount of local minimal could be identified. Even though we cannot verify it by our research, we believe that a longer "scanning" period of larger feasible search space can help SA to find a global optimum in the end – this, however, does not yet mean that always a "perfect" match is found with that.

LHS itself outperformed Random Search (*RS*), but had a worse convergence rate than the LHS-SA combination algorithm. As anticipated, it scattered the possible candidate solutions evenly over the entire feasible search space, which led to better results than we saw for RS, since search areas were not repetitively evaluated. Further, RS functioned as a sort of lower bound benchmark for algorithm convergence in our test set-up. LHS performed worse than SA, however, if only a small amount of experimental runs are allowed its results are still decent.

Based on the conclusions above, we suggest first to try to execute input validation with, e.g., the histogram frequency matching approach. Only if this leads to unclear validation results then calibration with an automatic calibration approach should be carried out. A model is normally better, i.e., closer to reality, if sufficient input data is incorporated originally.

If calibration attempts become necessary, we recommend the use of our version of Simulated Annealing (*SA*), in combination with the RSME measure for this purpose, since this is relatively easy to implement and showed that it can achieve decent calibration results in a rather short amount of time. It is essential to consider the incorporated cooling scheme with great care, since premature cooling might lead SA into a deadlock of a suboptimal local optimum. Thus, one should always try to use the longest possible time to calibrate a model, so that the cooling happens slowly – practical considerations and other circumstances in reality will, of course, restrain this aspect.

Random Search is the easiest automatic method to implement for calibration in an a-priori or in a sequential manner, however it should only be considered if high solution quality is not required. The better choice that sense is Latin Hypercube Sampling, however it needs more programming efforts to be implemented.

In general, expectations of a calibration model should be kept realistic, since a model remains a model no matter what. Nevertheless such a calibration model can help in many instances to understand reality better and to produce a more decently validated simulation models.

8.2 Recommendations

In addition to the model-outcome-related conclusions, we also want to mention several secondary findings that we discovered during this research. First of all, the main limitation of a calibration or simulation model of any kind is the availability of data input. The larger the amount of data incorporated in a model, the higher the likelihood that such a model approaches practice closer than another one which misses such data. Keeping this in mind, also the circumstance of a supposedly perfect fit is reflected in another light. This means that, even though a model shows to have a perfect calibration fit, there is still no guarantee that it is actually true for all possible situations.

Generally speaking, calibration attempts – automatic or manual – are only useful if a-priori data analysis cannot help to discover underlying statistical coherences. Thus, this is the case when a process that is to be modeled is lacking suitable input validation. Furthermore, automatic calibration has more advantages than manual calibration. The main benefits are that it is less arbitrary, transferable to other modelers, reproducible, shows often better solution performance, and is rather easy to implement if it is done systematically. Disadvantages of automatic calibration are that the search might take place near feasibility edges of the allowed parameter ranges. This might detach physical reality from the model if those ranges are not chosen with great care. Sensitivity analyses are therefore of high importance for a good calibration conduction. This is due to two reasons: non-collapsing experimental designs and a good choice of an objective weighing scheme in case of scalarized multi-objective calibration. Commonly, humans have trouble with choosing weights by “gut feeling”, thus a systematic approach such as Design of Experiments (DOE) or the Morris Method – if the focus is only put on main effects – are good and accepted methods for that.

Future research suggestions

We earlier noticed that the parameters of a certain distribution and the related objective value can be comparably visualized in 3D scatter plots through Delaunay triangulation (Delaunay, 1934). We show examples in Figure 53 to Figure 56.

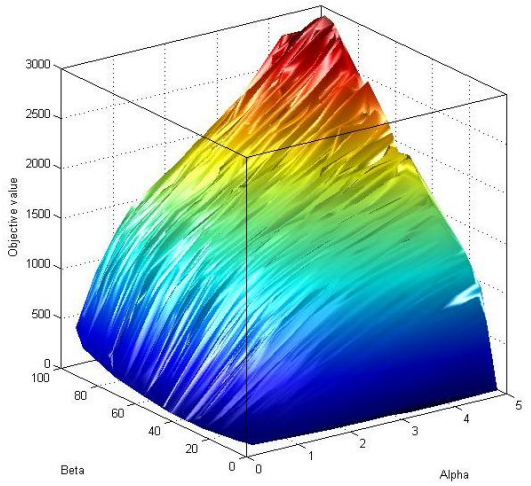


Figure 53: 3D plot of Gamma dist. (1500 rand. draws)

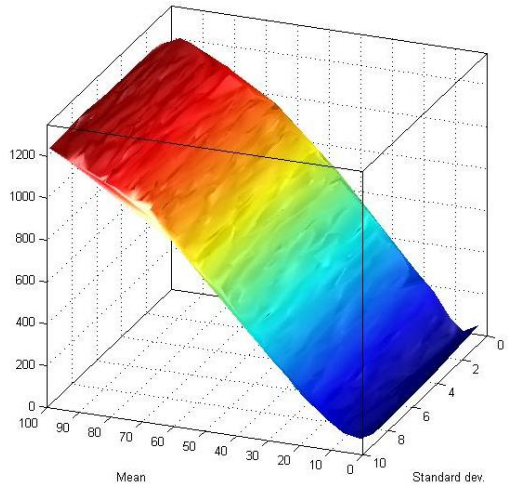


Figure 54: 3D plot of Normal dist. (1500 rand. draws)

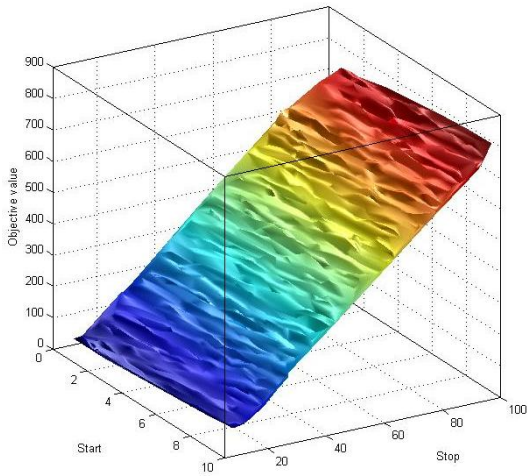


Figure 55: 3D plot of uniform dist. (1500 rand. draws)

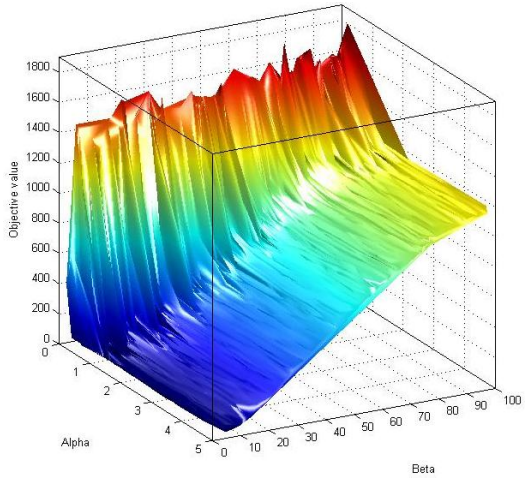


Figure 56: 3D plot of Weibull dist. (1500 rand. draws)

We believe that this knowledge can add to an improvement of our calibration methodologies. One of the various ways that we suggest to incorporate 3D gradient information regarding the tunable parameters is surface regression via a Vandermonde matrix. With this approach, a maximum smoothness of the studied surface can be achieved with a polynomial of the degree $n-1$, where n is the number of observations incorporated. We can formulate a surface function as shown in Equation (16). Knowing this formulation enables us to determine intermediate minima of the feasible search space that we defined a-priori. Nevertheless, the usage of this kind of regression in a time-efficient way for calibration still has to be researched.

$$Objective\ value(\alpha,\beta) = r_{n-1} * \alpha^{n-1} + \cdots + r_0 * \alpha^0 + p_{n-1} * \beta^{n-1} + \cdots + p_0 * \beta^0 \quad (16)$$

Also another look should be taken towards the most beneficial cooling scheme for Simulated Annealing. As we already noticed from the convergence comparison in Chapter 7, this is a key aspect for a decent final convergence quality. In general, we can remark that the slower SA is cooled, the better the eventual solution becomes. However, where exactly a proper trade-off exists between the cooling gradient and the solution quality still has to be determined in the future.

References

Abraham, A. and Goldberg, R. (2005). *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*. Berlin: Springer.

Agyei, E. and Hatfield, K. (2006). Enhancing gradient-based parameter estimation with an evolutionary approach. *Journal of Hydrology, 316*(1-4), 266-280. doi: 10.1016/j.jhydrol. 2005.05.010

Ayres, M. J. and Stamper, E. (1995). Historical development of building energy calculations. *ASHRAE Transactions, 101*(1), 47-55.

Balci, O. (1998). Verification, Validation and Testing. In J. Banks & J. Wiley (Eds.), *Handbook of Simulation* (pp. 243, 248). New York.

Banks, J., Carson, J. S., Nelson, B. L. and Nicol, D. M. (2005). *Discrete-event Simulation* (4th ed.). Upper Saddle River, Ney Jersey: Prentice-Hall.

Beck, M. B. (1991). Principles of modelling. *Water Science Technology, 24*(6), 1–8.

Bekele, E. G. and Nicklow, J. W. (2007). Multi-objective automatic calibration of SWAT using NSGA-II. *Journal of Hydrology, 341*(3-4), 165-176. doi: 10.1016/j.jhydrol.2007.05.014

Bendall, M. R. and Skinner, T. E. (1998). Calibration of STUD+ Parameters to Achieve Optimally Efficient Broadband Adiabatic Decoupling in a Single Transient. *Journal of Magnetic Resonance, 134*(2), 331-349. doi: http://dx.doi.org/10.1006/jmre.1998.1522

Beven, K. and Freer, J. (2001). Equifinality, data assimilation, and data uncertainty estimation in mechanistic modelling of complex environmental systems using the GLUE methodology. *Journal of Hydrology, 249*, 11-29.

Beven, K. J. (1993). Prophesy, reality and uncertainty in distributed hydrological modelling. *Adv. Water Resour, 16*(41-51).

Beven, K. J. and Binley, A. M. (1992). The future of distributed models: model calibration and uncertainty prediction. *Hydrol. Process, 6*, 279-298.

Boyle, D. P., Gupta, H. V. and Sorooshian, S. (2000). Toward improved calibration of hydrologic models: Combining the strengths of manual and automatic methods. *Water Resources Research, 36*(12), 3663-3674. doi: 10.1029/2000wr900207

Byers, J., Mack, C., Huang, R. and Jug, S. (2002). Automatic calibration of lithography simulation parameters using multiple data sets. *Microelectronic Engineering, 61-62*, 89-95. doi: 10.1016/s0167-9317(02)00438-0

Campbell, K. (2006). Statistical calibration of computer simulations. *Reliability Engineering & System Safety, 91*(10-11), 1358-1363. doi: 10.1016/j.ress.2005.11.032

Carson, J. S. (1986). Convincing Users of Model's Validity Is Challanging Aspect of Modeler's Job. *Ind. Eng., 18*, 74-85.

Carson, J. S. (2002). *Model Verification and Validation.* Paper presented at the Proc. 2002 Winter Simulation Conference, San Diego.

Chiu, Y., Zhou, L. and Song, H. (2010). Development and calibration of the Anisotropic Mesoscopic Simulation model for uninterrupted flow facilities. *Transportation Research, 44*(2010), 152-174.

Cooper, V. A., Nguyen, V. T. V. and Nicell, J. A. (1997). Evaluation of global optimisation methods for conceptual rainfall–runoff model calibration. *Water Science and Technology, 36*(5), 53-60.

Delaunay, B. (1934). Sur la sphère vide (Vol. 7, pp. 793–800). Izvestia Akademii Nauk SSSR: Otdelenie Matematicheskikh i Estestvennykh Nauk.

Doherty, J. and Johnston, J. M. (2003). Methodologies for calibration and predictive analysis of a watershed model. *Journal of the American Water Resources Association, 39*(2), 251-265.

Dorigo, M. (1992). *Optimization, Learning and Natural Algorithms.* PhD thesis, Politecnico di Milano, Milan, Italy.

Duan, Q. S., Sorooshian, S. and Gupta, V. K. (1992). Effective and efficient global optimization for conceptual rainfall runoff models. *Water Resources Research, 28*(4), 1015–1031.

Feltner, C. E. and Weiner, S. A. (1985). Models, Myths and Mysteries in Manufacturing. *Ind. Eng., 17*, 66-76.

Fishman, G. S. and Kiviat, P. J. (1968). The Statistics of Discrete-Event Simulation. *Simulation, 10*, 185-195.

Freedman, V. L., Lopes, V. L. and Hernandez, M. (1998). Parameter identifiability for catchment-scale erosion modelling: a comparison of optimization algorithms. *Journal of Hydrology, 207*, 83-97.

Gaume, E., Villeneuve, J. P. and Desbordes, M. (1998). Uncertainty assessment and analysis of the calibrated parameter values of an urban storm water quality model. *J. Hydrol, 210*(38–50).

Geng, Z., Yang, F. and Wu, N. (2011). Optimum design of sensor arrays via simulation-based multivariate calibration. *Sensors and Actuators B: Chemical, 156*(2), 854-862. doi: 10.1016/j.snb.2011.02.054

George, M. (2002). *Lean Six Sigma: Combining Six Sigma Quality with Lean Production Speed*. New York: McGraw-Hill Education - Europe

Gibbons, J. D. (1985). *Nonparametric Methods for Quantitative Analysis* (2nd ed.). Columbus, Ohio: American Sciences Press.

Gibbons, J. D. and Chakraborti, S. (2003). *Nonparametric Statistical Inference* (4th ed.). New York: Marcel Dekker.

Gill, M. K., Kaheil, Y. H., Khalil, A., McKee, M. and Bastidas, L. (2006). Multiobjective particle swarm optimization for parameter estimation in hydrology. *Water Resources Research, 42*. doi: 10.1029/ 2005WR004528.

Gilson, S. J., Fitzgibbon, A. W. and Glennerster, A. (2011). An automated calibration method for non-see-through head mounted displays. [Research Support, Non-U.S. Gov't Validation Studies]. *J Neurosci Methods, 199*(2), 328-335. doi: 10.1016/j.jneumeth.2011.05.011

Glover, F. (1989). Tabu Search - Part 1. *ORSA Journal on Computing, 1*(2), 190-206.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimisation, and Machine Learning*. Reading, MA: Addison-Wesley Publishing Co.

Graefe, J., Schmidt, S., Heißner, A., Rusin, W. and Wonneberger, C. (2005). Simulation of Soil Heating in Ridges partly covered with Plastic Mulch, Part II: Model Calibration and Validation. *Biosystems Engineering, 92*(4), 495-512. doi: 10.1016/j.biosystemseng.2005.08.001

Grenzebach Maschinenbau GmbH. (2012). Brochure - Automated Baggage Handling. Asbach-Bäumenheim, Germany.

Gupta, H. V., Bastidas, L. A., Sorooshian, S., Shuttleworth, W. J. and Yang, Z. L. (1999). Parameter estimation of a land surface scheme using multicriteria methods. *Journal of Geophysical Research D: Atmospheres, 104*(D16), 19491-19503.

Gupta, H. V., Kling, H., Yilmaz, K. K. and Martinez, G. F. (2009). Decomposition of the mean squared error and NSE performance criteria: implications for improving hydrological modelling. *Journal of Hydrology, 377*(1-2), 80.

Gupta, H. V., Sorooshian, S. and Yapo, P. O. (1998). Toward improved calibration of hydrologic models: Multiple and noncommensurable measures of information. *Water Resources Research, 34*(4), 751-763.

Harmon, R. and Challenor, P. (1997). A Markov chain Monte Carlo method for estimation and assimilation into models. *Ecological Modeling, 101*, 41-59.

Heerkens, H. and van Winden, A. (2012). *Geen probleem: Een aanpak voor alle bedrijfskundige vragen en mysteries*. Buren: Business School Nederland.

Henderson, D. A., Boys, R. J., Krishnan, K. J., C., L. and J., W. D. (2009). Bayesian emulation and calibration of a stochastic computer model of mitochondrial DNA deletions in substantia nigra neurons. *Journal of the American Statistical Association, 104*, 76–87.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Ann Arbor, MI: University of Michigan Press.

Huang, Y., Seck, M. D. and Verbraeck, A. (2010). Towards automated model calibration and validation in rail transit simulation. *Procedia Computer Science, 1*(1), 1259-1265. doi: 10.1016/j.procs.2010.04.140

Hughes, D. A. (2004). Incorporating groundwater recharge and discharge functions into an existing monthly rainfall–runoff model. *Hydrological Sciences Journal, 49*(2), 297–311.

Hughes, D. A., Parsons, R. and Conrad, J. (2007). Quantification of the Groundwater Contribution to Baseflow: RSA Water Research Commission.

Husslage, B., Rennen, G., van Dam, E. R. and den Hertog, D. (2006). Space-filling Latin Hypercube Designs for Computer Experiments. Tilburg, NL: Tilburg University.

Hyndman, R. J. and Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting, 22*(4), 679–688.

Ito, S.-i., Yoshie, N., Okunishi, T., Ono, T., Okazaki, Y., Kuwata, A., et al. (2010). Application of an automatic approach to calibrate the NEMURO nutrient–phytoplankton–zooplankton food web model in the Oyashio region. *Progress in Oceanography, 87*(1-4), 186-200. doi: 10.1016/j.pocean.2010.08.004

Jagner, D., Renman, L. and Stefansdottir, S. H. (1993). Determination of iron(III) and titanium(IV) as their Solochrome Violet RS complexes by constant-current stripping potentiometry: Part 1. Automated single-point calibration method for iron(III). *Analytica Chimica Acta, 281*(2), 305-314. doi: http://dx.doi.org/10.1016/0003-2670(93)85186-N

Janssen, P. H. M. and Heuberger, P. S. C. (1995). Calibration of process-oriented models. *Ecological Modeling, 83*, 55-66.

Jiang, Y., Li, X. and Huang, C. (2013). Automatic calibration a hydrological model using a master–slave swarms shuffling evolution algorithm based on self-adaptive particle swarm optimization. *Expert Systems with Applications, 40*(2), 752-757. doi: 10.1016/j.eswa.2012. 08.006

Jie, L., Van Zuylen, H., Chen, Y., Viti, F. and Wilmink, I. (2012). Calibration of a microscopic simulation model for emission calculation. *Transportation Research Part C: Emerging Technologies*. doi: 10.1016/j.trc. 2012.04.008

Kanso, A., Chebbo, G. and Tassin, B. (2006). ApplicationofMCMC–GSAmodelcalibration method tourbanrunoffqualitymodeling. *Reliability Engineering & System Safety, 91*, 1398–1405.

Kanungo, T. and Zheng, Q. (2004). Estimating degradation model parameters using neighborhood pattern distributions: an optimization approach. San Jose, CA 95120, USA: IBM Almanden Research Center.

Kennedy, J. and Eberhart, R. (1995). *Particle Swarm Optimization.* Paper presented at the 1995 IEEE International Conference on Neural Networks, Perth, Australia.

Kennedy, M. C. and O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society, 63*, 425-450.

Khu, S. T., di Pierro, F., Savic, D., Djordjevic, S. and Walters, G. A. (2006). Incorporating spatial and temporal information for urban drainage model calibration: an approach using Pareto preference ordering genetic algorithm. *Advances in Water Resources Research, 29*, 1168–1181.

Khu, S. T. and Madsen, H. (2005). Multiobjective calibration with Pareto preference ordering: An application to rainfall–runoff model calibration. *Water Resources Research, 41*(W03004).

Kirkpatrick, S., Gelatt Jr, C. D. and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220* (4598), 671-680.

Kleijnen, J. P. C. (1992). Sensitivity Analysis of Simulation Experiments: Regression Analysis and Statistical Design. *Mathematics and Computers in Simulation, 32*, 297-315.

Kleijnen, J. P. C. (1995). Verification and validation of simulation models. *European Journal of Operational Research, 82*, 145-162.

Kleijnen, J. P. C., Sanchez, S. M., Lucas, T. W. and Cioppa, T. M. (2005). A user's guide to the brave new world of designing simulation experiments. *INFORMS Journal on Computing, 17*(3), 263-289. doi: 10.1287/ ijoc.1050.0136

Kolmogorov, A. (1933). Sulla determinazione empirica di una legge di distribuzione (Vol. 4, pp. 83): G. Inst. Ital. Attuari.

Kong, C. Y., McMahon, P. M. and Gazelle, G. S. (2009). Calibration of disease simulation model using an engineering approach. [Research Support, N.I.H., Extramural, Non-U.S. Gov't]. *Value Health, 12*(4), 521-529.doi: 10.1111/j.1524-4733.2008.00484.x

Kuczera, G. (1997). Efficient subspace probabilistic parameter optimization for catchment models. *Water Resource Research, 33*(1), 177-185.

Kuroda, H. and Kishi, M. J. (2003). A data assimilation technique applied to "NEMURO" for estimating parameter values. *Ecological Modelling, 172*(69-85).

Lamb, R. and Kay, A. L. (2004). Confidence intervals for a spatially generalized, continuous simulation flood frequency model for Great Britain. *Water Resources Research, 40*(7), W075011-W0750113.

Larsen, R. J. and Marx, M. L. (2006). *An Introduction to Mathematical Statistics and Its Applications* (4th ed.). Upper Saddle River: Pearson Prentice Hall, Pearson Education International.

Law, A. M. (2005). *How to Build Valid and Credible Simulation Models.* Paper presented at the Proc. 2005 Winter Simulation Conference, Orlando.

Law, A. M. (2007). *Simulation Modeling & Analysis* (fourth ed.). New York: McGraw Hill Education.

Li, X., Weller, D. E. and Jordan, T. E. (2010). Watershed model calibration using multi-objective optimization and multi-site averaging. *Journal of Hydrology, 380*(3-4), 277-288. doi: 10.1016/j.jhydrol.2009.11.003

Li, Y., Brimicombe, A. J. and Li, C. (2008). Agent-based services for the validation and calibration of multi-agent models. *Computers, Environment and Urban Systems, 32*(6), 464-473. doi:10.1016/j.compenvurbsys. 2008.09.002

Linden, J., Vinsonneau, B. and Burnham, K. J. (2005). *Review and enhancement of cautious parameter estimation for model based control: a specific realisation of regularization.* Paper presented at the 18th International Conference on Systems Engineering, Luxembourg, Luxembourg.

Lith, I. v., McMenamin, L., Ploemen, M., Thoonen, P. and Wieringen, T. v. (2012). Functional Design Simulation Final Report. Veghel: Vanderlande Industries B.V.

Liu, G. and Liu, M. (2011). A rapid calibration procedure and case study for simplified simulation models of commonly used HVAC systems. *Building and Environment, 46*(2), 409-420. doi:10.1016/j.buildenv. 2010.08.002

Liu, Y. (2009). Automatic calibration of a rainfall–runoff model using a fast and elitist multi-objective particle swarm algorithm. *Expert Systems with Applications, 36*(5), 9533-9538. doi:10.1016/j.eswa. 2008.10. 086

Liu, Y., Khu, S. T. and Savic, D. A. (2004). A fast hybrid optimisation method of multi-objective genetic algorithm and k-nearest neighbour classifier for hydrological model calibration. *Lecture Notes in Computer Science (LNCS), 3177*, 546–551.

Liu, Y. and Sun, F. (2010). Sensitivity analysis and automatic calibration of a rainfall–runoff model using multi-objectives. *Ecological Informatics, 5*(4), 304-310. doi: 10.1016/j.ecoinf.2010.04.006

Madsen, H. (2000). Automatic calibration of a conceptual rainfall–runoff model using multiple objectives. *Journal of Hydrology, 235*(3–4), 276-288. doi: http://dx.doi.org/10.1016/S0022-1694(00)00279-1

Madsen, H. (2003). Parameter estimation in distributed hydrological catchment modelling using automatic calibration with multiple objectives. *Advances in Water Resources, 26*(2), 205-216. doi:10.1016/s0309-1708(02)00092-1

Matear, R. J. (1995). Parameter optimization and analysis of ecosystem models using simulated annealing: a case study at Station P. *Journal of Marine Research, 53*(571-607).

Mazzotti, F. J. and Vinci, J. J. (2007). Validation, verification, and calibration: Using standardized terminology when describing ecological models: Wildlife Ecology and Conservation Department, University of Florida.

McKay, M. D., Beckman, R. J. and Conover, W. J. (1979). A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics, 21*(2), 239-245. doi: 10.2307/1268522

McMenamin, L. and de Jongh, S. (2006). Functional Design Simulation Start Document. Veghel: Vanderlande Industries.

Montgomery, D. C. (1991). *Design and Analysis of Experiments* (3rd ed.). New York: Wiley.

Montgomery, D. C. and Runger, G. C. (2002). *Applied Statistics and Probability for Engineers* (3rd ed.). New York: Wiley.

Moore, C. and Doherty, J. (2006). The cost of uniqueness in groundwater model calibration. *Advances in Water Resources, 29*(4), 605-623. doi: 10.1016/j.advwatres.2005.07.003

Morris, M. D. (1991). Factorial Sampling Plans for Preliminary Computational Experiments. *Technometrics, 33*, 161-174.

Moussu, F., Oudin, L., Plagnes, V., Mangin, A. and Bendjoudi, H. (2011). A multi-objective calibration framework for rainfall–discharge models applied to karst systems. *Journal of Hydrology, 400*(3-4), 364-376. doi: 10.1016/j.jhydrol.2011.01.047

Mwelwa, E. M. (2004). *The application of the monthly step Pitman Rainfall–Runoff Model to the Kafue River Basin of Zambia.* Master thesis, Rhodes University, South Africa.

Nash, J. E. and Sutcliffe, J. V. (1970). River flow forecasting through conceptual models. Part 1:Adiscussion of principles. *Journal of Hydrology, 10*, 282-290.

Naylor, T. H. and Finger, J. M. (1967). Verification of Computer Simulation Models. *Management Science, 14*, 92-101.

Ndiritu, J. (2009). A comparison of automatic and manual calibration using the Pitman model. *Physics and Chemistry of the Earth, Parts A/B/C, 34*(13-16), 729-740. doi: 10.1016/j.pce.2009. 06.002

Oberkampf, W. L. and Roy, C. J. (2010). *Verification and Validation in Scientific Computing*. Cambridge: Cambrige University Press.

Paulo, R., García-Donato, G. and Palomo, J. (2012). Calibration of computer models with multivariate output. *Computational Statistics & Data Analysis, 56*(12), 3959-3974. doi: 10.1016/j.csda.2012.05.023

Pearson, K. (1900). On a Criterion That a Given System of Deviations from the Probable in the Case of a Correlated System of Variables Is Such that It Can Be Reasonably Supposed to Have Arisen in Random Sampling. *Philosophical Magazine, 50*, 157-175.

Pokhrel, P., Yilmaz, K. K. and Gupta, H. V. (2012). Multiple-criteria calibration of a distributed watershed model using spatial regularization and response signatures. *Journal of Hydrology, 418-419*, 49-60. doi:10.1016/j.jhydrol.2008.12.004

Qingyun, D., Sorooshian, S. and Gupta, V. (1992). Effective and efficient global optimization for conceptual rainfall- runoff models. *Water Resources Research, 28*(4), 1015-1031.

Rode, M., Suhr, U. and Wriedt, G. (2007). Multi-objective calibration of a river water quality model—Information content of calibration data. *Ecological Modelling, 204*(1-2), 129-142. doi:10.1016/ j.ecolmodel.2006.12.037

Rose, K. A., Megrey, B. A., Werner, F. E. and Wared, D. M. (2007). Calibration of the NEMURO nutrient–phytoplankton–zooplankton food web model to a coastal ecosystem: Evaluation of an automated calibration approach. *Ecological Modelling, 202*, 39-51.

Rykiel, E. J. (1996). Testing ecological models: The meaning of validation. *Ecological Modeling, 90*, 224–229.

Saltelli, A., Chan, K. and Scott, E. M. (2000). *Sensitivity analysis*. Chichester: Wiley.

Sargent, R. G. (1996). *Some Subjective Validation Methods Using Graphical Displays of Data.* Paper presented at the 1996 Winter Simulation Conference, Coronado, CA.

Sargent, R. G. (2004). *Verification and Validation of Simulation Models.* Paper presented at the Proc. 2004 Winter Simulation Conference, Washington, DC.

Sargent, R. G. (2008). Verification and Validation of Simulation models: Proceedings of the 2008 Winter Simulation Conference, pp. 157 – 169.

Schelsinger *et al.* (1979). Terminology for Model Credibility. *Simulation, 32*, 103-104.

Schrijver, A. (2003). *Combinatorial Optimization: Polyhedra and Efficiency* (Vol. 24): Springer.

Sen, M. K., Dattagupta, A., Stoffa, P. L., Lake, L. W. and Pope, G. A. (1995). Stochastic reservervior modeling using simulated annealing and genetic algorithms. *SPE Formation Evaluation, 10*(1), 49-55.

Shannon, R. E. (1975). *Systems Simulation: The Art and Science*. Enlewood Cliffs, New Jersey: Prentice Hall.

Shrestha, R. S. and Rode, M. (2008). Multi-objective calibration and fuzzy preference selection of a distributed hydrological model. *Environmental Modelling & Software, 23*(12), 1384-1395. doi:10.1016/j.envsoft. 2008.04.001

Singh, A., Minsker, B. S. and Goldberg, D. E. (2004). *Combining reliability and Pareto optimality – an approach using stochastic multiobjective genetic algorithm.* Paper presented at the 2004 World Congress on Water and Environmental Resources,, Salt Lake City, UT.

Skahill, B. E. and Doherty, J. (2006). Efficient accommodation of local minima in watershed model calibration. *Journal of Hydrology, 329*(1-2), 122-139. doi: 10.1016/j.jhydrol.2006.02.005

Smirnov, N. V. (1948). Tables for estimating the goodness of fit of empirical distributions. *Annals of Mathematical Statistics, 19*, 279.

Spear, R. C. (1997). Large simulation models: calibration, uniqueness and goodness of fit. *Environmental Modelling & Software, 12*(2–3), 219-228. doi: http://dx.doi.org/10.1016/ S1364-8152(97)00014-5

Statnikov, R. B. and Matusov, J. B. (1995). *Multicriteria Optimization and Engineering*. Ney York: Chapman and Hall.

Stephens, M. A. (1974). EDF Statistics for Goodness of Fit and Some Comparisons. *J. Am. Statist. Assoc., 69*, 730-737.

Straatman, B., White, R. and Engelen, G. (2004). Towards an automatic calibration procedure for constrained cellular automata. *Computers, Environment and Urban Systems, 28*(1-2), 149-170. doi: 10.1016/s0198-9715(02)00068-6

Thoonen, P., van Wieringen, T. and van Lith, I. (2012). 70MB Backbone - Detailed Performance Description. Veghel: Vanderlande Industries B.V.

Vallino, J. J. (2000). Improving marine ecosystem models: use of data assimilation and mesocosm experiments. *Journal of Marine Research, 58*, 117-164.

Van der Meulen, J. (2010). Software User Manual RA_LTM_BCM (Product Development SWH, Trans.) (pp. 40). Veghel, the Netherlands: Vanderlande industries.

Van Griensven, A. and Bauwens, W. (2003). Multi-objective autocalibration for semidistributed water quality models. *Water Resource Research, 39*(12), 1348. doi: 10.1029/2003WR002284

Van Griensven, A. and Bauwens, W. (2005). Application and evaluation of ESWAT on the Dender basin and the Wister lake basin. *Hydrol, Process*, 827–838.

Van Horn, R. L. (1971). Validation of Simulation Results. *Management Science, 17*, 247-258.

Vanderlande Industries B.V. (2010a). Zuid: Batching - regie, Niveau 3 Module 13R. Veghel.

Vanderlande Industries B.V. (2010b). Zuid: Routering - systeembeheer, Niveau 3, Module 4S. Veghel.

Vanderlande Industries B.V. (2012a). E-Commerce from A-Z: Zalando also places reliance upon Vanderlande Industries Retrieved 20.07, 2012, from http://www.vanderlande.com/en/ WarehouseAutomation /Industry-segments/Ecommerce.htm

Vanderlande Industries B.V. (2012b). Extensive V-model (Engineering). *Instructions & Training* Retrieved 12 December, 2012, from http://nlveg-st-office/download/documents/a_ doc069652/en/lastreleased/ viewable/a_doc069652-en.pdf

Vanderlande Industries B.V. (2012c). Manage a Project. *Instructions & Training* Retrieved 12 December, 2012, from http://nlvega53:8080/websites/VIProcessMappages/AreaManagea Project.htm

Vanderlande Industries B.V. (2012d). Material Flow Diagram (MFD) of Schiphol South. In M. E. Schiphol-Zuid (Ed.). Veghel, the Netherlands: Vanderlande Industries B.V.

Vanderlande Industries B.V. (2012e). Products & Solutions, from http://www.vanderlande.com/en/ Baggage-Handling/Products-and-Solutions.htm

Vanderlande Industries B.V. (2012f). Realize on-site. *Instructions & Training* Retrieved 12 December, 2012, from http://nlvega53:8080/websites/VIProcessMappages/AreaRealizeon Site.htm

Vanderlande Industries B.V. (2012g). VI Process map. *Instructions & Training* Retrieved 12 December, 2012, from http://nlvega53:8080/websites/VI%20Process%20Map.html

Vanni, T., Legood, R. and White, R. G. (2010). Calibration of Disease Simulation Model Using an Engineering Approach. *Value in Health, 13*(1), 157.

Wagener, T. and Wheater, H. S. (2006a). Parameter estimation and regionalization for continuous rainfall-runoff models including uncertainty. *Journal of Hydrology, 320*(1–2), 132-154. doi:http://dx.doi.org/ 10.1016/j.jhydrol.2005.07.015

Wagener, T. and Wheater, H. S. (2006b). Parameter estimation and regionalization for continuous rainfall–runoff models including uncertainty. *J. Hydrol., 320*(132-154).

Wang, Q. J. (1991). The genetic algorithm and its application to calibrating conceptual rainfall-runoff models. *Water Resources Research, 27*, 2467–2471.

Ward, B. A., Friedrichs, M. A. M., Anderson, T. R. and Oschlies, A. (2010). Parameter optimisation techniques and the problem of underdetermination in marine biogeochemical models. *Journal of Marine Systems, 81*, 34-43. doi: 10.1016/j.jmarsys.2009.12.005.

Weinstein, M. C. (2006). Recent developments in decision-analytic modelling for economic evaluation. *PharmacoEconomics, 24*, 1043–1053.

White, R. (1995). *Multi-scale spatial modelling of self-organzing urban systems*. Paper presented at the International Twin-Conference on Complexity and Self-Organisation, Stuttgart.

Wong, D. F., Leong, H. W. and Liu, C. L. (1988). *Simulated Annealing for VLSI Design*. Boston, MA: Kluwer Academic.

Yan, J. and Haan, C. T. (1991). Multiobjective parameter estimation for hydrologic models. *Trans. ASAE, 34*(1), 135-141.

Yapo, P. O., Gupta, H. V. and Sorooshian, S. (1998). Multi-objective global optimisation for hydrologic models. *Journal of Hydrology, 204*, 83-97.

Youssef, H., Sait, S. M. and Adiche, H. (2001). Evolutionary algorithms, simulated annealing and tabu search: A comparative study. *Engineering Applications of Artificial Intelligence, 14*(2001), 167–181.

Yu, P. and Yang, T. (2000). Fuzzy multi-objective function for rainfall-runoff model calibration. *Journal of Hydrology, 238*(1–2), 1-14. doi: http://dx.doi.org/10.1016/S0022-1694(00)00317-6

Yuan, J. and Szu Hui, N. (2013). A sequential approach for stochastic computer model calibration and prediction. *Reliability Engineering & System Safety, 111*, 273-286. doi: 10.1016/j.ress.2012.11.004

Yum, B.-J. and Lee, S.-H. (1991). Calibration procedures when both measurements are subject to error: A comparative simulation study of the unreplicated case. *Computers & Industrial Engineering, 20*(4), 411-420. doi: http://dx.doi.org/10.1016/0360-8352(91)90013-V

Zitzler, E. and Thiele, L. (1998). Multiobjective optimisation using evolutionary algorithms – a comparative study. *IEEE Transactions on Evolutionary Computation, 3*(4), 257-271.

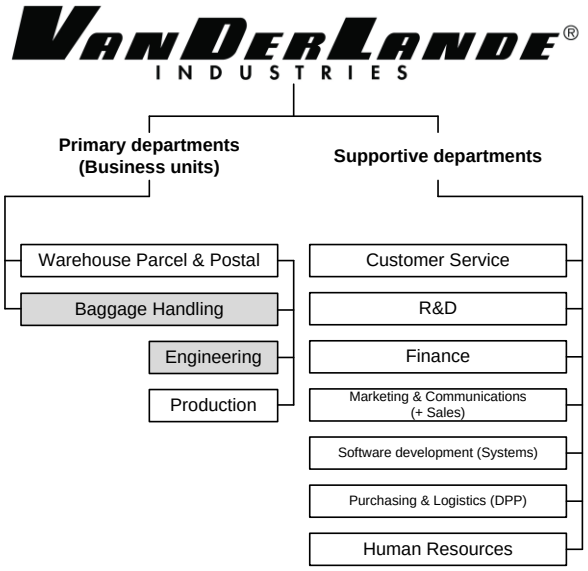
Appendix

Table of Appendices

A1 GENERAL INFORMATION ABOUT VANDERLANDE INDUSTRIES ..... K
A2 MAIN OUTLINE OF PROCESSES AT VANDERLANDE INDUSTRIES ..... K
A3 MANAGERIAL PROBLEM SOLVING METHOD (MPSM) – BRIEF SUMMARY & DEFINITIONS L
A4 CONTROL ENVIRONMENT OF THE LUGGAGE BATCHING PROCESS AT SCHIPHOL SOUTH ..... N
A5 PROGRESSION OF LUGGAGE BATCHING PROCESS AT SCHIPHOL SOUTH ..... N
A6 EXTENDED PROBLEM CHART ..... O
A7 LAYOUT OF SCHIPHOL SOUTH – LUGGAGE BUFFER TO BAGGAGE HANDLING ROBOTS ..... P
A8 SUMMARY OF DEVIATIONS BETWEEN SIMULATION MODEL AND IMPLEMENTED SYSTEM ..... Q
A9 ADDITIONAL DATA ANALYSIS PLOTS ..... R

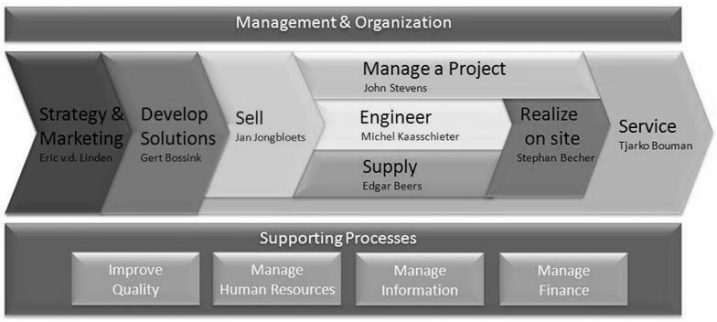
A1 General information about Vanderlande Industries

Company structure

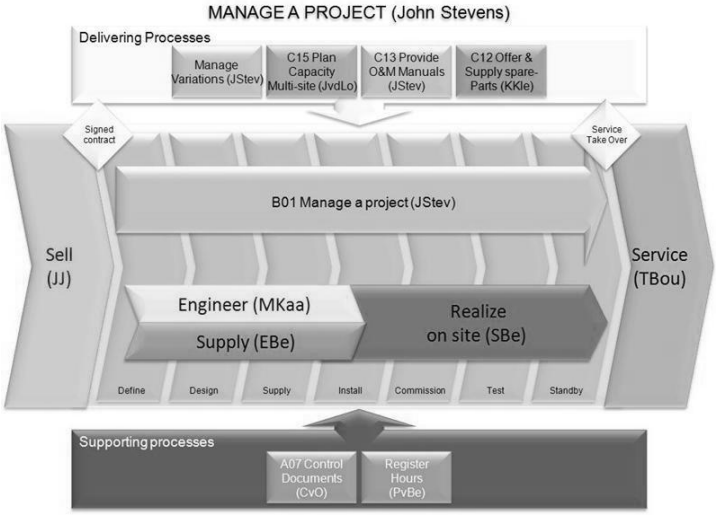


This research is conducted in the grey-shaded division / department.

A2 Main outline of processes at Vanderlande industries



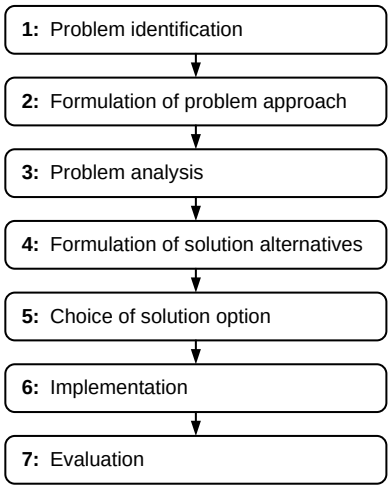
(Vanderlande Industries B.V., 2012g)



(Vanderlande Industries B.V., 2012c)

A3 Managerial Problem Solving Method (MPSM) – Brief summary & definitions

Heerkens and van Winden (2012) have developed a seven-stage approach to conduct an elaboration on an economical or process-related problem most efficiently (see figure below).



Initially, the above-mentioned method starts with the problem identification phase, where the actual core problem (definition follows below) is figured out of a set of possible acting problems that the principal of the anticipated research is facing. These acting problems normally will be visualized in a problem chart that acts like a funnel – on top general problems are mentioned while towards to bottom the problems become more specific. The problem chart can also be called a "fishbone" or "Ishikawa" diagram or a root-cause-analysis. For this research, the problem chart can be found on page 6. Out of the lowest level acting problems, the one with the highest impact factor is selected. This happens upon the following criteria: general feasibility (problem can be potentially solved by the researcher), there is no other lower level acting problem that is causing the potential core problem and the solution of the core problem has to significantly be able to add to the solution of the main problem that is experienced by the project principal.

Thereafter, the research goal and the related problem statement are formulated. The latter represents the main knowledge problem that has to be resolved by the ongoing research. A knowledge problem is a problem that normally has to be tackled by a literature research or experimentation (a more specific definition can be found below). The solution of the problem statement is further supported by the formulation of research (sub-) questions to keep track of various issues that play a role during the treatment of the problem statement. These questions can also be seen as additional in-depth knowledge problems.

After the definition of the knowledge problems, the actual research can begin, which is represented by the second phase of Heerkens et al.'s methodology; the "formulation of the problem approach". Within this phase, the research project is planned in detail, including the intended process steps and manners to resolve the prior-defined problem statement and the related research questions.

The succeeding phase "problem analysis" is trying to illuminate the depth of the prior-recognized core problem. In here the problem size and including components are generally quantified.

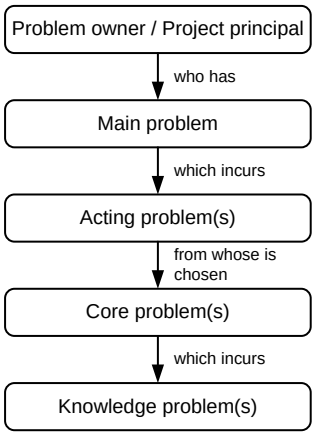
After knowing the depth of the research problem, the next mythological phase considers solution alternatives that can be generated to resolve the core problem. This phase also includes deeper data analysis and experimentation.

In the fifth phase of the MPSM, the added value of the obtained solution alternatives from phase four are weighed against each other and the "best" option – based upon the criteria that have been defined in corporation with the problem owner (respectively "principal" according Verschuren & Doorewaard).

When the choice of the problem solution alternative is confirmed, the following stage of the MPSM incurs a detailed action plan for the implementation of the chosen solution in practice. This includes temporal, financial and resource issues that are connected to a realization of the desired solution option.

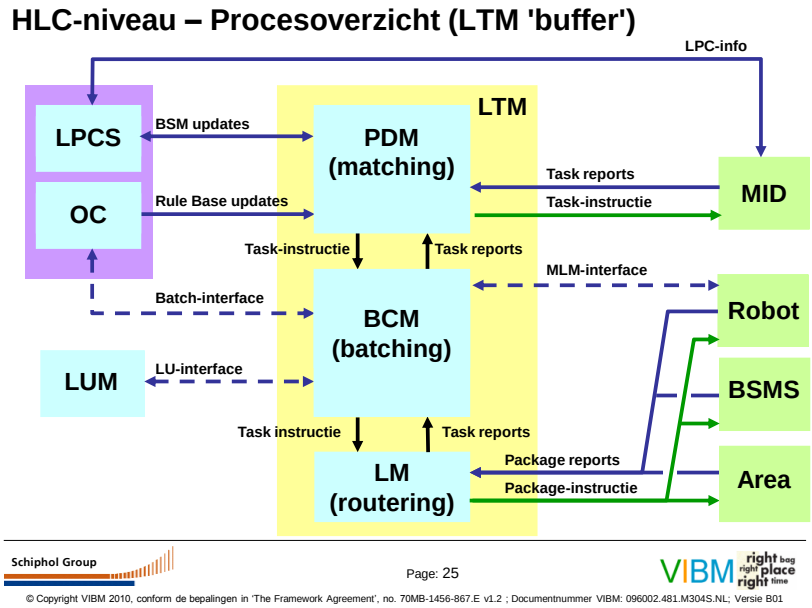
In the last phase of the MPSM, the entire project is subjected to a recapitulation that assesses the qualities, shortcoming and trade-offs that were experienced throughout the execution of the research and the implementation process.

In the further course of this thesis, the subsequent terms main problem, acting problem, core problem and knowledge problem, as well as problem owner or principal will be used, thus a brief definition is provided in the following paragraphs to reduce interpretation mistakes:



- **Problem owner or principal:**
  This is the person or group of people that is mainly responsible for the outcomes of the ongoing research and also the people that are mainly involved realization of the project goal.
- **Main problem:**
  The main problem is the most noticeable issue for the problem owner which ought to be resolved.
- **Acting problem:**
  An acting problem is a subsequent, thus lower level, issue to the main problem experienced by the problem owner, which is actually feasible to be resolved, but has not been considered / solved yet.
- **Core problem:**
  A core problem is the most significant and impactful in-depth (lowest level) acting problem that can resolved by a research. The formulation of this problem type is the basis of the definition of the project goal, as well as the general problem statement. This problem type is chosen upon the following four criteria: General feasibility (the problem and its outcome are able to be influenced somehow), highest added value towards the main problem resolution, no other deeper acting problem causing the core problem and the solution to this problem is non-trivial
- **Knowledge problem:**
  A knowledge problem will be subsequently defined after the core problem is chosen. The most prominent one will be the problem statement which overall constraints the project scope. Further, research (sub-) questions are formulated which assist to solve the problem statement eventually. A knowledge problem is in general non-trivial and normally has to be resolved by an extensive literature study or in-depth experimentation.

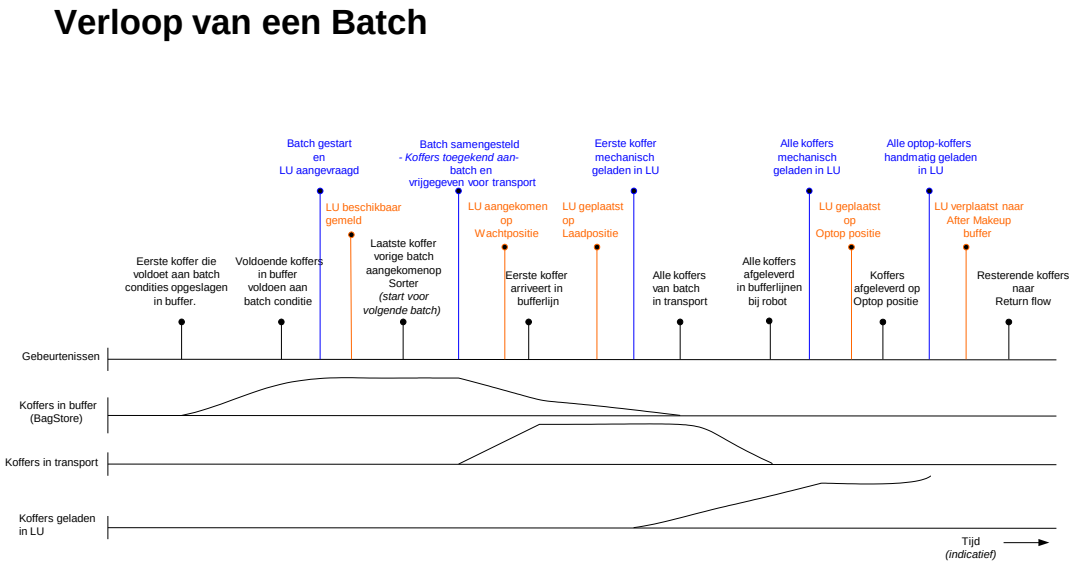
A4 Control environment of the luggage batching process at Schiphol South



(Vanderlande Industries B.V., 2010b)

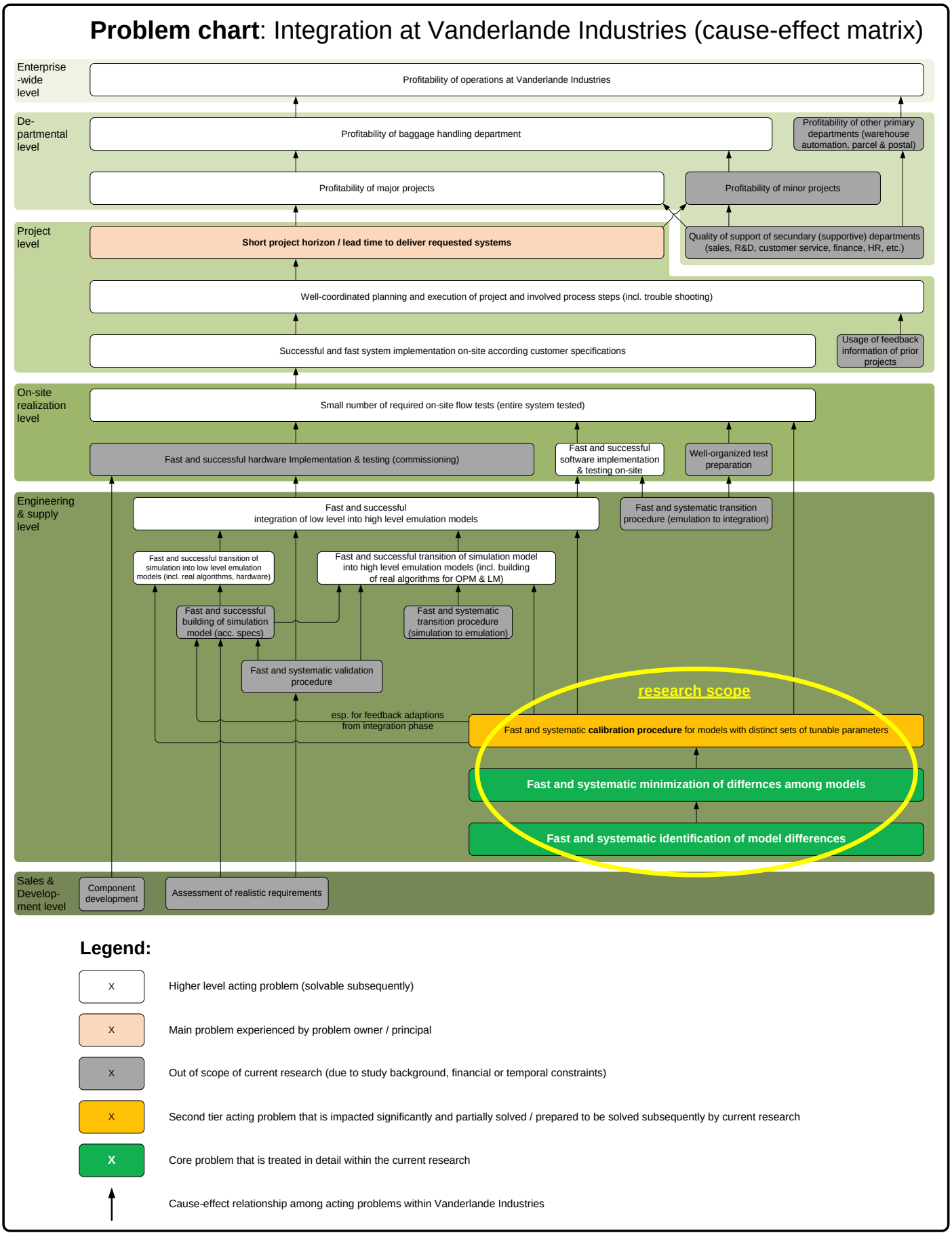
A5 Progression of luggage batching process at Schiphol South

(in Dutch)



(Vanderlande Industries B.V., 2010a)

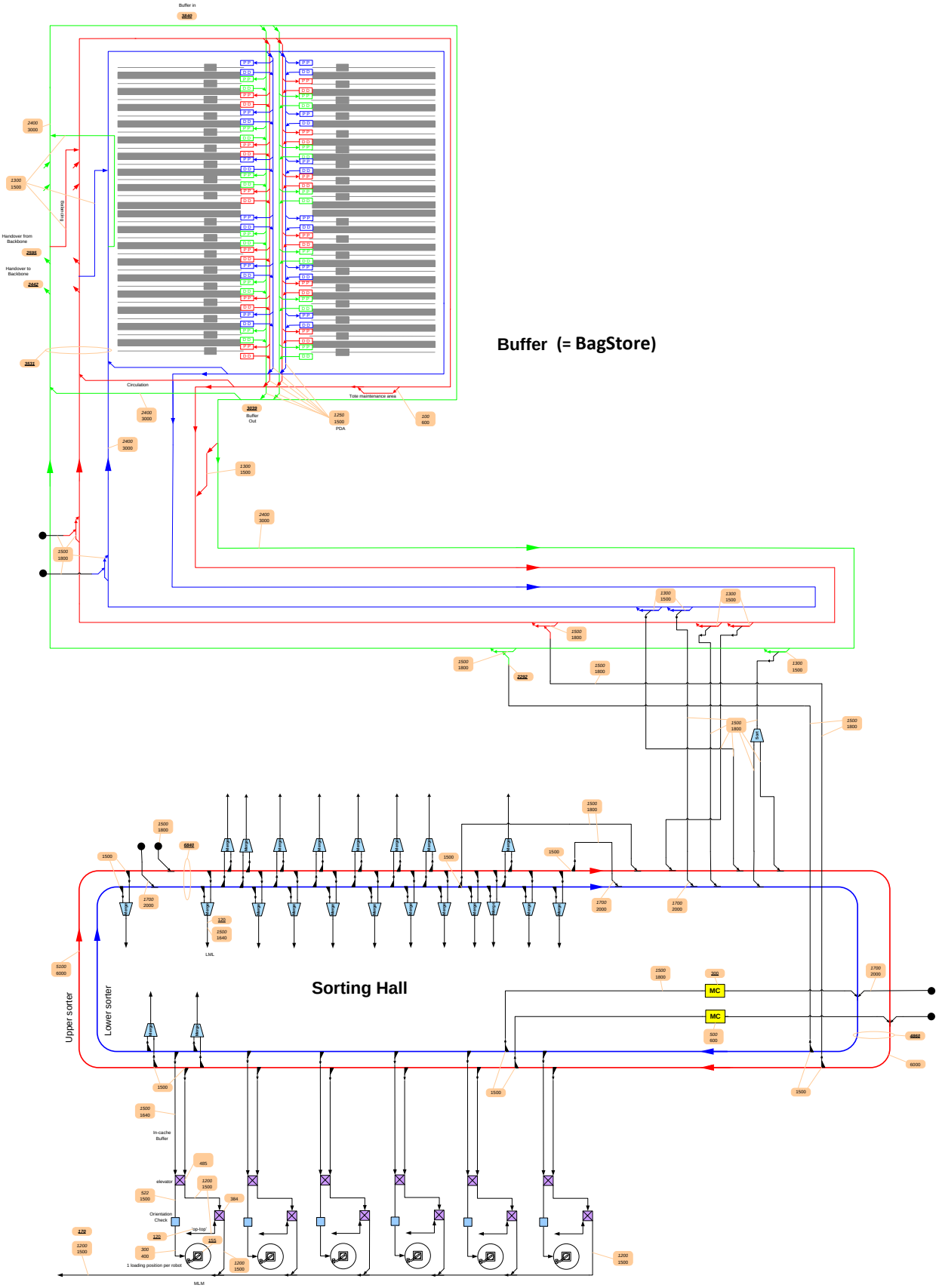
A6 Extended problem chart



Legend:

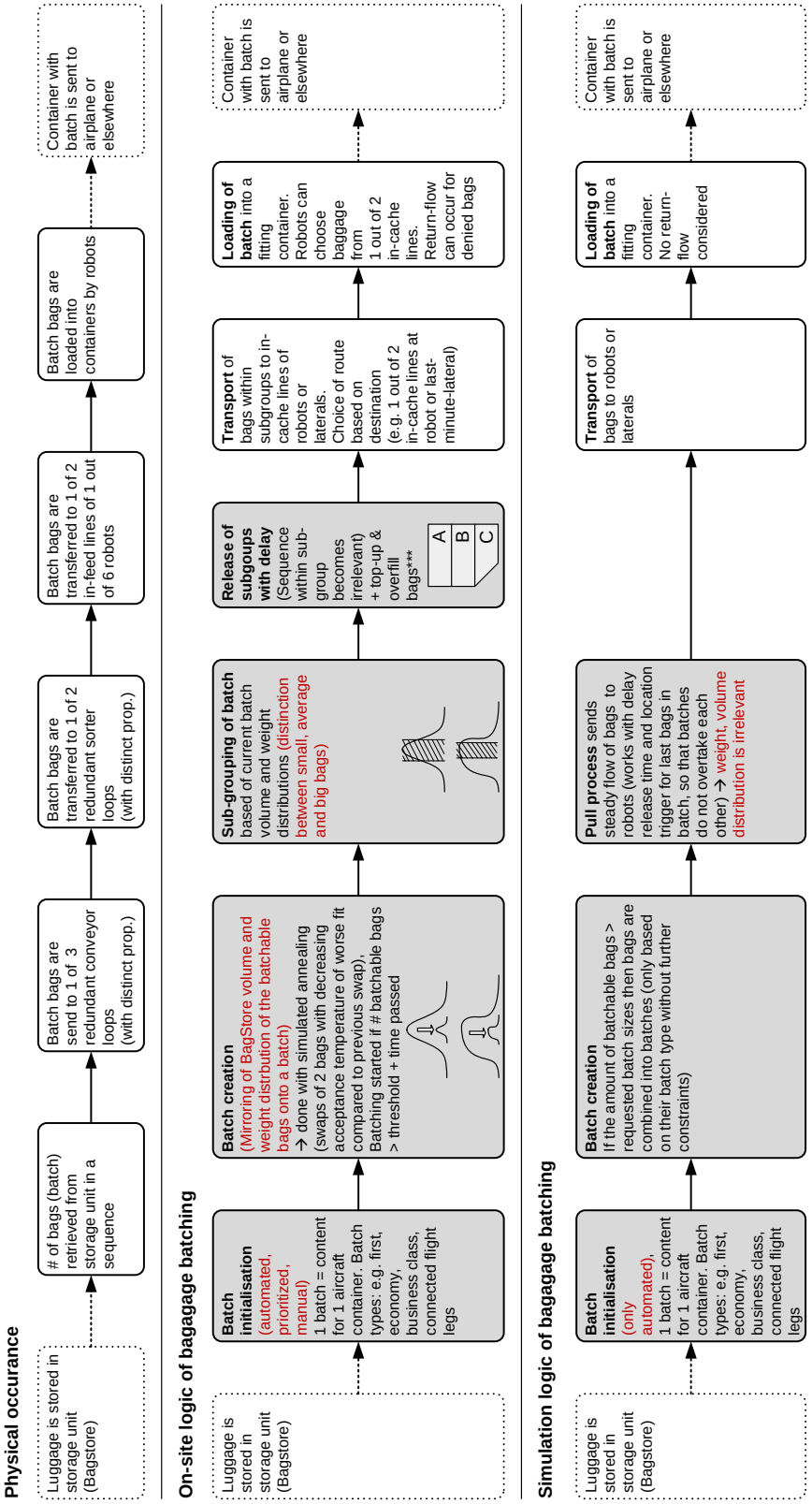
X	Higher level acting problem (solvable subsequently)
X	Main problem experienced by problem owner / principal
X	Out of scope of current research (due to study background, financial or temporal constraints)
X	Second tier acting problem that is impacted significantly and partially solved / prepared to be solved subsequently by current research
X	Core problem that is treated in detail within the current research
↑	Cause-effect relationship among acting problems within Vanderlande Industries

A7 Layout of Schiphol South – Luggage buffer to baggage handling robots



(Vanderlande Industries B.V., 2012d)

A8 Summary of deviations between simulation model and implemented system



*** “Top-up” = bags that are manually added to a container, if batch is too small, “Overfill” = additional bag sent to robots in case too many bags fall out of the batch

A9 Additional data analysis plots

Batch request arrivals

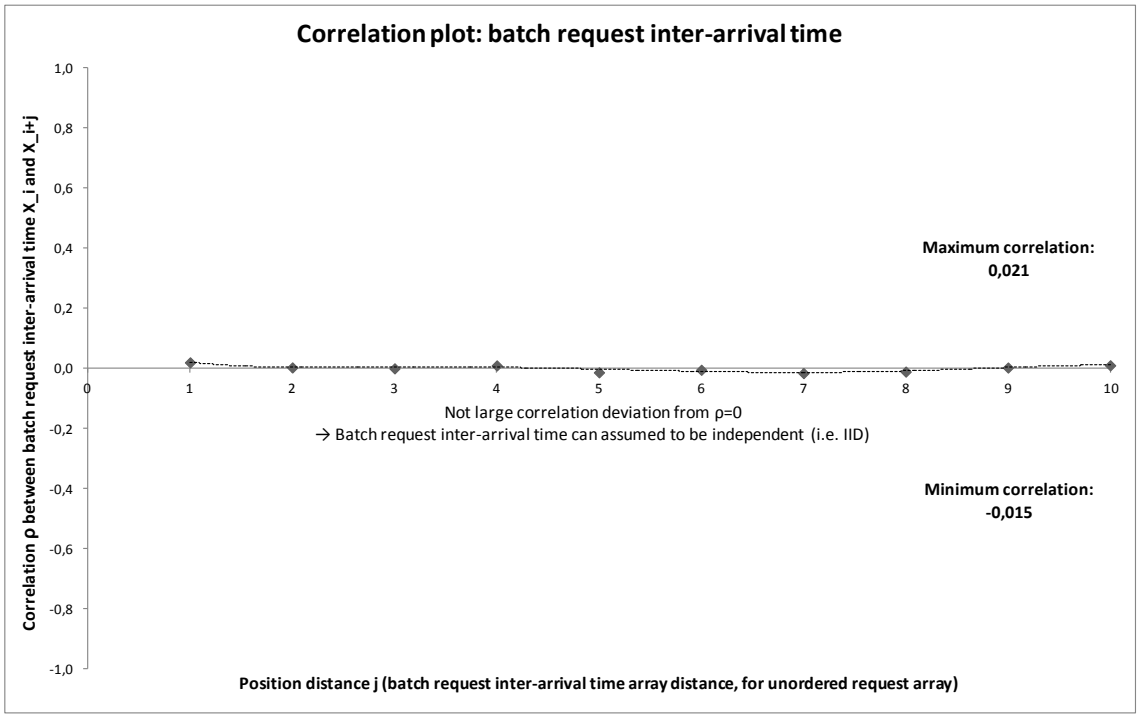


Figure 57: Test on auto-correlation of batch requests - Correlation plot

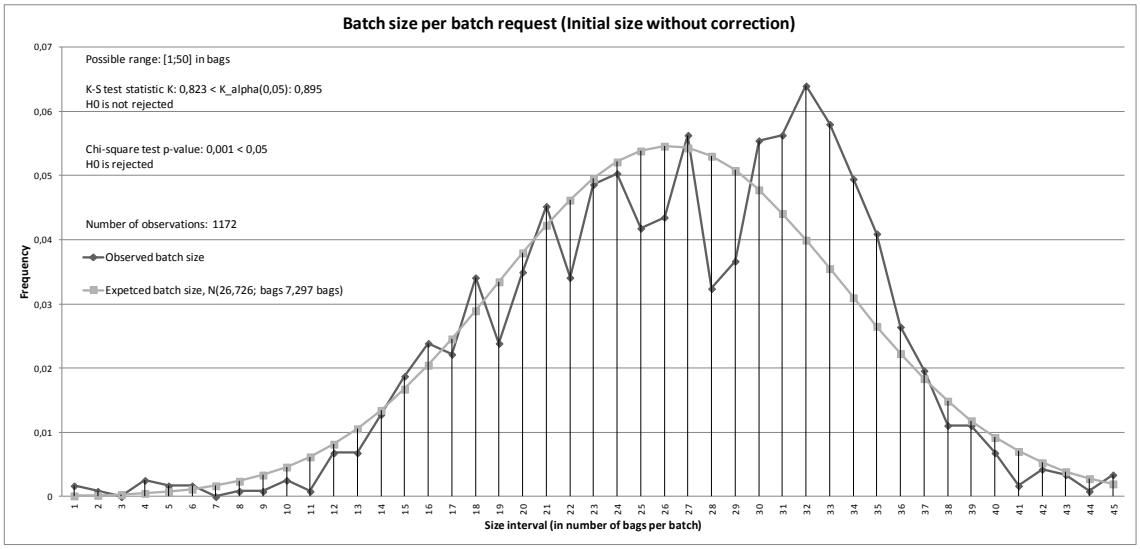


Figure 58: Probability density function of initially planned batch size

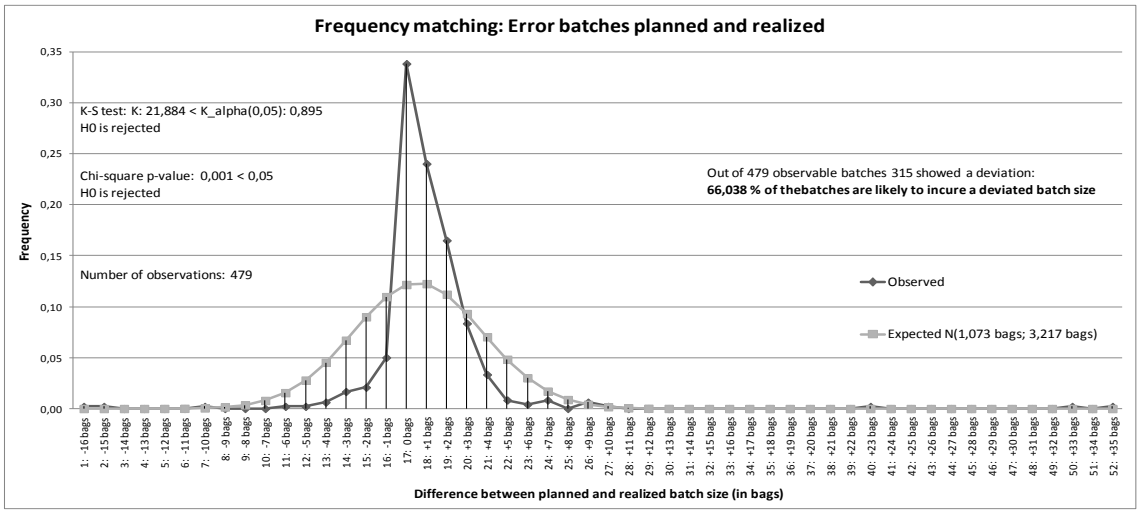


Figure 59: Unquantifiable error between planned and realized batch size

Batch composition

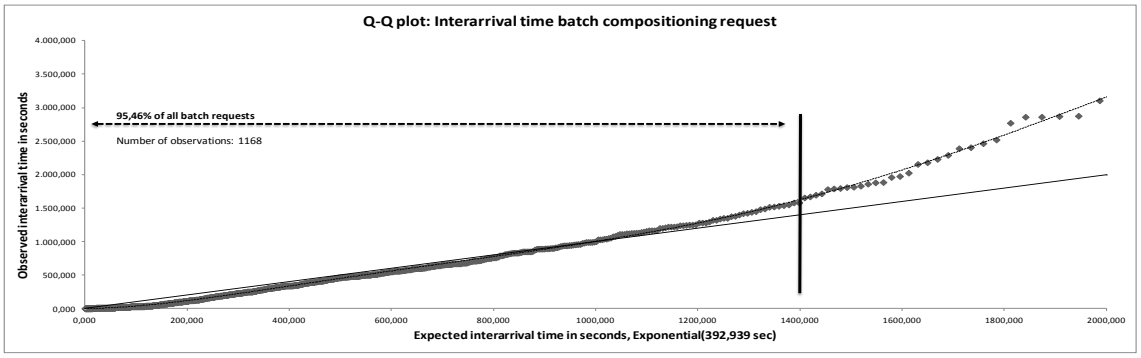


Figure 60: Q-Q plot of batch composition requests – focus on right tail matching

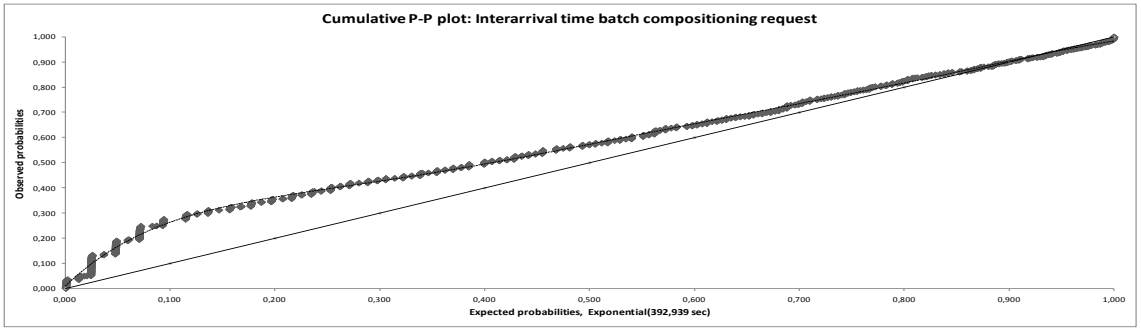


Figure 61: P-P plot of batch composition requests - focus on large value and middle curve matching

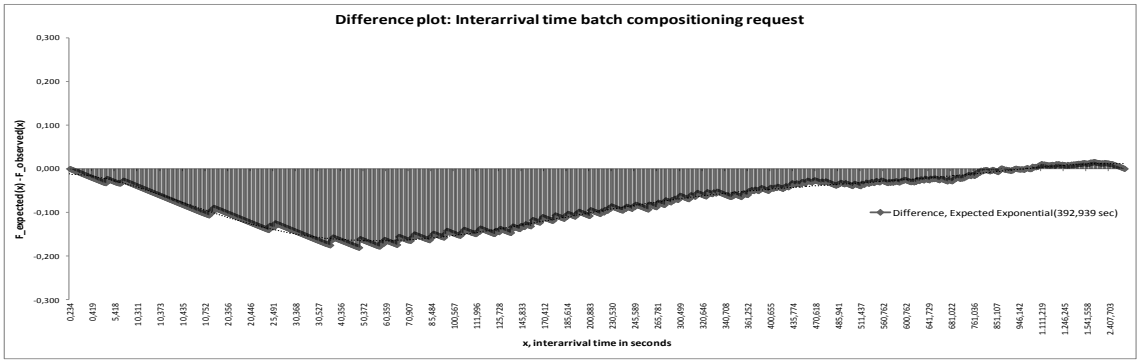


Figure 62: Difference plot of batch composition requests – Amplification of large size deviations

Batch size

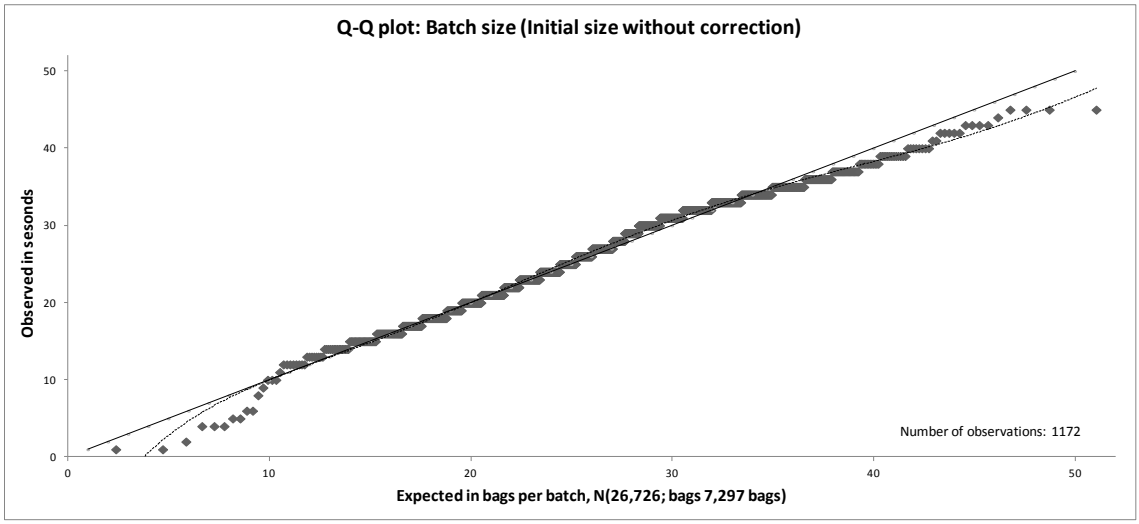


Figure 63: Q-Q plot of initially planned batch size

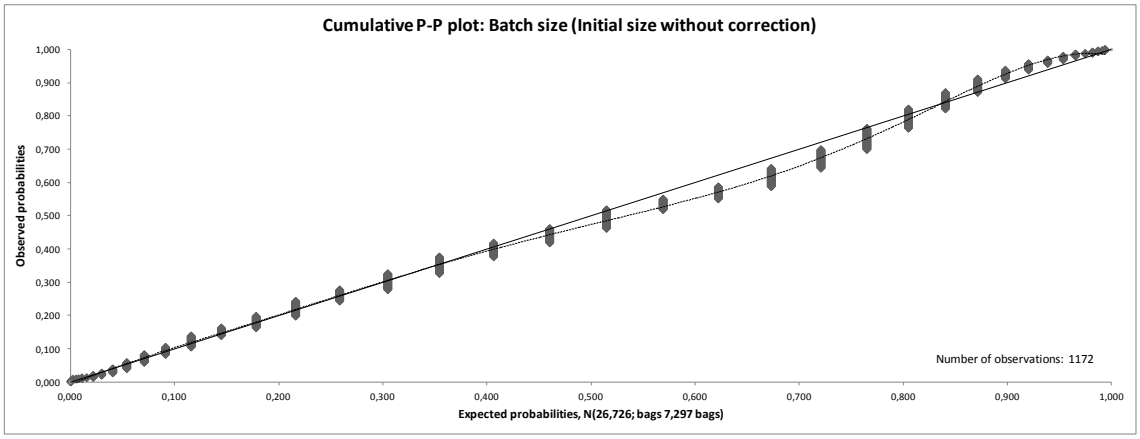


Figure 64: P-P plot of initially planned batch size

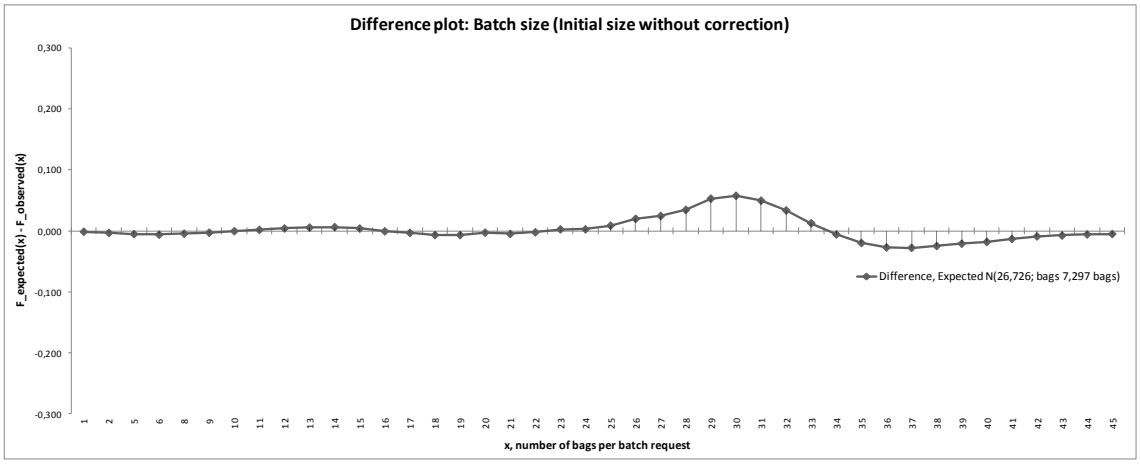


Figure 65: Difference plot of initially planned batch size

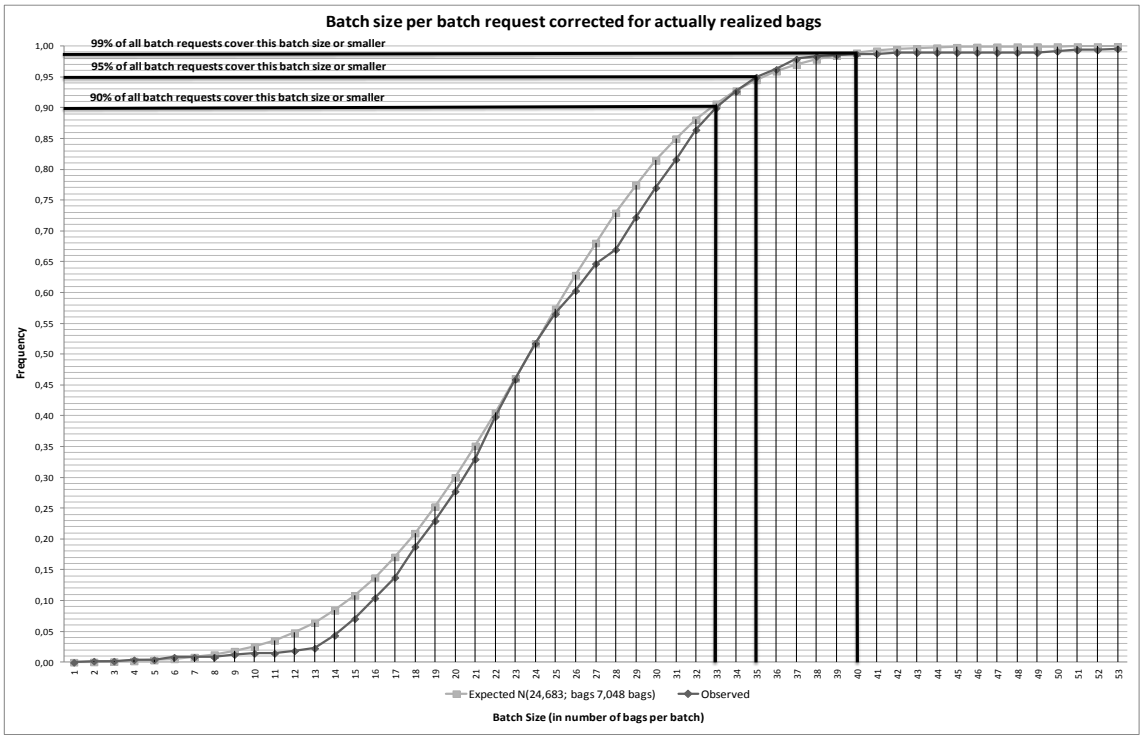


Figure 66: Cumulative density function of corrected batch size

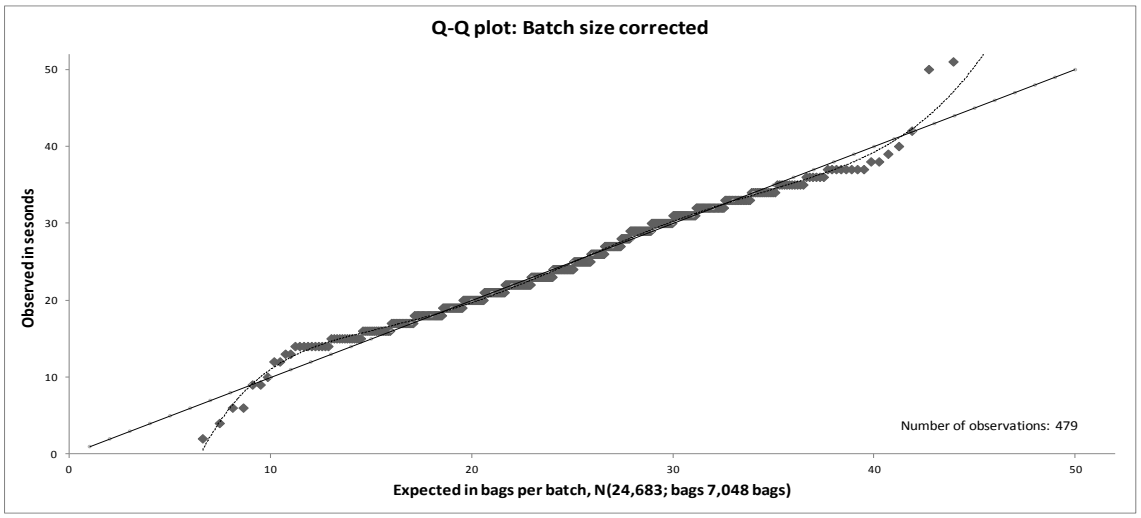


Figure 67: Q-Q plot of corrected batch size

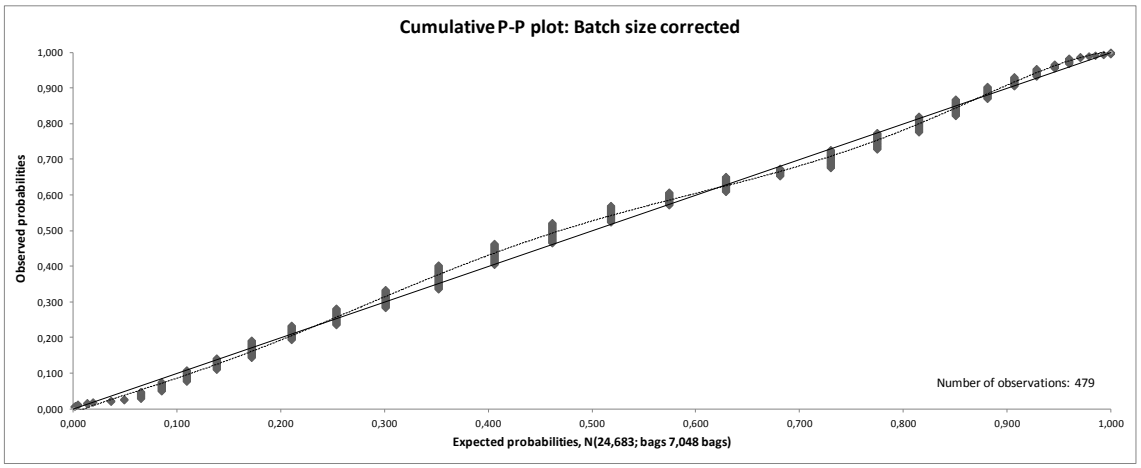


Figure 68: P-P plot of corrected batch size

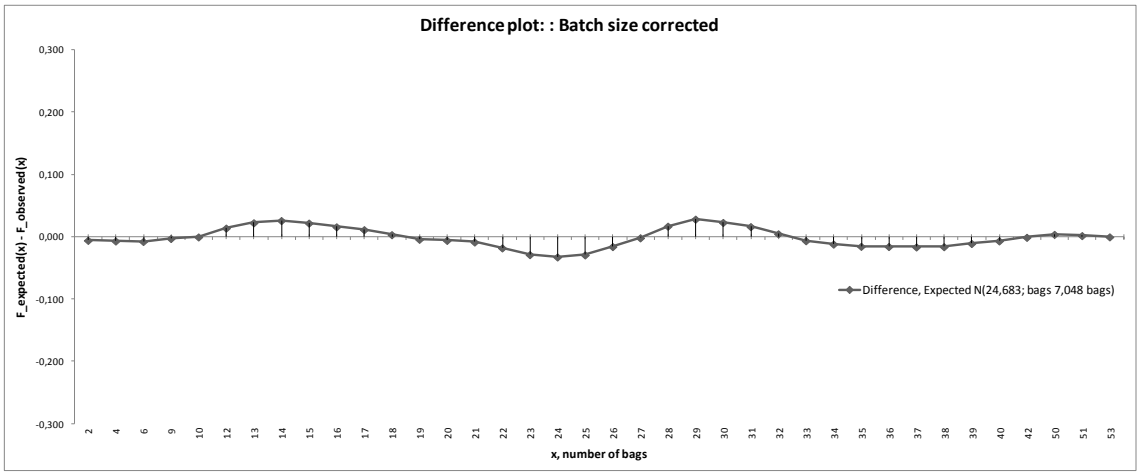


Figure 69: Difference plot of corrected batch size

Batching robot allocation

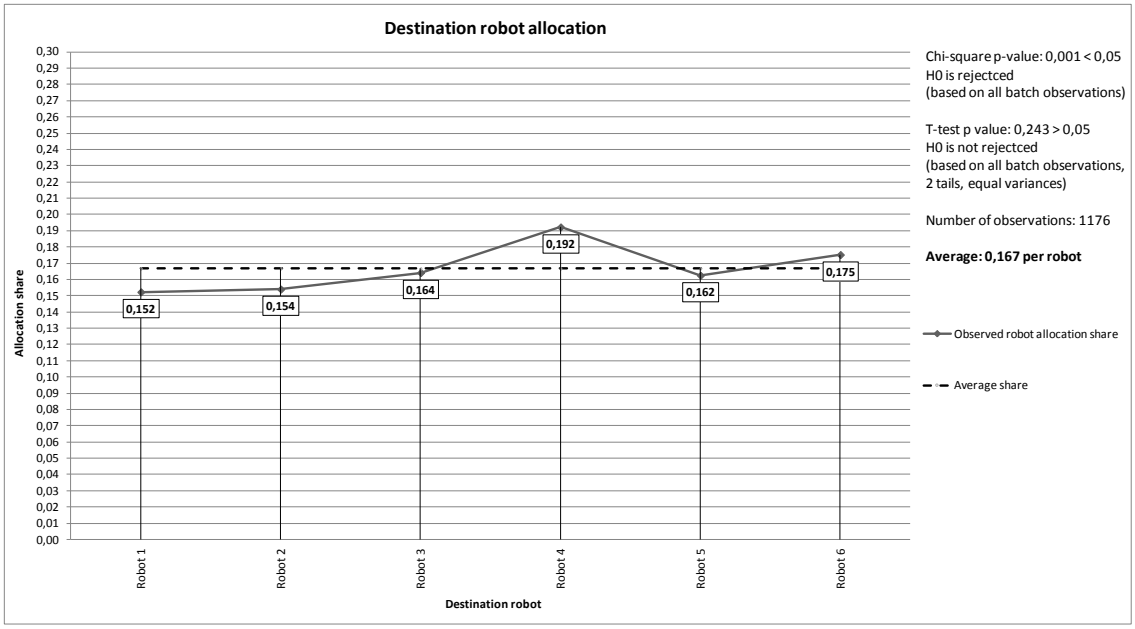


Figure 70: Allocation frequency of destination robots

Initial batch dispatch delay

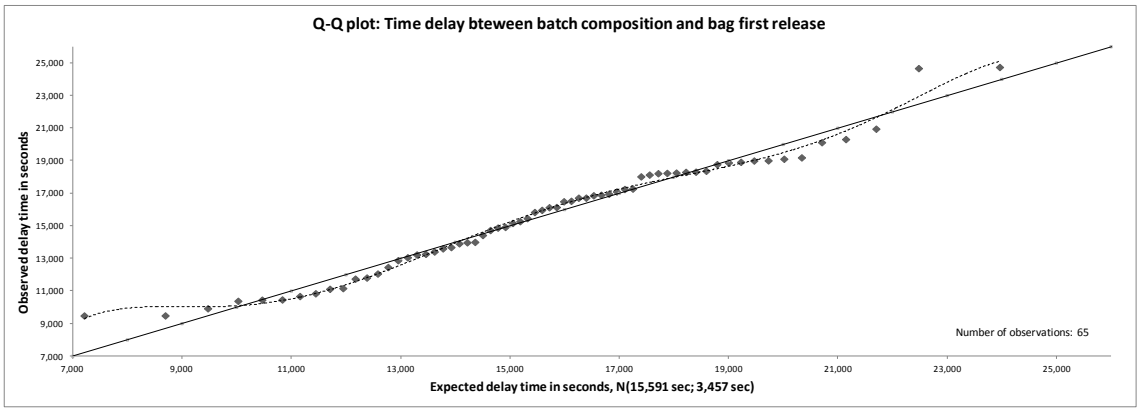


Figure 71: Q-Q plot of release delay between batch composition and first bag release

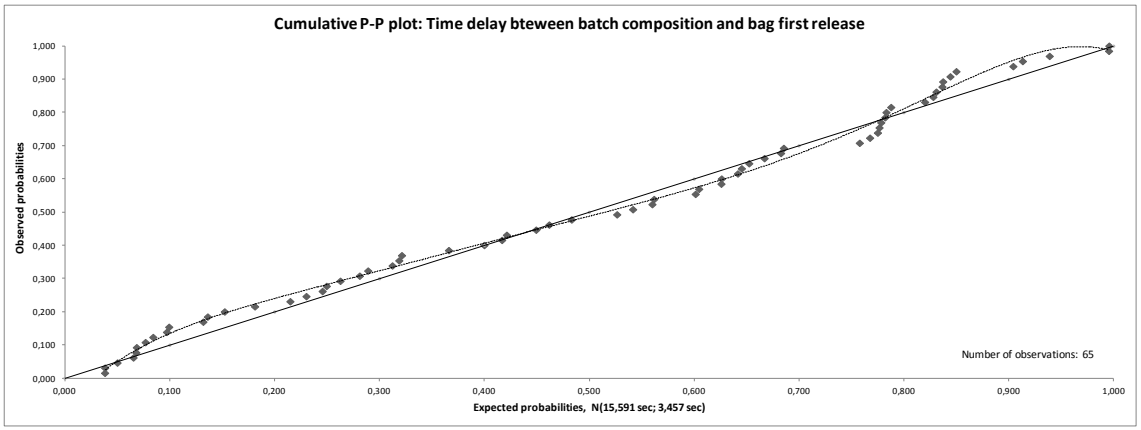


Figure 72: P-P plot of release delay between batch composition and first bag release

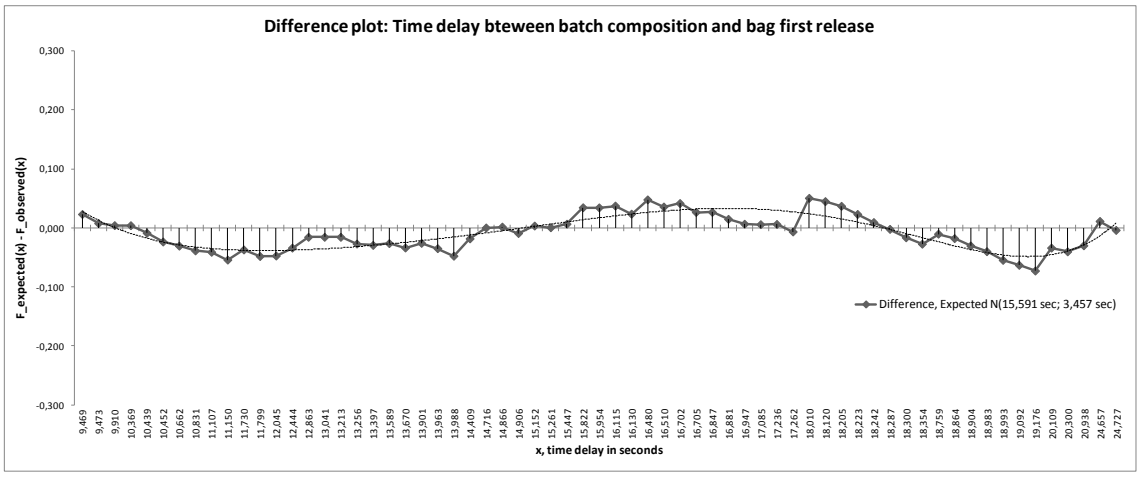


Figure 73: Difference plot of release delay between batch composition and first bag release

Travel times

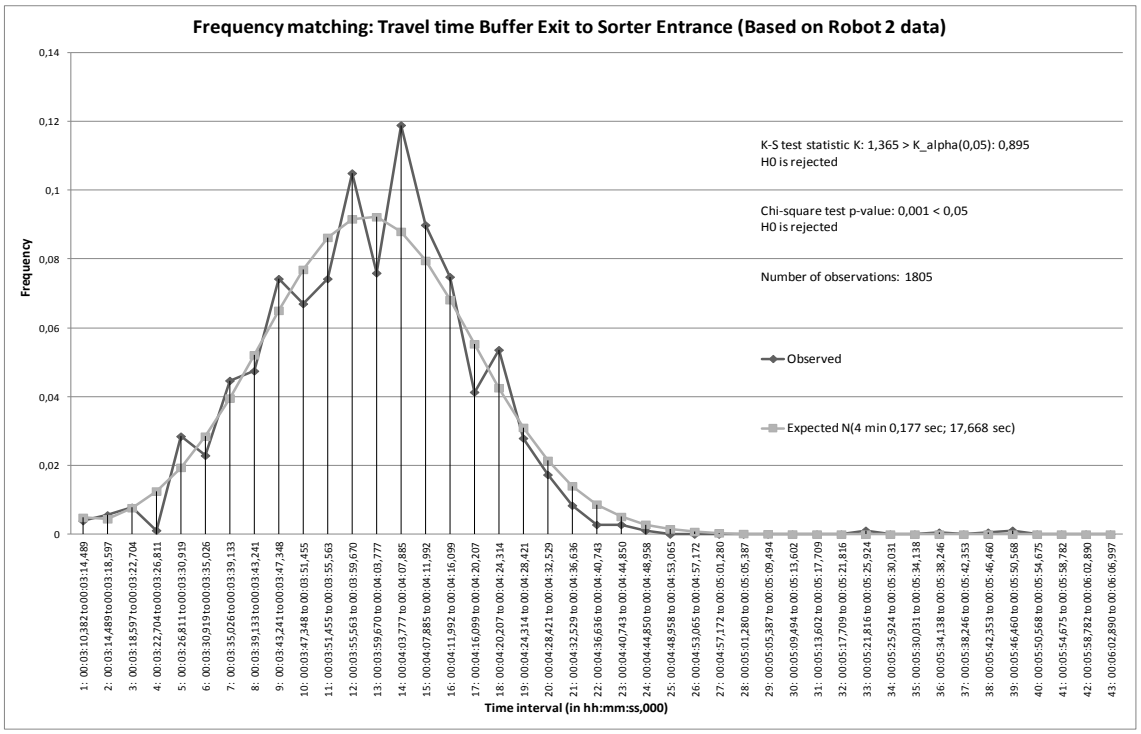


Figure 74: Probability density function of Buffer to In-cache lines travel time

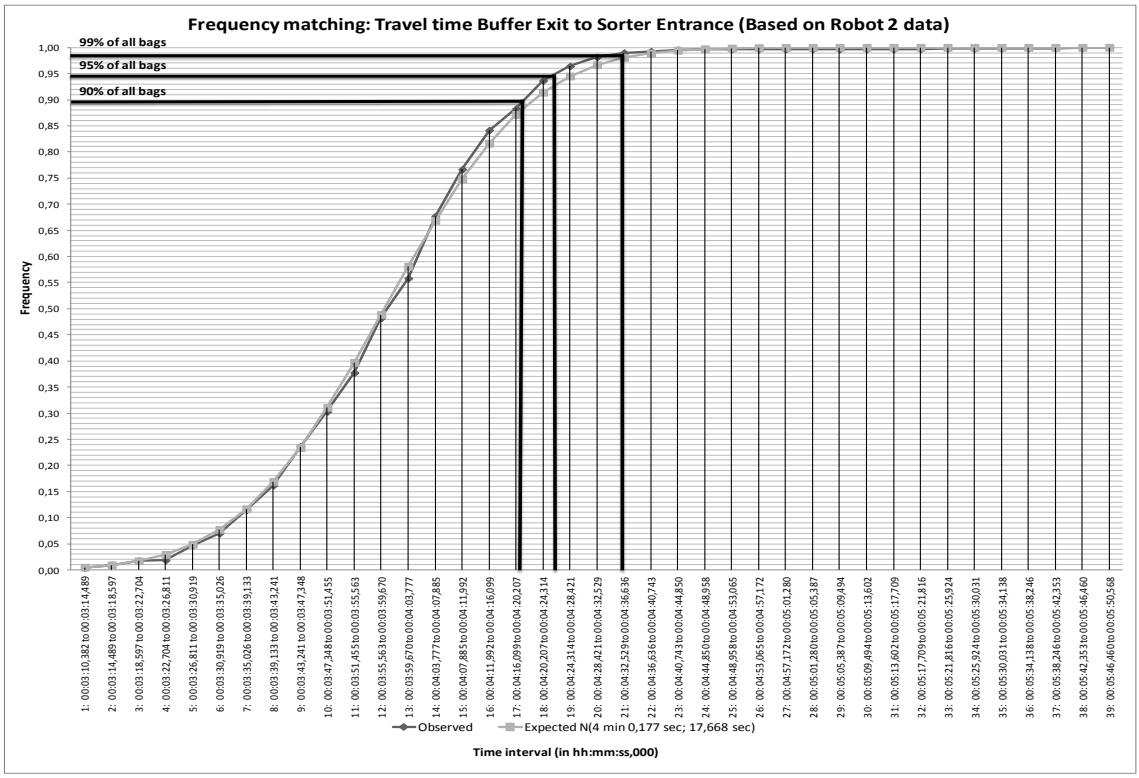


Figure 75: Cumulative density function of Buffer to In-cache lines travel time

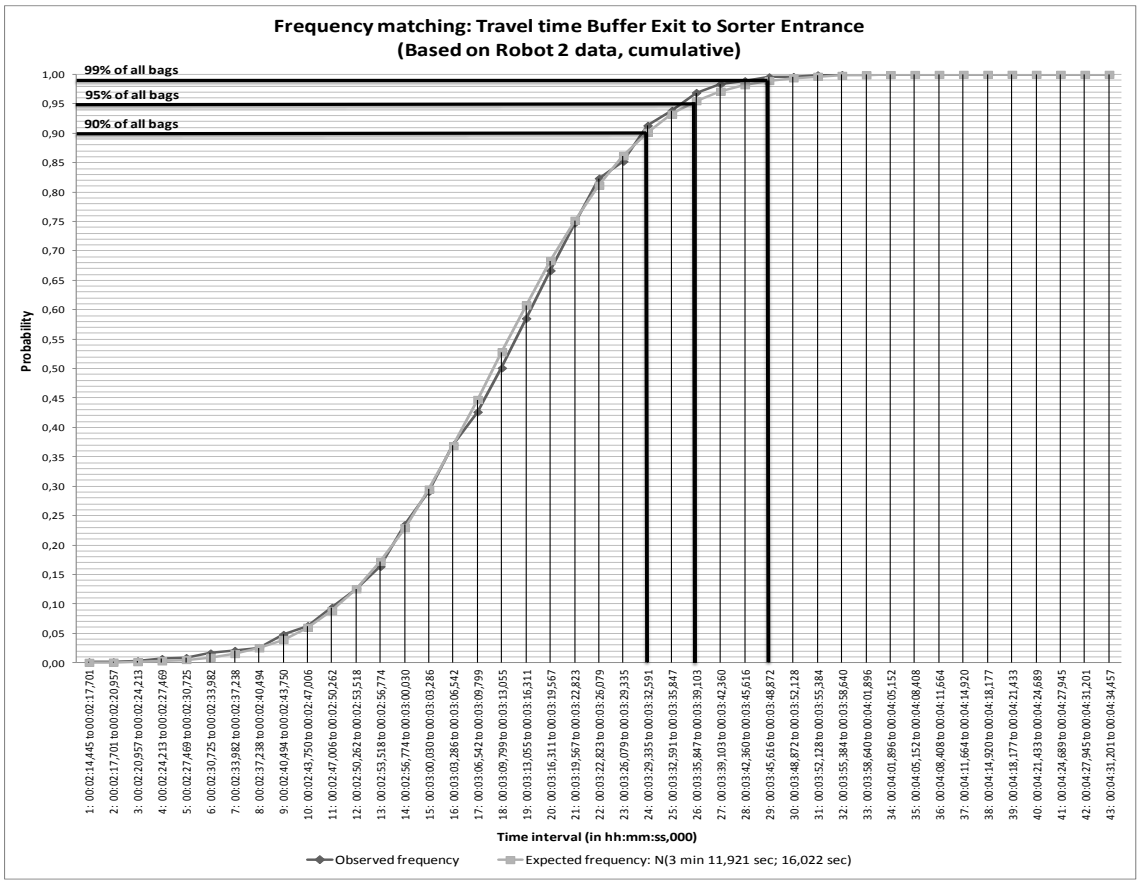


Figure 76: Cumulative density function of Buffer to Sorter travel time

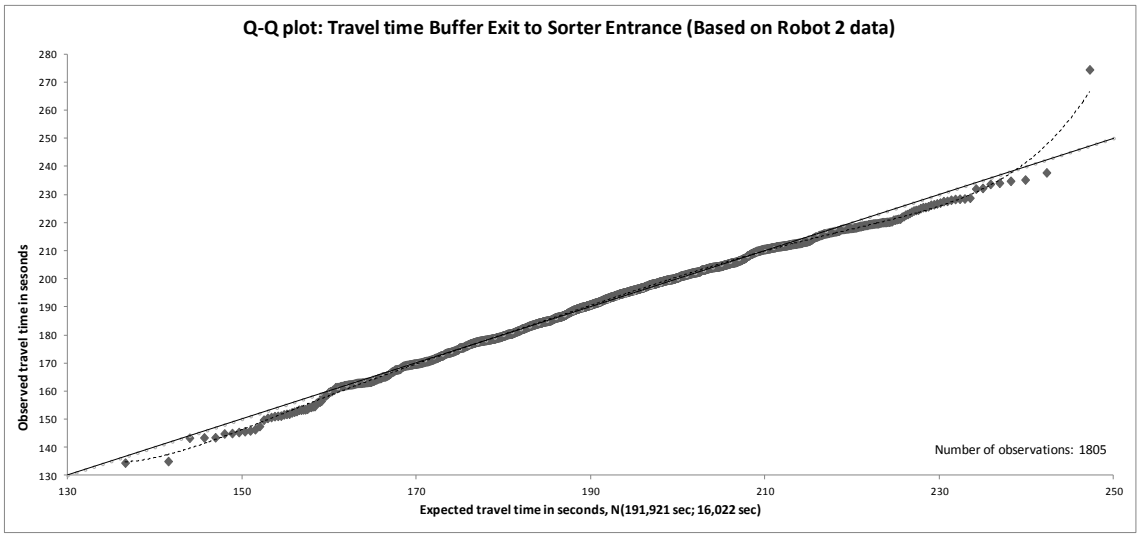


Figure 77 Q-Q plot of Buffer to Sorter travel time

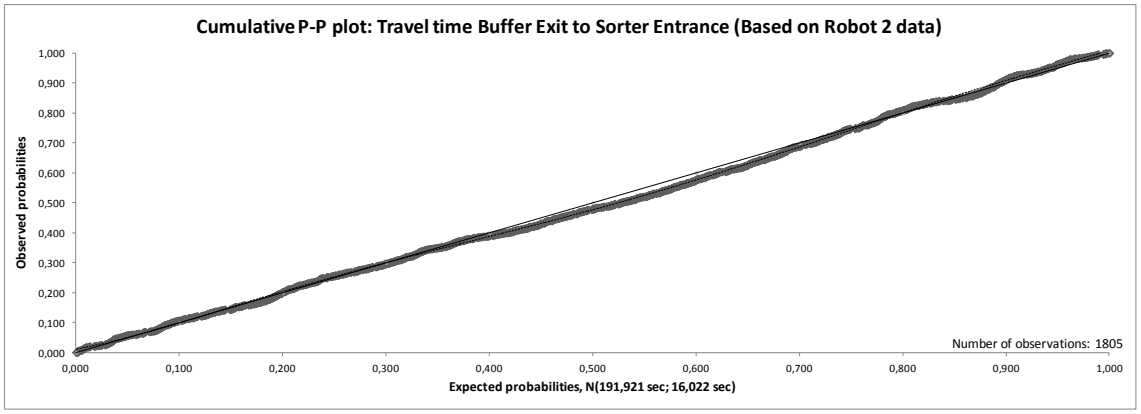


Figure 78: P-P plot of Buffer to Sorter travel time

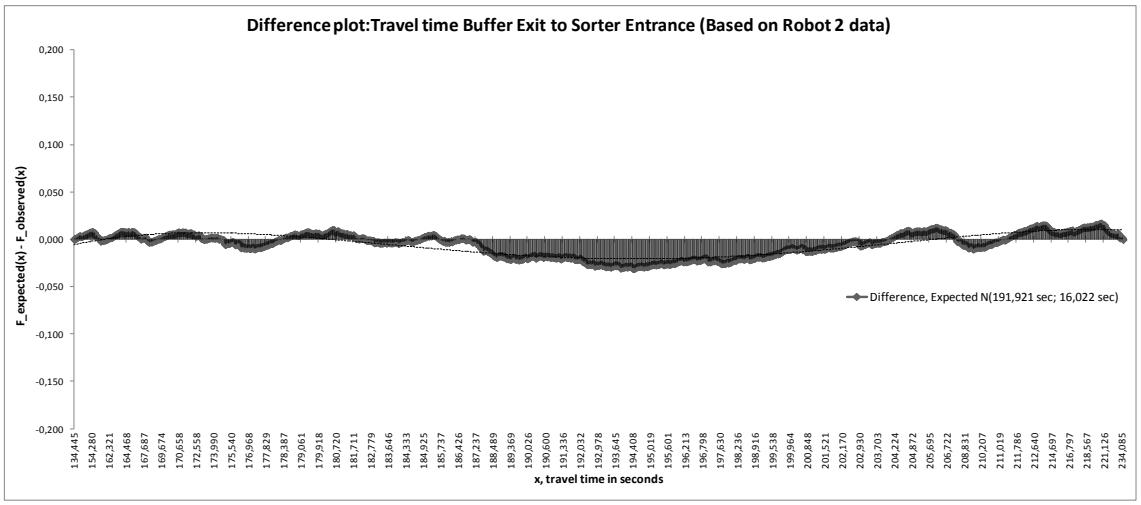


Figure 79: Difference plot of Buffer to Sorter travel time

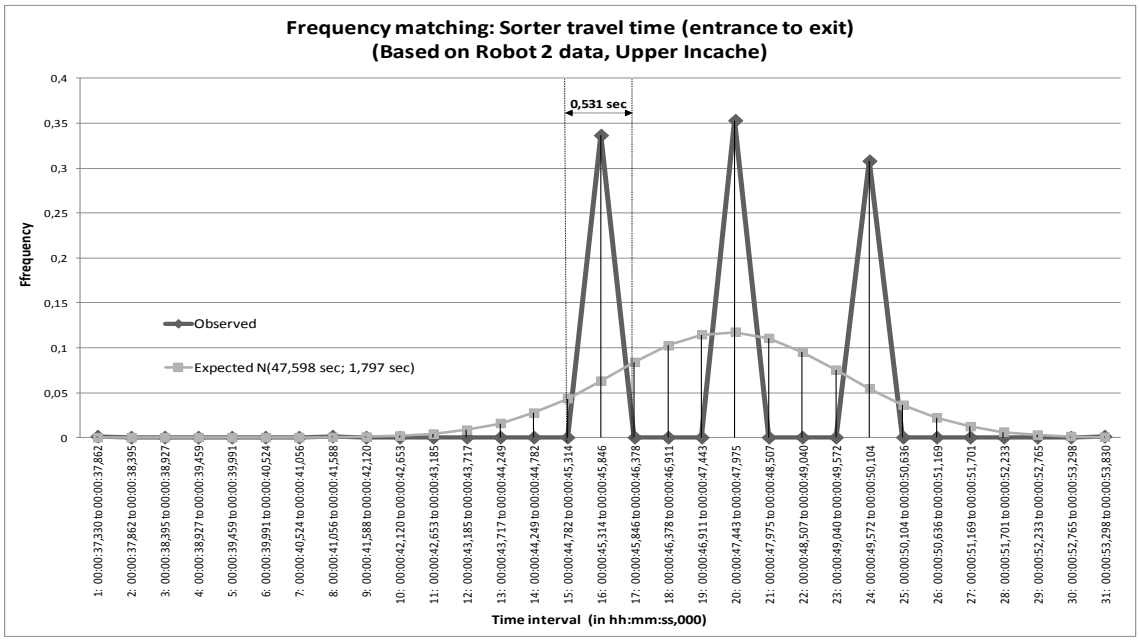


Figure 80: Zoomed-in inner-sorter travel time with focus on Robot 2 data

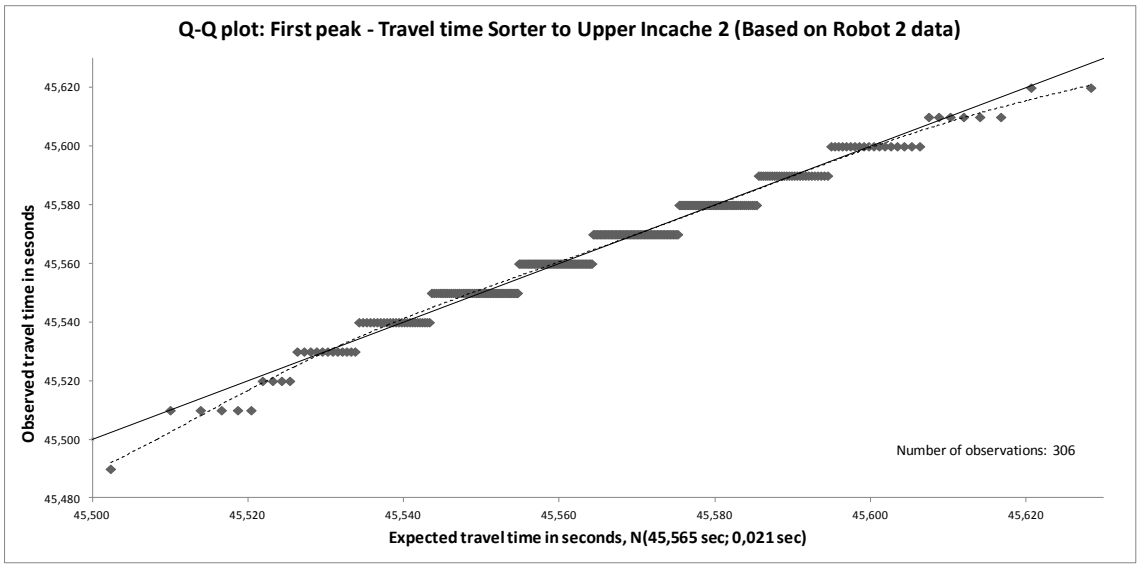


Figure 81: Q-Q plot of inner-sorter travel time with focus on first "data peak" of Robot 2 data

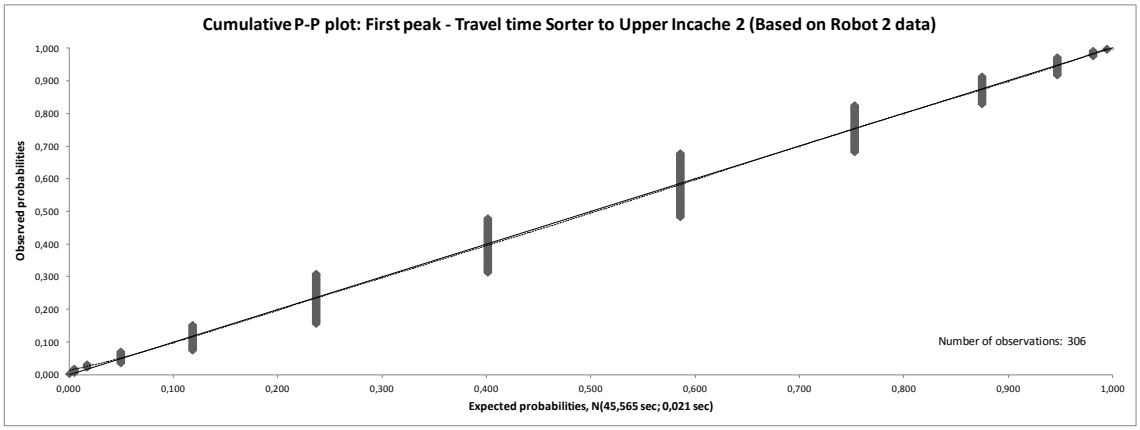


Figure 82: P-P plot of inner-sorter travel time with focus on first "data peak" of Robot 2 data

Table 14: Inner-sorter travel times

Travel time: Sorter (entrance to exit) (in hh:mm:ss,000)																		
Destination / Origin	Sorter entrance Closest		Sorter entrance Middle		Sorter entrance Farthest		Robot 1 Incache U+L		Robot 2 Incache U+L		Robot 3 Incache U+L		Robot 4 Incache U+L		Robot 5 Incache U+L		Robot 6 Incache U+L	
	Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation
Sorter entrance Closest	-	-	-	-	-	-	00:00:20,429	00:00:00,023	00:00:26,704	00:00:00,023	00:00:32,994	00:00:00,025	00:00:39,286	00:00:00,025	00:00:45,575	00:00:00,024	00:00:51,866	00:00:00,000
Sorter entrance Middle	00:00:01,952	00:00:00,004	-	-	-	-	00:00:22,411	00:00:00,026	00:00:28,688	00:00:00,026	00:00:34,977	00:00:00,025	00:00:41,269	00:00:00,024	00:00:47,556	00:00:00,024	00:00:53,849	00:00:00,026
Sorter entrance Farthest	-	-	00:00:02,357	00:00:00,004	-	-	00:00:24,768	00:00:00,022	00:00:31,047	00:00:00,025	00:00:37,337	00:00:00,024	00:00:43,626	00:00:00,025	00:00:49,916	00:00:00,025	00:00:56,206	00:00:00,023
Robot 1 Incache U+L	-	-	-	-	-	-	-	-	00:00:06,275	00:00:00,000	-	-	-	-	-	-	-	-
Robot 2 Incache U+L	-	-	-	-	-	-	-	-	-	-	00:00:06,290	00:00:00,002	-	-	-	-	-	-
Robot 3 Incache U+L	-	-	-	-	-	-	-	-	-	-	-	-	00:00:06,291	00:00:00,000	-	-	-	-
Robot 4 Incache U+L	-	-	-	-	-	-	-	-	-	-	-	-	-	-	00:00:06,289	00:00:00,001	-	-
Robot 5 Incache U+L	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	00:00:06,292	00:00:00,024
Robot 6 Incache U+L	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

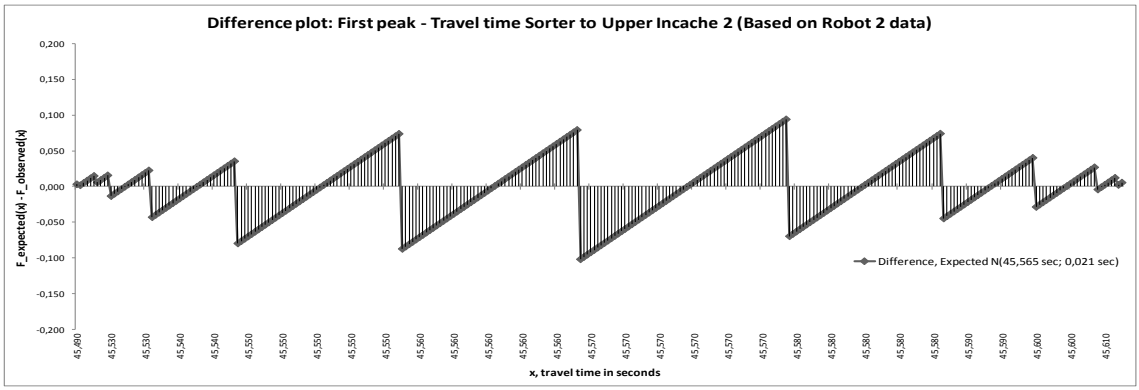


Figure 83: Difference plot of inner-sorter travel time with focus on first "data peak" of Robot 2 data

Batch inter-arrival time

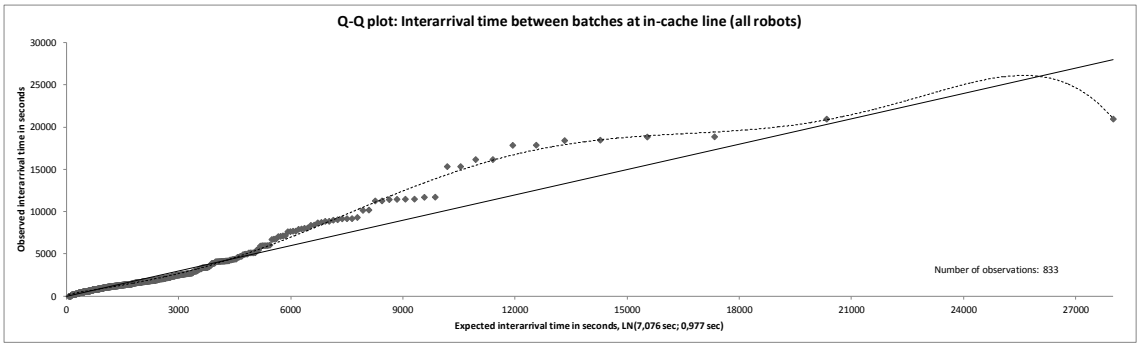


Figure 84: Q-Q plot of inter-arrival time between batches at in-cache lines

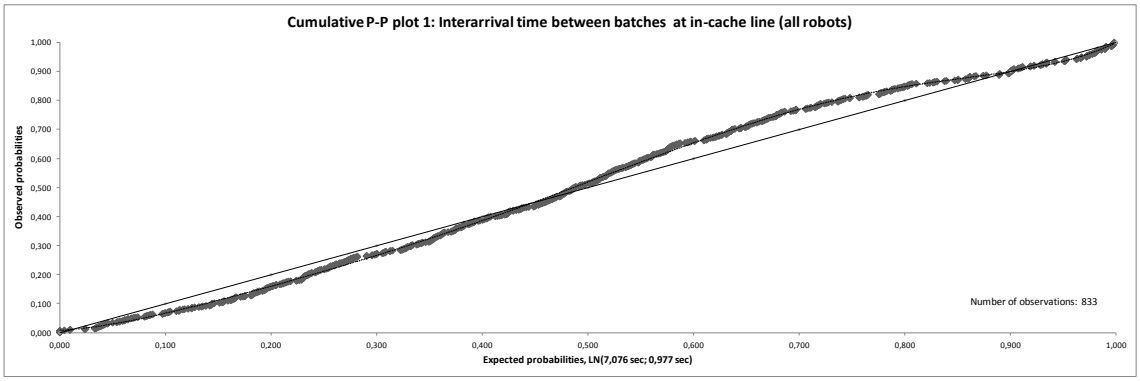


Figure 85: P-P plot of inter-arrival time between batches at in-cache lines

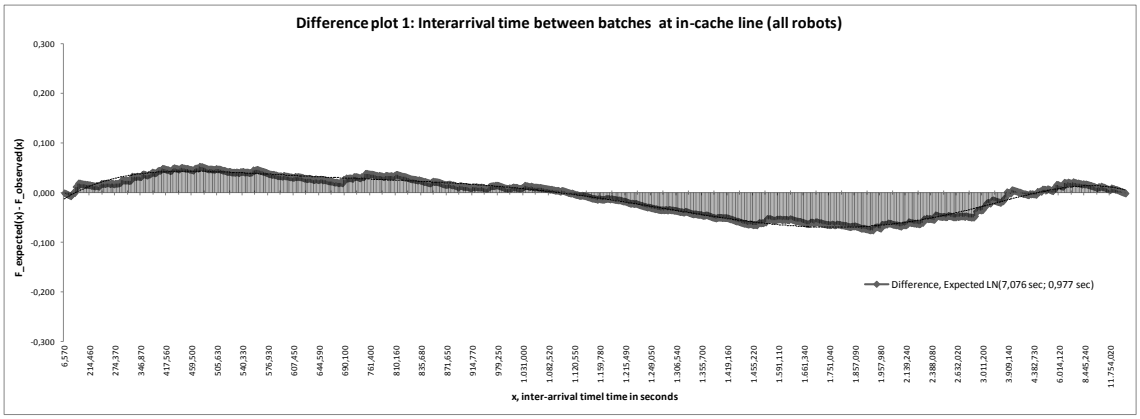


Figure 86: Difference plot of inter-arrival time between batches at in-cache lines

System component capacity

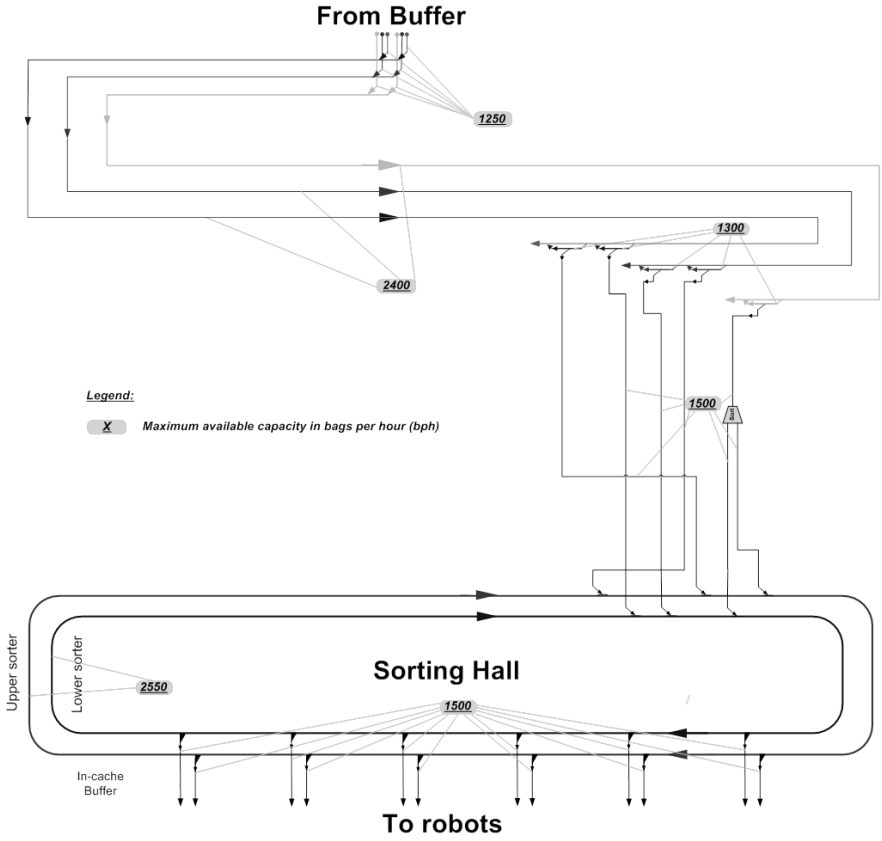


Figure 87: Equipment capacities restricting baggage batching

Batch in-system time

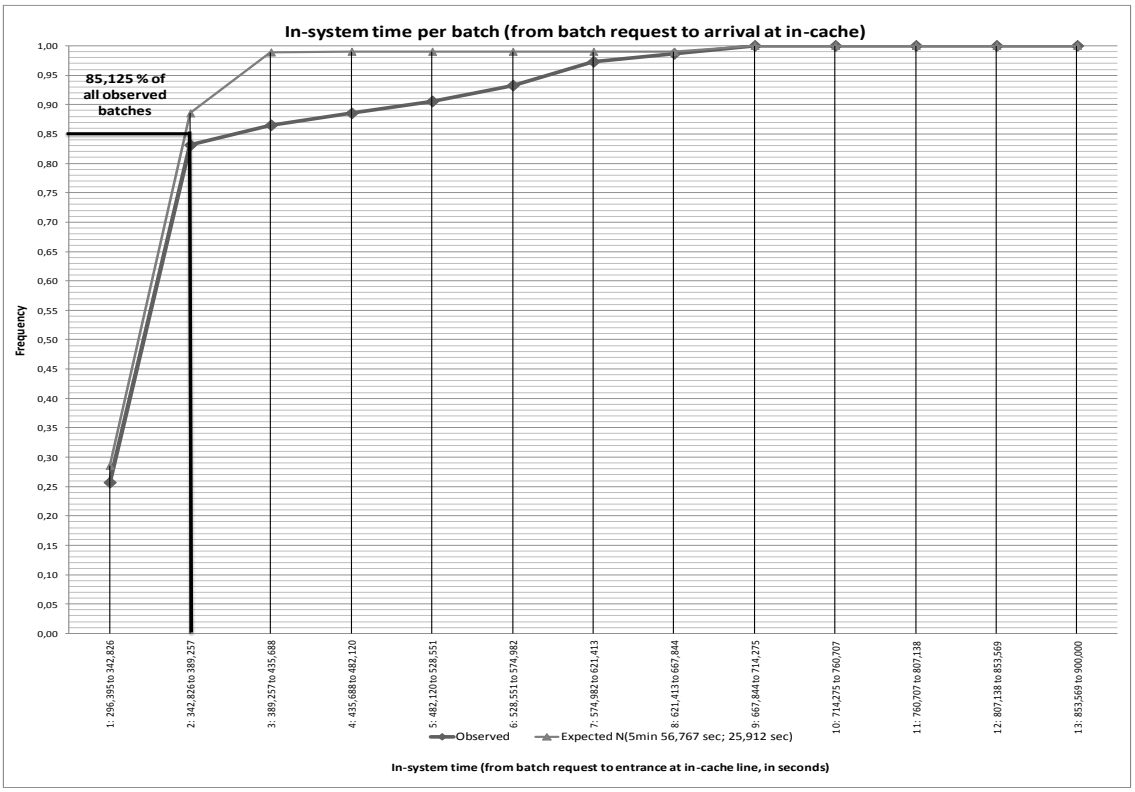


Figure 88: Cumulative density function of in-system time per batch

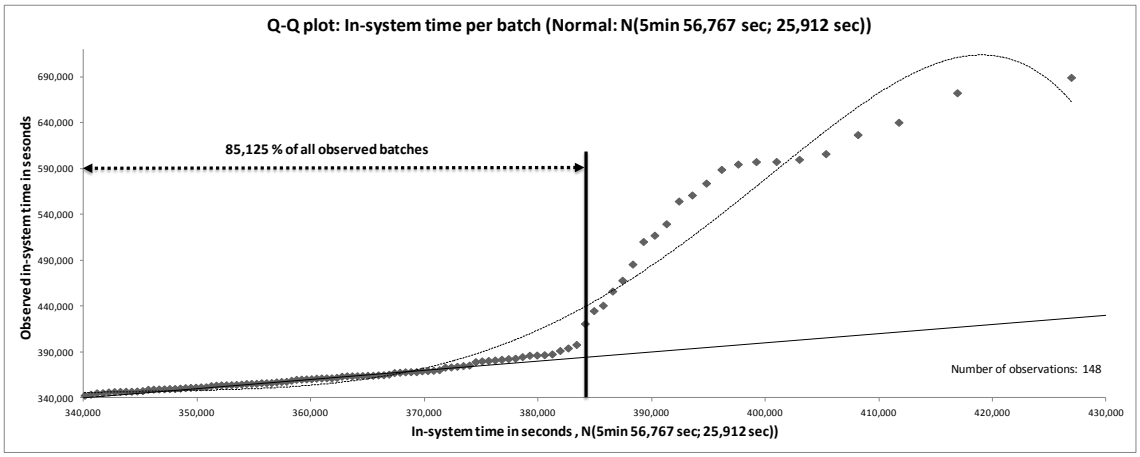


Figure 89: Q-Q plot of in-system time per batch

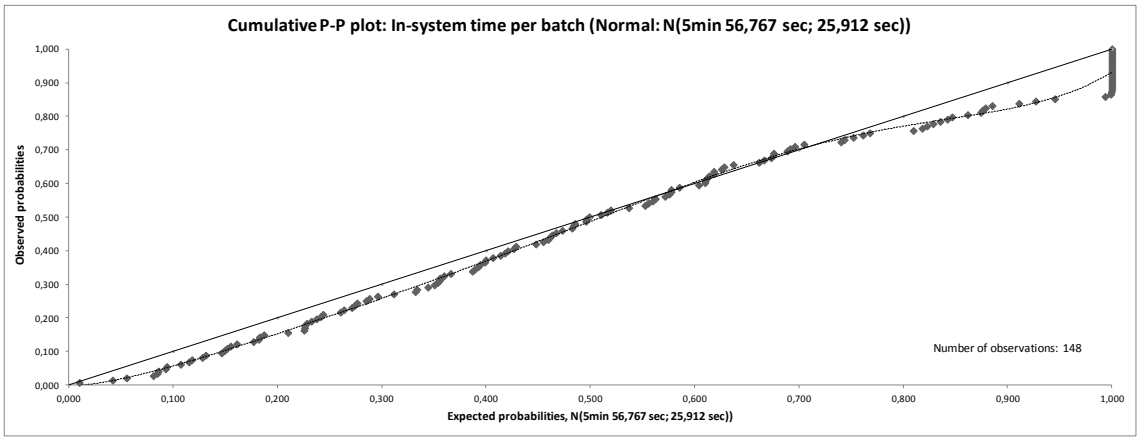


Figure 90: P-P plot of in-system time per batch

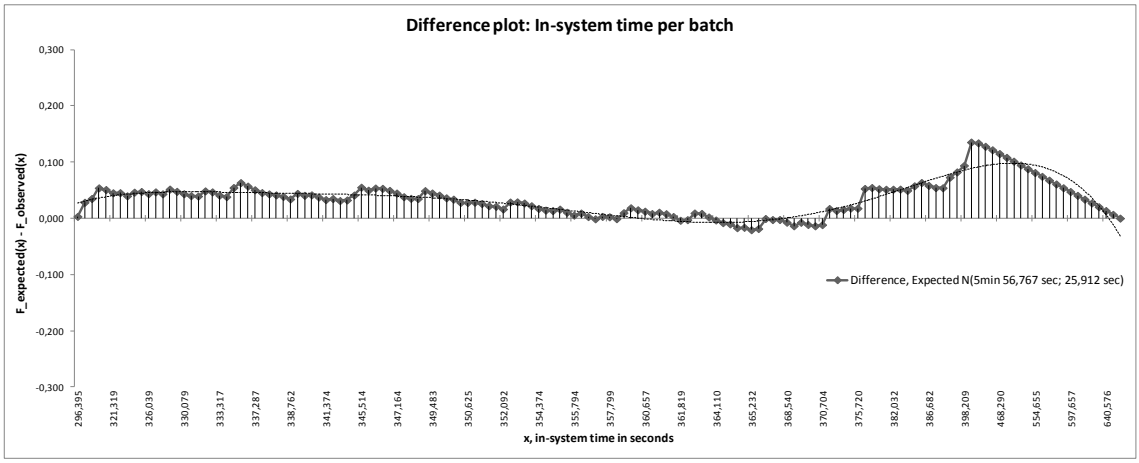


Figure 91: Difference plot of in-system time per batch

Inner-batch release delay

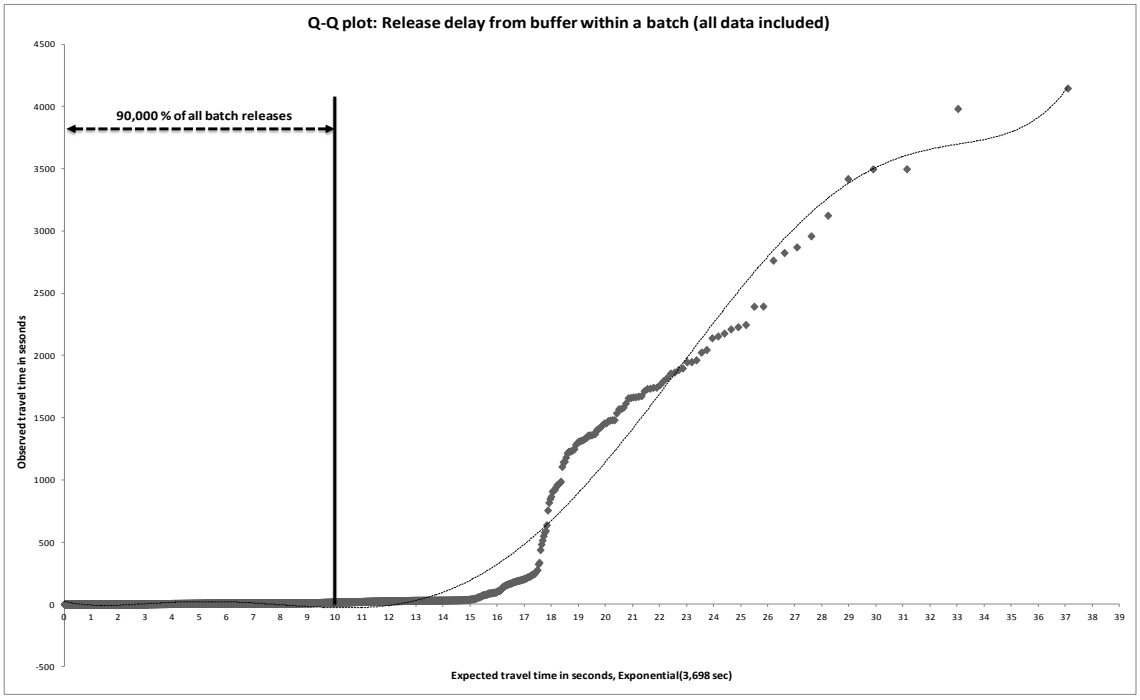


Figure 92: Q-Q plot of inner-batch release delay – large tail deviation

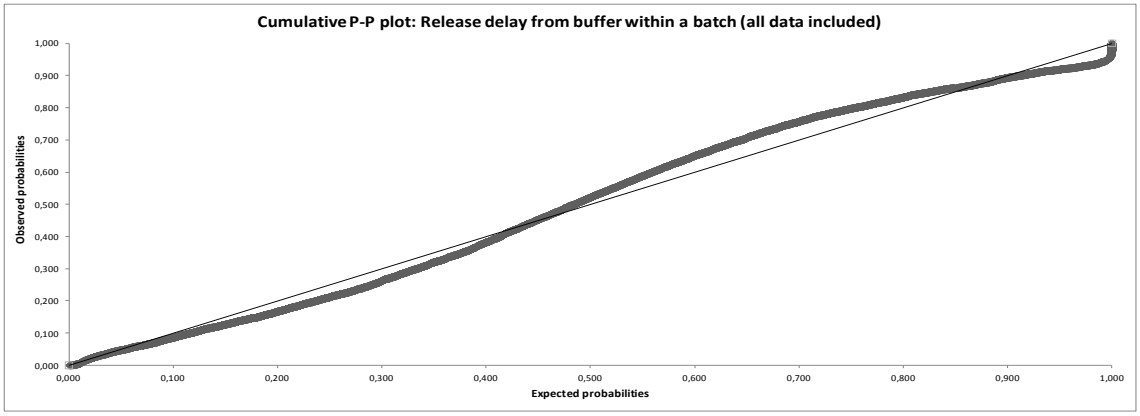


Figure 93: P-P plot of inner-batch release delay

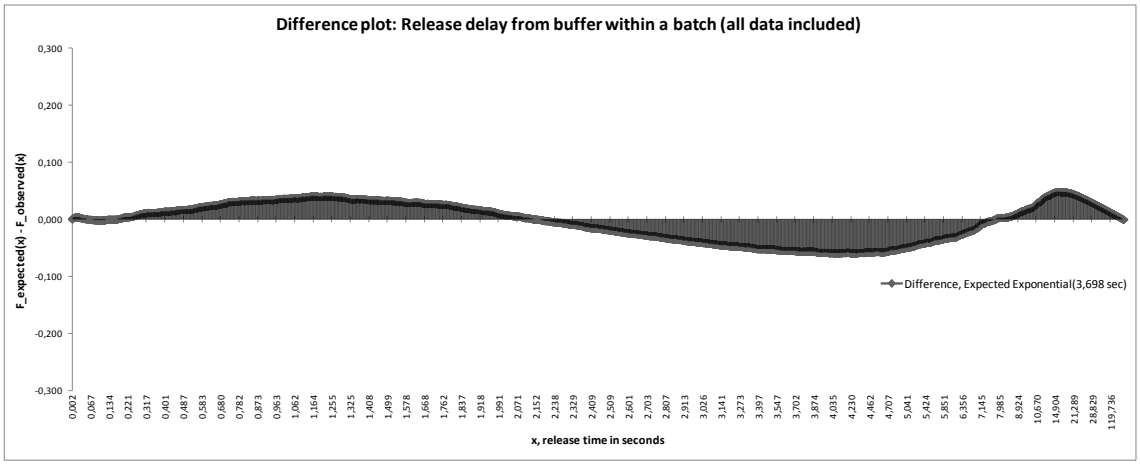


Figure 94: Difference plot of inner-batch release delay

Subject index

Algorithm convergence comparison 56
Applied statistical tests 29
Assumptions 38
Automatic calibration 22
Automatic calibration approaches 42–44
Bage Inter-arrivals at in-cache 34
Baggage batching process 3, 10–14
Baggage subgroup division 13
Batch composition 29
Batch in-system time 35
Batch size 31
Batching hours per day 29
Buffer to sorter travel time 32
Calibration ideology 3
Calibration model benchmarks 34–36
Calibration model implementation 44–46
Calibration model input 28–34
Calibration problem formulation 20
Categorization of optimization search schemes 25
Chi-square test 29
Combination flowchart LHS-SA 42
Comparison measures from practice 41
Comparison of calibration methodologies 49–56
Computational calibration results 49
Concept distinction 15
Concluding summary 58
Conclusions & Recommendations 58–60
Convergence speed & quality 53–56
Cooling scheme Simulated Annealing 43
Current calibration practice 8
Current situation 8–14
Data analysis 28–36
Data analysis sequence 28
Definitions 7
Desired situation 14
Destination robot allocation 31
Deviation measurements 16
Differences in abstraction level 14
Discussion - Calibration mdeol set-up 41
Discussion - Calibration model set-up 48
Discussion - Data analysis 36
Discussion - Literature review 27
Discussion - Problem analysis 14
Discussion - Results & Findings 57
Equipment capacity 34
Expected results & Contributions 5
Experimental distributions & feasible ranges 48
Experimental set-up 47–48
First bag delay 31
Frequency matching 30–33
Future research suggestions 60
Idea of Latin Hypercube Sampling 43
Importance of sensitivity analysis 26
Independence of batch composition 30
Inner-batch release delay 35
Introduction 1–5
Introduction of Vanderlande Industries 6
Kolmogorov-Smirnov test 29
KPIs and weights in objective function 40
Layout of model implementation 45
Limitations of calibration 23
Literature review 15–27
Manual and semi-automatic calibration 21
Minimum number of model replications 41
Model deviation comparison 16
Model set-up 37–41
Objective function and constraints 40–41
Observed relative error 48
Operational layout baggage batching 10
Outliers 28
Parameter bounds Simulated Annealing 44
PDF of batch in-system time 35
PDF of batch requests 30
PDF of Buffer to sorter travel time 32
PDF of corrected batch size 31
PDF of inner-batch release delay 36
PDF of of first bag delay 32
PDF of sorter to in-cache travel time 33
Phases of project management 6
Power regression example 53
Problem analysis 6–14
Problem chart 2
Problem identification 1
Problem statement 4
Process flow of calibration model 39
Project organization 6
Project phases of Major Projects 8
Public perception of Vanderlande Industries 6
Recommendations 59
Regression averaging 54
Regression averaging quality 55
Regression methodology 53
Research goal 4
Research motivation 1
Research question 4
Research scope 3
Results & Findings 49–57
Root Mean Square Error (RMSE) 18
Scattering of candidate solutions 50–52
Scope & Level of detail 37
Screenshot Plant Simulation 46
Search schemes in calibration 24–26
Sensitivity analysis 47
Single- & multivariate calibration 19
Solution approach & Experimentation 42–48
Sorter to In-cache travel time 33
Student t test 29
Test on auto-correlation of batch requests 30
Travel times 32
Type of calibration model 38
Verification & Validation 45
Worse candidate solution acceptance 43