UNIVERSITÉ DE STRASBOURG

ÉCOLE DOCTORALE ED 269

Laboratoire Icube - Équipe CSIP - Conception, Système d'Information et Processus inventifs

THÈSE présentée par :

Michael SCHLECHT

soutenue le : **18 Septembre 2023**

pour obtenir le grade de : **Docteur de l'Université de Strasbourg**

Discipline/ Spécialité : **Génie industriel**

Redesigning production systems by their digital shadow

(Reconcevoir les systèmes de production à l'aune de leur ombre numérique)

DIRECTEURS DE THÈSE :

Monsieur DE GUIO Roland	Professeur, INSA Strasbourg
Monsieur KÖBLER Jürgen	Professeur, Hochschule Offenburg

RAPPORTEURS :

Monsieur CHEUTET Vincent	Professeur, INSA Lyon
Monsieur SIADAT Ali	Professeur, ENSAM Metz

EXAMINATRICES :

Madame CHAABANE Sondes	Maîtresse de conférences, INSA Hauts-de-France
Madame GOEPP-THIEBAUD Virginie	Professeure INSA Strasbourg

Acknowledgements

Completing my PhD has been a challenging and rewarding experience, and I owe a debt of gratitude to the incredible individuals who have supported me throughout this academic journey. Their unwavering encouragement, guidance, and belief in my abilities have been instrumental in achieving this significant milestone.

First and foremost, I extend my deepest appreciation to my esteemed supervisor, Professor Roland de Guio. His exceptional mentorship, expertise, and relentless dedication to my academic growth have shaped me into a better researcher. I am grateful for the countless hours he invested in providing constructive feedback, refining my ideas, and helping me navigate through the complexities of my research.

I would also like to express my sincere gratitude to my co-supervisor, Professor Jürgen Köbler. His valuable insights, encouragement, and collaboration have been invaluable in enhancing the depth and breadth of my research. I am thankful for the fruitful discussions and academic support that he generously offered throughout this journey.

To my family and friends, I offer my heartfelt thanks for their unwavering love and support. Their understanding, patience, and belief in my abilities have been the driving force behind my perseverance during challenging times. I am grateful to my close-knit circle of friends, whose camaraderie and encouragement have provided much-needed balance in my life and distracted me from PhD stress, whenever it was needed.

I would also like to acknowledge the faculties and staff at the Institut National des Sciences Appliquées de Strasbourg and Hochschule Offenburg, whose dedication to education and research have provided me with a conducive environment to thrive academically. Finally, I would like to thank my colleagues from VIRTfac and the organization of Interreg Europe, who made this PhD possible.

Thank you very much all of you!

Michael Schlecht

31^{st} July 2023

Contents

List of Figures

List of Tables

List of Appendices

Abstract

This thesis deals with the redesign of manufacturing systems by data-driven simulation, optimization and problem-solving. Simulation of material flows is a widely spread tool for solving problems in the design of manufacturing systems. However, limitations are the high requirements in terms of time and knowledge to execute simulation studies, evaluate results and solve design problems. New chances arrives with the technologies of industry 4.0 and particularly the digital shadow, providing production data for modeling and simulation. However, the methods to use production data for the redesign of production systems are not available yet. Purpose of this work is providing the methods to a) automate modeling and simulation from digital shadow, b) use data-driven simulation to optimize systems and provide experimental data and c) solve problems in the design of manufacturing system. The strategy to provide the methods is studying the literature, extracting the methods, evaluation in two case-studies and providing the insights in the discussion. The result of this work is a framework for the application of the digital shadow in optimization and problem-solving as well as its validation in two case-studies.

Keywords: Digital shadow, data-driven simulation, optimization, inventive design, problem-solving.

List of publications

Automated Generation of Simulation Model in Context of Industry 4.0

Michael Schlecht, Roland de Guio, Jürgen Köbler; TJMS - International Journal of Modelling and Simulation; - DOI: 10.1080/02286203.2023.2206075

Mathematical model for the maintenance activities scheduling in the case of railway remanufacturing

Ayoub Tighazoui, Michael Schlecht, Roland De Guio, Bertrand Rose and Jürgen Köbler; ITSIS2022 - International Conference on Information Technology & Smart - Industrial Systems; DOI: 10.1109/ITSIS56166.2022.10118388

Flexibles Referenzmodell zur Planung und Optimierung der Produktion – Teil 2

Jürgen Köbler, David Wußler, Michael Schlecht, Sarah Kirchenbaur, Roland de Guio; Industrie 4.0 Management 05/2022; DOI: 10.30844/IM_22-5_45-48

Data-driven decision process for robust scheduling of remanufacturing systems

Michael Schlecht, Sara Himmiche, Virginie Goepp, Roland De Guio Jürgen Köbler; MIM2022 - IFAC Conference Manufacturing Modelling and Control; DOI: 10.1016/j.ifacol.2022.09.500

Optimization of Sequencing: Integration of machine learning and generic material flow models

Michael Schlecht, Sebastian Berger, David Wußler, Matthias Haun, Jürgen Köbler; ZWF Zeitschrift für Wirtschaftlichen Fabrikbetrieb; DOI: 10.1515/zwf-2022-1005

Flexibles Referenzmodell zur Planung und Optimierung der Produktion

Michael Schlecht, Jürgen Köbler, Roland de Guio; Industrie 4.0 Management 06/2021; DOI: 10.30844/I40M_21-5_S53-56

Industrie 4.0: Der Weg zu einem digitalisierten Produktionsunternehmen

Jürgen Köbler, Tobias Fischer, Benjamin Klerch, Michael Schlecht; Industrie 4.0 Management 03/2020, DOI: 10.30844/I40M_20-3_S57-60

Chapter 1 Introduction

1-1 Context

Producing organizations exist in a fast-changing environment. Organizations must respond to increased competition, continuously changing market-conditions, unpredictable demands, and a high variety in product mix. These challenges force organizations to continuously advance their production systems. With increasing size and complexity of the systems, the redesign of production system becomes an increasingly difficult task. Simulation of material flows is a recognized tool for generating knowledge about the critical systems and support of redesign through experiments using digital models. However, problem-solving in the redesign of material flows remains difficult and requires expertise.

Inventive problem-solving is in the focus of the research efforts conducted over the past fifteen years in the CSIP laboratory. The core of this methodology consists of reformulating problems by highlighting the system limitations, formulating the system of contradictions and changing the model to solve a problem and overcome system limitations. Methods and algorithms have been developed in previous works to reformulate problems and change models based on the systematic analysis of experimental data. To support problem-solving, experimental data is required and must provide links between action and evaluation parameters. Depending on the field of research or expertise, the term 'action parameters' is also referred to as decision variables (e.g. in experimental design, optimization area), design parameters or system parameters. In this work, all three terms are used interchangeably. The availability of experimental data is a major challenge to overcome.

New means of obtaining data have resulted from Industry 4.0, which has been marked by an elevated level of connectivity and intelligence through the adoption of ubiquitous information and communication technologies. Data are automatically measured in the physical production systems with sensors, smart machines, and various Industry 4.0 technologies. The volume of available data has increased, enabling new concepts, particularly the digital shadow, which represents the physical system in virtual space. Through data-driven simulation, the digital shadow can also provide data about the behaviour of the problematic systems. This data can support solving problems in design, but requires systematic methods to extract knowledge [1]. The present work is dedicated to exploring the use of the new technologies of Industry 4.0, particularly the digital shadow and data-driven simulation, as well as inventive problem-solving methods to solve problems in the redesign of production systems.

The first goal of this work is to provide the methods and tools to generate experimental data for problem-solving by simulation of material-flows on request. Providing experimental data entails two objectives. The first objective is to provide methods to build simulation models and execute experiments from the data, and to provide the methods to extract simulation data from the digital shadow in order to enable data-driven simulation. The second objective is to provide the methods to define ideal simulation experiments for providing the right data to detect system limitations and root-causes. The second goal is to provide the methods to analyse the generated simulation data, formulate the system of contradictions and eventually support model change in order to overcome system limitations.

In order to demonstrate the genericity of the methods, two illustrative case-studies will be presented. The case-studies focus on the design of a remanufacturing system for trains (TRM) and a medical emergency department (ED). Subsections 1-1.1 to 0 provide the background of this research. The focus of the background is the role of the digital shadow and data-driven simulation in the design of material-flow systems, particularly from the system type of job-shop manufacturing systems, simulation-based optimization, and data-based problem-solving methods. The challenges of this research are then presented through research questions on three different levels in section 1-2:

- Generating models of material flows and simulating experiments from data received via the digital shadow
- Providing data for problem-solving by systematic simulation experiments
- Highlighting system limitations, formulating system of contradictions and changing models to overcome problems in system design

To address the research questions, a research method is proposed and questions are answered by providing methods and tools in section 0. Presentation of the two previously mentioned case-studies support this research and process to conclude Chapter 1.

1-1.1 Problem-solving

Problem-solving is an elementary activity in numerous domains and a crucial challenge in the design and redesign of systems [2]. The ability to solve problems effectively and efficiently is essential in design. In the design of complex and large-scale systems, problems become complex and multidimensional. Simple, linear problem-solving approaches are not appropriate for solving these problems. Moreover, problem-solving requires a deep understanding of the underlying causes and a holistic approach to develop sustainable solutions. A systematic process for solving problems and providing new solutions was presented in [3], [4] and is illustrated in Figure 1.

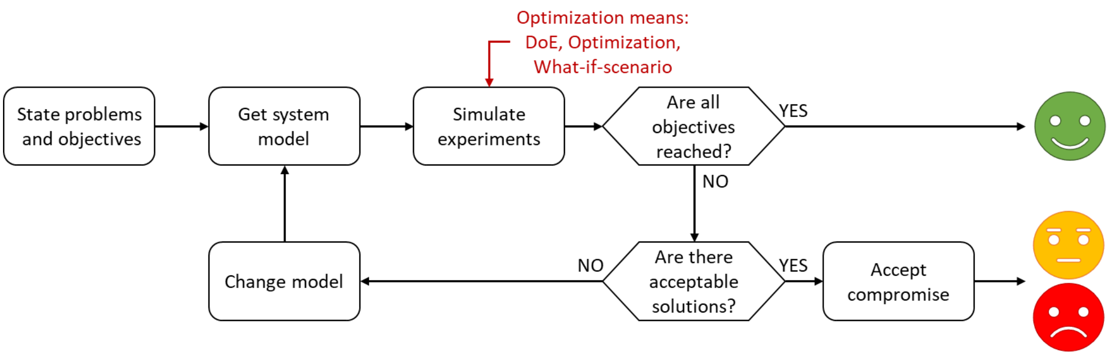


Figure 1: Problem-solving loop

The initial activity of problem-solving is the description of the problem and definition of objectives. In *get system model,* a simulation model of the system to be improved or changed is provided. In the activity *simulate experiments*, the model is used to evaluate the system's behaviour and determine its performance by exploring the design space using approaches. Optimization means for exploring the design space are what-if scenarios, design of experiments (DoE) or optimization algorithms. The objective is to find values for decision variables to satisfy the objectives in problem-solving by parameterization of the model. When all objectives are reached, the problem-solving process stops and the decision maker has a solution. When the objectives are not reached, the decision maker decides to accept or not accept a compromise. Accepting a compromise means to choose a solution among the results of the activity *simulate experiments*, which does not reach all objectives but is not unacceptable. In *change model,* the decision maker designs new solution concepts to satisfy all (at least more) objectives. Model change refers to generation of new solution concepts by structural changes beyond the simple change of decision variables. New simulation models are required to validate solution concepts. The loop repeats until a solution fulfils all objectives or provides an acceptable compromise.

In the presented problem-solving loop two approaches are appropriate to solve problems and create new solution concepts: *optimization* and *inventive design* [3]. Optimization increases the system efficiency by optimizing the system parameters. Inventive design introduces new parameters during the design process and changes the principle of work. In problem-solving, the formulation of a problem requires a statement of the system parameters; successive optimization of existing parameters is the first step of problem-solving. If optimization cannot provide acceptable solutions, then the problem is considered to be an inventive problem, for which the decision maker uses solving techniques of inventive design. Inventive design requires more profound changes of the model.

1-1.2 Optimization

In system design, the objective of optimization is to find solutions for design problems by exploring a predefined problem solution search space. For optimization, the decision maker provides a mathematical model, representing the idealized problem

[5]. The mathematical model provides the link between decision variables and the objective function for the critical system. The decision variables represent the decisions to be made by determining specific values. The objective function measures the consequence of the taken decisions by evaluating the mathematical model. The goal of optimization problems is to minimize or maximize the values for the objective function by determining values for the decision variables. During optimization, parameterization of decision variables moves the system through the problem solution search space, defined in [6], by adopting alternative states of problem-solving. Optimization browses a space of potential solutions that is limited by the problem space. The objective is to arrive at an optimal state without changing the model beyond determining parameters.

An illustration of an exemplary optimization problem is given in Figure 2 [3], with the decision space on the left and the solution space on the right. The illustrated problem has k objectives (y) in the solution-space Y and all objectives are to be minimized. In the example there is no additional knowledge about importance of the objectives available. The solution of this multi-objective problem is describable in terms of the decision variables (x) for n decision variables, formulated in a decision vector. All possible decisions determine the decision space X. As a consequence, a function can be expressed that evaluates the quality of a specific solution by calculating the values for the objective functions and providing an objective vector in the solution-space Y. The function for evaluation in mathematical optimization is expressed by the problem model. However, any model capable of evaluating a specific decision vector can provide the objective function, e.g. simulation in a model of material flows.

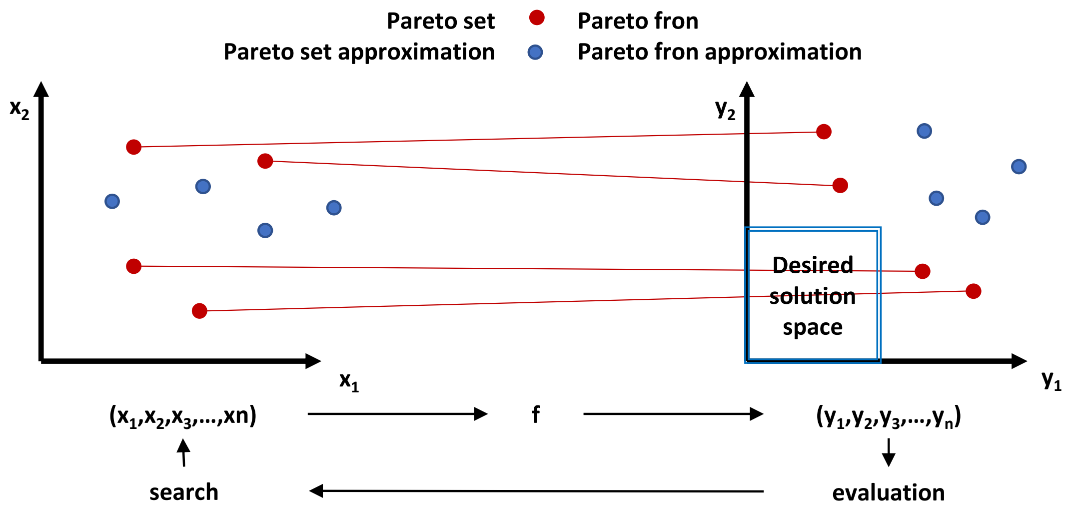


Figure 2: Optimization problem

The model can provide a set of solution vectors by evaluating a set of decision vectors. Within the set of solution vectors there are subsets of dominated and non-dominated solution vectors. The subset of non-dominated solution vectors describes the Pareto front of the optimization problem. Distinction between dominated and non-dominated solutions is required. Based on the concept of Pareto dominance [7], one objective vector Y_i dominates another objective vector Y_j, if Y_i achieves all objectives of Y_j plus at least one additional objective. Providing optimum solutions by optimization requires the identification of non-dominated solutions that clearly perform better than dominated solutions. However, comparing multiple non-dominated solutions does not provide one best solution; moreover, trade-offs between multiple objectives are provided. In this case, if all the objectives are not satisfied, an inventive problem occurs.

1-1.3 Inventive design (model change)

Optimization methods are proven to be effective in solving many problems [8], [9], but are not effective for problems of inventive design, which require improving systems by adding new decision variables and new links between decision variables. The objective of inventive design is to go beyond the Pareto front of optimization and obtain results in the desired objective space. To go beyond the Pareto front, Genrich Altshuller developed a systematic problem-solving methodology called TRIZ [10]. TRIZ, which stands for *teoriya resheniya izobretatelskikh zadatch* in Russian or 'theory of inventive problem solving' in English is based on the idea that innovation and creativity can be approached in a structured and methodical manner. To lead to new and inventive solutions to technical problems, TRIZ uses two assumptions about contradictions and inventive principles.

- **Evolution of technical systems through contradictions**: TRIZ is built on the premise that systems evolve by resolving contradictions. Contradictions occur when improvements in one aspect of a system lead to a degradation in another aspect. Resolving contradictions drives to innovative solutions.
- **Inventive principles:** TRIZ assumes that there are universal principles underlying inventive solutions that have been employed throughout the history of technical evolution. By identifying these principles and applying them to current problems, engineers and innovators can generate novel solutions.

When applying TRIZ to solve problems, the main objective is to detect contradictions by finding a situation where improvement of one aspect of a system leads to degradation of another aspect, and vice versa. After detection of the contradictory situation, application of inventive principles yields a solution to the problem by resolving the contradiction. For resolving contradictions, TRIZ provides inventive principles.

To illustrate the process of going beyond optimization, Figure 3 illustrates the Pareto front of an exemplary problem [3]. In the illustration, a) shows the Pareto front of an optimization problem. There is a conflict between different objective vectors in regard to the objective functions. For the exemplary problem in b), solution 1 satisfies evaluation parameter 1 and solution 2 satisfies evaluation parameter 2. For both solutions, the remaining evaluation parameter is not satisfied. In inventive design, this situation is named contradictory [11]. The conflict between objective functions is considered a technical contradiction. For solving the contradiction, this conflict is to be translated into the search space of possible solutions, which is to be considered as a physical contradiction.

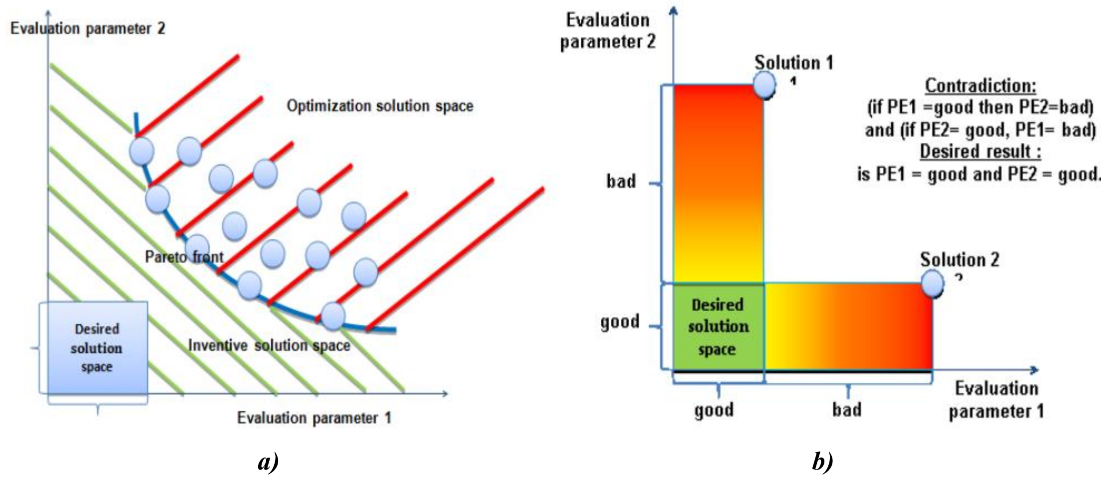


Figure 3: Pareto front and inventive problem

The definitions for the different contradictory situations are given in the literature [10], [12], [13]. The literature separates between administrative, technical and physical contradictions:

- An **administrative contradiction** describes a situation in which an objective is formulated but not satisfied.
- Paired **technical contradictions** describe a situation with two conflicting objective functions. For different solution vectors, the first objective function improves while the second objective function degrades, and vice versa. Satisfying both does not seem possible.
- A **physical contradiction** describes the conflicting situation on the basis of the decision variables. A decision variable must adopt the opposite states (values) simultaneously to satisfy both objective functions.

The idea of solving-problems by describing them through their contradictions obtained from experiments is used and applied in many works [7], [14], [15]. A general framework for the system of contradictions is provided in [16] and is illustrated in Figure 4. The system of contradictions describes the conflicting situation through the technical and physical contradictions and removes the administrative contradiction. It should be highlighted that during optimization, the technical contradiction appears before the physical contradiction. The physical contradiction explains the technical contradictions on a deeper level. There is an assumption that behind each paired technical contradiction there is a physical contradiction.

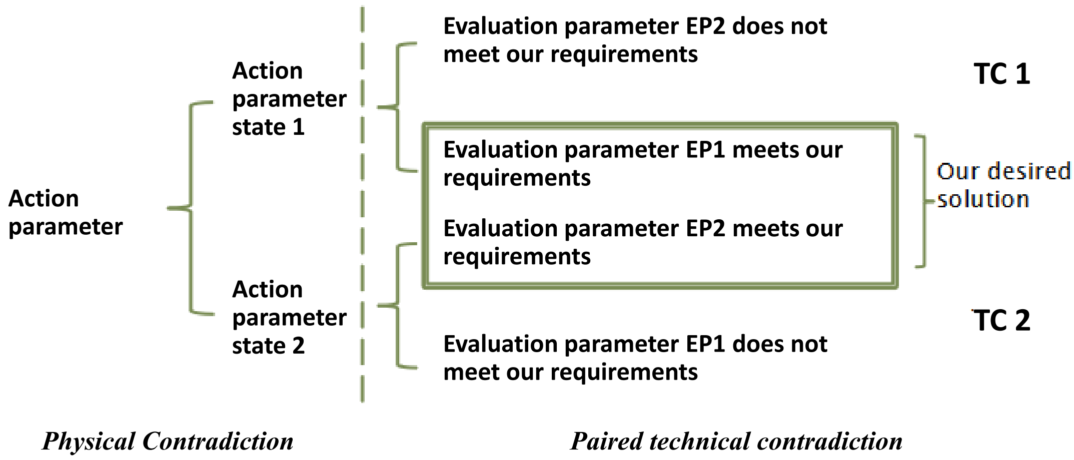


Figure 4: System of contradictions

Inventive design provides systematic methods to describe the problem through its contradictions on several levels. For solving the contradiction, [12] proposes separation principles. Separation principles include spatial, temporal and structural separation of decision variables. The decision maker introduces new parameters and functions in order to change the principle of work and solve the conflict.

1-1.4 Data-driven simulation

Both optimization and inventive design require the model to evaluate solution candidates and provide a link between decision variables and objective functions. For providing the links, a high number of experiments should be simulated. A common technique for analysis of material flows is modelling and simulation in discrete-event simulation (DES). DES enables simulating material flows in a dynamic environment, and analysing waiting lines while considering stochastic uncertainties. The technique can solve a wide range of problems in system design. However, simulation has significant requirements for the system modeler [17]. The process of setting up valid simulation models is time-consuming and requires expertise in the domain. The process of executing simulation studies is illustrated in Figure 5, based on [18].

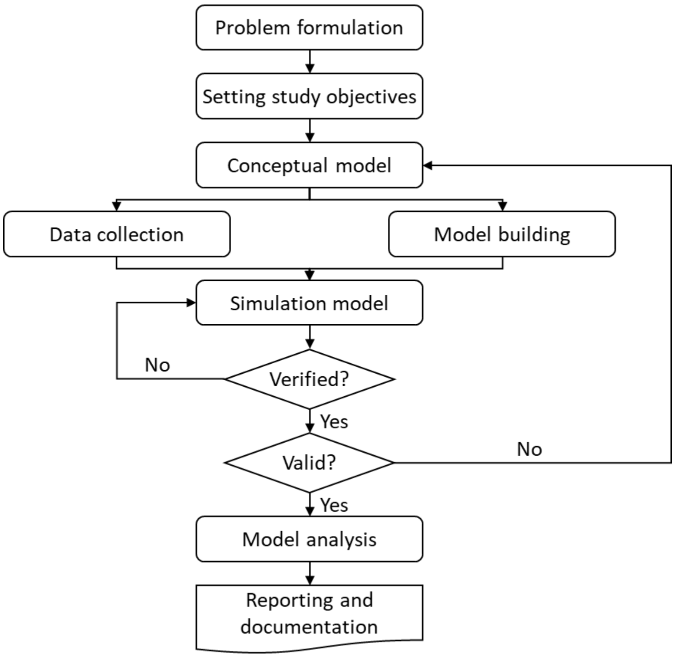


Figure 5: Modelling and simulation process

According to [18], the simulation process begins with problem formulation and setting study objectives. In these phases, the system modeler understands the system and problem, and designs a conceptual mode of the critical system. Based on the conceptual model, the model implementation and data acquisition are performed in parallel, ending with providing the final simulation model. In the verification and validation steps, the correctness and accuracy of the model are ensured. In the model analysis, the model generates new knowledge by evaluating action parameters and providing evaluation parameters by simulation. The system model provides reports for communicating results. Depending on the complexity and size of the system to be analysed, simulation studies can last from a few days to several weeks and months, particularly when proposing, implementing and evaluating new solution concepts.

New opportunities are offered with data-driven simulation, which enables setting up simulation models and executing simulation experiments instantly from the data [19]. However, a literature review in the context of this work demonstrated the current limitations of data-driven simulation [20]. In the literature, a range of methods (e.g.

reuse, parameterization and data-driven modelling) can provide models for simulation, while compromising between genericity and specificity. The strategy and methods that can be used to provide models for problem-solving by data-driven modelling are unclear. Assuming that the strategies and methods are available as data-driven simulation, automated evaluation of a wide range of systems can be obtained within short time.

The benefits and drawbacks of data-driven simulation depend on the quality of the simulation data [21]. Providing simulation data is a major issue in data-driven simulation, and in previous studies it was mainly linked to data-extraction from corporate business systems, called *system data* [22], [23]. In particular, since the rise of Industry 4.0, methods have been sought for extracting simulation data from actual material flows, called *flow data* [24], [25]. The added value of flow data can be seen in the increased precision of historic data.

1-1.5 Retrieval of simulation data

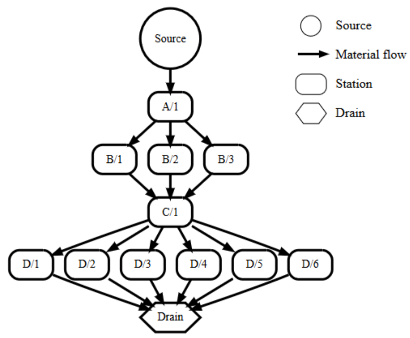
Studies dealing with data-driven modelling and simulation differentiate between the retrieval of simulation data as *system data* and *flow data*. System data describe the physical system. Data change when the configuration of the physical system changes; consequently, changes do not occur regularly. Flow data describe the historic material flows in the physical system. Data change with each finished activity. Consequently, flow data change regularly and are extended with each event in the physical system.

System data describe the products, processes and resources and the physical system. In scientific and industrial literature, there are many works available providing models, standards and exchange formats for exchanging and providing system data for data-driven simulation. Examples for these are the product, process and resource (PPR) model [26], AutomationML [27] and Core Manufacturing Simulation Data (CSMD) [28]. The standards entail predefined classes for definition of system properties and allow the description of entire material flow systems by data. In addition to PPRs, classes can be used to describe the organization of work by providing order schedules and shift calendars. Extensions of the standards are permitted to model system-specific properties. The methods for retrieval of system data in this work are considered as available.

Flow data describes the historic material flows that occurred in a physical system [25]. To provide them, material flows are recorded in event-logs [24]. Event-logs provide information about the begin time and end time of an activity by measuring the moment and writing the event into a database (see Figure 6(a)). Alternatively, to measure the material flows on the shop floor, event-logs can be provided by a workflow-oriented system, e.g. enterprise resource planning (ERP) systems or manufacturing execution systems (MES) [29] by recording feedback from the shop floor. Event-logs provide the historic flows inheriting simulation data, but do not explicitly provide simulation data. Transformation (e.g. a process map) is required to extract simulation data. Figure 6 illustrates the exemplary input and output of the transformation process based on [24]. Extraction provides process-maps of material flows (b), from event-logs with historic data from the shop floor (a). In addition to creating input for simulation, analysis of event-logs allows for extracting knowledge about the historic material flows.

Workpiece	Station	Time
BB90F	B/3	2020-10-08 08:00:0
0184A	B/3	2020-10-08 08:00:0
95A0C	C/1	2020-10-08 08:00:2

a)



b)

Figure 6: Extraction of material flows from event-logs

Given the availability of system and flow data, methods are required to extract simulation data for data-driven modelling and simulation. In particular, to describe material flows, the methods for extraction of the process-map from logs with recorded events are not clear.

1-1.6 Digital shadow

The retrieval of simulation data from physical systems and information systems is a major challenge in simulation modelling and is tackled in numerous works [21]–[23], [30]. A literature review highlighting the challenges of data acquisition on the shop floor from a technical perspective is provided in [31] and identifies challenges, particularly in the acquisition from non-automated manufacturing systems with manual data inputs during the manufacturing process. Surveys at winter simulation conferences in 2002 and 2012 with simulation experts point out that acquisition of simulation data is a major challenge in simulation studies [22], [23]. Between 40 % and 50 % of the time spent on simulation studies is dedicated to data acquisition and, out of these, 80 % of activities are manually executed [1]. Following [21], just 7 % of the simulation data is available when needed. Providing data for simulation remains a major challenge. Origins for the challenge are seen in [25] by the high number and variety of data sources and, in particular, corporate business systems.

New opportunities have been created by the new technologies of Industry 4.0 and particularly the digital twin concept. A ‘digital twin’ was first mentioned in 2002 by Grieves and is defined as a set of virtual information constructs that fully describes a potential or actual physical manufactured product from the micro atomic level to the macro geometrical level [32]. From 2017 to 2019 the digital twin was one of the strategic trends identified by Gartner [33]–[35]. However, the literature review in [36] highlights, beyond laboratory-scale case studies there are barely any implemented digital twins available. Additionally, the authors provided an extended definition: ‘The Digital Twin consists of a virtual representation of a production system, that is able to run on different simulation disciplines that is characterized by the synchronization between the virtual and real systems’ [36]. The core of their definition is the differentiation between a digital model, shadow and twin based on the level of integration between the physical and virtual system (see Figure 7).

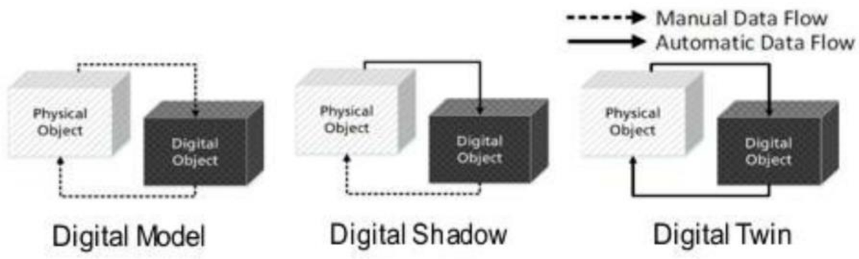


Figure 7: Digital model, shadow and twin

With the automated data flow from physical object to the digital object, the digital shadow is eligible to receive data from the physical object and provide it for modelling and simulation. Consequently, this work focuses on the digital shadow and its ability to provide data for simulation. The purpose of the digital shadow is to use simulations to forecast, optimize and evaluate the outcome of production systems at each life cycle phase [36], [37]. For the use of simulation in the context of the digital twin, Grieves [32] introduced the digital twin environment, an integrated, multi-domain application space for operating on digital twins for a variety of purposes. Consequently, in the context of the digital shadow there is a digital shadow environment providing a multi-domain application space for optimization and problem-solving. In this context, [1] highlights the possibility and gap in research for using big data generated by simulation for problem-solving and improvement. However, to use a digital shadow for problem-solving in manufacturing system design, there are multiple challenges to overcome. Challenges are linked to the retrieval of simulation data, data-driven modelling and simulation, and their application in optimization and problem-solving.

1-1.7 Discussion

Sections 1-1.1 to 0 provide the background for this work. The background covers the domains of problem-solving and data-driven simulation. In problem-solving, the background for both optimization and inventive design has been provided. In the context of simulation, the background has been provided for data-driven simulation, retrieval of simulation data and the digital shadow concept. In the design and redesign of material flows, the presented methods, techniques and tools can support the problem-solving process. The assembly of the retrieved knowledge establishes a theoretical framework for problem-solving based on the digital shadow. The framework is illustrated in Figure 8.

The presented framework relies on Kritzinger's concept of the digital shadow [36] and Grieves' concept of the digital shadow environment with the multi-domain application space [32], both of which are illustrated with a grey background. In the red and blue boxes, the domains of optimization and inventive design are shown. Optimization uses data-driven simulation in the multi-domain application space to solve design problems by optimizing system parameters. Experimental data from optimization is the enabler for detecting contradiction and adding new system parameters for changing the model and solving design problems.

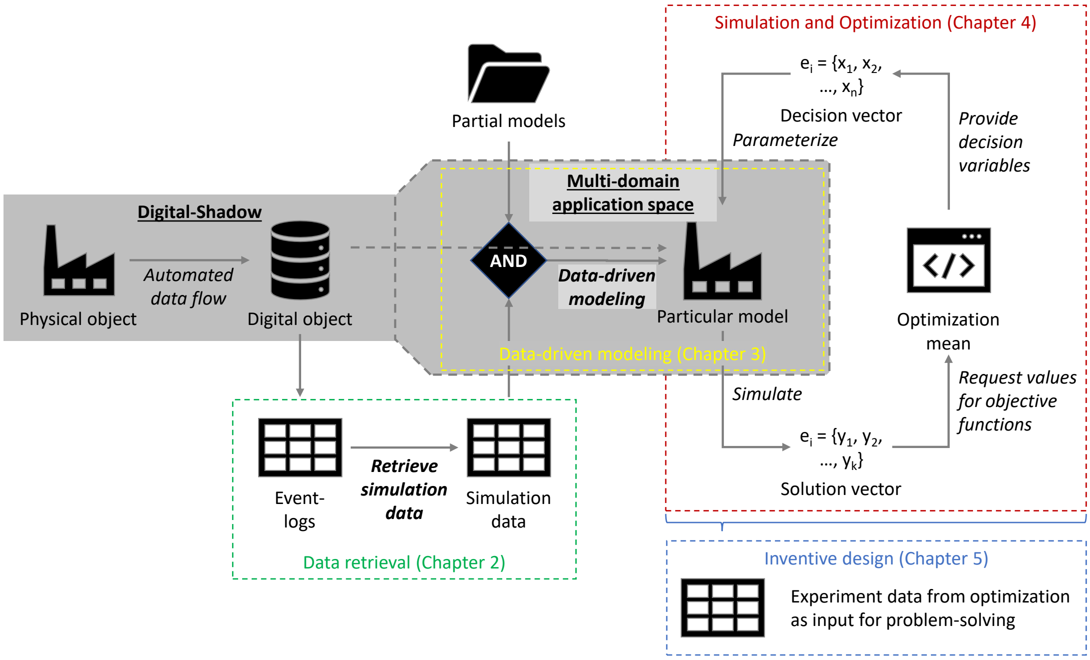


Figure 8: Technical framework for problem-solving with the digital shadow

The initial point in the presented framework is the physical object. In the digital shadow concept, it is represented by its digital object, receiving data via an automated dataflow. The digital shadow is the data source for simulation and provides system data (PPR-data) and flow data (event-logs). Methods of data-retrieval extract simulation data from the event-logs of the digital shadow. Data-driven modelling and simulation provides simulation models for optimization and experimentation. As input for data-driven modelling, there are simulation data and partial models, providing particular models, as proposed by the concepts of *enterprise modelling* [38]. Partial models provide reusable concepts for modelling in this work, particularly for job-shop systems. Simulation data provides the case-specific data of the physical object. Data-driven modelling provides particular models of the physical objects in the digital shadow environment by instantiating particular models from partial models according to the simulation data.

For optimization, the particular model allows the solution candidates to be evaluated by simulation. Optimization means (e.g. heuristics) suggest solution candidates by providing decision vectors with values for the decision variables. Parameterization of the particular model provides a model of the solution concept, while simulation evaluates the solution concept and provides a solution vector with values for the objective functions. Optimization means use data-driven simulation in a loop to evaluate solution candidates. Optimization experiments also provide data linking decision vectors and solution vectors. In problem-solving, when no viable solution is available, inventive design can be applied. Simulation-based experimental data and methods of inventive design enable the description of the technical and physical contradiction, changing the model by adding new decision variables. Changing the model and adding new decision variables have the consequence of extending simulation data and partial models to enable data-driven modelling. However, in this framework there are four major issues to be solved.

- Retrieval of simulation data: Techniques of Industry 4.0 enable the measurement and recording of data, particularly event-logs on the shop floor, and provide them in the digital shadow. Simulation requires data describing material flows. Extraction of simulation data is required to provide material flows from event-logs with recorded events. Nevertheless, the methods for extracting material flows are not yet clear, especially for systems with high uncertainties in the process.

- Data-driven simulation: Data-driven simulation allows scenarios to be evaluated in optimization and problem-solving by simulation on click. However, previous studies highlight a range of methods that are available to provide different types of models for different types of systems. The methods for generating models by data-driven modelling, automation and simulation experiments are not available.
- Simulation and optimization: Optimization provides solution candidates that minimize or maximize given objective functions. For design problems with multiple objectives, one best solution cannot be provided. Moreover, a set of non-dominated solutions describes trade-offs between objectives and is required as source data for solving the design problem by inventive design. The methods for receiving the experimental data from a digital shadow and data-driven simulation are not clear.
- Problem-solving: Beyond optimization, problem-solving benefits from the availability of experimental data, which provides the link between decision variables and values for the objective functions. Consequently, it inherits technical and physical contradictions that define system limitations. Support of the model change for extracting the contradictions is required. However, the methods for extraction of contradictions are not available.

1-2 Research statement

The global framework of this PhD thesis is the redesign of production systems by their digital shadow, which deals with the role of data-driven simulation of material flows in problem-solving. In this global environment, three research questions address the topic of data-driven modelling from the digital shadow (RQ1), using generated models for experimentation and optimization to provide data with the link between action and evaluation parameters (RQ2), as well as using data-driven modelling to support the problem-solving loop and changing the model by detecting contradiction and inventive design (RQ3). Based on literature research and its gaps, in the state of the art, there are three research questions that are addressed in this work.

RQ1: Modelling and Simulation	**What are the methods for generating simulation models for problem-solving from the digital shadow?**
RQ2: Simulation and Optimization	**What are the methods for optimizing material flows through the digital shadow and data-driven simulation?**
RQ3: Model Change	**What are the methods for solving problems in the design of material flows by the digital shadow and data-driven simulation?**

Additionally, there are three assumptions (AS1–AS3) defined to provide a framework for this research and define the type of manufacturing systems, availability of data and relevant stage in the manufacturing system lifecycle:

AS1: System type	**Systems from type job shop;**
AS2: Availability of data	**Data is available digitally and instantly (PPR-data and historic event-logs);**
AS3: Lifecycle stage	**It is focused on the stage of planning (design and redesign) in the lifecycle of material-flow systems.**

The formulated assumptions limit the scope of this research in regard to system type, lifecycle phase and data-availability. The literature research highlights that for different types of systems, different problems and different methods for modelling are available [20]; a common approach does not exist. Because the focus is on systems, whose material flows follow the paradigm of a job shop, data-driven simulation requires data from the physical system. Data-gathering from the shop floor is time-consuming and can include additional research questions [21]. Because data from the physical system are assumed to be available in the digital shadow instantly and digitally, during the system lifecycle there are many problems to be solved. To limit the scope, this work focuses on problems in the design and redesign of systems.

1-3 Research method

As a research method to address the stated questions, this work relies on action research, which uses a cyclic approach to tackle problems and generate knowledge [39]. In action research, an undefined and unlimited number of cycles sequentially addresses research questions. The phases of this cycle are *plan*, *action*, *overserve* and *reflect*. After each reflection, a revised plan can be formulated and allows entering another cycle. The cyclic approach allows us to sequentially address the research questions provided after each problem and the outputs that can be used for addressing the next set of problems. Figure 9 shows the phases of action research.

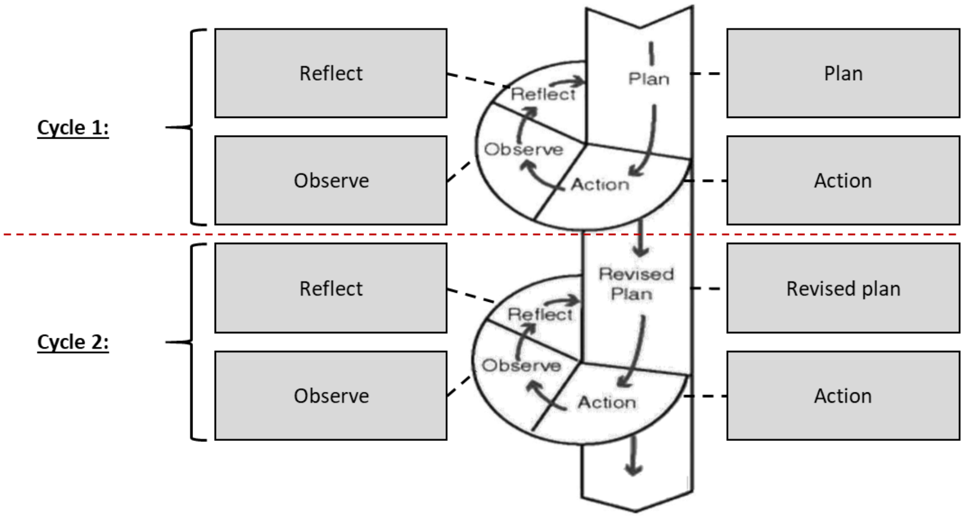


Figure 9: Action research cycle

- Plan: The researcher formulates the objectives and defines the method to address the objective.
- Action: The researcher executes the plan by implementing new methods, processes and technologies.
- Observe: The researcher gathers data to observe the impact of the implemented interventions.
- Reflect: The researcher analyses the data to evaluate the interventions, reflects on new insights and formulates new objectives.

In this work, there are four research cycles to address the three questions. The first question (RQ1: Modelling and simulation) is split into two cycles, where each cycle is described in one chapter. The first cycle addresses retrieving simulation data from the digital shadow (Chapter 2) and the second cycle addresses the generating and simulation models of material flows (Chapter 3). The outputs of the first question, the methods and tools to generate and simulate models of materials, are enablers for addressing the second question. The second question (RQ2: Simulation and optimization) is addressed in the third cycle and provides the methods to use simulation for generated models to address various optimization problems in the case studies, particularly in scheduling, resource allocation and constraint optimization (Chapter 4). The third question (RQ3: Problem solving) is addressed in the fourth cycle and provides the methods to use data-driven simulation and optimization to solve inventive problems (Chapter 5). An overview on the cycles and chapters of this work is given in Figure 10.

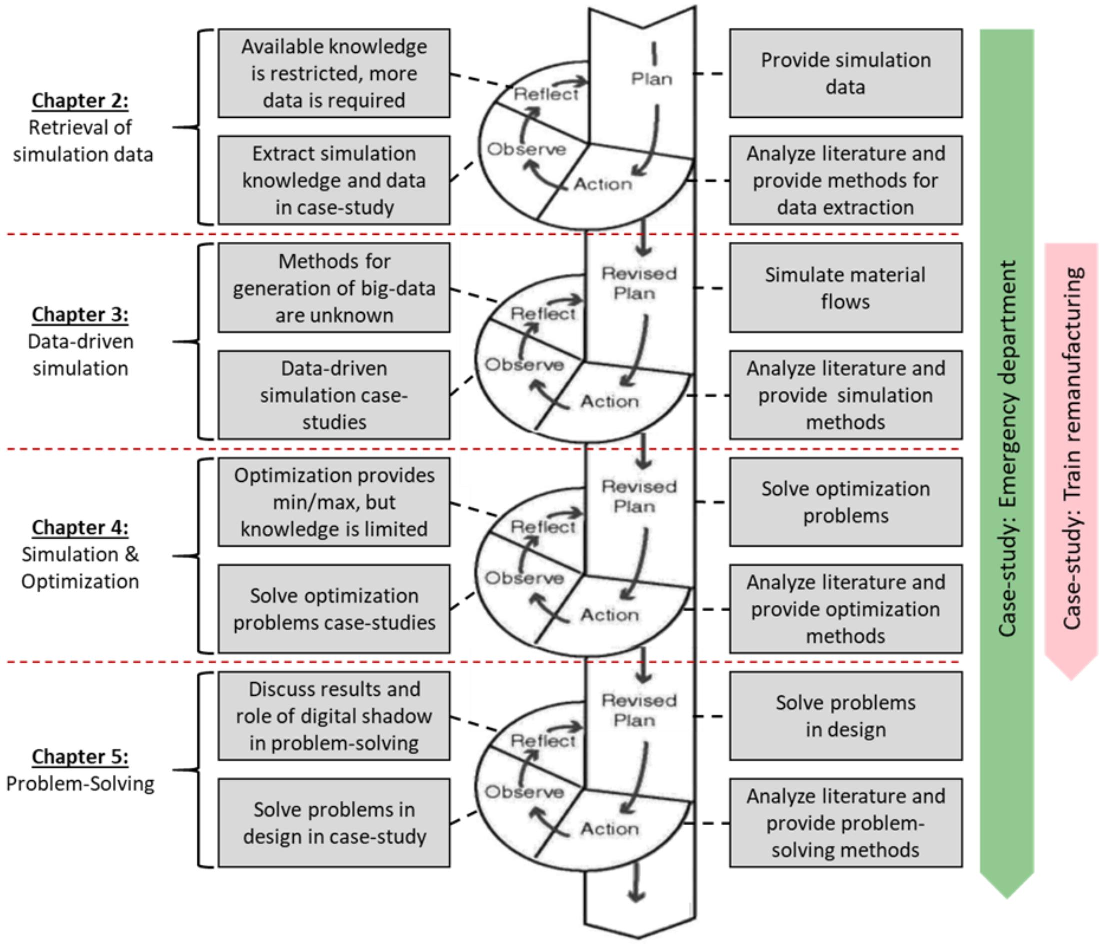


Figure 10: Research cycles in this work

1-4 Organization of the thesis

Chapter 2 (Cycle 1) aims at providing methods for retrieval of simulation data. The plan is to extract data with material-flows from event-logs with historical data provided by the digital shadow. In the action phase, a literature review is provided to understand the methods in the literature. Based on the literature review, new methods are proposed and implemented to receive simulation data. A case-study in the ED is presented to validate the methods and provide simulation data. The methods are evaluated, properties are discussed, and new objectives are formulated.

Chapter 3 (Cycle 2) is intended to provide methods for data-driven simulation. The plan is to provide models and execute experiments on request from data. Experiments evaluate sets of action parameters by simulation and provide sets of evaluation parameters. In the action phase, a literature review is provided to understand the methods in the literature. Based on the literature review, new methods are proposed and implemented to generate and simulate models. Two case-studies are presented in ED and TRM to validate the methods and provide simulation results. The methods are evaluated, properties are discussed, and new objectives are formulated.

Chapter 4 (Cycle 3) has the objective of providing methods to solve optimization problems by simulation. The plan is to provide methods to use models, generated from the digital shadow, to solve various optimization problems, particularly in scheduling, resource allocation and constraint optimization. In the action phase, a literature review is provided to understand the methods in the literature. Based on the literature review, new methods are proposed and implemented to solve optimization problems. Two case-studies are presented in ED and TRM to validate the methods and determine optima. The methods are evaluated, properties are discussed and new objectives are formulated.

Chapter 5 (Cycle 4) is dedicated to problem-solving, based on data-driven simulation and optimization. The plan is to provide methods to solve design problems, based on results from data-driven simulation and optimization. In the action phase, a literature review is provided to understand the methods in the literature. Based on the literature, a review of new methods for problem-solving is presented. A case study is presented in the ED to validate the methods and solve design problems. The methods are evaluated, properties are discussed and new objectives are formulated.

Chapter 6, (Conclusion) summarizes the results of the previous chapters, highlights contributions and discuss their benefits and drawbacks. Areas for future research are also given.

1-5 Case-Studies

This work uses two case-studies of redesign: an ED and a TRM. The focus of both case-studies is the redesign of physical flows, specifically, patient flows in the ED and train flows in the TRM. Despite the domains and differences in the system, the fundamental mechanics of both systems follow the paradigm of a job-shop production system. The system is characterized as a shop of different resources with similar functions for executing activities, i.e. doctors and nurses in the ED and installations in the TRM. Materials arrive at the shop floor and follow a sequence of activities. Each activity requires a resource to be executed; missing resources cause queues and waiting times, increasing the patient's length of stay (LOS) in the ED and the train's LOS in the TRM. When competing for a resource, priority is given by rules. A patient's prioritization is based on the emergency severity categorization in the ED and on the arrival sequence of the train in the TRM.

Owing to the nature of the system, the 'batch size of production orders' is equal to one. Each patient or train arrives individually with case-specific problems, i.e. diseases in the ED and technical defects in the TRM. The process in both cases is marked by assembly and dismantling activities. Indeed, during the ED process, patients arrive in the system and are 'assembled/dismantled' with stretchers, samples and reports. Patients need to be separated from samples and reports during their journey in the ED, and from the stretcher before leaving. In the TRM, during the remanufacturing process, the train, which is the *train à grande vitesse* (TGV) in our case study, is broken down into its two power cars and the eight or ten carriages that make up its trainset. In further descriptions, we use the word ‘wagon’ for both carriages and power cars of the trains. Before leaving the TRM, the wagons are reassembled to form a train.

Both systems are marked by a high degree of uncertainty. For both patients and trains, all the activities to be undertaken during the stay in the system are not clear at the moment of arrival. In the ED, the doctors decide during consulting which tests are required to make a diagnosis, and in the TRM the technicians decide which parts of the wagons are to be replaced. Consequently, the sequence and durations of activities are partially generated during the process. Table 1 displays an overview of both cases.

Table 1: Comparison of the two case studies

	Emergency department	*Train remanufacturing system*
Material	Patients (Unplanned arrivals)	Trains (Planned arrivals)
System type	Job-shop-production system	Job-shop-production system
Assembly/Dismantling	Stretchers; samples; reports	Wagons
Uncertainty of activity sequences	Sequence is unknown on arrival and in doctoral consulting	Sequence is unknown on arrival and determined in diagnosis
Uncertainty in activity durations	Activity duration is determined on the shop floor	Activity duration is determined on the shop floor
Resource	Doctors, nurses	Installations
Measure	Length of stay	Makespan (for a set of trains) Length of stay (for each train)

The case studies have the purpose of highlighting how data-driven simulation from the digital shadow can support problem-solving and validate the proposed models, even though the domains of the cases are completely different. We also questioned whether cross-fertilization among these domains is possible by comparing how the similar problems are solved in each area. Therefore, the case-study of the ED is used for retrieval of simulation (Chapter 2), data-driven simulation (Chapter 3), simulation and optimization (Chapter 4) and problem-solving (Chapter 5). The case-study of the train remanufacturing system is used for data-driven simulation (Chapter 3) and optimization (Chapter 4). Both case studies are introduced in this chapter.

1-5.1 Emergency department

This case study tackles the improvement of the patient flows in a hospital, in particular, the ED. Patients arrive at the ED with a given arrival stream, defined by the inter-arrival time (iat). The average arrival stream cycles; it depends on the day of the week and hour of the day. At the moment of arrival, there is no information about the pathology of the patient and the available pathway in the ED. The general pathway of the patient is illustrated in Figure 11. Patients register, see a nurse (triage), see a doctor, go through the testing loop and leave the ED. In the waiting area, the nurses assign the patient a priority level in accordance with a severity index. During the doctoral consulting, a doctor checks the patient, defines the tests to be performed and defines the patient's pathway (routing). The patient enters the testing sequence and executes the tests. After the sequence of tests, the patient sees the doctor for a final diagnosis and leaves, or further tests are required. In the last case, the patient enters a new testing sequence before seeing the doctor again. We call the 'see doctor and testing sequence' the 'testing loop'. The patient may perform several testing loops before leaving the ED. The pathways within a testing loop are illustrated in Figure 12.

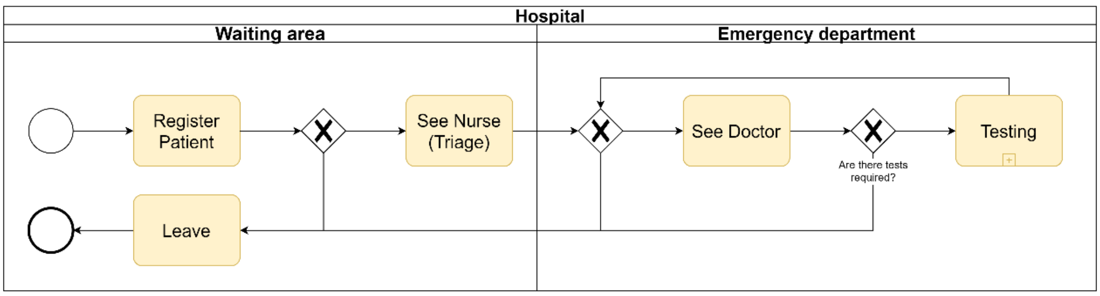


Figure 11: Patient pathways (from the point of view of sequence of activities)

Depending on the patient's pathway, there are different tests to be executed: blood sampling, scanning (RX) and imaging (Scan). If multiple tests are to be performed, there is a defined sequence. The sequence is blood sampling, scanning and imaging. The execution of activities requires resources. Unavailable resources cause waiting times and increase the patient's LOS. The severity index assigned to the patient in the waiting area (see Figure 11) defines the priority of patients for access to resources..

The patient flow is accompanied by an information flow. The information flow describes samples and reports created during the testing loop. At the end of each sequence of tests, patients must wait for all results to be communicated to the doctor. In the activity map in Figure 12, samples and reports of blood sampling, scanning and imaging are shown in purple, turquoise and orange, respectively.

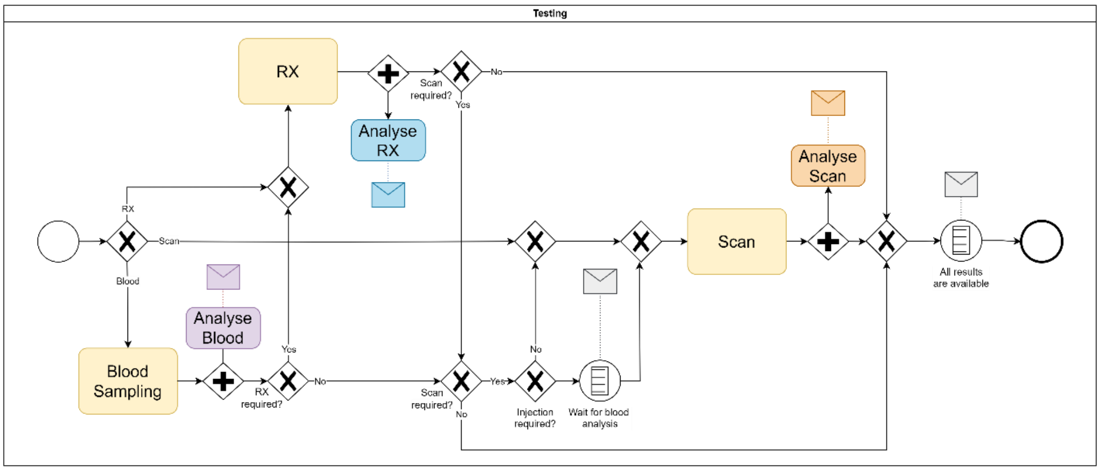


Figure 12: Testing loop

In the case study, there are two situations to be compared: the initial situation (S1), and a planned future situation (S2). Following the definition of the initial and future situation:

- In the initial situation, patients are organized with three increasing severity levels: green, orange, red. The severity levels are supplied by the hospital and reflect the history. In addition, the ED is organized in three zones. The patients are assigned to zones to balance the number of patients of each colour level in each zone (i.e. each colour should be equally represented among the zones). There are a given number of resources: stretchers, nurses, doctors and boxes to be assigned to the three zones for treating the patients.
- In the future situation, the patients are classified into two groups (blue and orange) using an algorithm based on the Emergency Severity Index (ESI), which

is supplied to us by the ED [1]. The ED is organized into two zones, dedicated to orange and blue patients, respectively. There is a given number of stretchers, nurses, doctors and boxes to be assigned to zones for treating the patients.

An overview of the scenarios and parameters for comparison are listed in Table 2. Regarding the resources, switching from the initial situation to the future situation is accompanied by the removal of four doctors. In the initial situation, there are ten doctors that are assigned to three zones in the future situation six doctors are assigned to two zones. In parallel, the number of nurses assigned to the zones is increased by 1 from three in the initial situation to four in the future situation. The set of arriving patients is identical and is defined by the sequence and inter-arrival time.

Table 2: Scenarios of the emergency department

<table>
<tr><th></th><th>Initial situation (S1)</th><th>Future situation (S2)</th></tr>
<tr><td>Categorization</td><td>green / orange / red</td><td>blue / orange</td></tr>
<tr><td>Number Zones</td><td>3</td><td>2</td></tr>
<tr><td>Patient assignment</td><td>Mixed coloured zones
Balancing stock level of each colour</td><td>Specialized zone,
one for each colour</td></tr>
<tr><td rowspan="2">Historic patient arrivals</td><td colspan="2">Same set of patients and inter-arrival times</td></tr>
<tr><td>13252 green patients
16877 orange patients
5111 red patients</td><td>19860 blue patients
11642 orange patients</td></tr>
<tr><td>Stretchers (beds)
(zones: 1 / 2 / 3)</td><td>10 / 10 / 10</td><td>10 / 20 / -</td></tr>
<tr><td>Doctors
(zones: 1 / 2 / 3)</td><td>4 / 4 / 2</td><td>2 / 4 / -</td></tr>
<tr><td>Nurses
(zones: 1 / 2 / 3)</td><td>1 / 1 / 1</td><td>2 / 2 / -</td></tr>
<tr><td>Boxes
(zones: 1 / 2 / 3)</td><td>5 / 5 / 5</td><td>5 / 10 / -</td></tr>
<tr><td>Stretcher-bearer</td><td>1</td><td>1</td></tr>
<tr><td>RX</td><td>1</td><td>1</td></tr>
<tr><td>Imaging</td><td>1</td><td>1</td></tr>
<tr><td>Prioritization
(low < high)</td><td>green < orange < red</td><td>blue < orange</td></tr>
</table>

[1] The Emergency Severity Index (ESI) is a five-level emergency department triage algorithm, initially developed in 1999. It was previously maintained by the Agency for Healthcare Research and Quality (AHRQ), but is currently maintained by the Emergency Nurses Association (ENA) [40].

1-5.2 Train remanufacturing system

This case study tackles the improvement of the material flow in a remanufacturing system for high-speed trains. Trains arrive at the remanufacturing system with a given schedule. The schedule defines the type of the arriving train and the inter-arrival time. Each type of train has specific processes (phases) to undergo, defined by the train type process (plan) routing. However, the exact activities and workload to be performed depend on the condition of the trains. During the routing, the trains are dismantled to the single wagons (DA, D1, …, DB), each wagon undergoes a sequence of processes, and the wagons are assembled to form the entire train (see Figure 13). During the dismantling and assembly process, there are technical constraints to be considered. Technical constraints describe precedence and successor relationships between phases of different wagons, e.g. the wagon DA has to be removed before removal of D1. In addition to dismantling and assembly, there are additional technical constraints during the entire process. Furthermore, the execution of processes underly start conditions. A list of start conditions is provided by the decision makers and defines permitted moments of start for each phase, e.g. painting cannot start before the weekend to not interrupt the process.

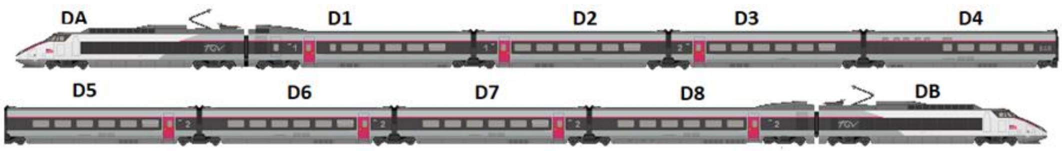


Figure 13: Train dismantling and assembly

The execution of phases requires resources (installations), e.g. installations for painting and washing. Each installation processes one wagon at the time. A bill of resources provides the resources, available on the shop floor. Each resource is available one or multiple times. A shift-calendar describes the temporal availability of resources. During execution of processes, resources are bound for given durations and cannot perform other tasks. Unavailable resources cause waiting times and increase the LOS of trains. Within the waiting lines, the arrival date of the trains at the shop floor defines the priority. Priority is given to trains with early arrival dates. On the shop floor, resources are arranged in a given layout. However, in this case study, the layout is not considered and traveling of wagons between resources is assumed as finished instantly.

In the case studies, there are several scenarios to be compared. The scenarios are defined by the scheduling parameters for the arrival of trains on the shop floor, particularly the *inter-arrival time* (iat) and the maximum number of trains on the shop floor (nbT). The inter-arrival time defines the temporal distance between the arrival of two trains, and the number of trains defines the maximum number of trains that is accepted to be on the shop floor at the same moment. For comparison of scenarios, there are two measures to be taken: the makespan of trains, and productivity of the system. The makespan of trains describes the average time between arrival and departure of trains (average LOS), and the productivity describes the system efficiency, defined as the takt time for departure of trains from the shop floor. An overview of the parameters and values is displayed in Table 3.

Table 3: Scheduling scenarios in train remanufacturing systems

	Future situation (S2)
Inter-arrival time	iat = {0, 1, …, 8, 9}
Maximum number of trains	nbT = {1, 2, …, 9, 10}

Chapter 2 Retrieval of simulation data

This chapter is dedicated to the question of the generation of models for material flow simulation from the data (RQ1): **What are the methods for generating simulation models for problem-solving from the digital shadow?** In particular, this chapter tackles the challenge of providing simulation data for data-driven simulation (see Figure 18). To address this question, this chapter provides the state of the art in the retrieval of simulation data (section 0), provides the methods to extract simulation data from event-logs (section 0), validates and evaluates the methods in the case study of the ED (section 2-3) and provides a discussion based on the retrieved knowledge from the case studies (section 0).

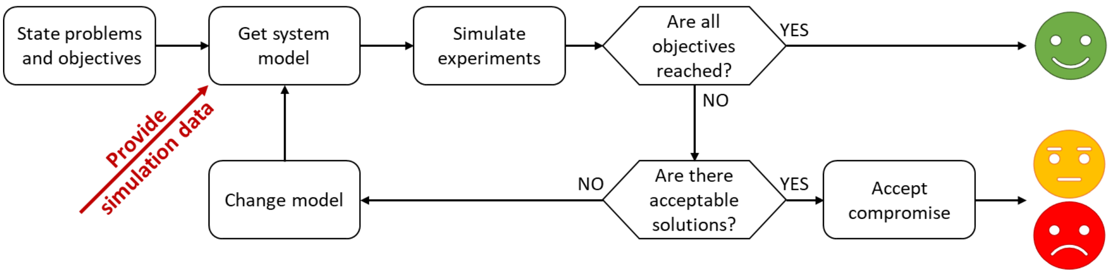


Figure 14: Providing simulation data for problem-solving

The state of the art provides definitions of simulation data and shows how simulation data is extracted in the literature. The contributions in the context of data retrieval are the methods for extraction of simulation data. The methods enable extracting product arrivals and process maps from event-logs with historical data. Product arrivals describe the arrival of materials and process maps describe the sequences of activities. The case-study presents retrieval of simulation data for the case of the ED. The case study describes patient flows on a generalized level, i.e. material flows. In the case study, the product arrivals describe the patient arrival and the process maps their pathways. The methods are generic and provide product arrivals and process maps from event-logs for cases in different domains.

2-1 State of the art

The retrieval of simulation data has the purpose of extracting relevant data from the information system and providing it for simulation. The state of the art has the purpose of identifying data for modelling and simulation, and providing methods for the extraction of simulation data from the literature. The state-of-the-art is separately provided for extraction of system data (section 0) and extraction of flow data (section 0). System data provides information about the non-static physical system, and flow data provides information about the dynamic material flows.

2-1.1 System data

In the context of job-shop production systems, Son and Wysk presented multiple works [41]–[43], addressing the problem of generating simulation models from data. In [38], they defined the data, particularly system data, that is required for model generation. For defining system data, the authors described the requirements for modelling and simulation of job-shop production systems. Based on the requirements, they derived the required data. In addition to providing the content and structure of system data regarding the physical system, the authors highlighted the need to define experimental data, which describes the experiments to be executed (decision vectors) and the results of the simulation (objective vectors). For validation, the authors provided a simulator to simulate a lab-scale system. The data for the simulation of job-shop production systems are as follows:

- **Header information** describes the experiments and the configuration of the systems in each experiment. The data provides sets of action.
- **Experimental information** defines the environment of the simulation experiments, e.g. duration and number of replications.
- **Shop-floor information** describes the resources of the shop floor by stating the amount, type and temporal capacity of resources.

- **Product and process information** describes processes for each product by stating durations and required resources.
- **Production information** describes the production schedule by defining orders, their quantities and moments of arrival.
- **Output information** manages and provides the simulation results, which provide sets of evaluation parameters for optimization and problem-solving.

The literature provides a wide range of standards and exchange methods to automate the exchange of data between information systems and simulation environments. From a technical perspective, [44] proposes to exchange PPR data between different product lifecycle management (PLM) environments with XML, which is a mark-up language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. XML is appropriate to exchange data between the digital shadow and a simulation. However, a mark-up language does not predefine classes for the exchange of particular types of data.

From a modelling perspective, PPR is used in multiple works to model production processes. In [45], [46] PPR is used to exchange simulation data between engineering software and a non-DES simulation environment. PPR is a common standard for the exchange of engineering data in the product lifecycle through various information systems. Therefore, the standard describes the products to be manufactured by the bill of materials and the resources of the shop floor by the bill of resources. The processes describe the manufacturing process by linking products and resources. The standard is supported by systems for PLM and MES, as well as various simulation tools [47]. PPR is the basis of ISA-95 and AutomationML.

ISA-95 is an international standard, developed for the integration of business enterprises and control systems [48]. In [46], the authors use ISA-95 to orchestrate data exchange between the systems of ERP and MES. The purpose of ISA-95 is to generate a common understanding for communication and integration of information systems. However, integration of simulation environments is not within the scope of this standard. However, studies have been conducted using the standard for exchange of system data for simulations of material flows [50], in particular, of a conveyor system in the continuous manufacturing industry.

Another PPR-based exchange standard in industry and research is AutomationML. In [48], the authors use AutomationML to exchange simulation data between PLM tools of the digital factory to evaluate a planned system by simulation in a dynamic environment. For the exchange, the authors provide interfaces to integrate AutomationML into the tools of the digital factory. Additional work is presented in [52] and uses AutomationML to model product, process, and resource data (PPR-data) of a production process and transfer these to a DES. With the standard, the authors provide system data for modelling and simulation of job-shop systems and evaluate a conceptual process through data-driven simulation.

The more specific standard, CMSD, was developed to provide data from corporate business systems, e.g. ERP and MES, for simulation in DES. In [50], the authors focus on exchanging data and generating models in DES based on CMSD. In another work, the same authors discuss the benefits and drawbacks, and provide the research agenda for the standard [54]. Additional work with CMSD addressing the exchange with data from the shop floor is presented in [55] and highlights the limitations in the definition and exchange of simulation results in CMSD.

In addition to the presented standards and formats, STEP has been developed to focus on the exchange of CAx data from computer-aided design systems [53], and SysML, a modelling language, has been used for simulation modelling [56], [57]. The authors also developed individual standards for exchange owing to individual demands, for instance in [58]–[61]. A common property of the presented standards is the objective of exchanging engineering data between corporate business systems and simulation environments. Case-studies succeeding in the data exchange of static engineering data are widely available.

2-1.2 Flow data

New challenges arise with dynamic production data, which describes the historic production flows on the shop floor and is recorded in event-logs with timestamps for the beginning and end of activities. According to [62], this data is gathered in 'non-standardized', self-built systems and is available as unfiltered raw data. There are two significant challenges: gathering the data from the shop floor, and extracting relevant data and knowledge for simulation. A survey from 2015 executed with German manufacturers shows that 80 % of the companies can provide timestamps for the end and 63 % additionally measure the beginning of activities. From the analysed companies, 53 % state have unlimited evaluation options regarding data quality [63]. However, in industry, data is gathered in different ways. To provide the state of the art, this section investigates available approaches from the literature.

The synthesis of the literature review is provided at the end of this subsection in Table 4. The publication metadata defines the authors and the area. The area defines the domain of application, which is either manufacturing (Ma), banking and insurance (BI) or ED. The retrieval method refers to the methods provided by [23]: direct entry (1), manually populated external data source (2), automatically populated external data source (3) and direct link to an external data source (4). The column data states the output of the data retrieval and the input for knowledge extraction. Methods for knowledge extraction are provided for product arrival, process and resource data. For blank fields, the authors either did not state the methods or the problem was not addressed in this work.

Goodall et al. [64] focus on the simulation of remanufacturing systems. As input for computer simulation, the authors use static engineering data and knowledge from corporate business systems and domain experts and extend this with data gathered with radio-frequency identification (RFID) tags during the manufacturing process. To gather data, the company equipped cores during the manufacturing process with RFID-tags and individually tracked the progress within the factory [65]. Data from RFID tags provides data about work in progress, the arrival of cores, activity times and failure rates. In simulations, this information initializes and parameterizes the model. However, extraction of event-logs and material flows is not within the scope. The main inputs for simulation modelling are the processes from static engineering data. Dynamic production data initializes the model to provide the initial stocks and provides parameters for the activities.

Reinhardt et al. [25] has highlighted model generation research gaps, especially in simulation from data describing dynamic production flows. To tackle this problem, the authors in [24] focus on gathering relevant information from the shop floor. To gather traceability, the authors use RFID-tags to provide event-logs with the historical material flows on the shop floor in data storage. An exemplary data extract is given in Figure 15 and provides information about historical flows by providing the workpiece, station and timestamp. To extract knowledge, the authors analyse the event-logs to directly provide new insights about the system and extract simulation data to arrive at new insights via simulation. For simulation data, they propose to extract the process map to describe the material flows. In their work, they present a data analysis and put utilization for modelling and simulation on the research agenda for future works.

Workpiece	Station	Time
BB90F	B/3	2020-10-08 08:00:07
0184A	B/3	2020-10-08 08:00:07
95A0C	C/1	2020-10-08 08:00:22

Figure 15: Exemplary event-log

Van der Aalst introduced the concept of object-centric process mining in [66]. As a challenge in process mining, it is highlighted that activities and events of the same case can involve multiple objects of different types, e.g. different products in the bill of materials. Additionally, when describing a process modelling language, e.g. Business Process Model and Notation (BPMN), AND and OR decisions can be included. Classical process mining does not address these challenges of nonlinear sequences with splitting and merging flows. To overcome limitations, the authors propose to use object-centric event-logs. These event-logs, in addition to the sequence of events, log the objects that are involved in the logged events. For illustration, they use an order handling process, involving orders, items, packages and routes. In focus of the paper was to provide a tutorial for dealing with object-centric process mining; therefore, the authors formulate the problems and provide abstract models to define data structures of event-logs and link them to physical objects. Simulation of the data is not the focus. An exemplary object-centric event-log based on [67] is illustrated in Figure 15.

		activity		**objects involved**	
event	**time-stamp**	***name***	***type***	***components***	***assembly***
1	0.00	1	start	A001	-
2	0.00	3	start	B001	-
3	0.35	1	finish	A001	-
4	0.35	2	start	A001	-
5	0.45	3	finish	B001	-
6	0.45	4	start	B001	-
7	0.51	4	finish	B001	-
8	0.62	2	finish	A001	-
9	0.62	5	start	{A001,B001}	D001
10	0.76	5	finish	{A001,B001}	D001
11	0.76	6	start	{A001,B001}	D001
12	0.88	6	finish	{A001,B001}	D001

Figure 16: Exemplary object-centric event-log

Lugaresi and Matta focused on the generation of simulation models and digital twins from mined production data [68]. The authors proposed object-centric event-logs for the description of assembly processes by linking events to the affected materials and final product via the bill of materials [69]. The object-centric event-log can provide all events that belong to a particular product. The authors extracted process maps from event-logs with historical material flows and used data-driven modelling to provide a digital model [70]. The origin of event-logs is in manufacturing systems. The authors assumed that data is measured via sensors in the physical system. By applying object-centric event-logs, the authors were able to describe complex assembly processes. A lab-scale case study is presented in [71]. As activities for future research, the authors added the extraction of statistic distributions from the event-logs to describe the duration of activities in stochastic models.

Multiple authors have tackled the challenge of providing laws with stochastic uncertainties. In [72], the authors highlighted the occurrence of uncertainty, particularly for production domains with a high percentage of labour-intensive processes, such as textile manufacturing. To provide laws with statistical uncertainties, the authors acquired data in a field study by measuring the duration of activities. For providing statistical laws, the authors used statistical software and evaluated the fitted distributions by applying the chi-squared test. The authors applied the fitted distribution to the simulation of the manufacturing system. With the background of uncertainty in processing times, failure and repair of machines, Law and McComas [73] provided a tool for distribution fitting. For validation, the authors fitted distributions for real-world data and provided specifications and rankings. In an additional study, Swanson [74] proposed the Monte Carlo method to generate samples for randomized simulations. To model orders in a case-study, they used a sample of deliveries that were planned for a given period and pulled random samples to increase the number of orders. By using this strategy, the authors generated additional orders and increased the duration of the simulated time in a case study.

Kumbhar et al. [72] focused on bottleneck detection from dynamic production data via direct data analysis and simulation. To provide data, the authors extracted and aggregated multi-source data from a manufacturing company's information system. The extracted production data provided event-logs with historical material flows. From the event-logs, the authors extracted three types of information: process maps, inter-arrival times, and activity durations. The authors extracted the process maps by extracting the sequence of events for each material on the shop floor. For each activity, they extracted the inter-arrival time (IAT) and duration. Therefore, for each activity, they measured the time between two product arrivals and the time between begin and end. They used this information to build a model and run simulation experiments to detect bottlenecks. In parallel, the authors attempted to detect bottlenecks through direct data analysis of the extracted data without simulation. In applications, they succeeded in extracting bottlenecks with both approaches, direct analysis and simulation. However, in [75] the authors highlight the added value of detecting bottlenecks without effort for modelling and simulation in the direct analysis, and the added value of execution of what-if scenarios in DES.

Cheaitou et al. [74] faced similar challenges in the context of modelling and simulation of patient flows. The authors received data about patient arrivals from the hospital's information system. From the information system, they extracted the arrival streams and determined the arrival rates for patients with different severity indices. However, the authors used average values for the simulation. For the modelling of patient flows, they modelled a flowchart of the patient flows with experts. Another challenge is seen in modelling treatments, particularly the estimation of activity durations. The authors gathered data and build a statistical distribution for use as an input parameter, e.g. for treatment durations given by the physicians of the ED.

In [76], Lay extracted data from a hospital's database using data mining technologies. Data is classified as administrative or medical information. The authors removed the patient information and focused on the extraction of the event-logs with patient pathways through the ED. Therefore, the authors grouped the event-logs by the cases to receive the sequence of events for each patient. As output, they received a process map of each patient variant. The authors introduced a *global filter* and *local filter* to apply the Pareto principle, first filtering infrequent variants, and then filtering the infrequent activities. In a case study they proved that despite removing 5 % and retaining 95 % of the variants, a good overall precision can be maintained. The author compared different gradations of filtering to determine how they affect the meaningfulness of data. The authors in [77] used the received knowledge to model material flows of the ED and used the cases from process mining as the patient pool. In a simulation they executed what-if scenarios with alternating patient mixes to evaluate stretcher allocation policies. As input for the simulation, the authors applied a global filter that took 90 % of the cases into consideration for simulation modelling. However, the reason for applying the filter and removing 10 % of the case variants is unclear and can bias the simulation results.

Liu et al. [29], in the area of banking and insurance, modelled and simulated the credit card application process. To receive data for modelling, the authors extracted workflow data directly from an ERP system to receive an event-log with timestamps for begin and end of activities by process-mining methods. They highlighted that the data can be retrieved by each workflow-oriented information system. The authors used the event-log to extract the historical processes by coupling pairs of events for the begin and end of the activity and determined the duration of the activity. From the historical processes, the authors derived a process map describing the entire credit card application process. Other than in manufacturing systems and EDs, the extraction of data and simulation focuses on a specific process. The competition of different processes for resources and waiting lines are not at the core of the simulation study because activities are digital and not a bottleneck. The authors built a simulation model based on the process map to analyse the process duration, throughput and acceptance rate of the process.

Wang et al. [77], [78] analogously presented two case studies of providing material flows from workflow data in the banking and insurance industry. However, other than previous works in manufacturing and emergency systems, in banking and insurance there is no physical material or patient flow. The 'manufacturing' process in banking and insurance is digitalized and, by nature, data is available or at least considered available. For simulation, the authors extracted event-logs and retrieved a process map from company databases. The authors built a model in DES to evaluate the received data. For future research, the authors proposed to couple data mining, modelling and simulation activities, and provide usable tools that are capable of applying these activities as a service.

The state-of-the-art created new insights regarding available methods and tools for the extraction of system and flow data. For the system data describing the physical configuration of the shop floor, there are many standards available to structure and provide simulation data from information systems. The methods for providing system data are considered to be available. A different situation is given for flow data, tackling mainly product arrivals and processes as well as uncertainties. The literature provides a range of methods to extract simulation data (see Table 4). Sources are generic event-logs for providing material flows and specific data sources, e.g. measurements from the shop floor for providing uncertainties of process durations. Using generic event-logs provides a wide range of information and enables the description of product arrivals and processes.

Table 4: Comparison of literature

Publication metadata		***Data retrieval***		***Knowledge extraction***		
Author	***Area***	***Extract method***	***Data***	***Product Arrivals***	***Process***	***Resource***
Goodall [64], [65]	Ma	(2) RFID	Specific Data	-	Estimate Distributions	Estimate Distributions
Reinhard [24], [25]	Ma	(2) RFID	Event-log	-	Extract Event map	-
Van der Aalst [66]	Ma	-	Event-log	Map products	Extract Activity map	
Lugaresi and Matta [62], [67]	Ma	-	Event-log	Map products	Extract Activity map	-
Kursun-Bahadir [72]	Ma	-	Specific Data	-	Fit stochastic laws	-
Law and McComas [73]	Ma	-	Specific Data	-	Fit stochastic laws	-
Swanson [74]	Ma	-	Specific Data	Monte Carlo simulation	-	-
Kumbhar [75], [78]	Ma	(3) Data-base	Event-log	Extract IAT	Extract Activity map	-
Liu [29]	BI	(3) Data-base	Event-log	-	Extract Activity map	-
Wang [79], [80]	BI	-	Event-log	-	Extract Event map	-
Cheaitou [81]	ED	(3) Data-base	Specific Data	Estimate arrival rate	Estimate Distributions	-
Lay [76], [77]	ED	(3) Data-base	Event-log	-	Extract Event map	-

For providing simulation from event-logs with historical data, two major challenges are identified: the extraction of product arrivals, and the extraction of process maps. Additionally, for product arrivals and process maps, there are methods required to provide static data from the event-logs for deterministic simulations and to create randomized variants for stochastic simulations. Creation of static product arrivals and process maps as well as random variants are addressed in the next section.

2-2 Methods for retrieval of simulation data

The purpose of this section is to present the generic methods to extract product arrivals and process maps from event-logs with historical data. The methods are derived from the data-mining and simulation literature. Both are provided as historic and randomized logs. Historic logs describe the historic material flows that occurred on the shop floor and randomized variants describe stochastic variants. The steps to extract simulation data from event-logs are illustrated in Figure 17. The upper part shows the extraction of historic and randomized product arrivals and in the lower part shows the extraction of historic and randomized process maps. Section 0 describes the methods for receiving product arrivals and section 2-2.2 describes the methods for receiving the process maps. Both product arrivals and process maps are input for data-driven simulation.

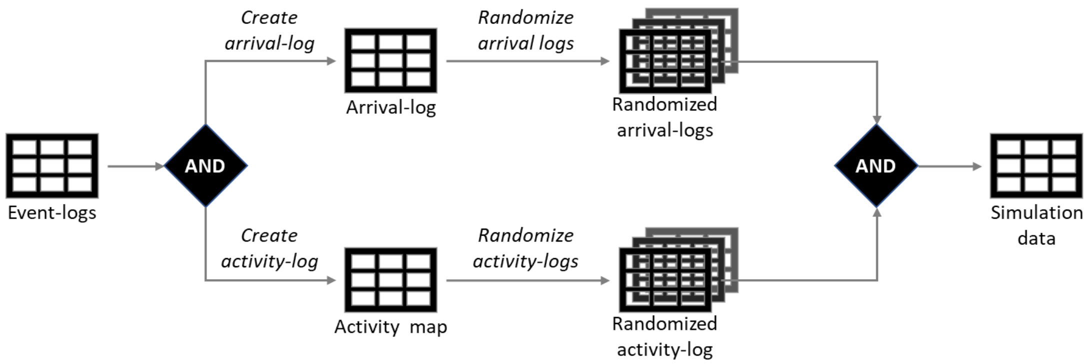


Figure 17: Methods of data-retrieval

The required data for the extraction of product arrivals and process maps are event-logs from the digital shadow. The general shape of the event-logs is displayed in Table 5. The property *event* is an incrementing number identifying each unique event of the event-log. In this work, event-logs are assumed to be available digitally and instantly (AS2). The attribute *case ID* assigns multiple events to a common case. When applying in the cases of the ED and train remanufacturing system, the neutral *case ID* can be, e.g. a patient ID or an order ID. The attribute *event-name* identifies the type of each event by referring to a finite list of predefined events. For each event, the *timestamp* gives the moment of occurrence. Additional properties can exist, for instance, the case-study of the ED the property *ESI* classifies the priority of the patients. The events of one *case ID*, sorted by the *timestamp* describe the sequence of events in the historic material flow.

Table 5: Event log in the emergency department

event	*case ID*	*event-name*	*timestamp*
1	13101	Register Patient	01.01.2021 00:23
2	14609	Register Patient	01.01.2021 00:28
3	13101	Start Triage	01.01.2021 00:29
4	13101	Stop Triage	01.01.2021 00:35
5	14609	Start Triage	01.01.2021 00:42
6	14609	Stop Triage	01.01.2021 00:44
7	14609	RX Realization	01.01.2021 00:48
8	14609	See Doctor	01.01.2021 01:08
9	22939	Register Patient	01.01.2021 01:09
10	22939	Start Triage	01.01.2021 01:12
11	22939	Stop Triage	01.01.2021 01:23

The event-log lists those events that occurred in the physical system in a table. Data cleaning is required to ensure data quality and can include multiple steps. Cleaning is particularly required when dealing with manually generated data. The process can inherit activities to correct spelling errors, missing data and duplicates. Filtering removes data that is not within the scope of the study, e.g. events outside the period of study. Aggregation allows grouping the events of a particular case and providing the routing of events for each case. Detection and removal of outliers removes cases that are not representative for the analysis and would distort the results. Filtering can be applied to use the Pareto principle and remove cases based on the frequency of their routing. The event-log is the assumed to be input in the subsequent chapters.

2-2.1 Extract product arrivals

The arrival stream describes the arrival of cases on the shop floor. Modelling of the arrival stream in the simulation requires as input for arrival the type of material and the time between arrival of two consecutive arrivals, i.e. the inter-arrival time (IAT). This information is available in the historic event-log and can be received through analysis of the timestamps (TS) for arrival of cases. Figure 18 illustrates how the event-logs describe historic arrival of cases and how the arrivals are linked to the IAT based on the events of Table 5.

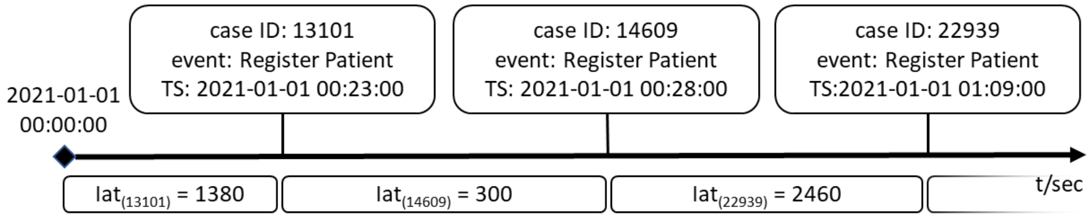


Figure 18: Arrival stream of patient in emergency department

Extracting product arrivals has the purpose of providing the historic arrivals from the event-logs as simulation input. The input is the event-log and the output is product arrivals, describing the arrival stream. There are three steps required to extract the product arrivals.

1. **Copy event-log**: Create a copy of the event-log as a draft for the creation of the product arrivals and to retain the initial event-log for further analysis.
2. **Filter arrivals**: The event-log contains all historic events (Register Patient, See Nurse, …). Filtering removes non-arrival events from event-log. Cases are filtered by the event-name; for the case of the ED, 'Register Patient'. The output is a log stating all arriving cases.
3. **Calculate inter-arrival time**: The inter-arrival time of each case is measured from the previous case. For the first case, the inter-arrival time measures the time from the beginning of the observation period.

The output is the historic arrival-stream, as illustrated in Table 6 for the case of the ED. The properties case ID, timestamp and inter-arrival time describe the sequence and moments of arrivals. The historic arrival stream directly provides knowledge or serves as input for data-driven simulation.

Table 6: Log with case arrivals

case ID	*timestamp*	*Inter-arrival time [sec]*
13101	2021-01-01 00:23:00	1380
14609	2021-01-01 00:28:00	300
22939	2021-01-01 01:09:00	2460

Direct analysis of the arrival stream for the case of the ED can, e.g. provide the arrivals of the patients during the hours of the day (see Figure 19). The figure provides a static view of the historic arrivals, highlights that arrivals are not continuous and underlies uncertainties. Different alternative arrival streams could provide the same distribution of arrivals during the hours of the day and could be input for simulation. Simulation of stochastic variants is required to understand the input of uncertainties. Because uncertainties do not include just the inter-arrival time, but also the sequence of cases, new randomized logs with arrival streams are to be created to enable stochastic simulation experiments.

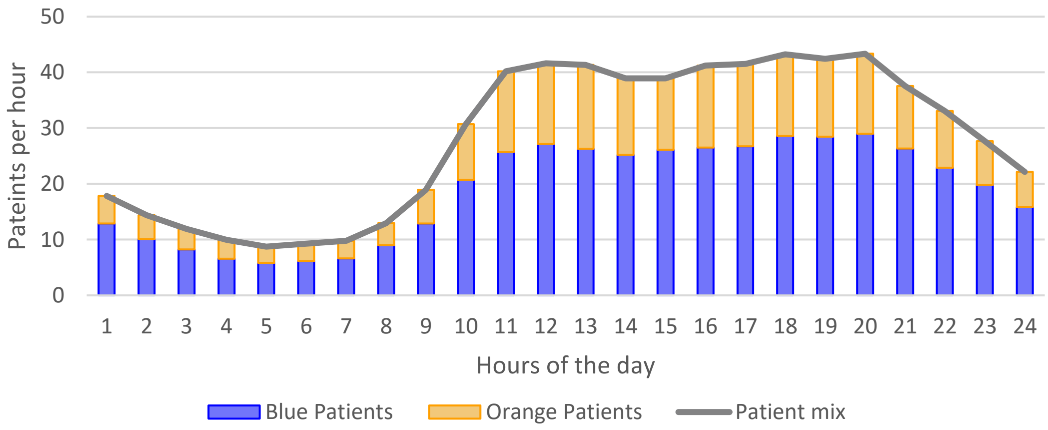


Figure 19: Hourly arrival stream for historic patient arrivals

Monte Carlo simulation is proposed to randomize arrivals logs and provide stochastic variants. In Monte Carlo simulation, an algorithm pulls random samples from datasets. Grouping the arrivals by the hours of the day, and potentially days of the week, provides datasets with the arrivals for periods of time. For the creation of a new arrival stream, an algorithm simulates the timeline and pulls random samples from the datasets according to the simulation time. In the simulation, the seed value is the origin for creation of stochastic variants. Changing the seed generates alternative random behaviour and is the origin for different stochastic variants. For the generation of the arrival streams, a three-step approach is proposed.

1. **Copy product arrivals as a template**: Create a copy of the historic arrival stream and remove all case arrivals to receive an empty template. Reuse the template to generate alternative arrival streams.

2. **Copy product arrivals as an arrival pool**: Create a copy of the product arrivals as an arrival pool that lists all historic arrivals. Group the arrival pool by the hours of the day. The grouped arrival pool is the source for pulling samples. An exemplary arrival pool based on Table 6 is illustrated in Table 7.

3. **Simulate arrivals**: Use Monte Carlo simulation to generate random arrival streams. Build an algorithm to simulate the arrival of patients during the observation time. Pull random samples from the classified arrival pool based on the time of simulation. Write the arrivals for each simulation run in a new template to create stochastic variants.

The procedure creates new logs with case arrivals. Each log has different sequences and inter-arrival times based on stochastic behaviour. In the case of the ED, random laws can increase and decrease the number of patients with blue and orange severity indices during sampling. Impacting the sampling process can provide data for simulation of scenarios. The log with the generated arrival stream has the same structure as the initial log, but describes stochastic variants of case arrivals.

Table 7: Arrival pool

arrival class	***case ID***	***timestamp***	***Inter-arrival time [sec]***
00:00:00 – 00:59:50	13101	2021-01-01 00:23:00	1380
	14609	2021-01-01 00:28:00	300
	…	…	…
01:00:00 – 01:59:50	22939	2021-01-01 01:09:00	2460
	…	…	…

2-2.2 Extract process maps

Material-flow simulation is activity-centric and simulates activities occurring over a period of time. The historical event-log is event-centric and describes the moments of occurrence. To bridge the gap between events and activities, transformation is required and provides a log with the historic activities. Each activity is defined through two events for begin and end. The time between begin and end defines the duration. Figure 20 illustrates the relationship between events and activities in the lifecycle of the exemplary case 13101 of Table 5. Time in the case that is not explicitly assigned to a specific activity is waiting time. Waiting time is caused by missing resources.

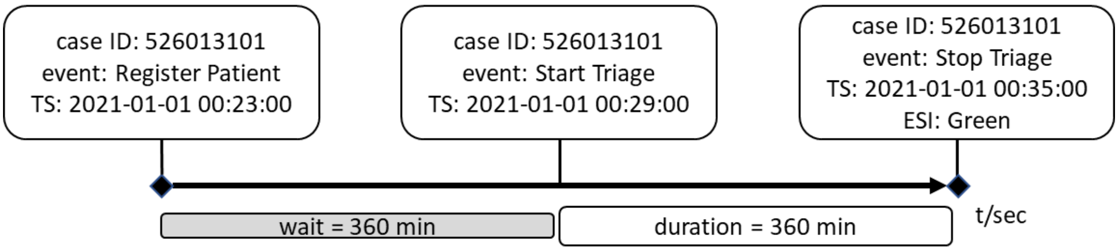


Figure 20: Event-centric view vs. activity-centric view

Extracting activities from event-log requires to map pairs of corresponding events for begin and end of activities. Time that is not assigned to an activity is waiting time. Waiting time is to be removed from the data and does not go explicitly as activity into the process map. Input for extraction of activities is the event-log. An algorithm can map events and generate activities. Output is the sequence of historic activities. For extraction a four-stepped approach is proposed:

1. **Copy event-log**: Create a copy of the event-log and retain the initial event-log for further analysis.
2. **Group by case**: Grouping of the historic events by case provides datasets with events for each case. This step is required to avoid pairing events of different cases.
3. **Create activities**: Map corresponding events for each case to define activities. Calculate the time between the beginning and end of event-pairs to obtain the duration.
4. **Determine process map**: Sort the activities of each case by the timestamp and provide for each event the successor as property by referencing the successor with the name.

An exemplary process map based on the example of Table 5 is illustrated in Table 8. The process map describes the activities and their sequences for all cases. Each activity is described by the attributes *case ID*, *activity-name*, *successor-name*, *duration* and *resource*. The attributes describe details of the historic material flow. The successor-name links activities and provides a routing for each *case ID*. The attributes case-ID and resource link the activity with the physical materials and resources of the production system.

Table 8: Output of create activities method

case ID	*activity-name*	*successor-name*	*duration [min]*	*resource*
13101	Registration	Triage	0	-
13101	Triage	-	360	Nurse
14609	Registration	Triage	0	-
14609	Triage	RX	120	Nurse
14609	RX	See Doctor	300	RX
14609	See Doctor	-	600	Doctor
22939	Registration	2 Triage	0	-
22939	Triage	-	660	Nurse

Visualization of the process map in a graph provides the structure illustrated in Figure 21. The figure illustrates the cases of Table 8. The activity map shows the activities that occurred during the observation period and their successor relations. The weight of the transitions denotes the number of patients that are described by the corresponding successor relationship as absolute and relative values.

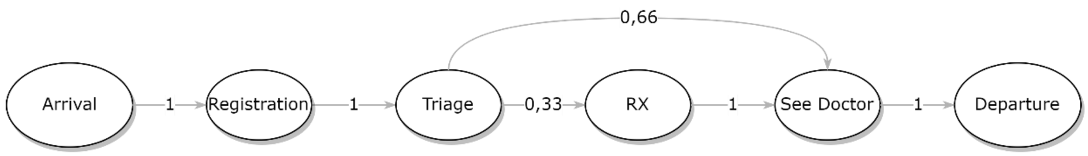


Figure 21: Activity map

Direct analysis of the process map for the case of the ED provides the durations of activities, e.g. the duration of the activity triage for orange and blue patients (see Figure 22). The figure shows the density for the appearance of values in the historic data. Simulation of stochastic variants is required to understand the input of uncertainties. To describe the uncertainty distribution, fitting can provide random laws by fitting a probability distribution to the historic data.

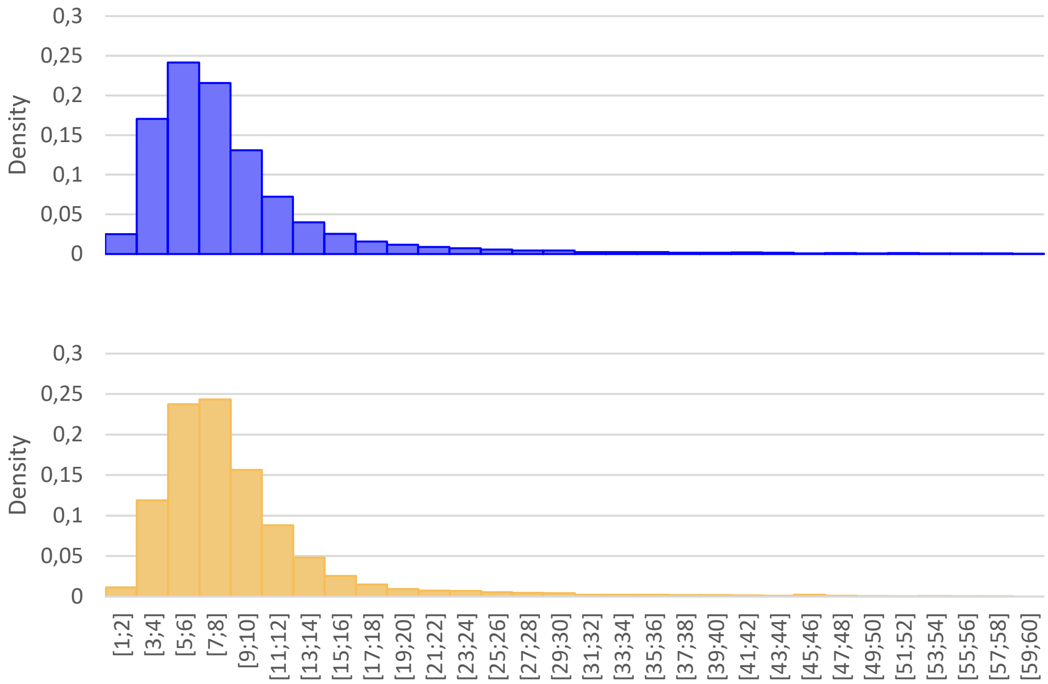


Figure 22: Probability distribution for triage based on severity index

In this work, distribution fitting is applied to fit stochastic laws to historic data and determine the parameters that enable the highest approximation. The inputs are the historic activities with the determined durations. The outputs of the fitting are the stochastic laws and parameters. To extract the laws, a four-step approach is used.

1. **Copy historic activities**: Create a copy of the dataset with the historic activities to provide analysis for each activity type.

2. **Group activities**: Group the list of activities by the activity-name. Grouping by additional attributes can provide more precise distributions, e.g. providing distributions individually for different severity indices in the ED-case.

3. **Create histogram and probability distribution**: Create a class diagram with the frequency of durations for each grouped element, fit a set of probability distributions to receive the best fit and provide parameters of the best fit. This work uses the Python library fitter package for distribution fitting [82].

An exemplary output of the distribution fitting is displayed in Table 9. The table shows the distribution and parameters for each activity. The property *distribution* states the type for each activity. Additional properties specify the parameters. The probability distributions generate random values or the activity duration in the simulation.

Table 9: Output of method: fit distributions

Activity name	*distribution*	*mean*	*standard deviation*	*minimum*	*maximum*
Triage	Erlang	9,22	5,83	1	103
Blood Analysis	Erlang	39,37	16,02	1	119

2-3 Case-Study: Emergency department

A case-study was performed in the ED. System data was received from the information system. The system data contained information about 64 technical and human resources and was provided to the simulation. For the retrieval of flow data, the presented methods for data-retrieval were applied. The IT department of the hospital provided the event-logs for the year 2021. The event-logs contained 297463 events for 36857 patients. A global filter was set to 95 % and removed 5 % of the patients with less frequent variants. The filter was needed because before the simulation, the validity each variant had to be checked manually. Focusing on 95 % of the patients reduced the number of variants from 968 to 103 and enabled reliably checking for invalid variants.

The event-logs provided for each event the *case ID*, the *event-name*, the *timestamp* and the additional property ESI. The generalized case ID was used in the case-study to represent the patient ID. The ESI describes the severity index of the patients in the initial (green, orange, red) and planned future system (orange, blue). The ESI provides additional case-specific, non-general information of the ED. In addition, expert interviews were conducted to receive information that are not available from the data, e.g. the operational rules on the shop floor, particularly for assignment, prioritization and displacement of patients. Assignment describes the rules of assigning patients to zones based on the severity level. Prioritization describes the priority rules of the laboratories. Displacement is linked to moving the stretchers with patients from the zones to laboratories.

The presented methods were applied to extract patient arrivals and pathways from the event-logs with historical data. For providing the historic patient arrivals, the events were filtered for the arrival event and the inter-arrival times were calculated. Datasets of patient arrivals for each hour of the week were developed to perform Monte Carlo simulation and generate randomized arrival streams. To provide the patient pathways, the historic activities were extracted from event-log, and durations were calculated. Owing to missing events for the start and stop of some activities, a dictionary was provided to generate activities from a single event and a duration, which was received during an interview with an expert. From the historic activities, a process map was derived. Figure 23 illustrates the activity maps as a first-order Markov chain. The weights of the transitions define the probabilities for patients to follow the pathway. Random laws for activity durations were developed by fitting stochastic distributions to provide uncertainty in the simulation.

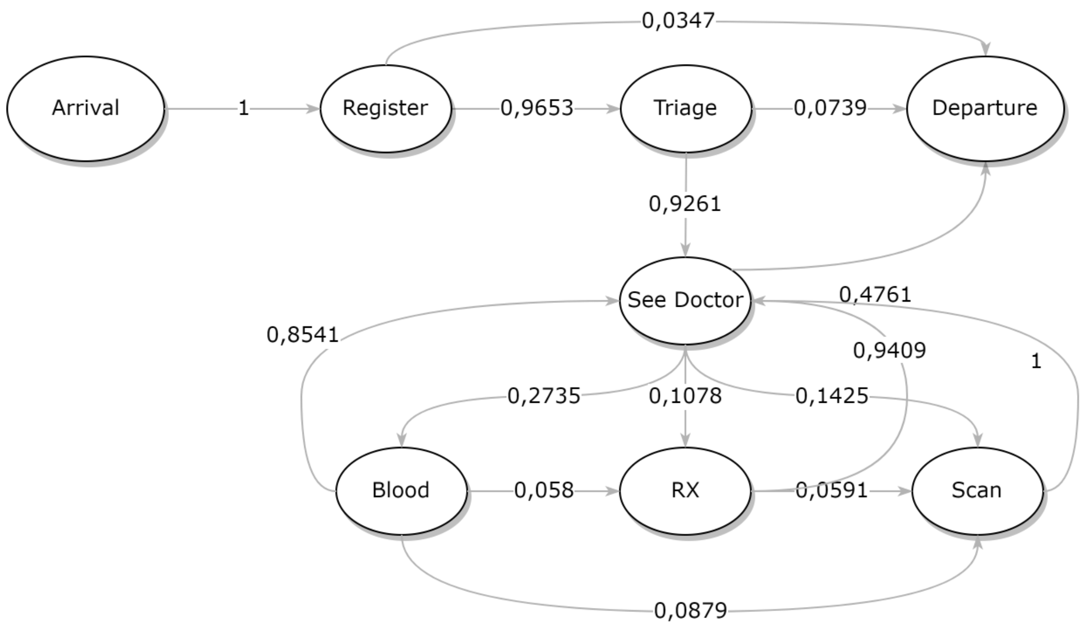


Figure 23: Markov chain with routing variants

In addition to providing simulation data as the input for data-driven simulation, the arrival and process maps enable static analysis of the historic material flows. In the design for the two ED situations, initial and future situations are to be compared according to the case description in section 1-5.1. A static analysis based on the historic arrival stream provides the number of patients arriving in each zone in the initial (green, orange, red) and future situation (blue, orange) (see Figure 24). The figure shows that the transition from the initial to the future situation creates more patients with blue than orange severity class (23598 vs. 11642). Without considering the workload per patient, the distribution indicates a lack of capacity for the patients in the blue zone.

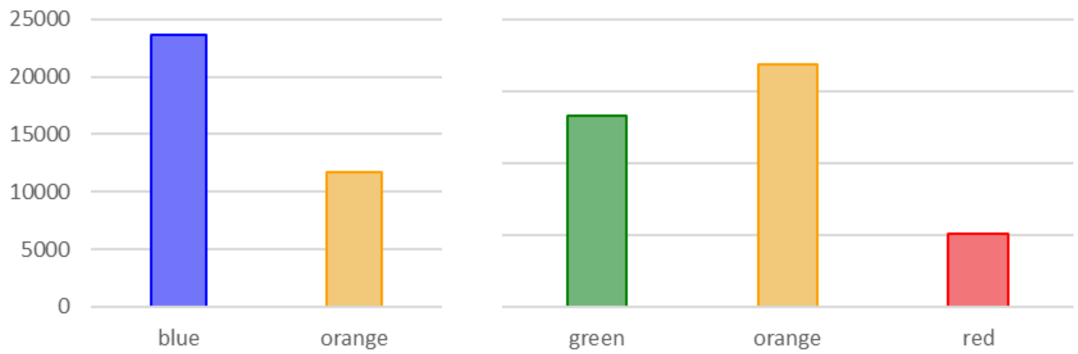


Figure 24: Patients per triage rule categories

To confirm the hypothesis, Figure 25 provides the static workload in the orange and blue zone based on the historic arrival stream and activities. The figure shows the average transit time per patient in minutes (left axis) and the cumulated workload in hours (right axis), depending on the number of loops (See Doctor) the patients complete during their stay. In the diagram, the *x*-axis describes the number of loops and 0 represents the triage.

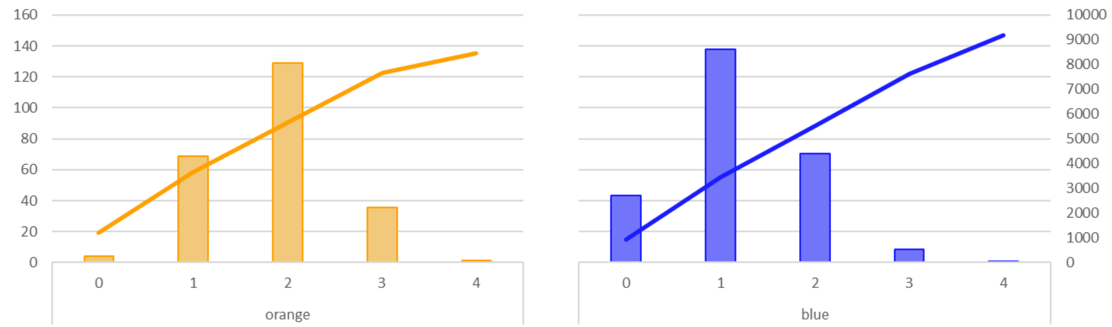


Figure 25: Workload and transit time for patients per triage rule categories

As seen in the figure, the workload in the orange zone is approximately equal to the workload of the blue patients (14920 vs. 16274 h). The workload is mainly generated by patients during triage in the first two loops for blue patients and in the first three loops for orange patients. However, for designing the workload, the colour provides an indication about an appropriate assignment of stretchers, doctors, nurses and boxes to zones. The planned configuration with 20 orange stretchers, 4 orange doctors, 3 orange nurses and 10 orange boxes can cause problems owing to a lack of capacity in the blue zone. In particular, when considering the distribution of arrivals in Figure 19, demand causes peak capacity between 11:00 am and 20:00 pm.

2-4 Discussion

The leading question of this chapter is RQ1: **What are the methods used to generate simulation models for problem-solving from the digital shadow?** This question covers the retrieval of simulation data and the generation of a model. Generating models are addressed in Chapter 3. This chapter addresses the simulation data. The event-logs with historical data are the inputs for retrieval of simulation data. Based on the event-log, this thesis uses multiple methods to extract product arrivals and process maps. Owing to the different requirements of simulation for both, deterministic historic variants and multiple stochastic virtual variants required. Randomization of the arrival streams and process maps generates random scenarios for simulation studies. The purpose is to randomize parameters of the model, such as inter-arrival times, routing sequences and activity durations.

Extract product arrivals has the purpose of providing schedules with historical material arrivals from event-logs. Extraction of the arrival stream requires arrival events from historical event-logs and the determination of the inter-arrival times for the arrivals. For the case of the ED, the extracted arrival stream describes the historic sequence of patients with their ESI and routing variants as well as the time between arrival of materials. This input is sufficient to determine the arrival stream for deterministic simulation. Problems in the case study arise from manual data inputs and cause problems with the data quality. Corrupted data causes patient arrivals that do not exist with the recorded routing in the history. In this case, the number of valid and invalid variants is increased. Because a validation check for all variants is required before simulation, a Pareto filter is introduced and removes 5 % of the patients without regularly presented variants. However, this can cause biases in data analysis and simulation. To fully exploit the event-logs and extract patient arrivals, valid data is required. When measuring events on the shop floor, measurements should be recorded automatically to ensure the correct timestamps and event descriptions.

Extract process map has the purpose of providing routing of arriving products or patients. Extracting the activities of the process map requires the events to be transformed into activities by mapping pairs of corresponding events for the beginning and end of an activity. Calculating the time between these events provides the duration of the corresponding activity. For the case of the ED, the extracted activities describe the pathways of the historic patient arrivals. Problems were experienced in the case-study, particularly regarding the data structure. The event-logs did not contain information about the affected objects for multiple parts of the same product. Consequently, when dealing with different parts associated to the same product, the sequence of events for the different parts was not clear from interpreting the event-logs. To facilitate retrieval of events of nonlinear routings, it is recommended to record object-centric event-logs. Without information of object-centric event-logs, additional effort is required to map the information about assembly and dismantling processes.

Randomizing patient arrivals: Uncertainty is introduced via the arrival streams by pulling random samples for patient arrivals. Random samples determine the inter-arrival time, routing and severity index. For each of the arrival patients, the routing is deterministic in the simulation algorithm once the patient arrives. For randomization of arrivals, this work recommends building datasets containing the product arrivals at each hour of the day, and to use Monte Carlo simulation to pull random samples according to the hour of the day and week in the simulation. However, the dependence of patient arrivals from the hour and day of the week is the specific behaviour of EDs.

An alternative approach, which is provided in several studies, is to separate **randomization of inter-arrival time and routings**. In some studies, the randomization of routings is indirectly performed owing to an analysis of the event logs, and provides a first-order Markov chain model of the routines (i.e. when an activity is finished, the next activity to be performed is defined by a probability). This approach is easy to implement, but requires changes to the data-driven model we are proposing. It does not, however, dispense with the initial data analysis described above. However, the interest in using our proposed approach lies not only in the re-use of the data-driven model described above, but also in theoretical considerations about the limits of the first-order Markov chain model of routines. Indeed, we have good reason to believe that patient routines do not fit into a first-order Markov chain. A preliminary study carried out at the CSIP Laboratory seems to confirm this hypothesis from a statistical perspective. A first-order Markov chain model could create variants or even provide a routing with an infinite number of loops, which does not exist in reality. If this hypothesis is confirmed, then the Markov chain model approach would require the identification of higher-order chains, making the approach much more difficult to implement; multiple additional preliminary analyses of event-logs would be required. Our approach avoids these potential biases. We plan to use our model to carry out a comparative study of the two approaches, particularly in terms of measured performance (e.g. LOS).

Randomization of activities focuses particularly on the durations of the activities. For randomizing activities, this work recommends building datasets with the historic durations for each activity and to pull random samples. This approach requires the fitting of stochastic laws on historic data. Alternatively, distribution fitting can be used to pull random samples by using stochastic laws, analogous to the process of pulling random patient arrivals from the arrival pool with Monte Carlo simulation.

In addition to the presented branches of introducing uncertainty more sub-variants are possible. Randomization could be performed for the inter-arrival time in the arrivals and for the routings in the activity map. For pulling samples of activity duration, the use of stochastic laws and Monte Carlo simulation is appropriate. Independently from these strategies, there are different ways of implementation from a technical perspective. Activities of randomization require the methods of data science, which are not necessarily available in simulation tools. Thus, it was advantageous in the case study to determine stochastic variants entirely with the data science tools and simulate these stochastic variants in a deterministic simulation model.

Independently from extraction of the product arrivals and process maps, another challenge was identified in the case study. For modelling and simulation system data, data for system behaviour was required. As stated in the literature, methods and standards for retrieval of system data are available. A different situation is encountered for the extraction of system behaviour. There is lack of methods to extract rules and behaviours. In the case studies, control logic was obtained from interviews and was reformulated using modelling languages such as BPMN. More standardized methods could allow us to extract control logic automatically, analogous to the spreadsheet extraction of the system configuration and event-logs with historical material flows.

Chapter 3 Data-driven simulation

This chapter provides further answers to the question of the generation of models for material flow simulation from data (RQ1): **Which are the methods to generate simulation models for problem-solving from digital shadow?** Particularly, this chapter tackles the data-driven modeling from the arrival stream and process maps, extracted in the previous chapter, see Figure 26. To address the question this chapter first provides a state of the art in enterprise modeling (Chapter 0), second provides the methods for data-driven simulation (Chapter 3-2), third applies the methods in the two case-studies (Chapter 0) and fourth provides a discussion based on the retrieved knowledge from the case-studies (Chapter 0).

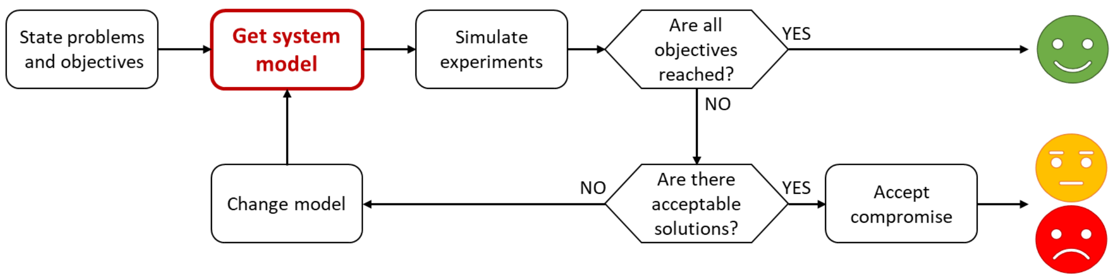


Figure 26: Automated modelling and simulation in problem-solving

The state-of-the-art provides approaches for the automated generation of material-flow simulation models and shows how models can be set up without expert knowledge. The methods, which are the contribution of this work, provide valid simulation models instantly from the data using a library of generic models and data-driven simulation. These methods enable the use of the digital shadow and simulation for optimization and problem-solving. The case studies present two applications of data-driven modelling, first in the ED and second in the train remanufacturing system. The two case-studies highlight how the generic methods can instantly provide models of material flows for systems from different domains and generate new knowledge for problem-solving that was not previously available.

3-1 State of the art

Simulation of material flows has the purpose of evaluating existing and planned systems by experimentation in dynamic models. When evaluating scenarios, decision variables to be tested are implemented in the model, and simulation provides measures of the performance. Performance measures can be interpreted as values for an objective function in optimization. To evaluate systems by simulation, a model of the material flows is required. Modelling and simulation have high requirements to the system modeler in the dimensions of time, knowledge and effort [17]. To facilitate simulation, the literature offers processes for execution of simulation studies and describes tasks and objectives in each phase [83]–[87]. Another approach is automated modelling and simulation. Assuming simulation data as available, techniques of data-driven simulation enable simulation without expert knowledge by using algorithms to build models from data and run simulation experiments [88]–[91]. This section has the purpose of providing the methods of data-driven simulation from the literature. Therefore, the state of the art is provided in two parts: section 0 provides the standard for enterprise modelling [38], and section 3-1.2 provides the methods for data-driven simulation. The state of the art in data-driven simulation has its origin in a review paper that was published in context of this PhD thesis [20].

3-1.1 Enterprise modeling

To provide methods for data-driven modelling and simulation, the state-of-the-art begins with the enterprise modelling cube [38], which provides a framework for defining generic concepts for creation of enterprise models. The framework proposes three dimensions: the enterprise model phase, genericity, and the enterprise model view (see Figure 27). The enterprise model phase describes the life cycle phases of model development, the genericity level describes the degree of abstraction, and the enterprise model view describes the selective perception of particular aspects of the enterprise. This work focuses on the dimensions of genericity and enterprise model views.

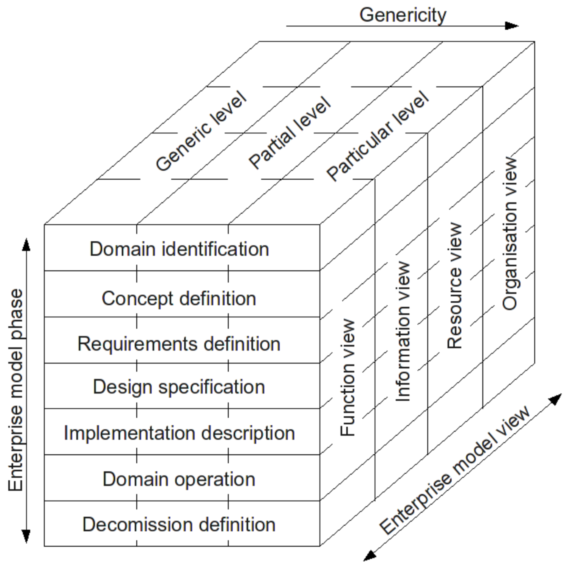


Figure 27: Enterprise modelling cube

Enterprise modelling phases describe the lifecycle phase of the model. During the model lifecycle, there are specific questions to be addressed. The lifecycle starts with defining the objectives, scope and limitations of the model. In the concept definition, the business concepts and operations are defined. The definition ends in the specification of the requirements for the model to be implemented. The modeler defines the model requirements, e.g. deterministic vs. stochastic model. In the design specification, the modeler defines the modelling concepts. Consequences of the requirements definition appear in the implementation description. Depending on the requirements of the model, individual data and concepts are required for implementation; in this work, deterministic and stochastic models. In the domain operation, the models are used for their actual purpose, e.g. scheduling or resource dimensioning. In the decommission phase, the system model seeks to enable reuse, recycling and conservation of the model. During the modelling phases, the modeler defines and details the modelling domain for the relevant genericity levels and modelling views explained in the upcoming sections.

The genericity dimensions describe through generalization and specialization mechanisms the transition from generic modelling constructs through reusable models, called partial models to, finally, particular models representing specific systems. The generic level is a set of given modelling constructs corresponding to a modelling language; the modeler chooses the language and/or technique and included constructs (in this work, DES). The choice of modelling language, techniques and model properties is elementary and provides the framework for the partial level. The partial level corresponds to specialized and aggregated industry and domain-specific models to represent common patterns. The modeler designs and reuses specific libraries with concepts, common to a set of enterprises or class of problems (in this work, job-shop manufacturing). The partial level facilitates the modelling when libraries are available because the modeler does not design the model from scratch. Available state-of-the-art works have been presented in this study. The presented works instantiate partial models to generate particular models of different enterprises. The particular level is the last-level genericity dimension. Partial models are used to generate a wide range of particular models of different enterprises to address specific questions (in this work, ED and train remanufacturing systems). The analysed literature identified two steps of setting up simulation models in data-driven approaches (see Figure 28). First, design of a library that provides partial models, which is generally implicit in the data-driven modelling literature. Second, data-driven modelling instantiates partial models to generate particular models. In the instantiation process, an algorithm chooses specific elements from the library and parameterizes them with data from the case-specific database. Design of libraries and data-driven modelling support the transition from generic to partial and from partial to a particular model.

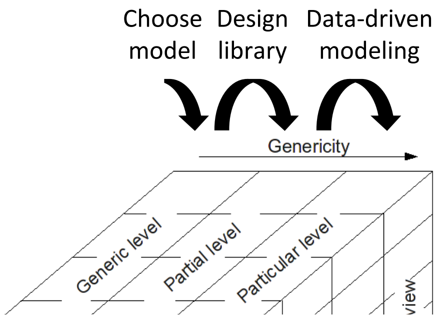


Figure 28: Transition between generic, partial and particular level

Enterprise modelling views describe the modelled elements that interact to describe the holistic behaviour of the system. The function view describes the activities of the production process. The information view describes the information of the production flows, which includes the representation of materials. The resource view describes the human and technical resources of the system. These three views are constant with the standards of the digital factory, e.g. the PPR model [54]. Challenges in data-driven modelling exist, especially in the organization view. The organization view describes responsibilities and organization of work. For standardized elements, generic partial models describe the general behaviour for data-driven modelling. For case-specific control logic, the literature uses modification to provide case-specific classes. The complex behaviour of the organization view is a bottleneck in data-driven modelling.

3-1.2 Automated modelling and simulation

Automated modelling and simulation deals with setting up simulation models and running experiments from data. This review has the purpose of understanding the strategies and methods that are available to automatically generate simulation models and execute experiments. For analysis of the literature, the classification framework in Table 10 is presented and has its origin in the genericity levels of enterprise modelling. During the review, all the approaches were classified according to the criteria of this framework. The *generic model* criterion describes the modelling language and technique of the approach. This review mainly considers works in DES modelling. DES models are developed with either process-oriented modelling (POM) or resource-oriented modelling (ROM) [92]. The *partial model* describes the library of the partial level. The authors provide libraries for different classes of problems and types of systems: job-shop, manufacturing line, assembly line. Additionally, there are case-specific libraries. For job shops, there is a distinction between job shops with linear flows, and job shops that can describe assembly and dismantling operations. The *particular model* criterion describes how the instantiation of particular models is performed. The available works generate models by data-driven modelling from data and libraries, parameterization of generic models or manual interactions. In particular, data-driven modelling customization allows the description of case-specific behaviour.

Table 10: Classification framework

<table>
<tr><th></th><th colspan="6">Solution Space</th></tr>
<tr><td rowspan="2">Generic model</td><td colspan="3">Process-oriented modelling (POM)</td><td colspan="3">Resource-oriented modelling (ROM)</td></tr>
<tr><td colspan="3">Resource-oriented modelling with logistics (ROM+L)</td><td colspan="3">Non-DES modeling (non-DES)</td></tr>
<tr><td rowspan="2">Partial model</td><td colspan="2">Job shop (with linear flows)</td><td colspan="2">Manufacturing Line</td><td colspan="2">Case-specific library</td></tr>
<tr><td colspan="2">Job-shop (with Assembly)</td><td colspan="2">Assembly Line</td><td colspan="2"></td></tr>
<tr><td rowspan="2">Particular model</td><td colspan="3">By data and library</td><td colspan="3">Parameterization of generic model</td></tr>
<tr><td colspan="3">By data and (customized) library</td><td colspan="3">Manual interaction</td></tr>
</table>

Hubl et al. addressed automated simulation modelling using a data-driven approach based on the bill of materials and ERP data [93]. This approach describes production processes and considers resources as machine groups with their associated capacities, as well as their control logic. The layout and logistics of the shop floor are not within the scope of this approach. By considering the bill of materials from ERP, the authors were able to simulate the assembly of parts, stocks and material supply. For validation, the authors provided an industrial case in an automotive plant and evaluated the performance of alternative scenarios within kanban and MRPII decisions [94]. The models were reused in multiple studies to support a wide range of decisions, such as capacity and material requirements [95], stochastic uncertainties in demand and customer orders [96], lot size and lead time [97], make-or-buy [98] and decision-making in workforce training [99].

Charpentier et al. referred to the idea of reusing and reducing simulation models [100]. The authors aimed to reduce the complexity of routing and describe multiple and similar work centres as aggregated blocks with given capacities. Similar to Hubl et al., the authors used the master production schedule (MPS) from ERP to provide simulation models of the production process. To implement decision-making in the simulation model, the authors reused predefined decision entities parameterized with the MPS data [101]. They applied their simulation approach in an industrial case of a job-shop system and provided schedules of manufacturing facilities.

Goodall et al. provided an approach for data-driven model generation for remanufacturing systems [64]. The challenge in their work was to overcome the complexity of the remanufacturing process, which is marked by dismantling, processing and reassembly. To overcome this issue, they designed reusable modules for each process type. To generate simulation models, they reused the predefined modules and generated a model of the process using a data-driven approach. The approach was validated in an industrial remanufacturing system for electrical equipment.

Schlecht et al. automated modelling and simulation for the scheduling of product arrivals in an industrial remanufacturing system [102]. The authors used a generic data-driven approach to generate process-oriented simulation models. POM allowed them to provide simulation models for complex remanufacturing processes. This approach can analyse the consequences of order scheduling under the dependency of task prioritization on the shop floor. Data-driven modelling enabled the authors to automatically generate simulation models, execute a set of simulation experiments, alternate systematic experimental parameters, and provide data for decision-making.

Arons and Boer presented an approach for model generation based on parametrization [103]. The authors provided generic simulation models along with simulation parameters in a database. The database model is described in [104], and the model generation is described in [105]. Users can parameterize the database. In the model generation phase, an algorithm pulls the generic model and parameters from the database and generates a simulation model by parametrization. The purpose of this approach is to provide simulation models to non-experts. Simulation models are simple, first-in first-out (FIFO)-managed job-shop systems with unspecified buffer dimensions. The modelling of complex control logic and logistics was not within the scope of their study.

Son and Wysk presented an approach for model generation for discrete manufacturing systems, especially job-shop systems [41]. They provided a data model and implemented it in a Microsoft Access database. The model describes experiments, products, processes, shop floor, and orders. In [42], the authors connected the model to the shop floor to receive real-time status updates from the shop floor and manage the shop floor. To reduce the complexity and overcome the inherent risk of deadlocks, the authors assumed the material flow to be linear. Products in the linear material flow do not pass the same machine more than once; consequently, the linear material flow cannot describe loops and feedback. In [43], a generic database was used to generate models for different simulation environments. Because the results in different environments coincide, the authors highlight the added value of data-driven online simulation.

In [106], Barlas and Heavey focused on online simulation and provide an online platform for simulation called DREAM. The project addresses data acquisition from corporate business systems, management of simulation data, and automated generation of the simulation model. In [107], data acquisition from corporate systems and the management of simulation data using the standard CMSD were addressed. In [108], the authors presented an approach to manage simulation data with the data exchange standard CMSD. Data-driven model generation was described in [109]. The project was dedicated to job-shop systems. The models are point-based and do not include logistics. The authors used DREAM to analyse the impact of default style prioritization rules on machining. The objective of this study was to provide a free-to-use online simulation tool. Industrial validation was not presented in the study.

Bergmann [110] presented a tool for automated model generation. As a source of simulation data, the authors focused on corporate business systems and exchanged data with the modelling standard CMSD [53]. The purpose was to provide simulation models for evaluating the scenarios described in corporate business systems. The author focused on job-shop and flow-shop systems, including the dismantling and assembly processes. In the case of a flow shop, unlike a job shop, the buffer capacities were not modelled. In [111], the author also addressed further questions regarding task prioritization, human resources, and machine groups. However, the authors listed complex buffer strategies and scheduling logic as future research problems. The validation of the provided tool was performed in a laboratory environment, and its application in an industrial case was not addressed. The logistic processes were not considered in this study. An application in a case study was presented in [112], in which the authors explicitly highlighted the advantages of data retrieval from ERP/MES and automated simulation modelling.

Zülch et al. presented an approach for automated model generation for job-shop production systems and implemented it in a usable tool called SIMULAST [113], [114]. The focus of the tool is on the simulation of the human workforce in manufacturing processes. In [115], the authors analysed the consequences of human errors on the operation time, product quality, and rework. They provide industrial case studies for a sheet-metal and pressing-plastic manufacturer. The simulation models were point-based. Logistics were modelled as ideal, and there was no delay in the travel of the workforce. Another study was based on the presented approach and analysed the influence of alternative personal structures on human errors [116]. The authors provided industrial case studies on sheet-metal and pressing-plastic manufacturing.

Tiacci et al. [117] presented an approach for model generation with a focus on resources and labour. The authors focused on linear assembly lines with ROM in a length-based system. This assumption reduces the complexity by excluding loops and feedback from the material flow. In this way, the authors removed the risk of deadlocks and were able to analyse the impact of buffers and blocking. Logistics were modelled in a length-based model with conveyors and labour transport. The authors used a parametric simulator and a genetic algorithm (GA) to improve the assembly lines in laboratory cases. The GA was used for the optimization of line balancing and buffer allocation [118], [119]. To model human processes, the authors varied the cycle times using statistical laws.

In [70], Lugaresi and Matta focused on the generation of models of manufacturing lines. The focus was on model generation and initialization from stream data. With stream data, the authors refer to data generated during the manufacturing process that are typically available in MESs. Unlike planning data, stream data contain timestamps for the start and stop of historical processes. For validation, they performed a case study using a LEGO demonstrator [71]. Using the generated data, the authors provided a digital shadow using DES. In [68], they implemented a bill of materials to generate the DES of an assembly line. These models are length-based resource models that describe the layout and material handling system with buffers and conveyors.

Wang et al. presented a data-driven modelling approach for manufacturing systems [120]. The authors designed a data model and built a generic database to manage manufacturing data. To generate simulation models, they populated the manufacturing data in the simulation environment. Simulation models describe production and logistics in a length-based model with a ROM view, including transportation by labour, conveyors, and forklifts. The assumption of linear systems enabled the authors to simplify the logistics system and model the system without using control logic. This approach was validated in an industrial case in an automotive manufacturing line.

Wy et al. presented another approach for data-driven modelling [19]. In addition to the input data, the authors used an AutoCAD model of the layout and flows and extended this with data on machine perturbation and workforce planning. The genericity of this approach is limited owing to the required availability of an AutoCAD model, which is typically manually modelled. The authors proposed reusing existing factory models to reduce time and effort. The approach was applied to compare different forms of organization for workshops and material handling in decision-making.

Popovics et al. focused on the generation of simulation models based on processing data from the MES and programmable logic controller (PLC) code [121]. The authors received product, process, and resource data, described in the ISA-95 modelling language from ERP and the behaviour of the system from PLC code. A parsing process was used to receive specific data from the PLC code and to describe the behaviour of the model. Using this data, they could model an industrial flow system with conveyors, buffers, and machines [50]. In contrast to previously analysed approaches, the PLC code allowed them to model complex behaviour. This approach was validated using an industrial conveyor system. In [122], the authors provided a database and a human interface to simplify the application and validate the approach for several test cases.

Ferrer et al. [123] focused on the mapping of product, process, and resource data through different domains during the engineering process of manufacturing systems. In [46], the authors focused on modelling complex manufacturing systems. They provided an ontology to exchange PPR data with AutomationML, from engineering to process simulation. The authors defined semantic rules and matched them to the instantiated objects to model functions. Ultimately, they used data-driven model building based on PPR data and semantic rules to generate complex simulation models [45]. Validation was performed at the engine assembly line of an automotive manufacturer with non-DES modelling.

Friedewald and Wagner presented a partially automated approach for scenario simulation [124]. In their proposed approach, experts design reusable modules and non-experts build a simulation model by reusing these elements. The reuse of predefined elements is performed manually. This procedure allows the integration of complex functions through the implementation of predefined modules. The authors validated their approach in an industrial manufacturing system and analysed capacity planning in scenarios [125], [126]. The purpose of their approach was to empower non-experts to build complex models in a short time rather than automate the generation of models for systematic experimentation.

Kallat et al. presented an approach for evaluating alternative scenarios in factory design using model generation [127]. The authors developed a library of reusable modules with individual behaviour and implemented control logic in these modules. A resource-oriented model was generated by reuse and parameterization. They presented a use case on an industrial manufacturing system. In another study, they applied their approach and generated a simulation model of a warehouse [128]. The authors used POM to simulate the packing process. The simulation of the layout and logistics processes in this model was beyond the scope of their study.

Butzer et al. provided a reusable simulation model for remanufacturing processes [129]. The model assumes a generic remanufacturing process comprising the following seven steps: inbound, disassembly, cleaning, checking and sorting, reconditioning, reassembly, and outbound. This approach aims to reuse the entire model, assuming that a new case is described by the same process. The model is reusable for remanufacturing systems described by the proposed generic process. This model is appropriate for investigating remanufacturing systems and their efficiency. It is not appropriate for analysing complex routing or remanufacturing processes. Validation was performed for a turbocharger remanufacturing system.

An overview and classification of the analysed studies based on the criteria defined in Table 10 is presented in Table 11. On a generic level, within modelling and simulation in DES, two major modelling techniques exist [20]. In POM, the authors model material flows by modelling processes that request resources during the simulation. In ROM, the authors model material flow by modelling the system. During simulation, parts move through the machines according to their processing routes. POM describes the process clearly and is widely applicable, although the size of models rapidly increases. ROM describes the physical structures and models the processes implicitly. In the review works, the replication of structure in ROM was enabled to model spatial organization and logistic processes. On the partial level, the works presented libraries for simulation of manufacturing systems. Libraries where identified for two types of manufacturing systems and two sub-types of material flows. Libraries of flow-shop systems describe manufacturing systems with linear routings, while libraries of flow-shops describe nonlinear routings with loops and feedback. Within both types, the consideration of assembly processes defines the sub-type. Material flows inherit split and merge activities for assembly and dismantling or are completely linear. In addition, case-specific libraries were used to represent specific behaviour. On a particular level, different strategies were used to instantiate models from generic libraries. Case-specific libraries were observed to be used as templates for manual modelling activities. Remaining libraries were instantiated from data. However, to adapt case-specific behaviour of systems, customization of libraries was observed.

Table 11: Overview of literature review

Author	Generic model - model type	Partial model – library type	Particular model – instantiation type
Hubl	POM in DES	Job-shop (with Assembly)	By data and (customized) library
Haouzi	POM in DES	Assembly Lines	By data and (customized) library
Goodall	POM in DES	Job-shop (with Assembly)	Instantiation from data
Schlecht	POM in DES	Job-shop (with Assembly)	By data and (customized) library
Bergmann	ROM in DES	Job-shop (with Assembly)	By data and library
Arons and Boer	ROM in DES	Job-shop (with linear flows)	By data and library
Son and Wysk	ROM in DES	Job-shop (with linear flows)	By data and library
Barlas and Heavey	ROM in DES	Job-shop (with Assembly)	By data and (customized) library
Zülch	ROM in DES	Job-shop (with linear flows)	By data and library
Tiacci	ROM in DES	Assembly Lines	By data and library
Lugaresi and Matta	ROM+L in DES	Assembly Lines	By data and library
Wang	ROM+L in DES	Assembly Lines	By data and library
Wy	ROM+L in DES	Assembly Lines	By data and library
Popovics	ROM+L in DES	Case-specific library	By data and (customized) library
Ferrer	Non-DES	Case-specific library	Manual interaction
Friedewald	ROM in DES	Case-specific library	Manual interaction
Kallat	ROM in DES	Case-specific library	Manual interaction
Butzer	POM in DES	Job-shop (with Assembly)	Parameterization of generic model

On a generic level, within DES modelling the review particularly exposed two modelling technique processes and ROM. The POM describes the manufacturing system by modelling processes. During the simulation, the manufacturing process requires resources. The resource travels to the process. The resource-oriented view describes the system by modelling the machines on the shop floor. During the simulation, the parts move through the machines according to their given processing routes. A detailed description of the benefits and drawbacks of the strategy are given in [92]. Within the resource-oriented works, some approaches additionally used techniques for modelling movement of materials in the layout. In the partial level, a range of libraries was provided for specific system types (job-shop, flow-shop and assembly lines) and specific assumptions (linear vs. nonlinear material flows). On the *particular level*, different strategies were used to instantiate models. Challenges are seen in providing models of specific cases from generic libraries, particularly when case-specific behaviour and control rules exist. To tackle these problems, customization of the partial models was used to provide case-specific libraries of partial models and instantiate models from these libraries.

3-2 Methods for modelling and simulation

The purpose of this section is to provide the methods for data-driven simulation, for which this work uses the generic, partial and particular levels of enterprise modelling. In Chapter 3-2.1, the generic level of simulation models is presented. The generic level defines the framework for the building of the partial library. On the partial level, Chapter 0 provides a library of modelling concepts for the different modelling aspects: processes, resources and material flow objects. On the particular level, in Chapter 0 this work provides the methods to generate models from data and libraries. Data-driven modelling is executed by an algorithm. After model-building, the algorithm manages simulation experiments and provides simulation results for optimization and problem-solving.

The validation of the proposed methods in section 0 uses the cases of the ED and train remanufacturing system. However, in this section, this work uses the neutral terminology of material flows to describe the generic methods. The flow of materials on the shop floor is appropriate for representing the processing of trains in the remanufacturing systems and movements of patients in the ED. The generic and partial models for both systems are identical. Customization of the partial models provides case-specific behaviour during the instantiation of the particular models.

3-2.1 Define generic model

Enterprise models in this work are modelled as material flows, which describe the movement of patients and trains in the ED and remanufacturing system. During the flow, the materials perform activities that are defined in the process map. For execution of activities, the materials can request resources. A lack of resources causes waiting lines and increases the time between arrival and departure of the material from system, that is, the LOS of the patient or makespan of the train. The systems are modelled as queuing networks to analyse the waiting line materials. To model flows, DES is used, particularly the plant simulation. A classification of models in this work is given in Table 12 based on [85], [130].

Table 12: Classification of models

Property	***Type***	
Simulation method	**Discrete-Event-Simulation**	
Temporal behaviour	static	**dynamic**
Time steps	**discrete**	continuous
State steps	**discrete**	continuous
Uncertainties	**deterministic**	**stochastic**
Termination	**terminating**	non-terminating
Model orientation	**process-oriented**	resource-oriented

This work uses models and simulation in DES. The dynamic nature of this technique enables analysing material flows over time and observing the effects of changes in real time. By considering the changes over time, one can examine the material flows under different operating conditions and identify critical bottlenecks or inefficient areas. In addition, the dynamics of the simulation allow different scenarios to be tested and alternative solutions for optimizing the material flows to be evaluated. The consequence of the model being static rather than dynamic is a limited ability to capture changes in material flows over time.

The proposed simulation models use discrete time and state steps to consider material flows at regular intervals. Using discrete time steps and states allows the analysis of the flow in clearly defined conditions and for changes in the system to be recorded. The occurrence of events and durations of activities is limited to discrete moments and periods. Both events and processes are simulated through the discrete representation of time and states. Conversely, in continuous simulation, states and steps change continuously, causing an increase in the complexity of modelling and computing power for simulation. Simulation of discrete time and states is sufficiently accurate for analysis of material flows.

The simulation models consider both deterministic and stochastic uncertainties in the material flow, depending on the simulation data provided. Deterministic uncertainties are modelled through precise knowledge of the parameters and conditions in the system. They make it possible to analyse the material flow under idealized conditions and to identify the potential for optimization. Moreover, stochastic uncertainties are modelled by random variations in the parameters. This makes it possible to examine the effects of uncertainties and fluctuations in real operation and to assess the robustness of the material flow processes and address specific questions in problem-solving.

An important aspect of the simulation models is the termination. This means that the simulation returns a final result after a set number of steps or when certain conditions are met. The terminating nature of the simulations indicates that the material flow can be analysed in a well-defined time frame to provide quantifiable results. Termination makes it possible to reach a defined end state at which simulation results are provided and allows simulation to be used as a black box for optimization. Simulation results are generated during simulation. To generate simulation results, an empty event-log is provided. Simulation writes events for the start and stop of activities in the event-log and generates an output having the same structure as the input of the data-retrieval process. The event-log enables the calculation of all relevant key performance indicators (KPIs) for optimization and problem-solving.

The modelling orientation is defined as POM. The simulation model describes a network of processes for each product by providing complex predecessor and successor relationships. In simulations, the material is represented by an information object that follows the process flow. For execution of activities, the processes request resources. Available resources travel instantly to the process and are released after processing. Traveling and material movements are modelled explicitly by use of processes. Waiting lines occur virtually when multiple processes request limited resources. Each material of information object waits in its specific process, since spatial limitations of the layout are not considered in simulation.

3-2.2 Design library of partial models

Partial models of enterprise modelling have the purpose of describing models and concepts for a group of common problems and systems. For this work, the partial models are eligible for the problem class of job-shop production systems (AS2). For simplification, the partial models do not consider material flows in space, but simulate material displacement as processes with fixed cycle times. The partial models implement the four modelling views of enterprise modelling: function view, resource view, information view, and organization view. Therefore, four specific classes are provided. Partial models of the generator class and drain class generate and destroy information representing material on the shop floor and describe the information view. The process class represents the activities of material routing and describes the function view. Resources describe the resource view and represent the capacities of technical and human resources. In the organization view, organizing the work on the shop floor has no specific partial model as represented. The organization of work is implemented by customization of the partial models of the function, resource, and information view, particularly by adding code and functions that are executed by resources when starting and stopping activities.

The generator class has the purpose of generating new products during simulation and moving them to the tasks of their process map. The generator is modelled from the generic DES elements: source, queue and process. The conceptual model of the generator is given in Figure 29. The input data for the generator are the product arrivals, describing the arrival stream. For each arriving material, the generator writes the associated information in the generated element. The information contains data to identify the material and the sequence of activities. After the simulation starts, the source instantly creates all arriving materials and sends them to the queue and the process. The process delays each material (p_i) based on the inter-arrival time of the arrival stream. After the delay, the process sends each material to the first activity of its routing. Generated materials wait in the queue before entering the process with the delay. On exit, the process writes the event of arrival together with the timestamp in the event-log of the simulation results.

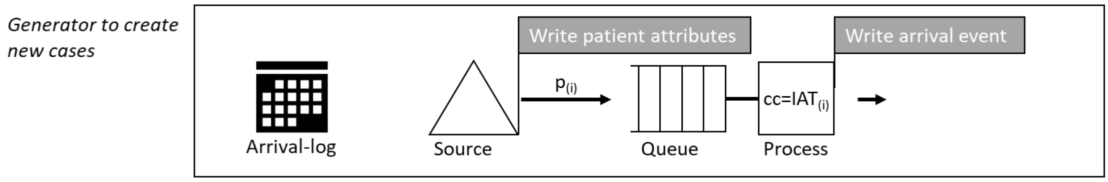


Figure 29: Case generator class

The drain class has the purpose of removing finished materials from simulation during experimentation; it consists of the generic DES elements queue and drain. A conceptual model of the drain is illustrated in Figure 30. The drain eliminates incoming materials and writes the moment of removal as a departure in the simulation results. The process of destroying materials occurs instantly and a waiting line is not required.

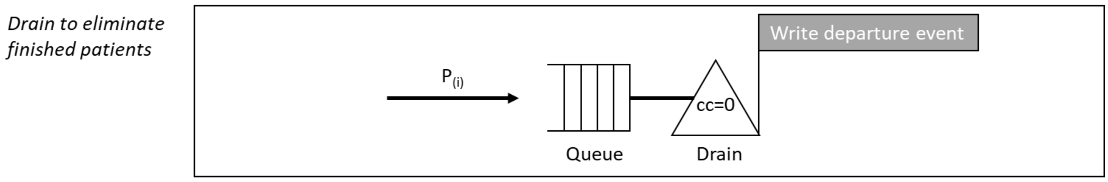


Figure 30: Drain class

The process class simulates activities of the material routing. The class is an instantiation of the generic DES class process. Each process describes an activity of the material routing. A conceptual model of the process class is given in Figure 31. In parameterization, the process receives the duration ($d_{a,s}$) of an activity (a) for a material (i) from the simulation data. Further, the process needs one or multiple assigned resources ($r_{i,s}$) to be executed. In the simulation, a material arrives at the process and requests a resource. If the resource is available, it moves instantly to the process and starts execution of the activity. When the resource is not available, the material waits in a virtual queue for arrival of the resource. On arrival, the resource writes the event of beginning the activity and a timestamp into the event-log of the simulation results. The resource is blocked until the activity is finished. When the activity is finished, the resource leaves instantly and writes the event and a timestamp into the event-log. The material moves to the successor activities from the simulation data.

Activity ($a_{i,s}$) of a patient (i) and the step (s)

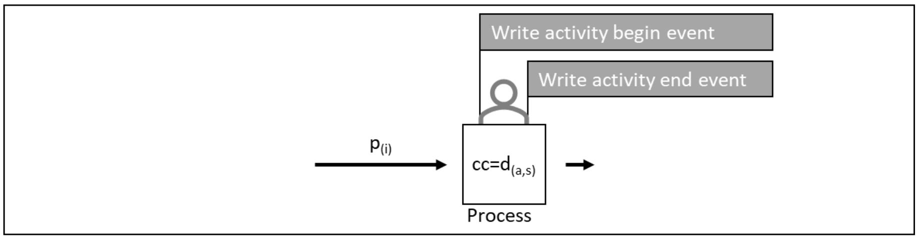


Figure 31: Process class

The resource class describes the capacities of the ED for groups of identical resources (r). The class is parameterized with the number of elements in the class, the shift calendar and the available services. Data is received from simulation data and assumed to be available. The shift calendar periodically defines the time where resources are in the states of active and passive. Passive resources are in an idle state and active resources execute activities. Resources are used to carry out transport and processing activities. For processing, they travel to the case. Traveling between resources is idealized and to be considered as performed instantly. A conceptual model, describing the resource class and interaction with the activity in the process class, is given in Figure 32. If resources are requested by multiple cases, the control logic defines the prioritization of cases; by default, there is FIFO-style prioritization. When changing the state owing to shifts or breakdowns, resources write an event in the event-log.

Resource (r) and interaction with activity

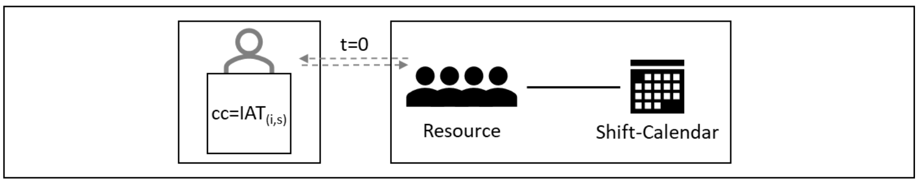


Figure 32: Resource class

In enterprise modelling, the organization view describes the organization of work. The organization of work is partially described by a partial model that describes job-shop manufacturing systems. Additionally, the generic aspect of the shift calendar is implemented in the resource class. Depending on the physical system, additional organizational information is stored in the partial models and, as a consequence, the partial models lose genericity and become case-specific partial models (in this work, partial models of the ED and train remanufacturing systems). Despite sacrificing genericity of the partial models, customization at one point is required to model the entire complexity of physical system. Therefore, implementation of the organization view is performed within the partial models of generator, drain, process and resource and provides case-specific partial models to generate particular models of the system by data-driven modelling.

3-2.3 Data-driven simulation

For data-driven simulation, an algorithm generates a model from simulation data, action parameters of optimization, partial models for the generator, drain processes and resources. The algorithm derives particular models from the partial models by instantiating objects of the reusable classes for the generator, drain, process and resource. Parameterization extends each object with case-specific attributes, such as inter-arrival time and required resources for execution of activities. The first step is generation of the generator and drain for creation and removal of materials; the second step is the generation of resources for execution of activities; and the third step is the generation of the activities of each material. The description of the model generation algorithm with pseudocode is given in Figure 33. After the generation of the model, the algorithm automatically executes simulation experiments.

```
# Generate materials
CREATE Generator
CREATE Drain

# Generate resources
FOR r = 1 TO R
    CREATE Resource(r)
    PARAMETERIZE Resource(r)
NEXT

# Generate processes
FOR p = 1 TO P
```

```
    FOR s = 1 TO Si
      CREATE activity(i,s)
      PARAMETERIZE activity (r)
      CONNECT activity(i,s), activity(i,s-1))
    NEXT
NEXT

# Start simulation experiment
EXECUTE SIMULATION
```

Figure 33: Data-driven modelling

In the first step, the algorithm instantiates objectives of the generator and drain classes. During instantiation, the algorithm parameterizes the generator with the product arrivals, inheriting the sequence and inter-arrival times for product arrivals. In the second step, the algorithm generates the technical and human resources. The input data is the bill of resources describing the temporal capacities of resources. The algorithm loops through all lines of the bill of resources, where 1 is the first line and R is the last line, instantiates objects from the resource class and gives parameters for amount and temporal capacity. In the third step, the algorithm generates the activities of the material flows for all incoming materials. The algorithm instantiates the activities of the steps *s* for each material p_i, and writes the parameters for cycle-time and resources in each activity. After parameterization, the algorithm links each activity with each predecessor and successor to describe the material flows. A conceptual model of the output is illustrated in Figure 34. After generating the model, the algorithm starts the simulation experiment that automatically stops when all materials have arrived at the drain.

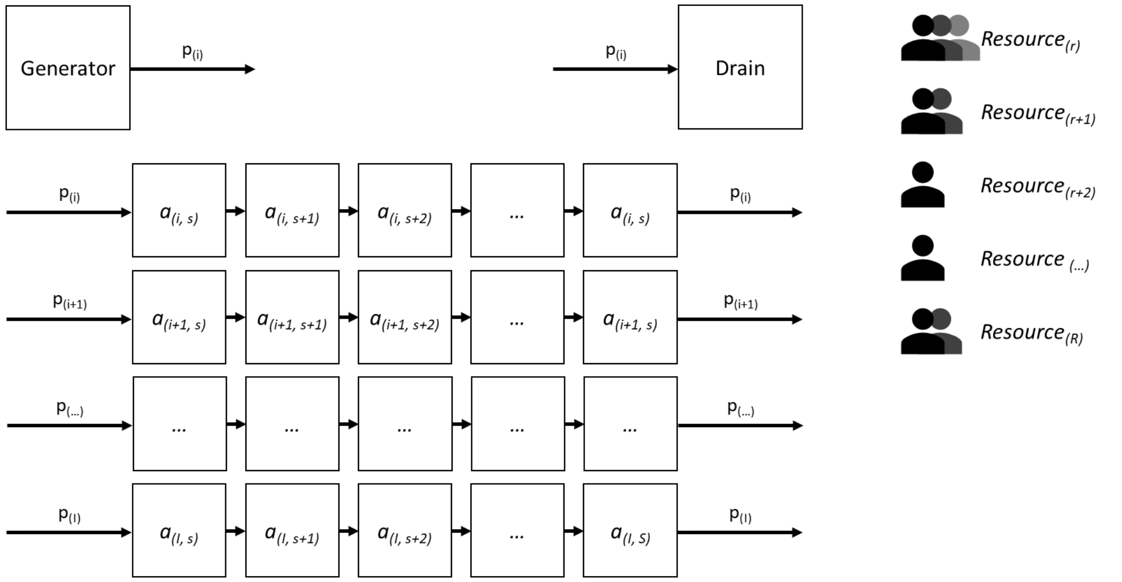


Figure 34: Generic simulation model

After simulation model building the simulation starts and the generator creates materials. With delay of the inter-arrival time, the generator sends the materials to their routings. At each activity, the material requests resources to execute processes. Missing resources cause virtual queues, while the materials wait in the activity for arrival of the resource. Available resources travel instantly to the place of need. Limitations of the layout are not considered. After finishing an activity, the resource is released. Materials move from activity to activity, according to their routing. After reaching the last activity, materials move to the drain. The drain removes the material from the simulation. During simulation, the generator, process and drain write events for arrivals and departures of materials as well as the beginning and end of activities in the event-log of the simulation results. The event-log follows the same structure as the event-log that provides the historical material flows as input for data-retrieval.

3-3 Case-Studies

This section has the purpose of showing how the generic methods can provide valid simulation models of different systems in different domains and enable simulation experiments without manual interactions. The case studies use the cases ED (section 0) and train remanufacturing system (section 0). Both cases are in different domains;

however, as highlighted in section 0, they follow the logic of a job-shop production system. Customization is used to adapt the generic models to the two specific cases. After illustrating the application of data-driven modelling, the case-studies highlight how simulation in dynamic models can generate new knowledge for problem-solving that was not previously available.

3-3.1 Emergency department

In the case study of the ED, the presented methods, the libraries of this chapter and simulation data from the previous chapter, were applied to model and simulate the system. The input covers system data and flow data. System data describes the system configuration and flow data describes the patient arrivals and their pathways through the ED. Automated modelling and simulation were applied in three steps according to enterprise modelling genericity levels. On the generic level, Table 12 defines the framework of using dynamic material flow simulation. On the partial level, a library with generic classes for the generator, drain, process and resource was designed as previously described. On the particular level, customization was used to adapt the partial model to the case study and data-driven modelling was used to build models and run simulation experiments in four steps.

1. Creation of generator and drain: The algorithm instantiated a generator and drain from the partial models to simulate the arrival and departure of 35 240 patients of the historic arrival stream during one year of simulation.

2. Creation of resources: The algorithm instantiated 64 technical and human resources (e.g. stretchers, doctors, nurses) in a loop, and the temporal capacities were defined for each resource by providing the shift-calendar.

3. Creation of activities: The algorithm instantiated 297 463 activities from the process map in a loop to model the pathways of patients, and linked the activities to describe predecessor and successor relationships.

4. Run simulation: The algorithm completed the simulation experiments and provided evaluation parameters for a set of action parameters by simulation.

The data-driven modelling exposed the benefits and drawbacks of the partial models. The approach allowed simulation models to be generated and simulated on request. Given the availability of data and partial models, this allows a time reduction for modelling and simulation from days and weeks to seconds and minutes. Reduction of modelling and simulation time allows the evaluation of a large number of solution candidates and the use of data-driven modelling and simulation for optimization. However, drawbacks in the data-driven modelling were identified and are caused by the generic nature of the partial libraries. The assignment of patients to zones and stretchers after triage requires the modelling of specific functions in the process class. For data-driven modelling, a new genericity level was introduced between the partial and particular level. The customized partial model enriches the partial models by case-specific functions of this specific ED.

Simulation of the historic product arrivals and process maps in the initial and future system evaluates the performance in both scenarios and provides the patient LOS, the time between arrival and departure of patients. Figure 35 provides and compares the average LOS for patients with blue and orange severity indices. The LOS defines the time between patient arrival and departure. The transition from S1 to S2 degrades the average LOS of blue patients from 108 to 132 min and improves the average LOS of orange patients from 257 to 241 min. The LOS of the patient mix degrades from 157 to 168 min. The results confirm the analysis made in section 2-3 that forecasted waiting lines after redesign, particularly in the blue zone. Independently, the box-plots show a high degree of uncertainty in the LOS, particularly for the blue patients.

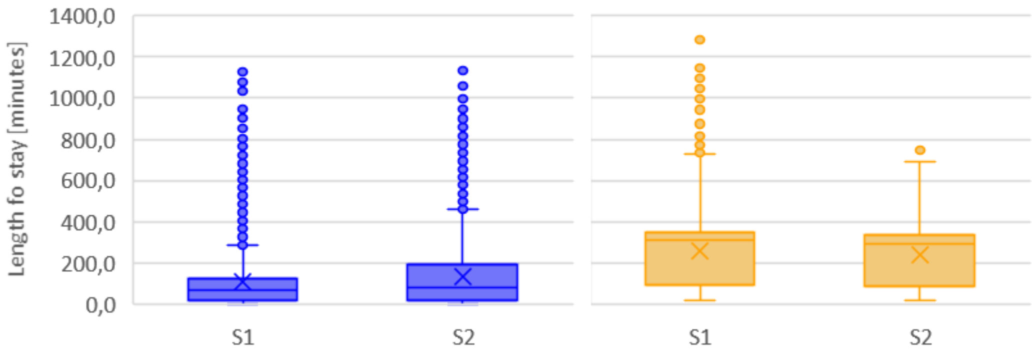


Figure 35: Average length of stay of orange and blue patients

Although this analysis is commonly used in the design of EDs, there are shortcomings. Comparing both scenarios does not provide an indication about the improvement and degradation of the different patient types and their dispersion. The simulation model of the ED includes the routings of 62 different patient types. Each routing variant has a specific, static LOS, in this work defined as reference. The reference is the ideal LOS of each patient. It is statically calculated for each patient with the routing of the patients and durations for activities (section 2-3). It is defined solely by the activity durations and does not entail waiting. The analysis in Figure 35 does not consider different patient types. It shows the average LOS of all patients, independent from the reference because the analysis of the global distribution of the LOS of all patients is not meaningful, owing to the variance of the reference.

To overcome this issue, this work proposes a comparison of the LOS of the entire patient population with the reference and an evaluation of the dispersion. Figure 36 illustrates this method. The graph in (a) plots the reference by plotting the reference vs. the reference. The reference is identical patients with the same routing, since the reference provides a straight line (red). For evaluation of simulation results, (b) plots the simulated LOS vs. the reference. The discrepancy between reference and simulated LOS is caused by the dynamics of the simulation, particularly by waiting times. Vertical structures in the graph represent multiple patients with the same reference but having waiting times to different degrees.

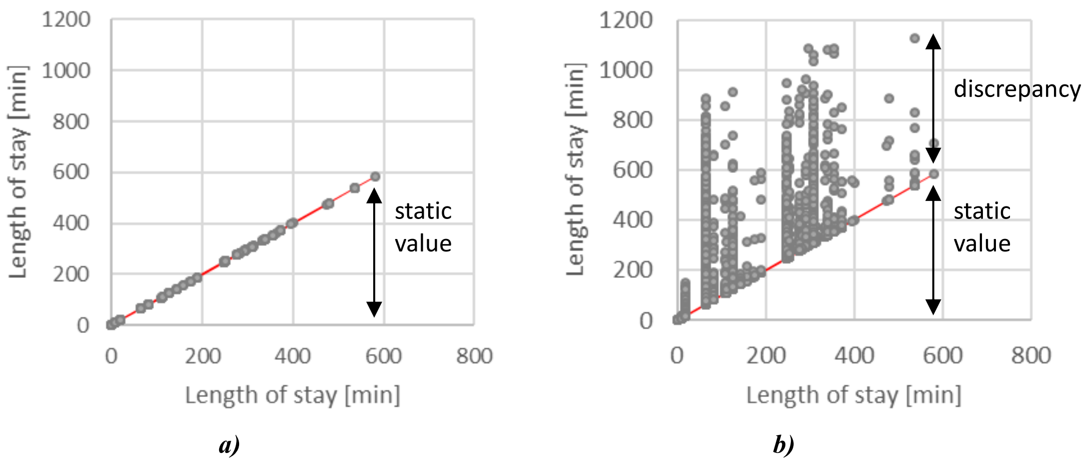


Figure 36: Length of stay – reference vs. reference

The illustration method of Figure 36 is applicable for the patient populations based on their severity indices in the initial and future scenario. Figure 37 plots the reference and LOS of the populations in the initial (top) and future scenario (bottom) in multiple sub-plots. Each sub-plot shows the populations of blue and orange patients based on their severity index in the initial system that was either green, orange, or red, highlighted in the background of the sub-plots. For example, sub-plot (a) shows the patients that initially had a green severity index but received a blue or orange severity index in the new triage system.

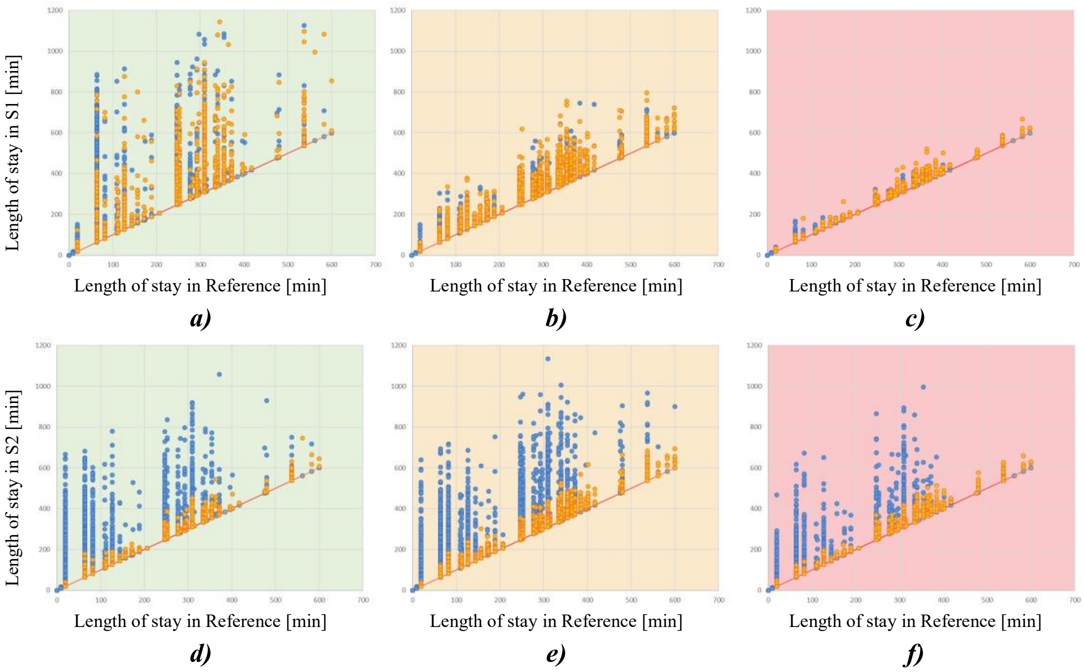


Figure 37: Length of stay – reference vs. simulation for patient groups

The plots reveal that the reference of the patients is not correlated to the severity index of the patients. The range of references is identical for all groups of patients, independent from the severity index. However, there is a correlation between the severity index impacting the priority and dispersion. In the initial situation, the dispersion decreases with increasing priority (green, orange, red) of patients, illustrated in (a), (b) and (c). In the future situation, the dispersion is linked to the new severity indices (blue, orange), illustrated in (d), (e) and (f). The plots explain the observations of Figure 35. When switching from the initial to the future situation, the LOS improves, particularly for patients that benefit from increased priority (orange patients in (a) and (c)) while patients with decreased priority have a lower LOS (blue patients in (e) and (f)). To improve some green patients by increasing priority, some red patients are scarified by decreasing the priority.

3-3.2 Train remanufacturing systems

In the case study of the train remanufacturing system, the presented methods (libraries in this chapter and simulation data from the previous chapter) were applied to model and simulate the system. The input covers system data and flow data. System data describe the system configuration and flow data describe the train arrivals and their routings through the train remanufacturing system. Automated modelling and simulation were applied in three steps according to enterprise modelling genericity levels. On the generic level, Table 12 defines the framework of using dynamic material flow simulation. On the partial level, a library with generic classes for the generator, drain, process and resource were designed as previously described. On the particular level, customization was used to adapt the partial model to the case study and data-driven modelling was used build models and run simulation experiments in four steps.

1. Creation of generator and drain: The algorithm instantiated a generator and drain for the planned arrival and removal of 20 trains to be remanufactured in the year 2022. Each train is an assembly of 10 wagons.

2. Creation of resources: The algorithm instantiated 50 resources in a loop, particularly installations for the processing of wagons, and provided the shift-calendar with temporal capacities for each resource.
3. Creation of activities: The algorithm instantiated 8 580 activities in a loop to model the remanufacturing process of each train with 429 activities and linked activities to describe predecessor and successor relationships.
4. Run simulation: The algorithm runs the simulation experiments and provides evaluation parameters for a set of action parameters by simulation.

The data-driven modelling exposed benefits and drawbacks of the partial models. The approach allowed the generation and simulation of models on request. Given the availability of data and partial models, this allows a reduction of the time for modelling and simulation from days and weeks to minutes. Reduction of modelling and simulation time allows for the evaluation of a large number of solution candidates and the use of data-driven modelling and simulation for optimization. However, drawbacks are linked to the generic nature of the partial libraries. In the case study, customization of partial libraries was required in the process class to add a new function to check the start conditions before the start of the activity. For data-driven modelling, a new genericity level was introduced between the partial and particular level. The customized partial model enriches the partial models by case-specific functions of this specific train remanufacturing system.

The simulation evaluates the planned system for the given action parameters of train arrivals every four weeks and a maximum of four trains in the system. The simulation results consider the waiting lines and waiting times of trains when competing for resources in the dynamic model and provide the makespan of each train. The results are illustrated in Figure 38 and compare the makespan of trains that was simulated in a queueing system with the calculated lead time of the remanufacturing process. The results show the need for validating static models and calculation by simulation of dynamic models. In particular, in a dynamic system the results of dynamic simulation are required to provide robustness of the measures.

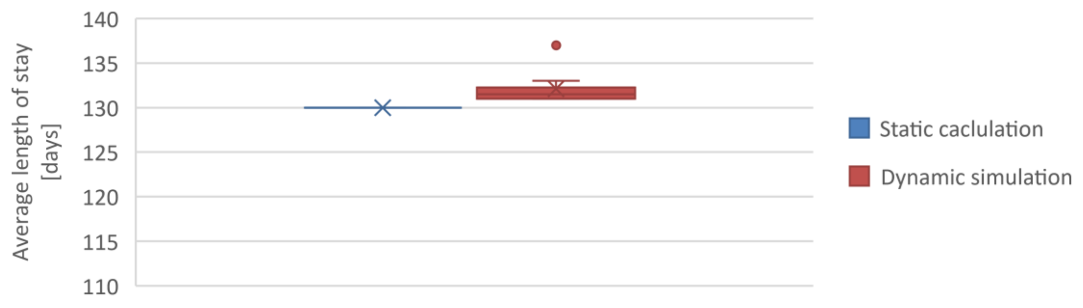


Figure 38: Average length of stay for trains in static calculation

The provided simulation results in this case study have the same structure as in the ED and enable the theoretical application of the illustrated dispersion analysis. However, because in this case there are only 20 trains to be remanufactured and all trains are of the same type, the dispersion analysis will not provide new knowledge about the performance of the population that is not visible in the box-plot. However, application in other manufacturing cases is of interest.

3-4 Discussion

The leading question of this chapter is RQ1: **What are the methods for generating simulation models for problem-solving from the digital shadow?** This question covers the retrieval of simulation data and the generation of a model. Retrieving simulation data was addressed in Chapter 2. This chapter addresses the automated generation of a material flow simulation model and execution of experiments. Before explaining the methods, the framework for modelling is provided, independent of the question of automated generation and simulation of models. The framework is based on the standard for enterprise modelling, particularly the genericity levels and modelling views (see section 0). Genericity levels define models through generalization and specialization mechanisms in the generic, partial and particular level. Modelling views describe different aspects of the system, particularly function, information, resources and organization. In this framework, four methods were identified to generate simulation models (see Figure 39).

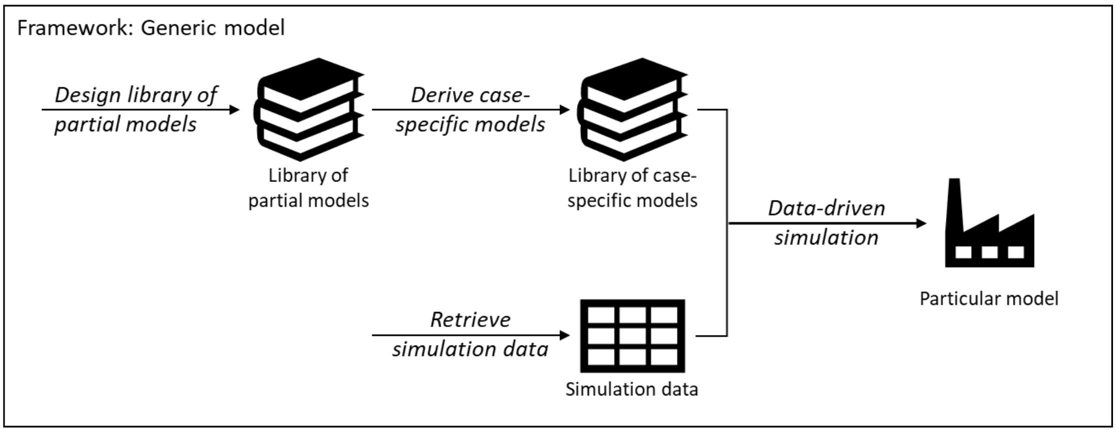


Figure 39: Methods of data-driven simulation

Definition of the generic model provides the framework for modelling and simulation. The system modeler gathers information about the system and understands the problem. The objective is to define the properties of the model, including the modelling technique language and modelling constructs. However, this method was treated in this work only on the application level because the framework was given. The framework of this thesis is defined by simulation of material flows in stochastic and dynamic environments (see Table 12). However, it should be mentioned that a wide range of models exist to address a wide range of problems. Defining the generic model is crucial and has a significant impact on the success of the simulation studies.

Design library of partial models provides a library of reusable models for data-driven modelling. Partial models are neutral, reusable models describing different aspects of the system, according to the enterprise modelling views. Aspects inherit function, information, resources and organization. Functions, information and resources were used in this work as synonyms for products, processes and resources. The organization view includes all information regarding responsibilities, e.g. the assignment of resources to zones in the ED. Partial models are common for a class of problems or type of system, e.g. job-shop or flow-shop. The library of partial models contains generic representations of products, processes and resources as well as aspects regarding the organization.

Derivation of case-specific models is required to customize the library of partial models to the requirements of the specific case. Despite belonging to the same class of problems, different systems to be modelled can feature case-specific behaviour of the particular system. In the two case-studies, both systems share common partial models for job-shop systems, but inherit case-specific behaviour, e.g. the assignment of patients based on ESI in the ED and the conditions to start activities in train-remanufacturing. Derivation of case-specific models provides customized, case-specific libraries that enable the algorithmic generation of models, entirely based on the generic simulation data. However, simple cases without case-specific control logic and behaviour do not require customization of the partial library. In these cases, the step becomes optional and data-driven simulation is possible from the generic partial models.

Data-driven modelling and simulation is the method used to generate models and run simulation experiments from the data. In data-driven simulation, an algorithm builds a simulation model by instantiating particular models from the library of case-specific models. The algorithm instantiates and parameterizes particular models according to the simulation data. After model-building, the algorithm starts simulation experiments and waits for termination of the experiment to provide results of the simulation data. The previous design of a library with partial or case-specific models is crucial and absolutely necessary for data-driven modelling and simulation.

This chapter provides the methods used to generate simulation models and run simulation experiments. The methods are evaluated in the case-studies of the ED and train remanufacturing system. Despite the obvious differences in the domain and systems, the models are similar. Materials (trains or patients) arrive in the system and follow their routings (remanufacturing process or pathways). During their routing, there are several activities to be executed by resources (doctors or installations) and, finally, the material leaves the system after a given lead time (makespan or LOS). For modelling, the similarities of both systems are enabled using common partial models.

The case-specific behaviour of the system was implemented by customization of partial models and provided a library of case-specific models. The scope of customization in the case studies included aspects of the organizational view. These elements were marked by heterogenous structures and data, which were in contrast to products, processes and resources not describable in a standardized data structure. Acquisition and modelling required expert interviews and problem-specific tools, e.g. functions, code and BPMN models. Customization is to be performed for the case-specific behaviour of each system and is a manual activity. The process of customizing for each of the two cases required time in the dimension of hours and days. After providing customized partial models and simulation data, modelling and simulation is available on request. By reusing the concepts of partial and particular models of enterprise modelling and introducing customization, the methods tackle the existing conflict in the literature [20] between genericity and specificity (see Figure 40). The available approaches are either generic or specific. Generic approaches are appropriate to model a wide range of systems, but lack capability to implement case-specific behaviour, and specific approaches implement case-specific behaviour but are not appropriate to model a wide range of systems.

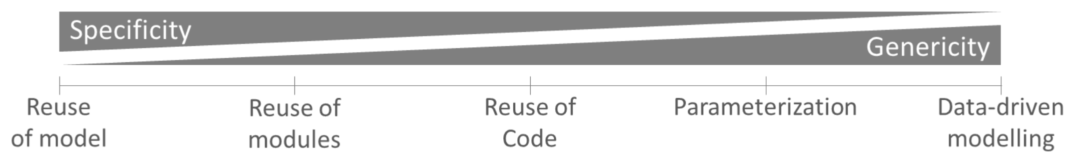


Figure 40: Trade-off between genericity and complexity

By using the concept of customizing partial models, the methods provide a trade-off by reusing modules and code via the case-specific libraries. Partial models (modules) describe a class of problems or systems (in this work, job-shop production systems). Specific partial models inherit the case-specific behaviour (code) of a particular system. Introducing customization of partial model enables the use of generic partial models to simulate a wide range of systems by data-driven simulation from the data.

In the case-studies, the methods were used to compare and evaluate planning scenarios. Data-driven simulation requires time in the dimension of seconds and minutes. Consequently, data-driven simulation allows the evaluation of a large number of solution candidates. Manually modelling and simulation of one solution candidate can require days and weeks. Thus, running a design of experiments is not possible without these methods, due to a lack of time. However, the number of experiments in the design of experiments is limited by the required time for data-driven simulation. Beyond executing the design of experiments, data-driven simulation is usable within optimization. The algorithm can implement values for decision variables coming from optimization algorithms in the model and provide the simulation-dependent results as values for the objective function. This allows going beyond simple optimization of parameters and enables the introduction of qualitative parameters, e.g. adding and removing resources, changing patient mix, or changing the control logic for assignment of patients by using different models from the library.

The case study of the ED was used to provide a new method for analysis of simulation results. The need for the method comes from the high variance of patients (materials) with individual routings. Data-driven simulation enabled the development of a model that included 62 different routings, which is barely possible manually. Analysis of the global LOS (makespan) does not take into consideration the inherent variance of the reference caused by the different routings. The method proposes an analysis of the dispersion of the entire population of patients by comparing the results from the simulation with the statically calculated reference. This approach enables an analysis to be performed that considers the initial dispersion of the different patient variants when evaluating the LOS for the entire patient population. The discrepancy between the reference and the simulated LOS can be explained by waiting times due to lack of resources. In the case study, this analysis allowed the improvement of orange patients and degradation of blue patients to be explained. This method is of particular interest in cases having a wide range of variants, e.g. different materials and patients. A similar analysis is possible with the simulation results of train remanufacturing. However, because there was just one type of train considered in the simulation, the analysis does not yield benefits. Application to new cases is of interest to evaluate the benefits and drawbacks in the domain of manufacturing.

Chapter 4 Simulation and Optimization

This chapter is dedicated to the question of using data-driven simulation to solve optimization problems (RQ2): **What are the methods for optimizing material flows through the digital shadow and data-driven simulation?** In particular, this chapter explains how simulation data (Chapter 2) and the methods for data-driven simulation (Chapter 3) can be applied to solve optimization problems (see Figure 41). To address this question, this chapter first provides a state of the art in optimization (section 0), provides the methods for simulation and optimization (section 0), applies the methods for solving three optimization problems (section 0) and provides a discussion of the methods and insights gained in the case studies (section 4-4).

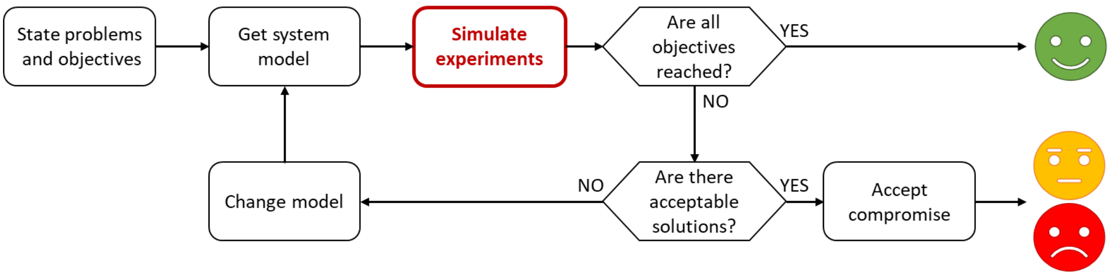


Figure 41: Optimization in problem-solving

The state of the art provides an overview of different optimization approaches and demonstrates how simulation can optimize complex systems. The methods of this work, couple data-driven simulation and algorithmic optimization, are used to solve optimization problems. For coupling, simulation evaluates the objective function of the optimization algorithm. The case studies present the application for three different optimization problems in train remanufacturing and ED. The case studies highlight how the methods can optimize different optimization problems without explicitly formulating mathematical models and reformulating the optimization problems by coupling data-driven simulation and optimization.

4-1 State-of-the-art

In problem-solving, optimization has the objective of providing solution candidates to minimize or maximize one or multiple objective functions. This section provides the state of the art to understand the available optimization methods and clarify how simulation can be used for optimization. The state of the art is separated into two sections. Section 0 presents the optimization methods and provides methods to solve different optimization problems and section 0 presents the method of simulation-based optimization, linking simulation and optimization.

4-1.1 Optimization methods

Optimization describes the process of searching for a solution that minimizes or maximizes an objective function towards a defined objective, called the optimum. During optimization, the impact of decisions towards the objective are evaluated, typically by using mathematical models [5]. Optimization can be used for a wide range of industrial and scientific decision problems. The purpose of optimization is typically to improve the efficiency, performance and effectiveness of a system or process. For finding optimal solutions, optimization uses mathematical methods. Especially in complex and large-scale systems, optimization can become difficult due to the high number of potential solutions. Depending on the type of system and problem, different methods are applicable for optimization. The choice of an appropriate optimization method is crucial for resolution of an optimization problem. Figure 42 provides an overview of the available optimization methods, based on [131]–[134]. At the top level, there are different branches of optimization. Algorithmic optimization uses algorithms to find minima and maxima for optimization problems, while design of experiments uses systematic experiments to explore the solution space.

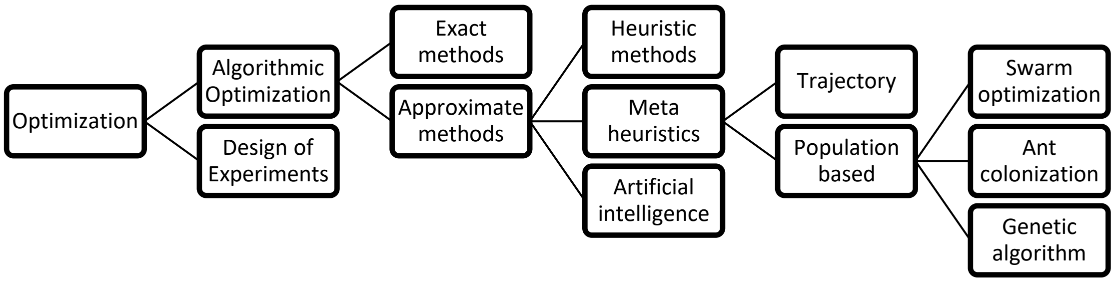


Figure 42: Optimization methods

Within algorithmic optimization, the literature in optimization distinguishes between exact and approximate methods. Exact methods classify methods that provide the optimum for the optimization problem, e.g. enumeration and branch and bound. Enumeration compares all possible solutions of an optimization problem, and branch and bound divides the solution space into subspaces and successively eliminates subspaces without optimal solutions. Common problems of using exact methods are the effort of testing solutions and the amount of computing power. Consequently, these methods are appropriate for problems with limited size and complexity of the solution space.

Approximate methods are non-exact methods, approximating towards the optimum without knowing the optimum. Examples of approximate methods include heuristic methods, metaheuristics and artificial intelligence. Heuristics include problem-specific methods. In contrast, metaheuristics are not problem specific. These methods use general, superordinate search strategies to provide solutions for a high variance of optimization problems. As a third group, artificial intelligence uses agent-based systems and neural networks to solve optimization problems. These methods attempt to mimic human intelligence and understand the behaviour of the system to be optimized.

Within metaheuristics, there are trajectory and population-based methods. Trajectory methods such as local-search start optimizing with an initial solution and improve the solution successively by searching within the neighbourhood. Limitations appear when optimizing for problems with local optima. More advanced methods such as tabu search [135] and simulated annealing [136] use additional functions to proceed from local optima to towards global optima. In contrast, population-based methods overcome local optima by generating new solution candidates with stochastic behaviour. Examples of these methods are swarm optimization [137], ant colonization [138] and GA [139]. These methods create and evolve a variety of solution candidates in parallel and are advantageous for problems with complex solutions spaces.

A crucial property affecting the choice of an appropriate optimization method is the number of objectives. There are mono- and multi-objective problems. Many real-life optimization problems are multi-objective problems, with multiple objectives to be minimized or maximized [138]. Additionally, conflicts between the objectives can appear [134]. In a multi-objective problem, there is not one single best solution, but a set of superior solutions. Within the solutions of a multi-objective problem, a set of dominating solutions is superior to a set of dominated solutions. The non-dominated solutions satisfy the different objectives to varying degrees. Various non-dominated solutions represent trade-offs between the different objectives of the optimization problem. In [132], complex problems are solved with multi-objective solution spaces, with a GA being the most common metaheuristic.

One strategy for handling multiple objectives is the introduction of weights. Weighting of multiple objective functions transforms multi-objective problems to mono-objective problems [140]. However, the method of using weights has challenges. First, it is not possible to satisfy all objectives in a multi-objective problem with a single solution; the solutions generated by weights compromise multiple functions. Second, weighting objective functions requires expert knowledge about the critical systems. Optimality of the solution is limited to the framework of the given weights. Changing the weights can provide completely different solutions in optimization.

Beyond compromising multiple objective functions by using weights, multi-objective optimization optimizes for multiple objectives simultaneously. With this approach, multi-objective optimization provides multiple solution candidates with trade-offs between multiple objective functions. The concept of Pareto optimality is used to provide solutions that are clearly better than others. Specific multi-objective algorithms attempt to provide Pareto optimal solutions [132] with trade-offs between multiple objectives. An algorithm for addressing the problem of multiple, conflicting objective functions is the non-dominated sorting genetic algorithm II (NSGA-II) [141], [142]. However, Pareto optimization is not exclusively addressed by GAs.

There are numerous optimization methods available. Strengths and weaknesses depend on the problem to be addressed. In particular, for the design of complex systems, where problems with multiple, conflicting objectives can occur and are to be optimized, the ability to deal with multiple objectives is crucial. Methods of interest for solving problems with multiple objectives are design of experiments and Pareto optimization. Design of experiments creates an understanding of the impact of parameters on multiple objective functions, and Pareto optimization provides sets of dominant solutions with trade-offs between multiple objective functions.

4-1.2 Simulation-based optimization

Despite the differences, heuristic methods and metaheuristics have properties in common [143]. During the search for a solution, both move from one solution to another and search for a solution inside the search space, although metaheuristics are not limited by restrictive assumptions about the search space. Another shared property is the principle of solution search. The principle of search with heuristics and metaheuristics is illustrated in Figure 43, according to [143]. The individualities of the methods are within the strategies for generating initial and new solutions.

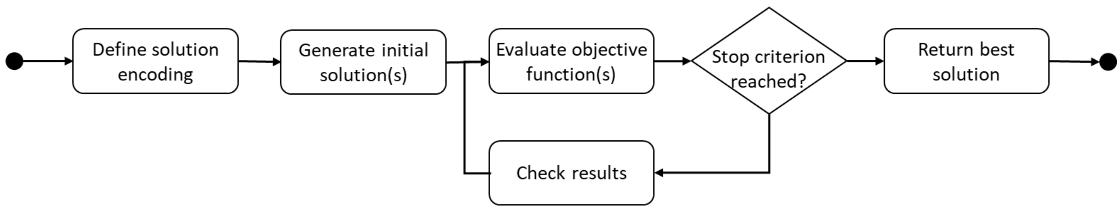


Figure 43: Principle of heuristics

The optimization principle begins with *define solution encoding*. In this activity, the decision-maker chooses the heuristics for optimization; these define the strategy for defining solution candidates. In *generate initial solutions(s)*, the algorithm creates one or multiple solution candidates to be tested, depending on the optimization heuristics. In *evaluate objective function(s)*, the algorithm evaluates the fitness of the new candidates(s). The evaluation takes one or multiple objective functions into consideration. In the decision *stop criterion reached*, the algorithm checks if more populations are to be evaluated. Decision criteria consider properties of the algorithms, for example, the number of iterations or properties of the solutions (e.g. a threshold of the objective function(s)). If the algorithm is to be stopped, the next activity is *return best solution*; the algorithm provides the best solution candidate.

Evaluation of the objective function is crucial for optimization heuristics. A wide range of techniques can be used to evaluate objective functions. Mathematical models use mathematical functions to formulate the objective functions. An example for mathematical models is (mixed) integer linear programming [144], [145]. In mixed integer linear programming, the objective function is formulated as a function of the decision variables. A solution candidate is described by a vector of decision variables. For evaluation of the objective function, the mathematical model is resolved. The objective functions of the mathematical model provide the fitness of the solution candidate.

Other approaches use complex modelling techniques. The authors of [146] used CAD models for optimizing welding points with the finite element method, and in [147] the authors optimize the layout of a facility by modelling the layout as a quadratic assignment problem. For optimization with computer models, a set of values for the decision variables is implemented in the model and a simulation evaluates the solution candidates. The model provides a set of values for the objective functions. Optimization requires modelling and simulation in a loop to evaluate decision vectors and provide objective vectors.

Optimization of material flows uses modelling and simulation in DES. For optimization, the values of decision variables are implemented in the model and simulation provides the values for the objective functions. For optimizing an assembly system in [139], the authors used data-driven modelling to model solution candidates of an assembly system and used simulation of material flows to evaluate the efficiency. By combining optimization with a GA and data-driven simulation, the authors optimized the scheduling of an assembly system. In an analogous approach in [136], the authors applied simulated annealing to optimize the layout of a machine shop and used a model in DES to evaluate solution candidates. The works using simulation and optimization to solve design problems separate both domains. The authors combined simulation in generic models and optimization by defining interfaces to exchange values for decision variables and objective functions [9], [132]. They applied metaheuristics to optimize without knowledge about the system and for evaluating the use of simulation in parametric and data-driven models.

Providing a model to evaluate the objective functions in optimization is complex and requires expert knowledge. Especially when dealing with complex systems, the requirements to the modelers increase significantly. Additional problems occur with the availability of dynamic systems having stochastic behaviour. However, a simulation of material flow is capable of evaluating the objective functions and providing an objective vector for a given decision vector. For data-driven modelling, the evaluation of the objective function is automatable. Simulation-based optimization couples parametric and data-driven simulation and optimization heuristics to optimize complex systems using simulation.

4-2 Methods for simulation and optimization

This work proposes simulation-based optimization to solve problems in the design of material flows. Owing to the complexity of production systems, the evaluation of solution candidates is not possible without simulating material flows in a dynamic environment. In this approach, simulation evaluates the objective functions for each solution candidate by using techniques of data-driven modelling and simulation. Data-driven simulation enables the replacement of entire models without reformulating the mathematical models, constraints and objective functions of the optimization problem by only replacing the data of the case-study. Enablers are generic interfaces that manage the exchange of decision vectors and objective vectors between optimization means and the simulator.

In simulation-based optimization, simulation and optimization heuristics function in a loop (see Figure 44). Generic optimization means provide values for the decision variables and evaluate the objective functions. To optimize without knowledge about the modelled system, this work recommends using metaheuristics. A conceptual architecture for combining simulation and optimization is given in Figure 44, and follows the study in [132]. In this approach, the optimization heuristics request the evaluation of solution candidates from the simulator, which implements the solution candidate in a model and executes a simulation experiment to evaluate the candidate. After simulation, the simulator provides the optimization parameters to the optimization heuristics, which interpret optimization parameters as objective functions.

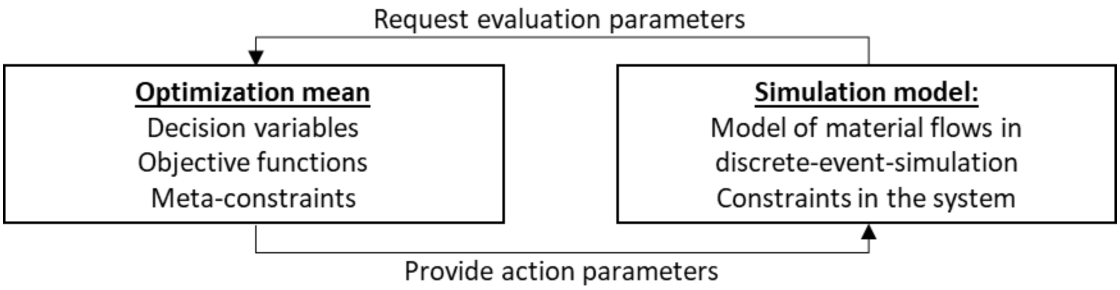


Figure 44: Architecture for simulation and optimization

The scheme illustrates the interaction between optimization means and the simulation model. The optimization means consists of the optimization heuristics, definition of decision variables and objective functions, as well as the constraints of the optimization problem. Decision variables and objective functions determine the decision and objective vectors and define the interface between optimization means and the simulation model. Meta-constraints describe relations between decision variables and limits for the objective functions. The simulation model is represented by the methods for data-driven simulation. These methods provide models of the material flow, execute experiments and provide the decision vector as a service for optimization. The simulation models inherit the constraints of the physical system. To provide these, the control logic of material flow is implemented within the methods for data-driven simulation. In simulation-based optimization, the constrains of the physical system are exclusively defined in the simulation model. However, models of material flows can be substituted by models of other domains.

4-3 Case-Studies

This section presents three applications of data-driven simulation and optimization from the digital shadow. The purpose of the case studies is to highlight the three added values of this approach. First, using data-driven modelling and simulation enables substituting the cases by replacing the simulation data of one case by the simulation data of another case. This allows optimization for a wide range of cases by only replacing simulation data. Second, data-driven modelling and provides complex models for optimization. These models can inherit complex behaviour and control logic of real systems and prevent explicit modelling of these constraints in objective functions. This allows generic simulation models to be used to address a wide range of optimization problems. Third, addressing different cases and problems can require different optimization means and algorithms. Consequently, coupling data-driven simulation and optimization enables the addressing of a wide range of problems without requiring the definition of optimization problems and algorithms. To highlight the abovementioned added values, this work uses data-driven simulation and optimization to solve three design problems in the two industrial cases of a train remanufacturing system and an ED. The following is an overview of the optimization problem.

- **Order Scheduling** (section 4-3.1): scheduling of product arrivals in the train remanufacturing system. In this problem, a fixed number of train orders are to be scheduled by defining the sequence and timing of arrivals. The objective is to minimize the makespan of trains and maximize productivity. The optimization means for addressing the problem is a full design of experiments.

- **Resource allocation problem** (section 0): allocation of resource to zones of the ED. In this problem, a fixed number of resources, i.e. stretchers, doctors, nurses and boxes, are to be allocated to two zones (orange and blue). The objective is to minimize the LOS of patients in the hospital. The optimization means for addressing this problem are using a GA and NSGA-II.
- **Constrained optimization** (section 0): allocation of resource to zones of the ED. In this problem, an undefined number of resources, i.e. stretchers, doctors, nurses and boxes, are to be allocated to two zones (orange and blue). The objective is to minimize the number of resources without violating the constraint of a maximum acceptable waiting time. The optimization means for addressing this problem are using a GA and NSGA-II.

The following sections describe the problems, models and heuristics for solving the optimization problems. The results in the sections for stochastic problems are the average values for experiments with 20 replications. The detailed simulation results of each individual experiment are in Appendices 1–5.

4-3.1 Scheduling problem

The scheduling problem tackles the optimization of the order schedule in the train remanufacturing system. The objective is to minimize the average makespan of trains and the takt-time for remanufacturing of trains. The problem is known as an order-scheduling problem [148]. The order-scheduling problem is defined as a problem where *n* customer orders with different products are to be processed on *m* different machines. Each production order comprises multiple jobs to be executed on different machines. Optimization attempts to schedule the arrival of a fixed number of orders and minimize/maximum the objective function [132], [140]. Objective functions can be the individual and overall makespans of the orders to be scheduled, as well as the productivity of the production system. The problem is classified as an NP-hard problem because: a) there are no steps known to arrive to the optimum solution; b) there is an exponential number of solutions; and c) there is no algorithm that solves the problem in polynomial runtime.

In this specific case, there are two objective functions. The first objective function describes minimizing the average makespan of all customer orders, and the second objective function describes minimizing the takt time between the departure of customer orders. There are two decision variables to act on. First decision variable is the inter-arrival time (iat) describing the time between the arrival of two trains, and the second decision variable is the number of trains to be allowed in the system at the same moment (nbT). The latter parameter reflects the stock level on the shop floor. When new trains arrive according to the inter-arrival time, but the stock level is reached, arriving trains wait before entering the system until the stock level permits the system to be entered. The focus of the case study is to obtain knowledge about the impact of both decision variables. A design of experiments is defined to systematically alternate both decision variables and provide knowledge about their impact.

The simulation model describes the scheduling principle and rules of the train remanufacturing system according to the description in section 1-5.2. Arriving trains are processed according to their workplans, following the production method of a job-shop production system. For the inter-arrival times, integer values between 0 and 9 are tested, describing the time between arrivals in weeks, with the inter-arrival time of 0 describing a scenario where all trains arrive in the same moment. The types, amounts and temporal capacities of resources are fixed. Values to be tested for the inter-arrival time are integer values between 0 and 9, defining the number of weeks between the arrival of trains. Values to be tested for the allowed number of trains are integer values between 1 and 10. To understand the impact of the decision variables, a full design of experiments is applied. There are 20 trains to be scheduled. The decision variables and constraints of the optimization problem are as follows:

iat: labels the inter-arrival time, iat = 1, 2, …, 9

nbT: labels the maximum number of trains, nbt = 1, 2, …, 10

i: Index for trains, $i = 1, 2, \ldots, 20$

The constraints of the decision variables are modelled in the optimization problem. The remaining constraints of the remanufacturing process, e.g. the sequence of processes, precedence rules and conditions to start activities, are implemented in the methods for data-driven simulation. The objectives of the optimization problems are to minimize the average makespan of trains (C_{max}), given by equation (1), and to minimize the takt-time (T), given by equation (2). The inter-arrival time and takt-time are measured in days.

$$minimize\ C_{max} = \frac{\sum_{i=1}^{I} C_{max,i}}{I}\ for\ 11 \leq i \leq 20 \tag{1}$$

$$minimize\ T = \frac{\sum_{i=2}^{I} t_i - t_{i-1}}{I - 1}\ for\ 11 \leq i \leq 20 \tag{2}$$

The simulation model describes the arrival, dismantling, processing reassembly and departure of trains. The input is the decision vector with the inter-arrival time and maximum number of trains. The output is the objective vector with the makespans of the trains *i*. The full design of experiments contains 100 experiments. To consider the warmup of the model, the makespan and takt-time are measured from trains 11 to 20. Simulation is self-terminating and stops when all trains have departed from the system. A deterministic model describes the scheduling principles because the experiments were not replicated. In the design of experiments, simulation was used to evaluate all possible combinations of decision variables and provide values for the objective functions. After executing the full design of experiments, the average makespan and takt-time were calculated for all experiments. Applying the concepts of Pareto dominance provided two sets of experiments, dominated and non-dominated solutions. The set of dominated solutions describes trade-offs between makespan and takt-time. Figure 45 illustrates the dominated solutions (red) and non-dominated solutions (blue). The Pareto front links the non-dominated solutions; the experiments of the Pareto front are provided in Table 13.

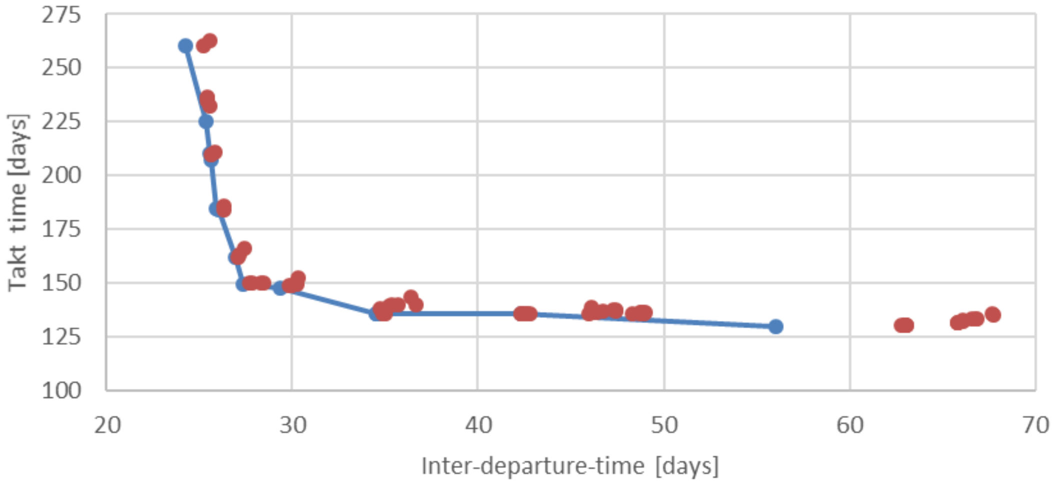


Figure 45: Optimization results for scheduling problem

Table 13: Non-dominated solutions for scheduling problem

	Action parameters		*Evaluation parameters*	
Experiment	***nbT [trains]***	***iat [days]***	***T [days]***	***C_{max} [days]***
e91	10	0	24	260
e84	9	21	25	225
e71	8	0	26	210
e74	8	21	26	208
e62	7	7	26	185
e61	7	0	26	184
e53	6	14	27	162
e75	8	28	27	150
e41	5	0	29	148
e46	5	35	35	136
e66	7	35	35	136
e76	8	35	35	136
e47	5	42	42	136
e29	3	56	56	130
e39	4	56	56	130
e49	5	56	56	130
e59	6	56	56	130
e69	7	56	56	130
e79	8	56	56	130
e89	9	56	56	130
e99	10	56	56	130

The Pareto front exposes a conflict between both evaluation parameters. In experiments with a relatively low takt time, the average makespan increases, and in experiments with relatively low makespan, the takt time increases. Minimizing both evaluation parameters at the same time is not possible. The experiments demonstrate that acceptable compromises are reachable when the inter-arrival time is equal to the takt-time, particularly for the experiments e46, e66, e76, e47, e29. In these experiments, the maximum number of trains loses impact on the evaluation parameters. In experiments with inter-arrival times greater than five weeks (35 days), there is no more waiting at the entrance, owing to the maximum number of trains.

4-3.2 Resource allocation problem

The resource allocation problem tackles the assignment of stretchers, doctors, nurses and boxes to zones of an ED. The objective is to minimize the LOS of the patients. The problem is known as a resource allocation problem, which is a problem where limited resources are to be distributed among competing demands to minimize or maximize an objective function (Katoh and Ibaraki, 1998). Owing to the domain, the literature considers the problem more precisely as a resource allocation problem in healthcare (Blake and Carter, 2002; Vlah Jerić and Figueira, 2012). The problem is a combinatorial optimization problem and attempts to find the best combination for the assignment. The problem is classified as an NP-hard problem [149]. There are multiple reasons for this classification: a) there are no steps known to arrive at the optimum solution; b) there is an exponential number of solutions; and c) there is no algorithm that solves the problem in polynomial runtime. Owing to its classification as NP-hard, metaheuristics are appropriate for optimization. However, without testing all possible solutions, there is no guarantee that the optimum will be found.

The optimization problem addresses the allocation of a fixed number of resources: stretchers, doctors, nurses and boxes to zones. Zones are spatial and organizational areas within the ED. In this particular case, there are three objective functions. The objectives are to minimize the LOS of orange patients, blue patients and the patient mix. For solving optimization problems with multi-objectives, the literature proposes two strategies. In [140], [150], it is recommend to reformulate multiple objective functions into one objective function by using weights, and to apply metaheuristics, e.g. with a GA to provide an optimum. In [132], it is recommended to formulate multiple objective functions and apply Pareto optimization, e.g. with a non-dominated GA, to explore the entire Pareto front of non-dominated solutions. In this section, the optimization problem is tackled by both approaches. The purpose is to understand the benefits and drawbacks for application in problem-solving. A comparison and insights are presented in the discussion.

The resources consist of 30 stretchers, 6 doctors, 4 nurses, and 15 boxes, which are assigned to the two zones of the ED. One zone is available for the orange patients, whose conditions are more severe, and one zone for the blue patients, whose conditions are less severe. The constraints for assigning the patients to the zones are inherited in the methods for data-driven simulation. During the observation time of one year, 35 240 patients arrive sequentially. Details about the system behaviour, e.g. patient arrivals, prioritization, etc., are modelled within the simulation model. The list of system parameters is as follows:

z: labels the zones, z = 1, 2, …, Z
Z: Number of zones

k: Index for resources, k = 1, 2, …, K
List of resources: M_R = {'stretchers', 'doctors', 'nurses', 'boxes'}

l: Index of units at resource r_k, l = 1, 2, …, L_k
L_k: Number of elements in the kth set of resources

i: Index for patients, i ∈ {1, 2, …, I}
I: Number of patients

The decision variables describe the decisions of the optimization problem. In equation (3), the decision variable x_{kz} determines the number of units of each resource k that is assigned to a zone z. In addition, meta-constraints describe the relations between the decision variables. The sum of units of each resource per zone must be equal to the number of units of each resource (4), and each resource must be represented in each zone with at least one unit (5). Exemplary for the doctors, each doctor must be assigned to one of the two zones (4), and there has to be at least one doctor in each zone (5). The constraints of the decision variables are as follows:

$$x_{k,z} = 1, 2, \dots, L_k \tag{3}$$

$$\sum_{z=1}^{Z} x_{k,z} = L_k, k = 1, 2, \dots, K \tag{4}$$

$$x_{k,z} \geq 1, \ z = 1, 2, \dots Z, k = 1, 2, \dots, K \tag{5}$$

The objective functions use the LOS of the patients ($C_{max,i}$) to describe the average LOS for orange (C_{orange}) and blue (C_{blue}) patients in (6) and (7). Equation (8) describes the average LOS of the patient mix (C_{mix}). All objective functions are to be minimized. The LOS of the patients are provided by simulation. Minimization is performed while changing values of $x_{k,z}$; the remaining system variables are considered as static. The decision variables o_i and b_i in (9) and (10) support calculating the LOS for orange and blue patients.

$$minimize\ C_{orange} = \frac{\sum_{i=1}^{I} o_i * C_{max,i}}{\sum_{j=1}^{I} o_j} \tag{6}$$

$$minimize\ C_{blue} = \frac{\sum_{i=1}^{I} b_i * C_{max,i}}{\sum_{j=1}^{I} b_j} \tag{7}$$

$$minimize\ C_{mix} = \sum_{i=1}^{I} \frac{C_{max,i}}{I} \tag{8}$$

$$o_i = \begin{cases} 1, & if\ severity\ e_i\ of\ patient\ p_i\ is\ \text{'orange'} \\ 0, & otherwise \end{cases} \tag{9}$$

$$b_i = \begin{cases} 1, & if\ severity\ e_i\ of\ patient\ p_i\ is\ \text{'blue'} \\ 0, & otherwise \end{cases} \tag{10}$$

The simulation model describes the arrival, treatment and departure of patients. After arrival, each patient follows an individual pathway. During the pathway, patients are assigned to stretchers and zones and are treated by the resources. The input is the decision vector with the assignment of resources to zones. The output is the objective vector with the LOS of the patients *i*. For optimization, two metaheuristics, a GA and NSGA-II with open source libraries [141] are tested. The objective function for the GA is the length of stay for the patient mix, compromising orange and blue patients. The objective functions for NSGA-II are the LOS for orange and blue patients as well as the patient mix. A warmup period was not used, since no extended transient behaviour was observed and empty states occurred regularly during the night. To consider uncertainties in the arrivals and pathways of patients, experiments were replicated 20 times. The results are provided in Table 14 and Table 15.

Table 14: Optimization results for resource allocation problem with GA

	Action parameters				*Evaluation parameters*		
exp	*Stretchers (o/b)*	*Doctors (o/b)*	*Nurses (o/b)*	*Boxes (o/b)*	*LOS(o) [min]*	*LOS(b) [min]*	*LOS(mix) [min]*
e0	14/16	3/3	1/3	4/10	-	-	161

In Table 14, the GA provides one experiment with the minimum LOS for the patient mix. Weighting the objective functions reduces the two objective functions for orange and blue patients to one objective function for the mix. The reduction enables the ranking of solution candidates and provides one minimum solution candidate. This solution candidate does not match with the proposed solution of the future situation (S2) for the ED that was to assign 20 stretchers, 4 doctors, 2 nurses and 10 boxes to the orange zone, and 10 stretchers, 2 doctors, 2 nurses and 2 boxes to the blue zone. At least for the objective of improving the LOS of the patient mix the planned, future situation does not provide the minimum.

Table 15: Optimization results for resource allocation problem with NSGA-II

	Action parameters				*Evaluation parameters*		
exp	*Stretchers (o/b)*	*Doctors (o/b)*	*Nurses (o/b)*	*Boxes (o/b)*	*LOS(o) [min]*	*LOS(b) [min]*	*LOS(mix) [min]*
e0	22/8	4/2	1/3	11/4	239	147	178

e1	12/18	1/5	2/2	2/13	752	104	326
e2	14/16	3/3	2/2	10/5	240	120	161
e3	7/23	2/4	1/3	1/14	453	106	225
e4	7/23	3/3	1/3	1/14	445	107	222
e5	6/24	2/4	1/3	4/11	612	105	279
e6	6/24	2/4	1/3	3/12	612	105	279
e7	20/10	4/2	1/3	11/4	239	134	170
e8	8/22	2/4	1/3	8/7	288	111	171
e9	7/23	2/4	2/2	2/13	339	108	187
e10	7/23	2/4	1/3	2/13	339	108	187
e11	7/23	3/3	1/3	3/12	327	109	184
e12	9/21	2/4	1/3	11/4	268	113	166
e13	19/11	4/2	1/3	11/4	239	131	168
e14	12/18	2/4	2/2	3/12	249	117	162
e15	18/12	3/3	1/3	11/4	239	123	163
e16	16/14	4/2	1/3	11/4	239	128	166
e17	10/20	2/4	2/2	2/13	260	115	164
e18	19/11	3/3	1/3	11/4	239	125	164
e19	11/19	2/4	2/2	2/13	254	116	163

In Table 15, NSGA-II provides multiple experiments with the minimum LOS for the orange and blue patients and the patient mix. Using multiple objective functions in Pareto optimization enables multiple non-dominated solution candidates on the Pareto front. The simulation results from section 0 provide average LOS values of 241, 132, and 168 min for orange and blue patients and the patient mix, respectively. Comparing these values with the optimization results shows that there are configurations that perform better for each of the three objective functions, particularly e15, e16 and e18. Comparison shows that the planned future situation (S2) is a dominated solution candidate that is not on the Pareto front. Optimization provides three experiments that perform better for each of the stated objective functions as S2.

4-3.3 Constrained optimization problem

The optimization problem tackles the assignment of stretchers, doctors, nurses and boxes to zones of an ED. In this problem, the number of resources is not fixed. The objective is to minimize the resources that are assigned to each zone. Additionally, there are constraints on the objective functions. Constraints define the acceptable

LOS. These constraints are to be maintained while minimizing the resources. Again, as in section 4-3.1, the problem is classified a resource allocation problem. However, constraints on the LOS classify the problem as a constrained optimization problem, which focus on minimizing or maximizing an objective function with constraints [151]. Constraints are not linked to specific optimization problems. Moreover, constraints can occur in different optimization problems, for example, resource allocation [152]. Based on the criteria of section 4-3.1, the problem is classified as a multi-objective NP-hard problem.

The optimization problem addresses minimizing the number of stretchers, doctors, nurses and boxes, that are assigned to each zone. Similarly, the objective function describes the number of stretchers, doctors, nurses and boxes that are assigned to each zone. Decision variables and objective functions are identical to those in the previous problem. The constraints define the acceptable LOS for orange and blue patients as well as the patient mix. The problem is a multi-objective problem. Analogous to section 4-3.1, two strategies are appropriate to optimize the multi-objective problem. The weighted sum method reformulates the objective functions and introduces weights [140]; Pareto optimization is applied for multiple objectives to provide a set of non-dominated solutions [132]. Both strategies are tested and compared in this section. To ensure that the optimization constraints are upheld, two strategies are available. The introduction of penalty functions allows the fitness of a solution candidate to be degraded, if the constraints are not maintained and unacceptable solutions are given [151]. Another approach is to include constraints along with the decision variables and objective functions as an additional entity in the optimization algorithm [141]. The insights from the case study are presented in the discussion (section 4-4).

As resources, 60 stretchers, 12 doctors, 8 nurses, and 30 boxes are available for both zones of the ED. One zone is available for the orange patients, whose cases are more severe, and one zone is for the blue patients, whose cases are less severe. The constraints for assigning the patients to the zones are inherited in the methods for data-driven simulation. During the observation time of one year, 35240 patients arrived sequentially. Details about the system behaviour, e.g. patient arrivals, prioritization, etc., are modelled within the simulation model. The following is the list of system parameters:

z: labels the zones, z = 1,…, Z
Z: Number of zones

k: Index for resources, k = 1, 2, …, K
List of resources: M_R = {'stretchers', 'doctors', 'nurses', 'boxes'}

l: Index of units at resource r_k, l = 1, 2, …, L_k
L_k: Number of elements in the k^th set of resources

i: Index for patients, i ∈ {1, 2, …, I}
I: Number of patients

C_{Crit}: Threshold for length of stay

The decision variables describe the decisions of the optimization problem. In (11) the decision variable x_{kz} determines the number of units of each resource k in a zone z. Additionally, the meta-constraints describe the relationship between the decision variables. The meta-constraints in (12) ensure that each resource is represented in each zone with at least one unit. Exemplary for the doctors, there is at least one doctor in each zone. Equation (13) describes the constraint of the acceptable LOS (C_{crit}) values, which are 132, 241, and 168 min for the blue and orange patients and the patient mix, respectively. The origin of the constraints are the simulation results for the planned future situation in section 0. The purpose is to understand if the same result is achievable with fewer stretchers, doctors, nurses and boxes. The following are the constraints of the decision variables:

$$x_{k,z} = 1, 2, \dots, L_k \;, x_{k,z} = 1, 2, \dots, \frac{L_k}{2} \tag{11}$$

$$x_{kz} \geq 1 \; for\; each\; z = 1, 2, \dots, Z \; and\; each\; k = 1, 2, \dots, K \tag{12}$$

$$\sum_{i=1}^{I} \frac{C_{max,i}}{I} \leq C_{crit} \tag{13}$$

The objective function describes the minimization of the number of resources that are assigned to both zones. In equation (14), *x* represents the vector of decision variables $x_{k,z}$. Minimization is performed with respect to $x_{k,z}$. While minimizing the sum of all resources that are assigned to all zones, the remaining system variables do not move. For optimization, the constraint is the critical waiting time.

$$\begin{aligned} &minimize\ f(x) = \sum_{k=1}^{K} \sum_{z=1}^{Z} x_{k,z} \\ &with\ the\ constraint\colon \sum_{i=1}^{I} \frac{C_{max,i}}{I} \leq C_{crit} \end{aligned} \tag{14}$$

The simulation model describes the arrival, treatment and departure of patients. After arrival, each patient follows an individual pathway. During the pathway, patients are assigned to stretchers and zones and are treated by the resources. The input is the decision vector with the assignment of resources to zones. The output is the objective vector with the LOS of the patients *i*. For optimization two metaheuristics, a GA and NSGA-II with open source libraries [141] are tested. The objective function for the GA is the sum of the number of resources that is available in each zone. The objective functions for NSGA-II are the number of resources that are assigned to each zone. A warmup period was not used, since no extended transient behaviour was observed and empty states occurred regularly during the night. To consider uncertainties, experiments were replicated 20 times. Table 16 and Table 17 provide optimization results of the GA and NSGA-II.

Table 16: Optimization results for constraint optimization problem (GA)

	Action parameters & Evaluation parameters				***Values for constraints***		
exp	***Stretcher (o/b)***	***Doctors (o/b)***	***Nurses (o/b)***	***Boxes (o/b)***	***LOS(o)***	***LOS(b)***	***LOS(m)***
e0	19/11	4/3	1/2	3/3	239	*125*	*164*

Table 16 provides the optimization results of the GA with the minimum number of resources. Summing up the number of each resource reduces the eight objective functions to one objective function. Providing one objective function enables the solution candidates to be ranked and provides one minimum. The minimum solution candidates have LOS values of 239, 125, and 164 min for orange, blue and mixed patients, respectively, with the resources of e0. The results are better than in the future situation with values of 241, 132, and 168 min, respectively (section 0). In the experiment, the number of stretchers, nurses and boxes was smaller than in the future situation; however, an additional doctor was required. The decision-maker must decide if the minimum from the optimization represents a preferable solution.

Table 17: Optimization results for constraint optimization problem (NSGA-II)

	Action parameters & Evaluation parameters				*Values for constraints*		
exp	***Stretcher (o/b)***	***Doctors (o/b)***	***Nurses (o/b)***	***Boxes (o/b)***	***LOS(o)***	***LOS(b)***	***LOS(m)***
e0	16/10	4/4	2/1	4/4	238	126	164
e1	16/11	3/4	2/1	3/3	239	123	163
e2	16/12	4/3	2/1	4/4	238	122	162
e3	15/13	4/4	2/1	3/4	239	121	161
e4	16/11	3/3	2/1	5/4	239	124	163
e5	17/10	4/4	2/1	3/5	239	126	164
e6	15/11	4/4	2/1	3/5	239	123	163
e7	17/10	3/4	2/1	4/3	239	126	165
e8	14/10	4/4	2/1	5/4	239	125	164
e9	15/11	3/4	2/1	4/3	239	123	163
e10	17/13	4/2	2/1	4/3	238	128	166
e11	14/11	4/3	2/1	5/4	239	124	163

In Table 17, the NSGA-II provides multiple experiments with the minimum number of stretchers, doctors, nurses and boxes in the blue and orange zone. Using multiple objective functions in Pareto optimization enable multiple non-dominated solution candidates on the Pareto front. However, in the optimization, only 12 non-dominated solution candidates were identified. Following the concept of the Pareto optimum, there is no best solution. However, the results show that there are two experiments, e4 and e10, that use fewer resources than the future situation S2 from section 0 and provide better performance (see Table 18).

Table 18: Comparison of S2, e4, and e10

exp	Stretchers (o/b)	Doctors (o/b)	Nurses (o/b)	Boxes (o/b)	LOS(o) [min]	LOS(b) [min]	LOS(mix) [min]
S2	20/10	4/2	2/2	10/5	241	138	168
e4	16/11	3/3	2/1	5/4	241	126	165
e10	17/13	4/2	2/1	4/3	240	131	168

4-4 Discussion

The leading question of this chapter is RQ2: **What are the methods for optimizing material flows through the digital shadow and data-driven simulation?** For addressing this question, simulation data (Chapter 1), libraries of partial models and methods for data-driven simulation (Chapter 2) are assumed to be available. The method to tackle optimization problems is implementation of data-driven simulation from data of the digital shadow in the method of simulation-based optimization. Simulation and optimization provide a generic architecture consisting of optimization algorithms and simulation models [132], where the exchange of action and evaluation parameters realize the interface between both. However, in the proposed method of ***data-driven simulation and optimization***, the simulation model is represented by the library of case-specific models, simulation data and the methods for data-driven simulation (see section 3-2). The interface between optimization algorithms and simulation models is realized via the simulation data. Values for action parameters are edited into the simulation data. For each new set of action parameters, data-driven modelling provides an individual model, runs simulation experiments and provides evaluation parameters to the optimization algorithm.

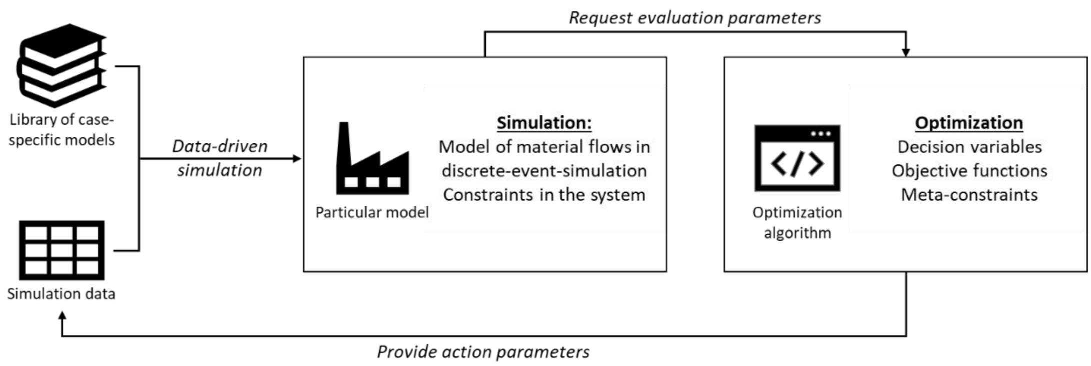


Figure 46: Data-driven simulation and optimization

The generic architecture of simulation and optimization enables optimization for a wide range of problems by using simulation in parametric models for evaluating the objective function. The proposed method of data-driven simulation and optimization enhances the decision space of the optimization problem. By writing action parameters in the simulation data and instantiating models from libraries, non-parametric changes in the model can be evaluated, e.g. changing the product mix, adding and removing resources and changing the organization of work. Examples will be presented in the case-study of the ED in Chapter 5, in which new assignments of stretchers to patients and the changes in the assignment of processes to activities are described. Both represent qualitative aspects of system control.

The method of data-driven simulation brings added value to optimization by generating an increased amount of genericity. The methods independent replacement of the simulation model of the case, optimization problem to be solved, and the optimization algorithm. This degree of freedom enables a wide range of optimization problems to be solved. The details of substituting simulation models, optimization problems and optimization algorithms is described in the following paragraphs.

Replacing the simulation model enables optimization problems to be addressed in different systems. In this chapter, optimization for three problems in two case-studies is presented, a *train remanufacturing system* and an *emergency department*. The models of the two systems were substituted by replacing the models of the case-specific library and simulation data. The algorithm for data-driven simulation and the models of the partial library remained consistent for both problems. To address additional cases, the derivation of new models in the case-specific library and the replacement of simulation data are required. The limitations are the partial models that are designed for simulating material flows in job-shop production systems.

Replacing optimization problems enables different optimization problems to be addressed. The problems in this chapter belong to three general groups: *scheduling problems*, *resource* allocation problems, and *constraint optimization problems*. The replacement of optimization problems is enabled by using generic simulation models that simulate experiments for different values of the decision variables and provide results for the objective functions. The required activities to replace the problem are defining the decision variables, objective functions and meta-constraints of decision variables and objective functions. The remaining constraints of the system are implemented within the simulation model to avoid the formulation of mathematical models and reformulation when switching between different optimization problems. An additional problem addressed in [139] provides an application and describes using a GA to optimize for a single-objective sequencing problem.

Replacing the optimization algorithm allows substituting the algorithm with different optimization means. In this work, what-if scenarios (section 0), design of experiments (section 4-3.1), and optimization (section 0 and 0) with GA and NSGA-II were tested. Benefits were experienced when using NSGA-II, especially for multi-objective problems. The algorithm provides non-dominated solutions on the Pareto front, representing the trade-offs between multiple objective functions, different from GAs and design of experiments. The GA provides the optimized solution for the weighted objective functions, and design of experiments provides the entire set of solutions, including dominated solutions; non-dominated solutions are to be identified manually.

For optimizing the ED, two optimization algorithms, GAs and NSGA-II were compared. Limitations of the GA were encountered when formulating the objective function. Weighting multiple objectives to arrive at a single objective function is not possible without expert knowledge. The NSGA-II enabled optimization for multiple objectives without weighting and provided a set of non-dominated solutions representing trade-offs between the objective functions. However, by using weights, the GA creates a ranking and provides the best solution, while the NSGA-II provides a set of non-dominated solutions and requires expert knowledge to choose a solution for application. Compared to the design of experiments, the NSGA-II provides solutions exclusively on the Pareto front. In the case-study of the scheduling problem, the solutions were manually classified as dominated and non-dominated. For optimizing complex systems, manual classification can become difficult and time-consuming, while Pareto optimization provides only the set of non-dominated solutions.

The train remanufacturing system was modelled in parallel with mixed-integer linear programming [153]. The objective of this model was to tackle two problems in optimization: the scheduling of production tasks, and minimizing the resources on the shop floor. New insights were retrieved by comparing optimization with mixed-integer linear programming and simulation. Providing a mathematical model to evaluate the objective function is a manual activity with high requirements to the system modelers, particularly when modelling complex systems in a dynamic environment while considering uncertainties. Data-driven simulation provides values for the objective functions automatically on request, based on the data from the digital shadow. However, limitations are linked to the high requirements for computing power for simulation. The mathematical model was able to provide the performance of solution candidates instantly, while simulation of complex systems requires multiple seconds. Consequently, limitations in terms of the search space occurcur.

Chapter 5 Problem-Solving

This chapter is dedicated to the question of model change in context of the redesign of manufacturing systems (RQ3): **What are the methods for solving problems in design of material flows by the digital shadow and data-driven simulation?** This chapter explains how the data-driven simulation from Chapter 3 and optimization from Chapter 4 can be applied to solve problems in design by changing the model (see Figure 47). To address this question, this chapter provides the background of inventive design (section 0), provides the methods to support model change (section 0), applies the methods in the case-study to solve a design problem (section 0), and provides a discussion of the methods and insights gained in the case-studies (section 0).

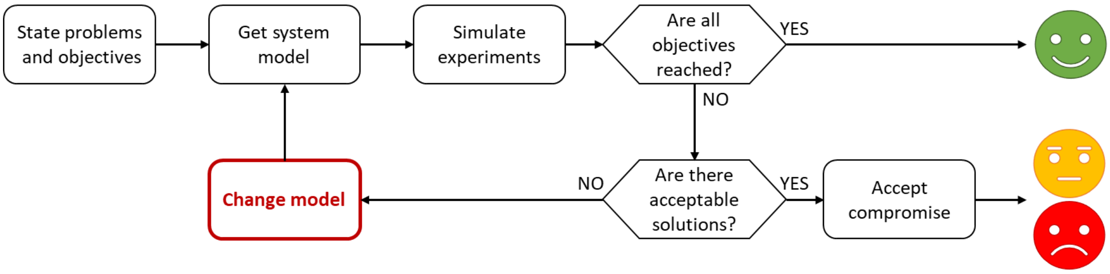


Figure 47: Changing the model in problem-solving

The background provides an overview of the methods of inventive design and shows how inventive design uses the concept of model change to improve systems when optimization cannot achieve the objectives. The methods of this work use the results from optimization to solve problems by highlighting the system limitations, extracting contradictions, and changing the model. The presented case study applies the methods to solve problems in the design of the ED. The case-study highlights the added value of applying data-driven simulation (Chapter 3) and optimization (Chapter 4) in problem-solving and shows how the methods of inventive design can address problems that are not solvable by simulation.

5-1 Background

In the problem-solving process illustrated in Figure 47, two approaches are considered separately or combined to solve problems and create (new) solution concepts: *optimization* and *inventive design* [3]. Optimization increases the system efficiency by optimizing the system parameters. Inventive design introduces new parameters during the design process. As stated in section 1.1.1, optimization is the first step of problem-solving. Experimental data from optimization are required to change the model by inventive design methods. To provide the background, section 0 describes the methods for retrieving experimental data for problem-solving, and section 0 describes the methods to exploit experimental data and provide model change. The methods from state of the art will be applied in the case study to solve problems in design.

5-1.1 Provide experimental data

In the literature, the problem-solving process of Figure 47 is applied in multiple works in different domains. Optimization has the purpose of generating data with a link between action and the evaluation parameters. In inventive design, the data are used to formulate contradictions and solve the problem. Optimization and execution of experiments is crucial for problem-solving. The purpose is to evaluate alternative solution concepts and provide data about the link between the decision vector and the objective vector in the optimization problem. To obtain this knowledge, a wide range of techniques can be applied.

In the domain of train design, the authors in [154] use expert interviews to evaluate alternative configurations of trains, represented by three action parameters versus ten evaluation parameters. The authors use the gathered data to identify multiple technical contradictions in the system. Despite recognizing the problem as not solvable, they are able to state the problem by using expert interviews providing the evaluation parameters for different solution candidates.

In the domain of electrical circuit breakers, the authors in [14], [155]–[157] use a design of experiment with physical prototypes to understand the impact of five action parameters on six evaluation parameters. From the experimental data they provide the technical and physical contradictions of the system. To solve the problem, the authors use the TRIZ-principle separation in space and provide a solution concept [14].

In the domain of lattice structures, the authors in [158], [159] extract the link between action and evaluation parameters from scientific research papers. For the extraction of action and evaluation parameters, they use algorithms for natural language processing and formulate a problem-graph. From the action of evaluation parameters, they extract the technical and physical contradictions and solve a design problem having a lattice structure.

In the domain of cutting processes, the authors in [150] use a design of experiments with physical prototypes to understand the impact of three action parameters on two evaluation parameters. The authors provide a mathematical model to describe the link between action and evaluation parameters and use optimization with a GA and the weighted-sum method to provide a solution candidate for the multi-objective optimization problem. However, the results of the optimization are not able to solve the design problem. Therefore, the authors propose the use of TRIZ principles.

In the domain of cutting tools, the authors in [7] use a design of experiments with physical prototypes to understand the impact of five action parameters on six evaluation parameters. From the experimental data they can highlight, that there is no solution that satisfies the design problem. Based on the experimental data, they describe the system of contradictions and provide a new solution concept by using the TRIZ principle of separation.

In the domain of warehousing, the authors in [4], [15] use simulation of material flows to understand the link between action and evaluation parameters. To define simulation experiments, they use a design of experiments. Experimentation in virtual models allows them to execute a large number of experiments in a short time. With the experimental data, they provide the Pareto front and describe the system of contradictions. Based on the reformulated problem, the authors apply methods of TRIZ to redesign the material flows and provide new solution concepts.

The analysis of previous works shows that to obtain experimental data, different approaches can be used: expert interviews, experiments in physical prototypes and experiments in digital models. For analysing complex and large-scale systems with a big decision space, expert interviews and prototypes have limitations. Digital models and simulation can be used to analyse a large number of solution concepts without harming the system to be designed. To define the action parameters in optimization, the studies use design of experiments and what-if scenarios. These studies did not use algorithmic optimization to provide experimental data. Therefore, is should be determined by testing if optimization (particularly, multi-objective optimization) can provide the relevant data to extract contradictions and support model change.

5-1.2 Bridge between optimization and model change

Inventive design addresses the overcoming of system limitations by providing a qualitative change of the model or the system such that it fits the solution requirements. The actual model change is a creative process. In contrast, optimization provides solution candidates by quantitative changes of action parameters. The focus of inventive design is on exploiting experimental results and generating knowledge about critical systems. The literature provides methods to support this process. Based on the simulation results from the design of experiments in the warehouse, a dominance analysis is used in [4] to classify the experiments as dominated and non-dominated. Based on the concept of Pareto dominance, an experiment is better (non-dominated experiment) than another (dominated experiment) if it has at least as good performance for all objectives and is more successful for at least one objective. The set of all non-dominated solutions describes the Pareto front, which represents the best compromise solution, provides choices for the decision-maker and represents various trade-offs between goals.

When the objective is not attainable with optimization, model change can be applied to solve the inventive problem. In [155], to bridge the gap from detecting the Pareto front to model change, a system of contradiction is formulated to define the problem, as illustrated in Figure 48 (based on [16]). For describing the system of contradictions, the authors first identify the paired technical contradiction by finding a situation where improvement of one aspect of a system leads to degradation of another aspect, and vice versa. After identifying the paired technical contradiction, a physical contradiction is identified within which one or multiple action parameter(s) must take on different values at the same moment. The system of contradictions is then formulated. The concept of the system of contradictions is part of OTSM-TRIZ, which is presented in [16]. The figure illustrates a 'classical' contradiction and a generalized contradiction. The classical contradictory situation describes a conflict between two evaluation parameters that is caused by a conflict with one action parameter. A paired generalized technical contradiction describes a conflict between two solution concepts with multiple evaluation parameters. The origin, the generalized physical contradiction, is expressed by a set of action parameters.

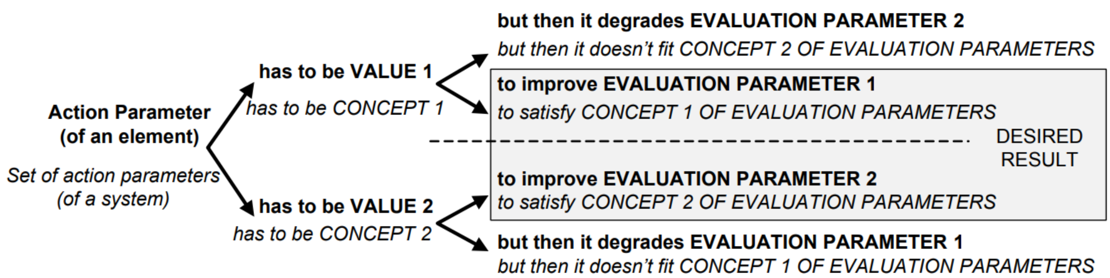


Figure 48: System of contradictions

As initially stated in section 0, the TRIZ inventive design method solves problems through the resolution of contradictions. To resolve contradictions, the first step of TRIZ is to describe the problem through paired technical contradictions and the underlying physical contradictions. The second step is to apply inventive principles to solve the problem by resolving the contradiction. Solving the contradiction remains a creative task. The exact description of the contradiction is crucial for attaining its resolution. For describing the system of contradictions, the authors in [7] use the concept of Pareto dominance to separate experiments with dominated and non-dominated solutions. Within the set of non-dominated solutions, they use binarization to describe the satisfaction of evaluation parameters and identify the technical contradictions. From the technical contradictions, they use discriminatory analysis to identify the action parameters that are involved in the conflict and provide the physical contradiction.

Algorithms have been developed to automate the process of extracting technical and physical contradictions. For identification of contradictions, the authors in [157] and [156] provide algorithms to extract the technical and physical contradictions, respectively. Additional work uses machine learning techniques to provide contradiction from data [160]. The work provides methods to explore contradictions from experimental results by detecting technical and physical contradictions to support the formulation of system of contradictions. For validation, the methods are tested in a set of test functions. In [161], the authors present an algorithm to rank the action parameters by their impact on the contractionary situation. Challenges remain, particularly in providing data about system behaviour to formulate the system of contradictions and the bridge between optimization and model change.

5-2 Methods for problem-Solving

The purpose of this section is to provide the methods to support model change in the problem-solving loop shown in Figure 47. The initial situation is the availability of the experimental data from optimization and having the data and methods to execute data-driven simulation for generating additional experimental data. To bridge from optimization to model change, this section presents methods to highlight the system limitation as a reference (section 0), the methods to reformulate the problem for providing the system of contradictions (section 0) and to change the model for solving the problem (section 0).

5-2.1 Define solution space

The starting point for inventive design is the experimental results from simulation and optimization. This experimental data provides information about the performance of the solution candidates, but does not provide data about the ideal fitness of the system because when analysing the experiment results one cannot determine if the target value being approached in optimization is reachable. To tackle this problem, this work presents methods to define the solution space. For illustration, Figure 49 depicts the solution space for an exemplary optimization problem with two objectives, f_1 and f_2, for the makespan of two products.

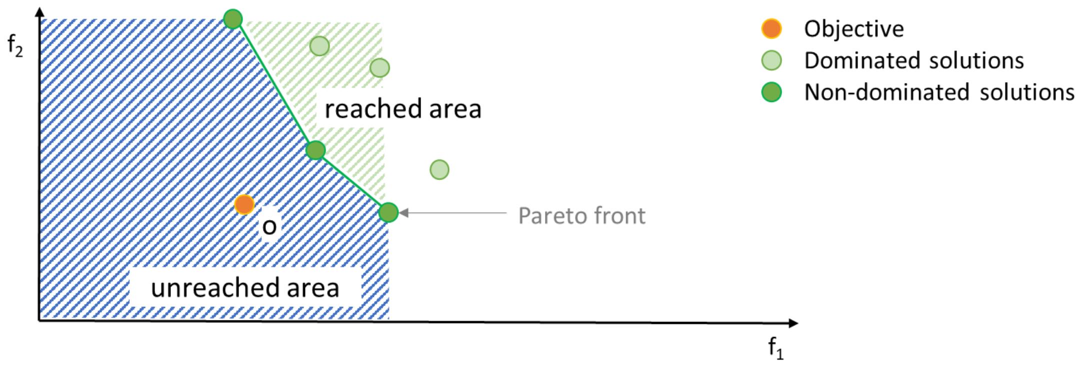


Figure 49: Definition of solution space

The solution space is divided by the Pareto front into two subspaces: a green space hosting the dominated and non-dominated solution candidates, and an empty blue space. The green space, the *reached area*, describes the solution space that was made accessible through optimization by parameterizing the decision variables. After optimization, the decision-maker has solution candidates available to reach points within the *reached area*. When the objective (e.g. o) is not reached by optimization, it is located within the blue area, the *unreached area*. In this case, the decision-maker has no knowledge about the accessibility to this point of the solution space, only that the objective is not reachable by optimization of the present model of the system.

To tackle the question of accessibility of objectives, this work proposes the concept of *ideality*. Ideality is a virtual system state in which the materials flow according to the process without any restriction by the physical system. This state cannot be reached in reality. In the ideal queuing network, there are no waiting lines. Consequently, the performance of the ideal system is determined by the modelled process, particularly the routing and the processing times. The properties of the physical system do not degrade the performance of ideality. To provide performance measures for the ideal system, this work uses the methods of data-driven simulation. For simulating ideality, the system is modelled as ideal by defining the number of resources as infinite. This means providing a large number of resources without restrictions in availability. In ideal system, there is no competition for resources, there are no waiting lines, and the performance is exclusively driven by the underlying process. The impact of providing the measure for ideality is illustrated Figure 50.

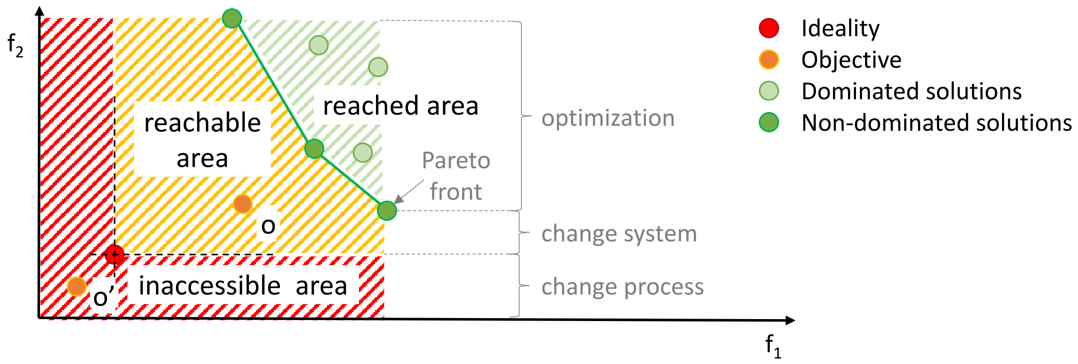


Figure 50: Illustration of the concept of ideality

The performance of the ideal system (red point) sub-divides the blue area of the unreached space in Figure 49 into two subspaces in orange and red in Figure 50. The orange space, the *reachable area* between ideality and the Pareto front, describes the part of the initially unreached area that might be achieved by changing the model. Changing the model refers to changes in the system configuration and organization, e.g. by adding resources. The impact of model change ends at the point where all resources are 'infinite' and no waiting lines exist. At this point, the system arrives at the concept of ideality and the performance is equal to the reference that was introduced in section 0. The red space, the *inaccessible area*, is due to fixed values of system parameters whose modification is not accessible or allowed to the designer (e.g. cycles times of machines or of a given activity, cycle times for subcontracted activities). To access the unreachable area, the decision-maker needs to change the underlying process that defines the reference. The reference is defined for each product by the sequence of activities and the activity durations. It is unique for each variant of routing (section 0) and assumed as static.

In the exemplary figure, the green data points are dominated and non-dominated solution candidates of the optimization. The objective o is reachable by changing the model of the physical system and o′ requires changing the model of the process. Changing the model of the system aims to minimize the waiting times (reachable area) and is limited by ideality. To go beyond ideality and access the inaccessible area, changing the process is required. Transferred to the case of the ED, the green points are solution candidates that were reached by optimizing the assignment of stretchers, doctors, nurses and boxes. The reachable area requires changing the model of the system, e.g. by adding additional resources or changing the prioritization of patients. The inaccessible area requires changes in the process that are not yet allowed to the designer (e.g. removing the activity triage and sending all patients directly to the doctor, reducing the cycle time for blood samples analysis by sending each sample through an automated transport system instead of a human bringing them by batches to the laboratory). The last changes would impact the reference of ideality.

The decision-maker uses the simulation results and methods from simulation to analyse the solution space. The Pareto front from optimization and the reference from simulation of the ideal system are used to understand if the objectives are within the reachable area and are feasible by changing the system model, or if the objective requires a change of the process. With the received knowledge, a decision is made to either accept a compromise and stop problem-solving or to change the model and overcome the limitations of the physical system by inventive design.

5-2.2 Contradictions as bridge between optimization and inventive design

The starting point of model change is the experimental results from optimization. To bridge the gap between optimization and changing the model, there are three steps to be executed: extracting the technical contradiction (TC), extracting the physical contradiction (PC), and providing the system of contradictions. The first two steps, i.e. extraction of TC and PC, are illustrated in Figure 51, and an exemplary system of contradictions is provided in Figure 52. Before the extraction of contradictions, there are two steps required to extract the table from the simulation results. First, the concepts of dominance and Pareto optimality are used to identify non-dominated solutions on the Pareto front. Second, for the non-dominated solutions, a binarization is executed. To binarize, the quantified objectives for problem-solving are required. For each non-dominated solution candidate and each objective, it is checked whether the objective is reached. The result of this binarization is illustrated in (a), where the values 0/1 indicate the (un)satisfaction of objectives.

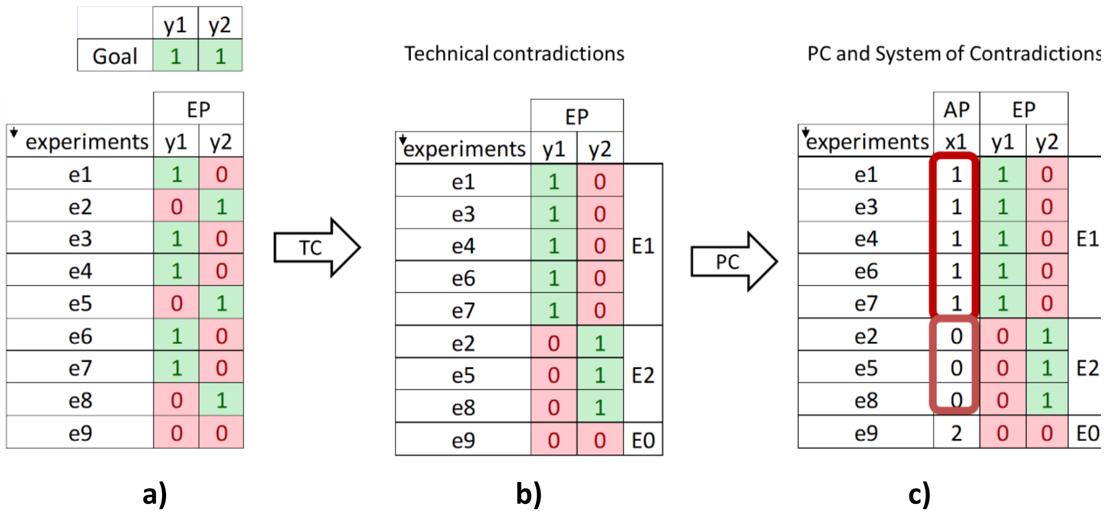


Figure 51: Steps of contradiction detection

The initial point in (a) for detecting technical contradictions in the exemplary illustration are the binarized experiment results, particularly the satisfaction of evaluation parameters y1 and y2 in nine experiments. To detect the technical contradiction, the decision-maker extracts two solution concepts, that together satisfy all objectives y_j of the design problem. In (b), solution concepts e1, e3, e4, e6, e7 of the group E1 satisfy the objective y1 and the solution concepts e2, e5, e8 of the group E2 satisfy objective y2. Solution concept e9 satisfies neither y1 nor y2. There is a paired technical contradiction between the experiments of E1 and E2. The paired technical contradiction is the initial point for detecting physical contradictions. The physical contradiction is a conflicting situation in one decision variable and is the origin of the paired technical contradiction. Detecting a physical contradiction includes analysing the corresponding action parameters. To detect the physical contradiction, a discriminatory analysis is performed. The analysis provides the action parameters x_i

that are origin for satisfaction of evaluation y_j. In (c), the value 1 for the action parameter x1 satisfies y1, and the value 0 for x1 satisfies y2. There is a physical contradiction; the action parameter x1 value should be **0 and 1** to satisfy both objectives. Identifying technical and physical contradictions provides the system of contradictions, as illustrated in Figure 52. The system of contradictions provides a comprehensive understanding of the problem and the conflicting requirements. By recognizing the contradictions, the decision-maker can explore innovative solutions and overcome trade-offs of the Pareto front. Additionally, further action parameters can provide the context of the contradiction. In the illustrated case with one action parameter x1, there is no context available for the contradiction.

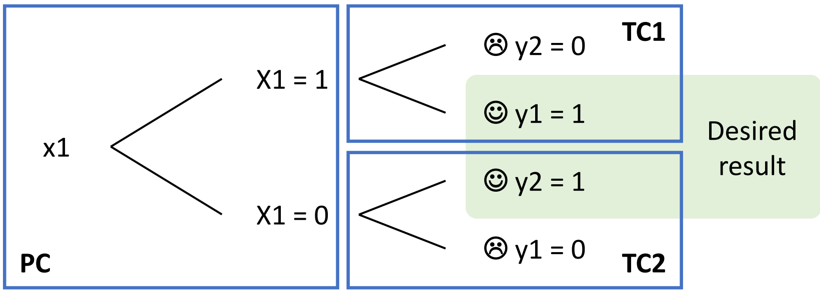
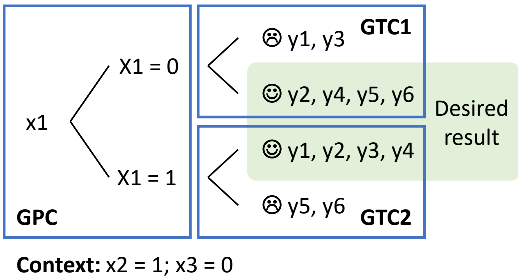


Figure 52: System of contradictions

As mentioned previously, the concept of the TRIZ system of contradiction is generalized when dealing with more than two evaluation parameters and when more than two action parameters might be involved in the conflict [16]. Let us illustrate it by the example shown in Figure 53. In this example, six evaluation parameters (EP) and five action parameters (AP) are involved in the problem (see (a)). The objective is for all evaluation parameters to equal to 1. The Pareto set is provided by four solutions that define three generalized technical contradictions (GTCs) defined by experiments E1 (e1), E2 (e4) and E3 (e6, e9). Having chosen the pair of GTCs defined by E2 and E3 as components of the system of technical contradictions, it remains to define the conflict with the action parameters using discriminant analysis. In the example, the conflict can be expressed as follows: with x2 = 1 and x3 = 0, the objectives of parameters y4 and y2 are achieved, but **x1 = 1 and x = 0** are also required to achieve the other objectives. To achieve all the objectives, the contradiction concerning the value of x1 when x2 = 1 and x3 = 0 is to be overcome. The generalized system of contradictions with the paired GTCs and generalized physical contradiction (GPC) are illustrated in (b).

		AP					EP					
		x1	x2	x3	x4	x5	y1	y3	y5	y6	y2	y4
E1	e1	1	1	0	0	1	1	1	1	1	0	1
E2	e4	1	1	0	0	0	1	1	0	0	1	1
E3	e6	0	1	0	1	2	0	0	1	1	1	1
	e9	0	1	0	0	2	0	0	1	1	1	1
	e2	0	1	1	1	1	0	0	1	1	1	0
	e3	1	0	1	0	0	1	1	0	0	0	0
	e5	1	0	1	0	1	1	1	1	1	0	0
	e7	1	0	1	1	0	1	1	0	0	0	0
	e8	1	0	0	0	1	1	0	1	1	0	1

a)



b)

Figure 53: Generalized system of contradictions

This system of contradictions is the starting point for guiding model change. The physical contradiction expresses the contradiction to be resolved by the decision-maker. To resolve this contradiction, the system needs to be qualitatively transformed, which will result in a change of the system model, for which the decision-maker uses the inventive design principle presented in the next section.

5-2.3 Change model

The objective of changing the model is to improve the system by using inventive design principles. Changing the model is a creative process that exploits knowledge from the system of contradictions to find new solution concepts. Inventive design provides multiple methods to support the creative process. This work uses the separation principles of TRIZ, including spatial, temporal and systematic separation [10].

- Spatial separation focuses on separating parts of a system that work together. Separation reduces interference and interaction between these parts
- Temporal separation changes the timing of events. This means to activate or deactivate parts of a system at different times to gain and avoid specific effects.

- Systematic separation divides a system into independent functional units. This increases the flexibility and adaptability of the system.

The application of separation principles will be developed when required within the case-study. Model change inherits modifications in the physical system by adding new relations among the existing parameters and/or new action parameters that were not part of the optimization problem. After developing new solution concepts, these are to be tested by simulation. For simulation, the model change can be implemented in the simulation data or library of partial models for data-driven simulation. Applying simulation and optimization again in a new cycle of the problem-solving loop provides a new Pareto front, as illustrated in Figure 54.

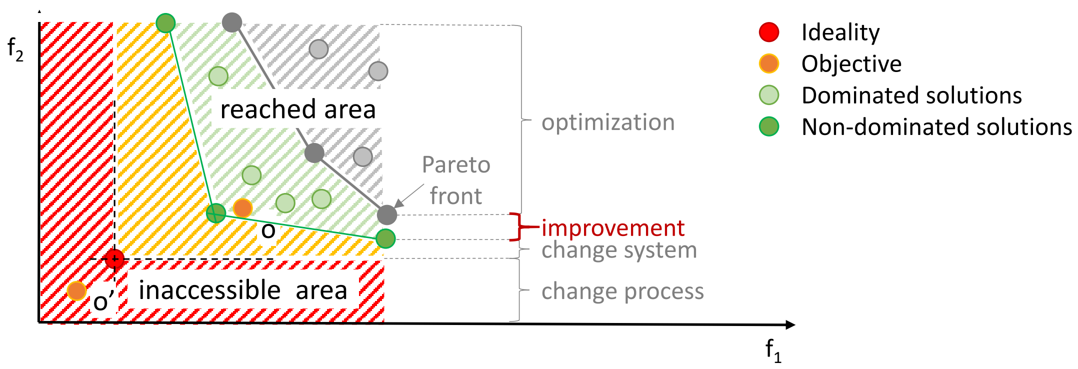


Figure 54: Moving Pareto front

A new solution concept can move the Pareto front by either improving or degrading the system. In the exemplary illustration, model change moves the Pareto front closer to ideality. Grey data points provide the Pareto front and dominated solution candidates shown in Figure 50. Green data points provide the moved Pareto front and reveal improvement by model change. The decision-maker checks if the objectives are reached. In this illustration, model change reduced the space of the reachable area (yellow) by extending the reached area (green). The unreached area remains unchanged as long as the process and the reference are not changed. The decision-maker can again evaluate if the objectives are satisfied and decide whether or not to continue with problem-solving. When the objectives are not reached and the decision-maker continues with problem solving, a change to the model should be made. To change the model again, there are two options.

- First, the decision-maker can move again to the initial situation and use the initial experimental data that was obtained before the model change (grey data points in Figure 54). In the initial data, the decision-maker searches for another contradiction and provides an alternative model change or attempts to use inventive principles to provide an alternative model change for solving the same contradiction.
- Second, the decision-maker can continue from the as-is situation and try to solve the problem by optimization in the changed model by using the action parameters. After optimization, the decision-maker can use the new experimental data (grey data points in Figure 54) to extract contradictions and solve the problem by inventive design.

To continue from the as-is situation, optimization is required to continue with inventive design to provide new experimental data. The contradictions from the initial experimental data are not valid after the model change. Once the model is changed, the action parameters and/or their relations are changed. As a consequence, the relations between action parameters and evaluation parameters are different. Even if the technical contradictions remain the same, the underlying physical contradictions are different. Consequently, in the changed system, there is a new inventive problem(s) to be solved, which requires new experimental data. As a result, the decision-maker can either go back and search for alternative inventive problems in the initial experimental data (first option) or optimize in the changed model and provide experimental data with new inventive problems (second option). However, optimizing in the changed model might provide a satisfying solution concept.

5-3 Case-Study: Emergency department

This case study illustrates the application of the problem-solving loop. Th purpose is to highlight how the methods for data-driven simulation (Chapters 2 and 3), simulation and optimization (Chapter 4) and inventive design (Chapter 5) support problem-solving. The case study of the ED was introduced in section 0. The decision-maker wants to improve the ED because the initial system (S1) was not found to be satisfactory. Patients with a green severity index have a long LOS. To solve this problem, the decision-maker provides a future configuration (S2). The starting point of the case study is a comparison between S1 and S2. After comparing both configurations, the decision-maker enters the problem problem-solving loop. Table 19 displays an outline of the case study.

Table 19: Outline for case study

		Optimization		***Inventive design***	
	Scenario	***Optimization***	***Objective / compromise reached?***	***Physical contradictions***	***Model change***
Loop 1: (section 0)	***S2***	Optimize: priority-rule, stretchers, boxes, doctors, nurses	not satisfied	Stretchers, doctors	Assign stretchers to patients
Loop 2: (section 5-3.4)	***S3***	Optimize doctors, nurses, priority	not satisfied	priority	Have nurses move patients
Loop 3: (section 5-3.5)	***S4***	Optimize doctors, nurses, priority	not satisfied	priority rule, doctors	Introduce a new priority rule
Loop 4 (section 5-3.6)	***S5***	Optimize doctors, nurses, priority	satisfied		

The decision-maker enters problem-solving by comparing scenarios S1 and S2. The S2 scenario improves the LOS for green patients, but degrades the LOS for orange and red patients. To solve this, the decision-maker proposes to optimize the assignment of resources. Descriptions of each cycle are as follows.

- In the first loop, the decision-maker optimizes stretchers, doctors, nurses, boxes and the priority strategy. Physical contradictions for stretchers and doctors are found. The conflict is solved by assigning stretchers to patients and moving both to the zone instead of having a fixed assignment to zones.

- In the second loop, the decision-maker optimizes doctors, nurses and the priority strategy. A physical contradiction is found in the priority strategy. The conflict is solved by providing additional capacity from the nurses to reduce competition for stretcher bearers, where the priority rule causes conflict.
- In the third loop, the decision-maker optimizes doctors, nurses and the priority strategy. A physical contradiction in the priority strategy and doctors is found. The conflict is solved by providing a new priority strategy to improve the patients with degraded LOS.
- In the fourth loop, the decision-maker optimizes doctors, nurses and the priority strategy. It is realized that there are multiple experiments satisfying all objectives. The problem is solved; the LOS of green patients is improved without degrading the LOS of the remaining patients.

After the fourth loop, the decision-maker arrives at an acceptable scenario (S5), in which the average LOS of green patients improves and the average LOS of the remaining patients does not degrade. The following sections illustrate the cycles of the problem loop, where the decision-maker solves the stated problem. Optimization is used to optimize the action parameters. It is recognized that optimization cannot provide an acceptable solution and an inventive problem is identified. Optimization provides experimental data. From the experimental data, binarization is applied to validate that the objectives are satisfied. The binarized data are used to highlight the contradictions and provide a system of contradictions. Based on the system of contradictions, methods of inventive design are used to solve the problem. After the last cycle of the problem-solving loop, the initial situation (S1) and the new situation (S5) are compared, thus solving the problem.

5-3.1 Stating the problem

The case study begins with the decision-maker highlighting the problems of the ED. In the initial situation (S1), patients with a low severity index (green) have long waiting times. The decision-maker formulates the objective of reducing the LOS of the green patients, corresponding to the administrative contradiction of inventive problem-solving. For solving the administrative contradiction, the decision-maker proposes the future scenario (S2). Table 20 contains the description of both scenarios S1 and S2, as well as the ideality for comparison.

The case study is based on the historic data from the year 2021. During this year, 35240 patients arrive at the ED. Of these, 13252 patients are classified as green, 16877 are classified as orange, and 5111 are classified as red. The remaining 2514 patients leave the ED before receiving an ESI. In the future scenario, these patients are classified as 19860 blue and 11642 orange patients, again with 2514 patients leaving the ED before receiving an ESI. During their stay, the patients receive resources for treatment based on their severity level; the sequence with decreasing importance is red, orange, green and blue.

In the initial situation (S1), the ED has three zones. Patients receive a severity classification (green, orange, red) in triage from the nurse and are assigned to a zone. For the assignment, the nurse uses the stock level of each zone for balancing the number of patients of each colour among the three zones. In each zone, there is a given number of stretchers (beds), doctors, nurses and boxes available (see Table 20). For solving the administrative contradiction, the decision-maker proposes a future configuration (S2) of the ED. In the future configuration, there are two levels for the patient categorization (blue, orange) which are based on the ESI. In the future situation, patients with a blue severity index are sent to the blue zone and patients with an orange severity index are sent to the orange zone. The number of resources in each zone are listed in Table 20. In addition, there are shared resources for patients of all zones, particularly stretcher-bearers, RX-laboratory and imaging. When competing for shared resources, priority is given to patients having the highest severity index. The sequence of descending priority in S1 is red, orange, green, and in S2 the sequence is orange, blue. In addition to the configuration of the zones, Table 20 provides the definition of the ideal scenario. In the case study, the scenarios are compared to the reference, which is measured by simulating the system with 'infinite' resources (see Table 20).

Regarding the resources, switching from the initial situation to the future situation is accompanied by the removal of four doctors. In the initial situation, there are 10 doctors that are assigned to three zones in the future situation, and six doctors are assigned to two zones. In parallel, the number of nurses assigned to the zones is increased by 1 from three in the initial situation to four in the future situation. The set of arriving patients is identical and is defined by the sequence and inter-arrival time.

Table 20: Initial and future situation

	Initial situation (S1)	***Future situation (S2)***	***Ideality (Ref.)***
Categorization	green / orange / red	blue / orange	blue / orange (has no impact)
Number Zones	3	2	2 (has no impact)
Patient assignment	Mixed coloured zones Balancing stock level of each colour	Specialized zone, one for each colour	Specialized zone, one for each colour (has no impact)
Historic patient arrivals	Same set of patients and inter-arrival times		
	13252 green patients 16877 orange patients 5111 red patients	19860 blue patients 11642 orange patients	19860 blue patients 11642 orange patients
Stretchers (beds) (zones: 1 / 2 / 3)	10 / 10 / 10	10 / 20 / -	99 / 99 / -
Doctors (zones: 1 / 2 / 3)	4 / 4 / 2	2 / 4 / -	99 / 99 / -
Nurses (zones: 1 / 2 / 3)	1 / 1 / 1	2 / 2 / -	99 / 99 / -
Boxes (zones: 1 / 2 / 3)	5 / 5 / 5	5 / 10 / -	99 / 99 / -
Stretcher-bearer	1	1	99
RX	1	1	99
Imaging	1	1	99
Prioritization (low < high)	green < orange < red	blue < orange	blue < orange (has no impact)

5-3.2 Comparison of scenarios

The initial formulated objective of the decision-maker is to improve the LOS of green patients by changing S1 and developing a new configuration S2. The switch from S1 to S2 includes two changes: changing the configuration of the ED, and changing the triage system. The former is a change from two to three zones, including the reassignment of their resources. The latter is the change of patients' categorization from green, orange and red to blue and orange. To assess the effects of the two changes in greater detail, the LOS is measured based on the initial and future triage system, considering the severity indices of blue, orange, and red, as well as blue and orange. Thus, patients are categorized in six groups defined by their colour in S1 and S2, respectively. Consequently, the reformulated objective is to improve the LOS of green patients that become orange, and green patients that become blue. Figure 55 provides the average LOS of all patient groups in a radar plot. The results were generated by data-driven simulation. For implementing the two triage systems of initial and future situation, the triage levels in both systems were provided in the simulation data and functions for assignment of patients to the zones were implemented in the partial models.

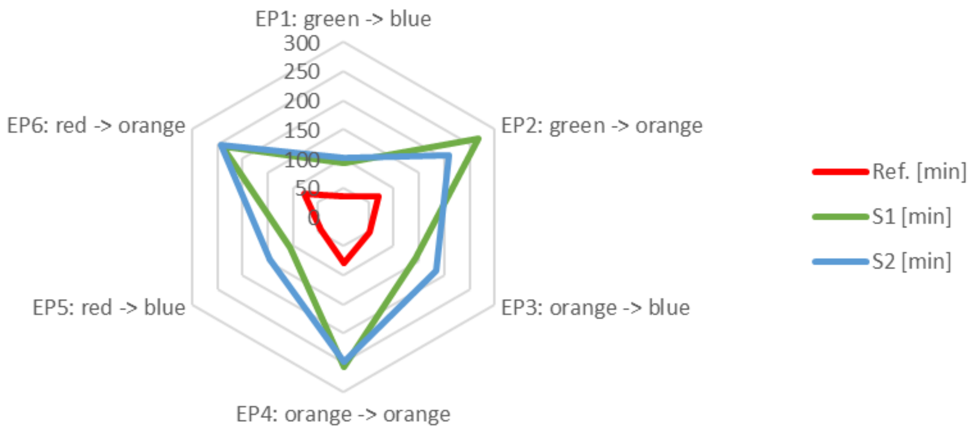


Figure 55: Length of stay for each patient group

The radar plot compares the LOS for all patient groups for S1, S2 and the reference. The reference refers to the 'ideal' model of Chapter 2. It provides the static LOS of each patient variant and is statically calculated by using the sequence and duration of activities. In this chapter, the reference is provided using simulation. In the reference model, all resources are immediately available when needed, which is achieved in our data-driven model by sufficiently increasing the number of resources in model S1 or S2. Thus, there are no queues nor waiting times for activities; the LOS is solely the product of the sequence and duration of activities. An alternative way to obtain the reference of the ideal model is to keep the resources off the model, run the patients through the system and measure the average throughput time. As illustrated by the radar plot, improvement and degradation within the patient groups is not homogenous. In each group, there are two sub-groups becoming blue and orange. The green patients becoming orange improve from 267 to 210 min, and the green patients becoming blue degrade from 92 to 101 min. In S2, patients with high priority (orange) improve, while patients with low priority (blue) degrade. The numbers are provided in Table 21. The table provides the evaluation parameters and binarized evaluation parameters.

Binarization is obtained for each parameter and scenario by comparison with the initial objective. LOSs of green patients have to be better than those for S1, and others have to be at least at the level of S1. Thus, for the green patients, a 0 value is assigned to the LOS if the value is greater than or equal to the those of S1, and a 1 is assigned if the value is lower than those of S1. In an analogous manner, for the orange and red patients, a 1 is assigned to the LOS if its value is less than or equal to those of S1, and 1 is assigned if the LOS is greater than those of S1. Binarization shows global improvement and degradation. The comparison between S1 and S2 highlights gaps, caused by waiting times (EP1 +9,3 %; EP2 −21,4 %; EP3 +28,5 %; EP4 −3,8 %; EP5 37,6 %; EP6 +1 %). This increases the global waiting time by 10,04 % from 61 164 h in S1 to 67 306 h in S2. Moreover, globally, S2 performs worse than S1 for the average LOS of the whole set of patients (+ 6,67 %). Another remark can be made by observing Figure 55. Whatever the patient’s colour in the S1 model, the LOS is already greater for the S2 model orange patients [i.e. LOS (colour S1,blue) < LOS (colour S1,orange)]. This observation holds for the three scenarios S1, S2 and the ‘ideal’ reference. Analysis of the reference curve also provides new insights for the designer (who is not a physician); when ranking the LOS of the reference model, the new categorization correlates the LOS almost positively with S2 blue and orange at the first level, and then almost with colours (green, orange, red) of the first model S1.

Table 21: Technical contradictions for S1 and S2

							Objectives:	<92	<267	≤143	≤258	≤106	≤242
		Evaluation Parameters						***Binarized Evaluation Parameters***					
		EP1	***EP2***	***EP3***	***EP4***	***EP5***	***EP6***	***EP1***	***EP2***	***EP3***	***EP4***	***EP5***	***EP6***
	Exp	***LOS(g,b)***	***LOS(g,o)***	***LOS(o,b)***	***LOS(o,o)***	***LOS(r,b)***	***LOS(r,o)***	***LOS(g,b)***	***LOS(g,o)***	***LOS(o,b)***	***LOS(o,o)***	***LOS(r,b)***	***LOS(r,o)***
E1	S1	92	267	143	258	106	242	0	0	1	1	1	1
E2	S2	101	210	184	248	146	244	0	1	0	1	0	0
	Ref	36	68	53	79	45	78	1	1	1	1	1	1

From the binarized simulation results, the decision maker learns several things about the new scenario. First, changing the system does not benefit all green patients. In particular, S2 improves the green patients that become orange. Second, improving these patients requires sacrificing remaining patients, except orange, that become orange. Performances of patients that become blue are worsened. From this situation, the decision-maker realizes that reformulation of the objective is required. The objective of improving the LOS of green patients is replaced with that of improving the LOS of green patients without worsening that of orange and red patients. Referred to the terms of inventive design, this is referred to as replacing the administrative contradiction by a technical contradiction.

To solve the new, technical contradiction, the decision-maker must understand the system's features that cause degradation of the patient's LOS and cause S2 to perform worse than S1. Nevertheless, at this stage, it is not known whether the problem can be solved within an optimization model or if it requires deeper model change. In order to clarify this point, the problem-solver must understand the links between the system's model parameters and the evaluation parameters (the different LOS). To this end, the decision-maker states the parameters that are modelled in the simulation model for scenarios S1 and S2 that could cause degradation of the system performance in comparison to the reference. The fishbone diagram in Figure 56 list the parameters that might impact the LOS; further simulations may validate it. The parameters in green are those that have be changed between S1 and S2, particularly the green parameters; those with red margins are parameters the decision-maker can act on within the conditions of S2.

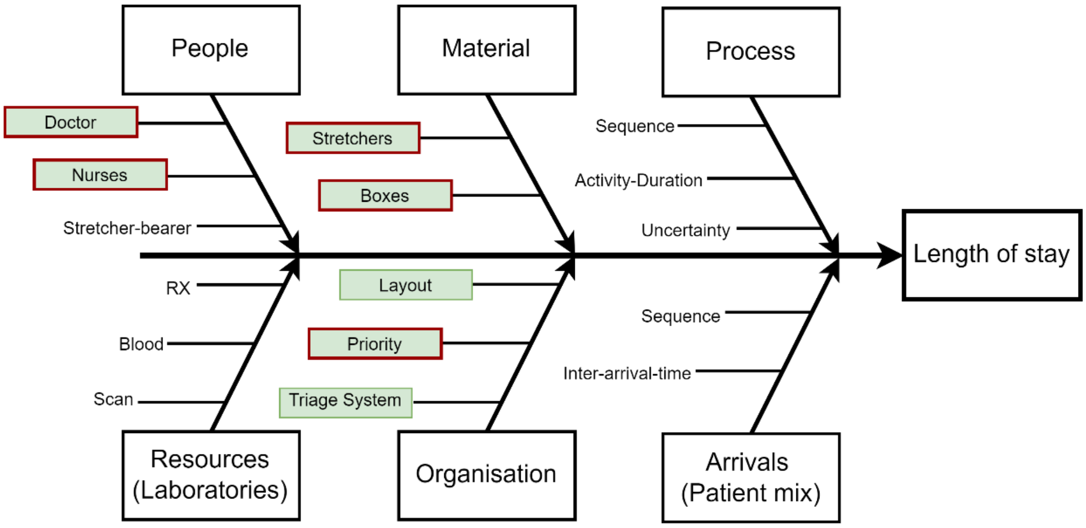


Figure 56: Action parameters influencing LOS

The potential impacting parameters are structured into six groups: arrivals, process, material, organization, resources and people. The arrival consists of the sequence and inter-arrival times of the patients. The process describes the sequence of activities, their durations and uncertainties. Both arrivals and process are assumed as given. The resources (laboratories) are theoretical parameters of the system model, but the decision-maker has no power to act on them. The parameters on which the designer can act remain as the organization, material and people. Material includes stretchers and boxes assigned to zones. The people group includes the doctors and nurses in the areas and the stretcher-bearers who move the stretchers. In the organization group, the building layout is not changeable. In the scenarios, there are either two or three zones. The triage system is given by the decision-maker for S1 and S2. Prioritization of patients at the organization is also considered as a parameter; it is a qualitative parameter and defines which patients, based on the severity index, receive priority when waiting for resources, materials and people. At this stage of analysis, it was decided with the decision-maker to improve S2, which already provides an improvement of green patients compared to S1 and try to lessen the worsening of orange and red patients. Additional reasons for this choice are the following.

- S2 ESI categorization becomes very popular and, as already shown on the ideal model, there is a positive correlation between the average LOS and S2 categorization system.
- The influence of the value of the quantitative parameters on S2 performance is not well known to the decision-makers, as it is a new model for them. The S2 model remains to be optimized by changing some of its quantitative parameters' values. Optimization might improve the performance of S2.

For optimization, the decision-maker provides the stretchers, doctors, nurses and boxes, as well as the prioritization policy. From a modelling perspective, we are referring to changing the model, since adding new action parameters was given in the activity of model change in section 0. However, these parameters have previously been modelled as static system parameters. Their definition as action parameters moves them into the scope of the optimization problem.

5-3.3 Loop 1: Change optimization parameters

To design a simulation-optimization configuration for S2, the decision-maker must define the action parameters among S2 parameters. Action parameters are a subset of system parameters, whose values can be modified independently of each other. The action parameters to be varied are highlighted in green with red margins in Figure 50; these are the number of stretchers, doctors, nurses and boxes assigned to each zone, and the priority rule. The decision-maker wants to determine the required number of stretchers, doctors, nurses and boxes in each zone and wants to understand the impact of the priority strategy. The number of resources in the future situation are static system parameters and are set to 30 stretchers, 6 doctors, 4 nurses and 15 boxes. For each resource, the number of elements in the orange zone is an action parameter to be acted on in the optimization. The number of the elements in the blue zone is a system parameter that is changed in accordance with the action parameter.

Additionally, the priority policy is an independent, qualitative action parameter. For prioritization, two strategies two strategies are tested: prioritizing patients having a high severity index (prio = 1), and treating all patients equally (no priority; prio = 2).

Optimization performed with data-driven simulation and Pareto optimization with NSGA-II provides the non-dominated experiments on the Pareto front. There are no solutions performing better than the initial situation for all measures simultaneously. In order to find properties that cause the Pareto front and the main features of the Pareto front, our approach proposes to keep points nearby the Pareto front. Table 22 displays the experiments with the best performance near the Pareto front. A value is highlighted in green if it satisfies the objective, and is red otherwise. The objectives are recalled in Table 21. Binarization of the evaluation parameters expresses in a binary manner whether an objective is reached or not for a given solution and evaluation parameter. The value 1 denotes satisfaction of the objective and 0 denotes dissatisfaction.

Table 22: Paired technical contradictions in S2

							Objectives:	<92	<267	≤157	≤284	≤117	≤266
		Evaluation Parameters						*Binarized Evaluation Parameters*					
		EP1	*EP2*	*EP3*	*EP4*	*EP5*	*EP6*	*EP1*	*EP2*	*EP3*	*EP4*	*EP5*	*EP6*
	Exp	*LOS(g,b)*	*LOS(g,o)*	*LOS(o,b)*	*LOS(o,o)*	*LOS(r,b)*	*LOS(r,o)*	*LOS(g,b)*	*LOS(g,o)*	*LOS(o,b)*	*LOS(o,o)*	*LOS(r,b)*	*LOS(r,o)*
E1	e5	70	265	144	305	114	300	1	1	1	0	1	0
	e11	71	252	145	292	114	286	1	1	1	0	1	0
	e12	71	258	146	298	115	293	1	1	1	0	1	0
	e18	71	252	145	292	114	286	1	1	1	0	1	0
E2	e3	82	224	160	266	127	261	1	1	0	1	0	1
	e8	81	225	159	267	127	262	1	1	0	1	0	1
	e14	82	224	160	267	127	261	1	1	0	1	0	1

First, it can be seen in Table 22 that an initial improvement of green patients is obtained for all selected, but that a conflict remains when trying to satisfy additional requirements for orange and blue patients simultaneously (S2 ESI categorization). Either we can satisfy the EP3 and EP5 requirements (S1 orange and red patients that become blue in S2) or the EP4 and EP6 requirements (S1 orange and red patients that become orange in S2). The experiments e5, e11, e12, e18 satisfy the LOS of the orange patients and the experiments e3, e8, e14 satisfy the LOS for the blue patients. The technical contradictions of the sets E1 and E2 describe the conflict between the LOS of orange and blue patients, particularly for patients that were orange or red in the initial situation. At this stage, there are two options for improving the systems performance.

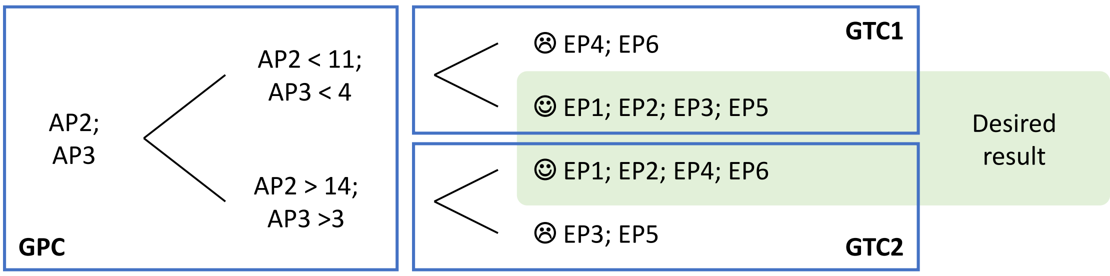
- First, extend the possible values for the fixed resources, that is, relaxing or changing constraints in the optimization problem (i.e. allow a higher total number of stretchers, doctor etc.). This approach leads to a solution from the perspective of the LOS evaluation parameters and requires new costs for the additional resources. This shows that an additional evaluation parameter, 'resources cost', could be added. This approach can provide acceptable results if the cost for the additional resource is acceptable. We are not developing this approach in this PhD thesis because it is a classical approach in simulation and optimization.
- Second, address the contradiction(s) and try to reach the objectives while keeping the same number of resources. As this approach is less classical and is part of the proposed approach of this thesis, it is developed next.

According to our problem-solving approach, we now need to find the physical contradiction at the root of this performance conflict. Table 23 provides the value of the decision variable for each of the experiments. One can notice that number of stretchers (Stret.), doctors (Doc.), nurses and boxes are only provided for the orange colour. Indeed, as the total number of each resource is fixed, once these values are defined, the values of resources of the blue zone are defined as well. A discriminatory analysis between the two sets of experiments, E1 and E2, of Table 23 is performed, which provides the physical contradiction.

Table 23 Physical contradiction in S2

							Objectives:	<92	<267	≤157	≤284	≤117	≤266
		Evaluation Parameters						*Binarized Evaluation Parameters*					
		AP1	***AP2***	***AP3***	***AP4***	***AP5***	***AP6***	***EP1***	***EP2***	***EP3***	***EP4***	***EP5***	***EP6***
	Exp	***Scenario***	***Stret. (o)***	***Doc. (o)***	***Nurse (o)***	***Box (o)***	***Prio***	***LOS(g,b)***	***LOS(g,o)***	***LOS(o,b)***	***LOS(o,o)***	***LOS(r,b)***	***LOS(r,o)***
E1	e5	S2	9	2	2	12	2	1	1	1	0	1	0
	e11	S2	10	2	3	10	2	1	1	1	0	1	0
	e12	S2	9	3	3	12	2	1	1	1	0	1	0
	e18	S2	10	2	3	10	2	1	1	1	0	1	0
E2	e3	S2	17	4	2	13	2	1	1	0	1	0	1
	e8	S2	15	4	2	12	2	1	1	0	1	0	1
	e14	S2	16	4	2	13	2	1	1	0	1	0	1

The discriminatory analysis in Table 23 shows that there is a physical contradiction in the number of decision variable stretchers AP2 and doctors AP3 values. The remaining APs are not involved in the conflict. Moreover, it seems that the AP1 and AP6 values should be S2 and 2 in order to be near Pareto points. There has to be more than 15 stretchers in the orange zone to satisfy the LOS of orange patients, and there has to be fewer than 11 stretchers in the orange zone, meaning more than 19 stretchers in the blue zone to satisfy the LOS of the blue patients (analogously for the doctors). Figure 57 provides the system of contradictions.



Context: AP1 = S2; AP6 = 2

Figure 57: System of contradictions

Let us explain the conflict at the level of the design parameters (APs). When analysing results of simulation linked to the number of stretchers, one type of cause of the physical contradiction can be highlighted. The conflict is caused by the temporary lack of stretchers and doctors when needed in high numbers. Owing to the uncertainty in the patient arrivals and routings, a high demand for stretchers and doctors there can temporarily occur. If they are not available, then waiting lines occur and the LOS degrades. The waiting times are illustrated in Table 24 Consequently, orange patients benefit from a high number of orange stretchers, and doctors and blue patients benefit from a high number of blue stretchers and doctors. Again, an easy solution to this problem would be to increase the total number of stretchers and doctors in each zone. However, as already mentioned, we shall first try to overcome this conflict by keeping the same resources. This is the purpose of the next section.

Table 24: Waiting for resources in S2

Waiting times [minutes]									
	Stretchers		Doctors		Nurses		RX	Scan	Stretcher bearer
Exp	blue	orange	blue	orange	blue	orange			
e5	6	43	0	2	0	0	0	4	14
e11	0	28	0	2	0	0	0	4	17
e12	0	41	1	0	0	0	0	4	16
e18	0	28	0	2	0	0	0	4	17
e3	4	1	5	0	0	0	1	5	19
e8	2	2	6	0	0	0	1	5	19
e14	3	1	6	0	0	0	1	5	19

5-3.4 Loop 2: Change assignment of beds

To address the physical contradictions, TRIZ proposes several principles and tools. One of these tools consists of specifying in time the need that is considered as contradictory, referred as separation in time. Here, one needs to answer the following question: When are the 15 stretchers for the orange zone and the 19 stretchers for the blue zone needed? This question is posed with the viewpoint of finding out whether the 15 stretchers for the orange zone and the 19 stretchers for the blue zone are needed at the same time. If this is not the case, then we have an indication of how to change the model and improve system performance. Simulation helps to answer the question. In the historical example of 35000 patients, we never need all 15 stretchers for the orange zone and all 19 stretchers for the blue zone at the same time. Furthermore, at times, there are unused stretchers in one zone, while there is a need for more in the other zone (S2).

To overcome the temporary shortage of resources, it is proposed to make the resources available in the zones where they are needed at the moment when they are needed by assigning stretchers to patients instead of the zones. As each patient is assigned to a zone regarding the severity index colour, the stretcher will be assigned to the same zone as the patient. Because the same conflict potentially appears for the boxes, it is decided to make the box generally available, similar to the stretchers. Despite the doctors having the same conflict, it is decided in the first step not to follow and check this solution direction. Reasons for not solving this conflict are given by the decision-maker who highlights that solving this conflict would lead to a change in the management of doctor assignment, which is not desired. Implementing the proposed model changes provides a new future situation (S3) for optimization.

In optimization, the previous action parameter of the number of stretchers and boxes in the orange zone is removed. For notation, we state the number of stretchers and boxes in the orange zone as dynamic (dyn.). With the changed model, the decision-maker optimizes the remaining action parameters, nurses and doctors, in the orange zone and the priority policy of the patients. Simulation and optimization with NSGA-II provides the experiments on the Pareto front of the new solution. Binarization visualizes the satisfaction of the objectives (see Table 25). There are still no solutions that perform better than the initial situation S1 for all measures. For accepting 11 % of degradation, all objectives are satisfied. By going from complete satisfaction, the decision-maker proceeds towards a problematic situation by lowering the acceptable degradation of the objective. When accepting 10 % of degradation, a conflict is found between E1 (e2, e4, e5) and E2 (1). Either we can satisfy the EP3 and EP5 requirements (S1 orange and red patients that become blue in S2) or EP6 requirements (S1 red patients that become orange in S2). Nevertheless, we are now capable of reaching five of the six objectives simultaneously, including the initial main objectives, while in section 5-3 and previously, only two or four objectives were reachable with S2 or S1. According to our methodology, the decision-maker can either accept the compromise represented by one of these solutions or start a new loop by seeking to resolve the new system of contradictions that prevents an improvement in the performance of this new system.

Table 25: Paired technical contradictions in S3

		Evaluation Parameters						Binarized Evaluation Parameters					
	Objectives:							<92	<267	≤157	≤284	≤117	≤266
		EP1	EP2	EP3	EP4	EP5	EP6	EP1	EP2	EP3	EP4	EP5	EP6
	Exp	LOS(g,b)	LOS(g,o)	LOS(o,b)	LOS(o,o)	LOS(r,b)	LOS(r,o)	LOS(g,b)	LOS(g,o)	LOS(o,b)	LOS(o,o)	LOS(r,b)	LOS(r,o)
E1	e2	90	209	172	247	137	243	1	1	0	1	0	1
	e4	82	219	162	257	127	253	1	1	0	1	0	1
	e5	83	210	164	248	129	244	1	1	0	1	0	1
E2	e1	72	234	147	277	116	271	1	1	1	1	1	0

Let us try to improve the system by solving the contradiction. To do this, we must first identify the physical contradiction(s) associated with the technical contradictions of Table 25. The technical contradiction between the sets E1 and E2 describes the conflict between EP3, EP5 and EP6. The experiments e2, e4, e5 satisfy the LOS of the initially orange and red patients becoming orange and the experiment e1 satisfies the LOS for the initially red patients becoming orange. Table 26 provides the physical contradiction using a discriminatory analysis. The stretchers and boxes in AP2 and AP5 are assigned to the patients instead of as previously to the zones (dyn.).

Table 26: Physical contradiction in S3

		Evaluation Parameters						Binarized Evaluation Parameters					
	Objectives:							<92	<267	≤157	≤284	≤117	≤266
		AP1	AP2	AP3	AP4	AP5	AP6	EP1	EP2	EP3	EP4	EP5	EP6
	Exp	Scenario	Stret. (o)	Doc. (o)	Nurse (o)	Box (o)	Prio	LOS(g,b)	LOS(g,o)	LOS(o,b)	LOS(o,o)	LOS(r,b)	LOS(r,o)
E1	e2	S3	dyn.	4	1	dyn.	1	1	1	0	1	0	1
	e4	S3	dyn.	3	1	dyn.	1	1	1	0	1	0	1
	e5	S3	dyn.	2	2	dyn.	1	1	1	0	1	0	1
E2	e1	S3	dyn.	3	2	dyn.	2	1	1	1	1	1	0

The discriminatory analysis in Table 26 shows that despite the fact that the conflict of the doctors has not been addressed, after the model change it is not the most pertinent conflict in the system. The more relevant action parameter is the priority strategy, AP6. The remaining APs are not involved in the conflict. When waiting for shared resources, the priority strategy of prioritizing orange patients satisfies the LOS of orange patients and the priority strategy of treating all patients equally satisfies the blue patients. Figure 58 provides the system of contradictions for the future scenario (S2).

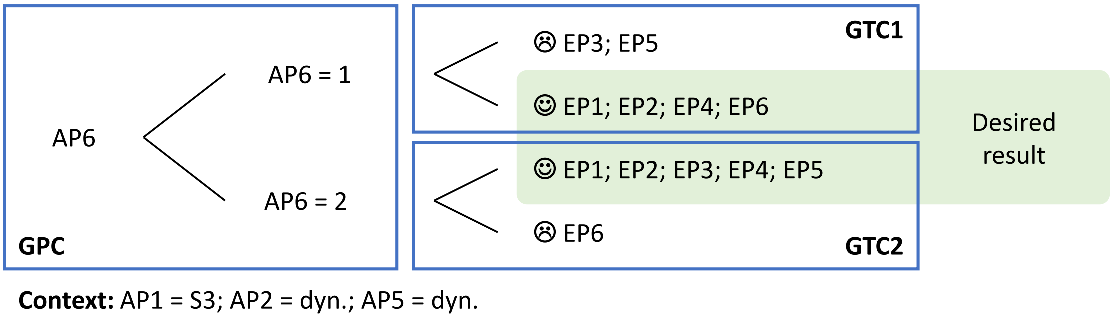


Figure 58: System of contradiction in future scenario (S3)

The conflict is caused by the different priority strategies. The first strategy of prioritizing orange patients improves the LOS of orange patients, and the second strategy of not prioritizing orange patients improves the LOS of blue patients. Let us explain the conflict at the level of the design parameters (APs). When analysing the results of the simulation linked to stretcher-bearers, the one cause of the physical contradiction becomes visible. Table 27 shows the average waiting times while waiting for different types of resources. Waiting times are mostly linked to the stretcher-bearers who move the patients between zones and laboratories. When multiple patients wait for the stretcher-bearer, the priority strategy determines which patients are moved first. When prioritizing orange patients (e2, e4, 5), the LOS of initially orange and red patients becoming blue is not satisfied (EP3, EP5), and when not prioritizing orange patients (e1), the LOS of the initially red patients becoming orange (EP6) is not satisfied. Increasing the number of stretcher-bearers might improve the LOS of both patient groups and was tested in the experiments. When applying this approach, it was found that one additional stretcher bearer improves the results. However, as defined by the decision-maker, we shall first try to overcome the conflict by keeping the same resources. This is the purpose of the next section.

Table 27: Waiting for resources in S3

Waiting times [minutes]									
	Stretchers		Doctors		Nurses		RX	Scan	Stretcher bearer
Exp	blue	orange	blue	orange	blue	orange			
e5	0	1	6	0	0	0	1	6	18
e11	0	0	1	1	0	0	0	6	20
e12	0	0	0	5	0	0	1	5	19
e18	0	0	1	1	0	0	0	5	21

5-3.5 Loop 3: Change activities of nurses

To address the physical contradictions, TRIZ proposes several principles and tools. One of these tools is systematic separation, which divides a system into independent functional units to increase flexibility. The lack of capacity for stretcher-bearers avoids fulfilling the function of moving patients without waiting. To provide the required capacity, the decision-maker needs to answer the question: Is there capacity available for moving patients between zones and laboratories? The provided measures for the waiting line show that there are no waiting lines at the nurses, neither in the blue nor in the orange zone. The decision-maker proposes to change the model by changing the activities of the nurses. If there is no stretcher-bearer available and there are no waiting lines for nurses, then the nurses move patients between zones and laboratories. Implementing the proposed model changes provides a new future situation (S4) for optimization. However, we keep in mind that the initial activities of the nurses might not fully reflect their actual tasks and discuss this point later.

In optimization, the number of nurses and doctors in the orange zone remain as action parameters. The number of doctors and nurses in the blue zone are again resulting system parameters. Simulation and optimization with NSGA-II provides the Pareto front of the new solution for S4. Binarization visualizes the satisfaction of the objectives (see Table 28). There is still no solution candidate that satisfies all the objectives. In addition, there remain three objectives that are in conflict: EP4 and EP6 are not reachable in the same moment as EP5. However, the discrepancy is again reduced. The experiments e7 and e8 satisfy all objectives, except that of EP5. The gap to satisfaction is 4 min. Because the remaining evaluation parameters increase, there could be the possibility of accepting this compromise. Further, the decision-maker must evaluate the model change.

Table 28: Paired technical contradictions in S4

							Objectives:	<92	<267	≤143	≤258	≤106	≤242
		Evaluation Parameters						*Binarized Evaluation Parameters*					
		EP1	*EP2*	*EP3*	*EP4*	*EP5*	*EP6*	*EP1*	*EP2*	*EP3*	*EP4*	*EP5*	*EP6*
	Exp	*LOS(g,b)*	*LOS(g,o)*	*LOS(o,b)*	*LOS(o,o)*	*LOS(r,b)*	*LOS(r,o)*	*LOS(g,b)*	*LOS(g,o)*	*LOS(o,b)*	*LOS(o,o)*	*LOS(r,b)*	*LOS(r,o)*
E1	e11	62	220	136	259	105	256	1	1	1	0	1	0
	e10	62	220	136	259	105	255	1	1	1	0	1	0
	e12	62	220	136	259	105	256	1	1	1	0	1	0
E2	e7	65	206	143	243	110	240	1	1	1	1	0	1
	e8	65	206	143	243	110	240	1	1	1	1	0	1

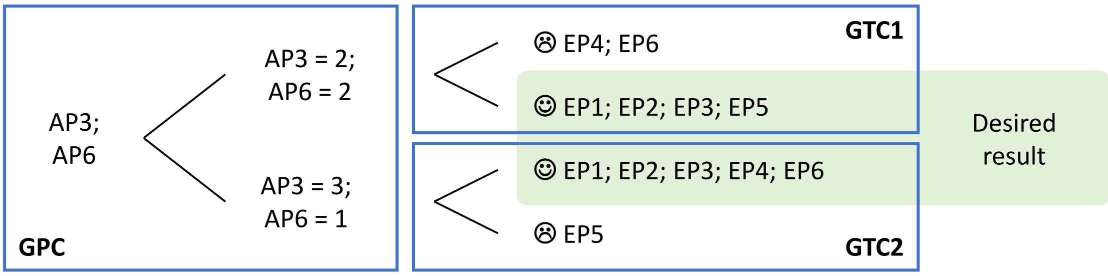
When assigning nurses to the activity of moving patients, the available capacity for the remaining activities of the nurses degrades. In this model, there is no critical impact; however, except for taking blood samples, there are no nursery tasks modelled. At this point, the activity model of the nurses might not fully reflect reality, where nurses perform tasks in addition to blood sampling. Taking away capacity could cause waiting lines for blood sampling and the remaining nursery tasks. The decision-maker might react to this problem by adding nurses, for which a cost evaluation should be made to determine if adding stretcher-bearers is a better solution. This could make the stretcher-bearer, who was until this time a static system parameter, into an additional action parameter.

Let us try to improve the system by solving the contradictions. To do this, we must first identify the physical contradiction(s) associated with the technical contradictions of Table 28. The paired technical contradiction between the sets E1 and E2 describes the conflicts between EP4, EP6 and EP5. Experiments e10, e11, e12 satisfy the LOS of the initially orange patients becoming red. Furthermore, experiment e8 satisfies the LOS for the initially orange patients becoming orange and red patients becoming orange. Table 29 provides the physical contradiction by discrimination. The physical contradiction(s) are the action parameters, explaining the contradiction between the experiments of E1 and E2.

Table 29: Physical contradiction in S4

							Objectives:	<92	<267	≤143	≤258	≤106	≤242
		Evaluation Parameters						***Binarized Evaluation Parameters***					
		AP1	***AP2***	***AP3***	***AP4***	***AP5***	***AP6***	***EP1***	***EP2***	***EP3***	***EP4***	***EP5***	***EP6***
	Exp	***Scenario***	***Stret. (o)***	***Doc. (o)***	***Nurse (o)***	***Box (o)***	***Prio***	***LOS(g,b)***	***LOS(g,o)***	***LOS(o,b)***	***LOS(o,o)***	***LOS(r,b)***	***LOS(r,o)***
E1	e11	S4	dyn.	2	3	dyn.	2	1	1	1	0	1	0
	e10	S4	dyn.	2	2	dyn.	2	1	1	1	0	1	0
	e12	S4	dyn.	2	1	dyn.	2	1	1	1	0	1	0
E2	e7	S4	dyn.	3	2	dyn.	1	1	1	1	1	0	1
	e8	S4	dyn.	3	1	dyn.	1	1	1	1	1	0	1

The discriminatory analysis in Table 29 shows again a conflict in the number of doctors, which previously appeared, but was not addressed. In addition, it shows the conflict in the priority strategy. Despite the availability of additional capacity for moving the patients, the priority strategy remains part of the physical contradiction. Besides stretcher bearers, priority policies are also applied in the laboratories. Similar to the stretcher-bearers, the capacity of the laboratories is limited. Depending on the priority strategy, either orange or blue patients wait before getting tested in the laboratories. Figure 59 provides the system of contradictions for the future scenario (S4).



Context: AP1 = S4; AP2 = dyn.; AP5 = dyn.

Figure 59: System of contradictions for S4

Again, the conflict is caused by the different priority strategy. Unlike the previous situation, the waiting times do not occur when waiting for the stretcher-bearer, but when waiting for the activities of scanning (RX and Scan) (see Table 30). Additionally, the waiting lines of the orange and blue doctors either occur when waiting for orange or blue doctors. To solve the contradiction, two scenarios appear: increasing the number of doctors, or the decision-maker can work on the priority strategy. Again, an easy solution to this problem would be to increase the total number of stretchers and doctors in each zone. But as already mentioned, we shall first try to overcome this conflict by keeping the same resources. This is the purpose of the next section.

Table 30: Waiting for resources in S4

	Waiting times [minutes]								
	Stretchers		***Doctors***		***Nurses***		***RX***	***Scan***	***Stretcher bearer***
Exp	***blue***	***orange***	***blue***	***orange***	***blue***	***orange***			
e11	0	0	8	5	0	0	2	6	0
e10	0	0	1	5	0	0	1	6	0
e12	0	0	0	5	0	0	1	6	0
e7	0	0	1	1	0	0	1	7	0
e8	0	0	1	1	0	0	1	7	0

5-3.6 Loop 4: Change priority policy

To address the physical contradictions, TRIZ proposes several principles and tools. One of these tools is systematic separation, which was used in the previous section to change the functionality of the nurses. In this section this principle is used to apply model change in the priority strategy. Priority is given for patients based on the common severity index. Either priority was given to orange patients or no priority was given. For changing the priority strategy by applying the separation principle, we propose to divide the groups of orange and blue patients into sub-groups: orange patients that previously were red and orange patients that previously were green or orange (and analogously for the blue patients). Laboratories and stretcher-bearers prioritize patients in the waiting line. For prioritization, the resources should use a new priority strategy based on the initial and future severity index. The first priority goes to red patients becoming orange, the second priority goes to red patients becoming blue, the third priority goes to orange patients becoming blue or orange, and the fourth priority goes to green patients becoming blue or orange. Implementing the proposed model changes provides a new future situation (S5) for optimization.

For optimization, the number of nurses and doctors in the orange zone remain as action parameters. The number of doctors and nurses in the blue zone are again resulting system parameters. Simulation and optimization with NSGA-II provides the Pareto front of the new solution for S5. Binarization visualizes the satisfaction of the objectives (see Table 31). The table shows that there are multiple experiments satisfying all objectives. However, beyond satisfying the objectives, there are differences in the performance of the different experiments. The experiment e2 provides better performance for the evaluation parameters EP1 and EP3, and e5 provides better performance for the evaluation parameters EP2, EP4 and EP6. These differences could be used when increasing the objectives to search for further conflicts and inventive problems to be solved.

Table 31: Solution candidates of S5

							Objectives:	<92	<267	≤143	≤258	≤106	≤242
		Evaluation Parameters						*Binarized Evaluation Parameters*					
		EP1	***EP2***	***EP3***	***EP4***	***EP5***	***EP6***	***EP1***	***EP2***	***EP3***	***EP4***	***EP5***	***EP6***
	Exp	***LOS(g,b)***	***LOS(g,o)***	***LOS(o,b)***	***LOS(o,o)***	***LOS(r,b)***	***LOS(r,o)***	***LOS(g,b)***	***LOS(g,o)***	***LOS(o,b)***	***LOS(o,o)***	***LOS(r,b)***	***LOS(r,o)***
E1	e2	64	240	135	256	103	242	1	1	1	1	1	1
E2	e5	66	224	136	247	103	237	1	1	1	1	1	1

The conflict that was caused initially by different priority policies was solved by adding a new strategy to prioritize patients. Unlike the previous model changes, this change did not add a new action parameter, but added a new qualitative behaviour, that was accessible via the action parameter AP6. Based on the two experiments, the decision-maker can choose one scenario for implementation. Owing to the better results for EP2, the decision-maker chooses to implement the experiment e5. A comparison of scenarios S1, S2 and S5 (e2) follows in the next section.

5-3.7 Compare the scenarios S1, S5 and reference

At the end of the problem-solving loop, this section has the purpose of comparing the scenarios. For improving the system, the problem-solving loop was used and five cycles in the loop were executed, using simulation to provide experimental data and inventive design methods to extract contradictions, provide the system of contradiction and change the model. The executed model changes were to change the assignment of the stretcher from the zone to the patients, changing the tasks of the nurses to support stretcher bearers and changing the prioritization policy of the ED. All model changes were provided by analysis of the experimental data and extraction of contradictions. To compare the three scenarios S1, S5 and the reference, a radar plot is provided in Figure 60. The main objective of the decision-maker, increasing the LOS for the green patients, was satisfied. The LOS of green patients becoming blue improved by 28,33 % from 92 to 66 min and for green patients becoming orange by 16,21 %. The second goal of not degrading the makespan of the remaining patients was also satisfied. Not only did the remaining patients not degrade, on the contrary, an improvement between 1,97 % and 5,19 % was achieved. Additionally, all system parameters were respected and no additional resources were added to the system. Moreover, when from changing from the initial scenario to the future scenario, four doctors were removed from the system.

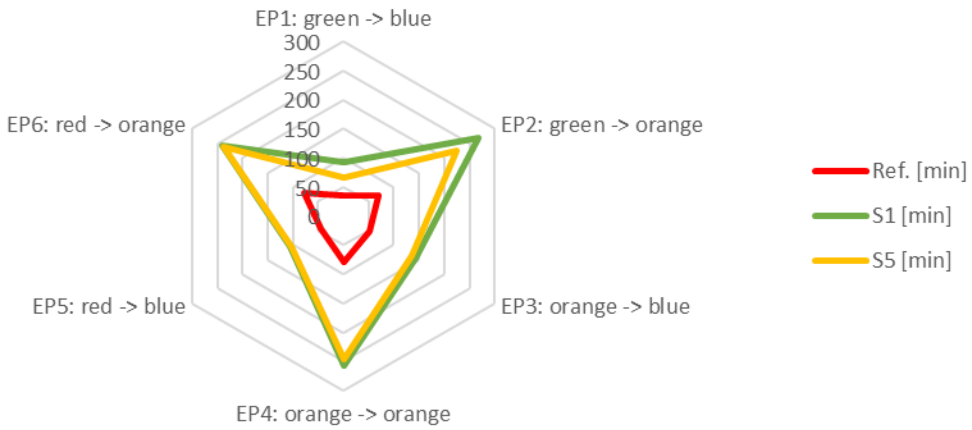


Figure 60: Radar plot for S1, S5 and reference

For comparing different patient groups in S1 and S5, the average LOS for each group based on the severity index in the initial and future system is plotted Figure 61. The top row shows the results from S1 in sub-plots (a), (b) and (c) and the bottom row shows the results from S5 in (d), (e) and (f). The background of each subplot illustrates the colour in the system initial severity index and the colour of the data-points the colour in the new system. The plots and numbers provide average improvements through all patient groups and almost all statistical measures for blue and orange patients. However, the box-plots and statistical analysis do not show how the changes improved the patients' LOS.

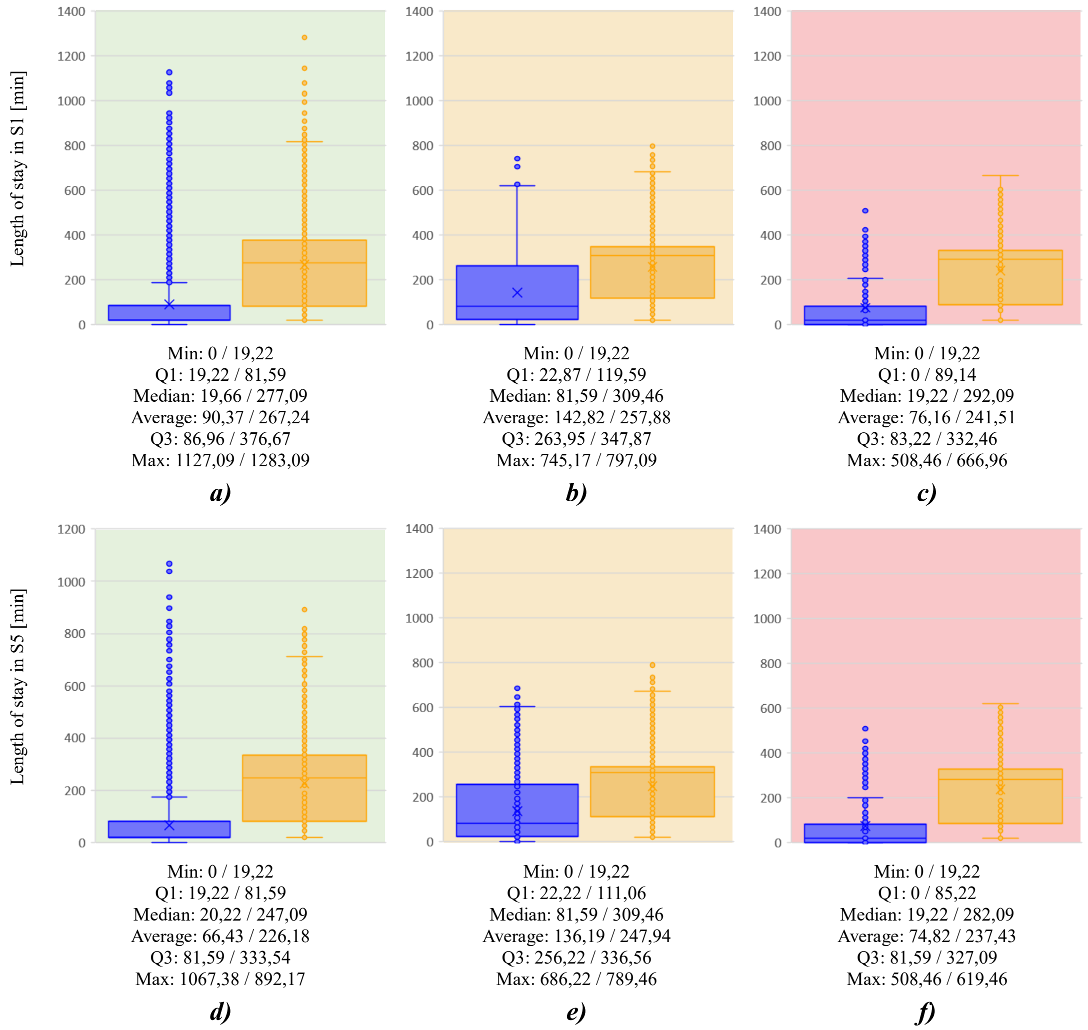


Figure 61: Length of stay per severity index in S1 (a, b, c) and S5 (d, e, f)

By providing the dispersion, as we introduced in Chapter 3, we can analyse the patients' LOS in dependence of their routing and reference. The comparison between scenarios and the reference still shows a discrepancy between the static value of the ideal system and the dynamic LOS from the simulation. To explore the origin, Figure 62 provides the dispersion introduced in Chapter 3. Again, the experiments of the bottom row in (a), (b) and (c) provide the LOS for S1 and the experiments (d), (e) and (f) provide the LOS for S2. In all plots, the red line illustrates the reference with the static LOS.

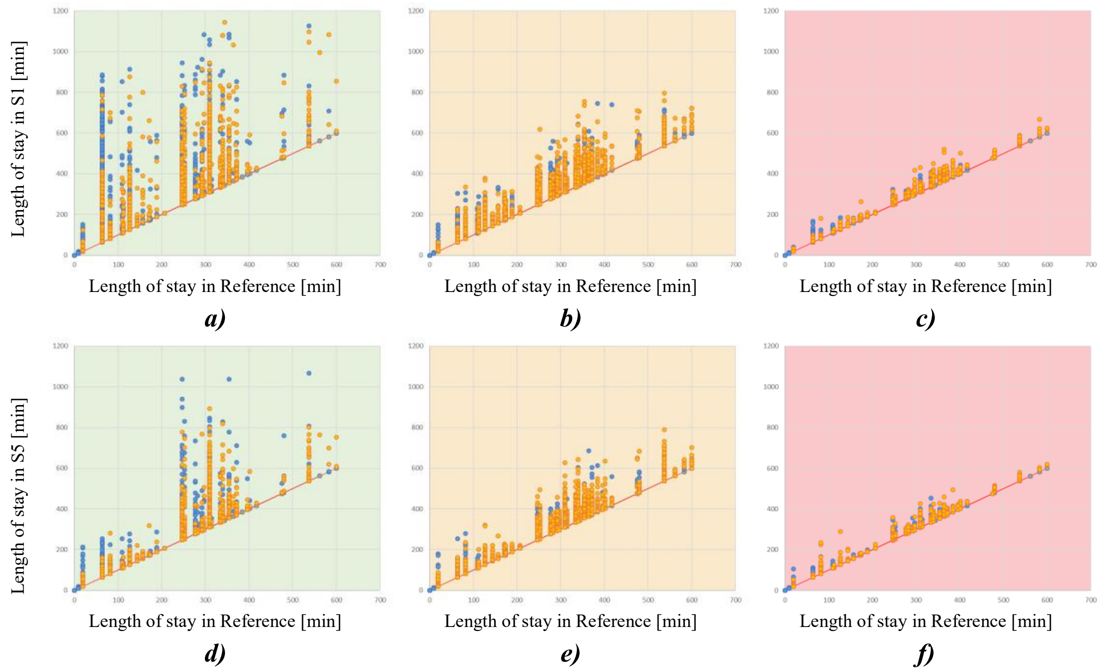


Figure 62: Dispersion for S1 (a, b, c) and S5 (d, e, f)

For the patients with an initially red severity index, there are minor improvements from (d) to (f). These patients benefit from the high priority level, owing to the final model change. In the group of the initially orange patients in (b) and (e), there are minor improvements visible, especially for the orange patients having static LOS values below 200 min. The most significant improvement is visible for the patients having an initially green severity index. From the initial situation in (a) to the future situation in (d), there is an obvious improvement for patients having static LOS values below 200 min. An improvement is also visible for patients with static LOS values above 200 min. Furthermore, in (d), the number of green patients with a dynamic LOS of more than 800 min strongly decreases. In S5, the correlation between priority level and LOS is less pronounced than in S1. For S1, with decreasing prioritization from red patients in (c), via orange patients in (b) to red patients in (a) the LOS degrades. Thus, in the initial situation, the green patients had long LOSs. In S5 with the new triage system, the waiting time of low-priority patients in blue is less pronounced.

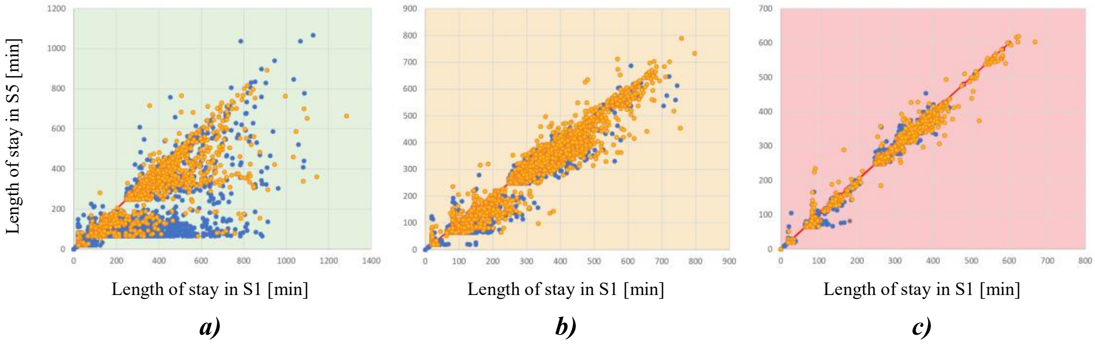


Figure 63: Improvement and degradation between S1 (a, b, c) and S5 (d, e, f)

A direct comparison between S1 and S5 is illustrated in Figure 63. The three subplots (a), (b) and (c) show the LOS for initially green, orange and red patients. In these plots, the diagonal line is not the reference of the ideal model, but describes patients having an identical LOS in S1 and S5. Consequently, patients that are plotted left/above this line are patients that have in S5 a longer LOS than in S1, and patients that are below/right this line are patients that have an improved LOS in S5. Sub-plots (b) and (c) show a slight shift to the right/below the reference. In particular, in (a), for the green patients there is just a minority of mostly blue patients that are in S5 worse than in S1. For the blue patients, significant improvement is visible for those with the short LOS, while for the orange patients, the major improvement is visible for those with a longer LOS.

5-4 Discussion

The leading question of this chapter is RQ3: **What are the methods for solving problems in the design of material flows by the digital shadow and data-driven simulation?** For addressing the question, simulation data (Chapter 1), libraries of partial models, methods for data-driven simulation (Chapter 2) and methods for optimization (Chapter 3) are assumed to be available. The methods to tackle problem-solving rely on the problem-solving loop, and are **data-driven simulation** from partial models and **multi-objective optimization** to provide data with links between action and evaluation parameters. With experimental data, ***providing the system limitation*** evaluates the feasibility of objectives, and ***extracting the contradictions*** bridges between optimization results and model change by providing technical and physical contradictions. ***Changing the model*** is the creative process of solving the problem. The methods for problem-solving are illustrated in red in Figure 64.

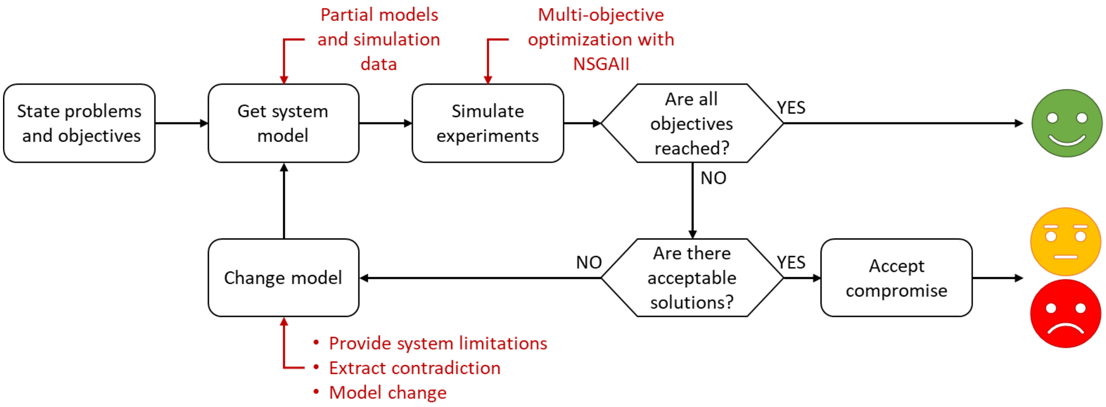


Figure 64: Methods of problem-solving

In the problem-solving loop, coupling data-driven simulation from partial models and simulation data with multi-objective optimization with the NSGA-II allows experimental data to be provided for analysis on request. The experimental data describes the link between decision variables and values for objective functions. Experiments within this data are located on the Pareto front of the solution space and provide optimum solutions for satisfying multiple objectives as well as trade-offs between these objectives. This allows the relevant experiments to be provided, leading to model change by highlighting the system of contradictions. A discussion of the methods was presented in the discussions in the corresponding chapters. The benefits and drawbacks of using different optimization means for problem-solving are as follows.

- **What-if scenarios** are appropriate for comparing scenarios. The method allows beneficial scenarios to be identified in a short time. The number of experiments is low, since the decision variables are given manually. Consequently, the amount of experimental data is limited. Additionally, it is unclear if the experiments are dominated or non-dominated.

- **Design of experiments** allows knowledge about the impact of each parameter to be provided. However, in systems with many parameters, the number of experiments and required time and computing power significantly increases. This is problematic and worsens when dealing with quantitative action parameters. A full factorial design of experiments is not possible in material flow systems. The identification of non-dominant solutions is a manual process.

- **Genetic algorithms** provided a minimum solution in the case studies in a short time. To deal with multiple objective functions, weighting of the objective functions is required. The weights compromise multiple objective functions by formulating a common objective function. Optimization for the compromised objective function does not provide data about the entire solution space. The solutions that are on the Pareto front do not represent the entire solution space.

- **Non-dominated sorting genetic algorithm II (NSGA-II)** is optimized for multiple objective functions. As a result, it provides numerous solutions representing different trade-offs between the multiple objective functions. This has an advantage for problem solving. First, there is no difficult weighting of objective functions required; second, the trade-offs between the objectives inherit the technical and physical contradictions that are required for changing the model and solving the problem.

When qualitative model change is required, the recommended optimization method in this thesis is optimization with multi-objective algorithms, particularly the non-dominated sorting algorithm. A comparison between the algorithms based on the criteria of definition of experiments, solution space and dominance of solutions is provided in Table 32. The grey background highlights the advantageous properties of the different optimization means.

Table 32: Comparison of optimization means

	What-if scenario	*(Full) Design of experiments*	*Genetic algorithm*	*Non-dominated sorting algorithm*
Experiment definition	**Low** number of specific experiments	**High** number of systematic experiments	**Low** number of computer-generated experiments	**Low** number of computer-generated experiments
Solution space	**Limited** to the specified experiments	**Entire** solution space	**Limited** to the (weighted objective function)	Representation of the **entire** solution space
Dominance of solution	**Unclear**	**Dominated and non-dominated** solutions	**Non-dominated** solutions	**Non-dominated** solutions

Providing the 'right' experiments for the creation of relevant data for problem-solving requires knowledge about the system and the definition of action parameters. Executing a design of experiment is often not possible, especially when there are numerous quantitative action parameters. Pareto optimization provides experimental data, including the conflicts of the system, by optimizing for multiple objective functions. In this way, Pareto optimization is capable of providing experimental data for problem-solving without knowledge about the system. The remaining challenge after optimization is solving the problem. Optimization, independent of the algorithm, provides experimental data for analysis. Additional methods were provided to solve problems through the analysis of data, particularly by highlighting system limitations, extracting contradictions, and changing the model.

Highlighting system performance limitations has the objective of evaluating the potential of model change and the feasibility of the objectives. To highlight the system limitations, the concept of ideality was introduced in Chapter 1 and was statically calculated. In Chapter 5, it was shown how simulation can facilitate and accelerate the process of determining the reference. In the case study of the ED, the reference was used to evaluate the improvement in performance for the different types of patients with different static references. Without the reference, the problem-solving would be based on the average values, with a loss of information about the distribution of discrepancies for each routing reference. When analysing the dispersion, the reference allowed us to evaluate how improvement activities impacted different types of patients. This level of detail in performance makes sense for the stakeholder, at least in the ED case, and we think that it could make sense for other problems as well.

Extracting contradictions bridges the gap between optimization and model change, which is a creative process. To bridge to and support model change, knowledge about the root causes is required. TRIZ uses the concept of contradictions to prepare for model change. To benefit from contradictions, this work proposed to detect technical contradictions to explain conflicts between experiments, detect physical contradictions to explain conflicts in the action parameters and, finally, provide the system of contradictions by stating the context of the conflict and the set of contradictions. The case study illustrated how this concept supports solving problems in the design of material flow systems. The extracted contradictions highlight the critical system mechanics and enable the search for inventive solutions. Without the concept of contradictions, the initial point for problem-solving is not obvious. Contradictions provided guidance on the way to model change.

Changing the model uses the concept of contradiction to solve problems. After providing the contradiction, TRIZ principles provided a start when searching for model change. For changing the model, this work used the separation principles: separation in space, separation in time and systemic separation. Specifically, it was used to change the assignment of stretchers (section 5-3.4), to change the responsibilities of nurses (section 5-3.5) and to provide a new system of prioritization policy (section 5-3.6). Model change is a creative task with high requirements in terms of expertise. TRIZ principle can guide the decision-maker to model change. However, separation principles were used only at the application level. When addressing real-world problems, additional TRIZ techniques can help to find solution concepts.

The methods provided a systematic approach to depart from experiment results and arrive at model change. Without these concepts, problem-solving would be driven by a long process of unsystematic data analysis and modelling of solutions. The systematic methods allowed us to solve the design problem in the case study by using inventive design methods within a few weeks, even as non-experts in emergency systems. To apply these methods, no expert knowledge is required. The time requirements without these methods are estimated to be higher.

Chapter 6 Conclusion

This work focused on problem-solving in the redesign of production systems, particularly on the improvement of material flows. To contribute to this global problem, in the introduction (Chapter 1) a theoretical framework for using digital shadow optimization and problem-solving was derived from the literature (see Figure 8). The core of this framework is the ability to simulate material flows and evaluate solution candidates in design on request. Instant evaluation of solution candidates enables using simulation to optimize material flows and solve problems in design. To enable simulation on request and use it in optimization and problem-solving, the following three research questions were stated.

RQ1: Modelling and Simulation:	**What are the methods for generating simulation models for problem-solving from the digital shadow?**
RQ2: Simulation and Optimization:	**What are the methods for optimizing material flows through the digital shadow and data-driven simulation?**
RQ3: Model Change:	**What are the methods for solving problems in the design of material flows by the digital shadow and data-driven simulation?**

These research questions were addressed by action research. To address the research questions, literature reviews were executed in each domain, and the relevant methods were derived, developed and implemented. To evaluate the methods, two industrial case-studies were executed. The purpose of these case studies was to evaluate the methods, assess the benefits and drawbacks and plan the next steps of the research to provide contributions and conclusions. The following sections present the contributions for each research question, and is followed by the conclusion and perspectives.

6-1 Contributions of RQ1: Modelling and Simulation

The question of modelling and simulation addressed two problems: the retrieval of simulation data, and data-driven simulation. In Chapter 2, this work presented the methods for extracting simulation data from historical event-logs of the digital shadow, and in Chapter 3, the methods for generating models of material flows and conducting experiments by data-driven simulation. The contributions are provided for data-retrieval and data-driven modelling in section 0 and sections -.

6-1.1 Data-retrieval

To address the question of retrieving simulation data, simulation was defined, based on a literature review and multiple case-studies. Based on the studied literature, the methods of this work were derived and developed. The methods addressed the extraction of historic product arrivals and activity maps. To enable stochastic simulation, methods for randomizing product arrivals and activity maps were presented. For providing data for stochastic simulation, random variants of the product arrivals and activity maps were given by using Monte Carlo simulation.

The application of the methods was presented in the case of the ED and provided two added values. First, a static analysis offered new knowledge about the system. It showed that the system is marked by a high degree of dynamicity and entails a large number of product variants with specific routings, which requires simulation for analysis. The routings enabled calculation of the static LOS (reference), which was crucial for the analysis in the solving chapters. Second, a static analysis provided simulation data, particularly the patient arrivals and activity maps for simulation. Both were provided as deterministic and stochastic variants and are crucial for data-driven simulation.

The initial literature review revealed that developing simulation data is time-consuming and is marked by a high degree of manual activities [1], [23]. The presented methods addressed both problems by extracting simulation data from event-logs.

Event-logs are available in workflow-oriented information systems, e.g. ERP, and were assumed as available in this work. The methods extract simulation data and support the process of setting up simulation models from the data. However, there are limitations regarding data quality and structure. Three following observations were made.

- High-quality data is absolutely necessary for providing valuable analyses. Ensuring sufficient data quality was a manual activity and major challenge in the case study. To reduce this effort, it is recommended to retrieve data from the shop floor by automated input rather than by manual input.
- Receiving assembly and dismantle activities from event-logs was a major challenge. The recommendation is to use object-centric event-logs to provide information about assembly and dismantling to reduce the effort for manually adding this information, as in the case studies, by using BPMN models.
- A dictionary with corresponding events and relevant resources was needed when mapping events and extracting activities. Providing and applying the dictionary are manual activities. The recommendation is to record additional attributes for each event to mark the beginning and end of activities and to record all involved technical and human resources.

6-1.2 Data-driven simulation

To address the question of data-driven simulation, a literature review was performed. The literature review provided the state of the art in enterprise modelling and automated modelling and simulation. Based on the review, the methods of this work were derived and developed. The methods use a library of partial models and customization to provide specific partial models. Data-driven simulation generates models and runs experiments by instantiating particular models of the system from partial models and simulation data. By reusing the concepts of partial and particular models of enterprise modelling and introducing customization, the methods handle the existing conflict in the literature between genericity and specificity. By using the concept of customizing partial models, the methods provide a trade-off by reusing modules and code via case-specific libraries. Partial models (modules) describe a class of problems or systems (in this work, job-shop production systems). Specific partial models inherit the case-specific behaviour (code) of a particular system. Introducing customization of partial models enables the use of generic partial models as a draft to simulate a wide range of systems by data-driven simulation from the data. The methods were evaluated in the following two case studies.

- In the ED, data-driven simulation was used to compare two scenarios in the design. Customization was exclusively used to implement the triage system and assign patients to stretchers and zones, based on the severity index.
- In the train remanufacturing system, data-driven simulation was used to evaluate a scheduling scenario. Customization was exclusively used to implement the technical constraints of the remanufacturing process.

The case studies showed that a common partial model is capable of describing the general behaviour of systems in different domains, despite the obvious differences. Furthermore, they highlighted that customization of partial models is capable of adapting the generic partial models and implementing the case-specific behaviour of particular systems. In this manner, the methods increase genericity and specificity of the generated models. Without reusing common partial models, the design of libraries for data-driven simulation must be done from scratch. After providing partial and specific partial models, data-driven simulation is entirely automatable. However, customization requires expert knowledge about system behaviour and expertise in modelling and simulation to implement this behaviour in partial models. In the future, new ways should be explored to provide case-specific behaviour in a generalized way to facilitate customization.

The case-study of the ED provided a new method to exploit simulation results by analysing the dispersion of the entire population of products. The method is appropriate for analysing particularly systems with a large variety of different variants and routings. As input, the method used the reference for the LOS and compared this with the simulated LOS in the dynamic system. The method yields added value by taking into consideration the different patient variants, rather than performing analyses using average values. In this way, it allowed the assessment of the impact of activities in the system design for different groups of patients. Owing to the common models, the method is also applicable to manufacturing systems, e.g. train design. However, it was not applied because of the low number of variants.

6-2 Contributions of RQ2: Simulation and Optimization

In Chapter 4, this work addressed the question of using data-driven simulation to optimize material flows. To address this question, a literature review was performed to understand how simulation can be applied to optimize systems. Based on the studied literature, the methods of this work were derived and developed. The proposed methods for handling optimization problems combine data-driven simulation and simulation-based optimization. In simulation-based optimization, an algorithm provides decision variables, and simulation evaluates the decision variables in a dynamic model to provide values for the objective function. The simulation inherits the constraints of the system and avoids building complex mathematical models.

Implementing the proposed methods allows, on request, the evaluation of solution concepts by data-driven simulation. Data-driven simulation represents the objective function in optimization. This enables the implementation of a wide range of scenarios, expressed by decision variables. Simulation and optimization in the literature uses parameterization to implement solution candidates, and the decision space is limited to changing values. Data-driven simulation enhances the decision space by its ability to generate new models based on changing decision parameters. In the case study of the ED, new forms of organizing the work were introduced via partial models.

In the case studies, data-driven simulation and optimization was used for solving three optimization problems (order scheduling, resource assignment, constrained optimization) in two cases (train remanufacturing, ED) by applying three different optimization means (design of experiments, optimization with GA, multi-objective optimization with Pareto optimization). The case studies showed that the method allows the substitution of the case study, optimization problem and optimization algorithm independently from each other as follows.

- Substituting the case study is enabled by replacing simulation data and partial models of the specific case.
- Substituting the optimization problem is enabled by providing vectors with decision variables, objective functions and optimization, as well as constraints between decision variables and objective functions.
- Substituting the optimization algorithm is enabled by using different methods to propose decision variables and process values for objective functions.

The ability to substitute case studies, optimization problems and algorithms individually provides an added value to optimization. The methods enable the solution of a wide range of optimization problems by the ability to use the right tools. Without these methods, optimization problems and models are to be stated individually.

6-3 Contributions of RQ3: Problem-Solving

In Chapter 5, this work addressed the question of solving problems in the design of material flows. To address this question, a literature review was performed to understand how optimization and inventive design are used to solve problems in the literature. Based on the studied literature, the methods of this work were derived and developed. Methods were provided on two levels. On the level of the problem-solving loop, data-driven simulation and optimization support problem-solving by providing experimental data with a link between decision vectors and objective vectors for inventive design. On the level of inventive design, methods were presented to highlight the system limitations by using the concept of the reference from the static analysis, to bridge between optimization and model change by extracting technical and physical contradictions and to change the model by TRIZ separation principles.

The case-study of the ED showed how these methods support problem-solving in the redesign of patient flows. In the case study, several iterations of the problem-solving loop were applied to continuously improve the system. Finally, the improvement of patients was achieved, while reducing the number of resources. In the case study, optimization and inventive design were applied. In the optimization of different methods, particularly what-if scenarios, the design of experiments and optimization were applied to improve the system by parameterization and to provide experimental data. In inventive design, the concept of model change was applied to add new action parameters and change the functionality of the systems. In the following loops, qualitative changes were considered in the optimization problem. The case study illustrated the design loop. With each improvement made, new problems appeared. Using optimization from the very beginning did not result in a solution to the problem, because the decision-maker was not able to state the objectives and decision variables. A loop was required to gather knowledge and add decision parameters.

In the loop, different optimization means were tested to provide experimental data for problem solving: what-if scenarios, design of experiments and optimization with GAs and NSGA-II. The optimization means were compared and highlighted, and NSGA-II was particularly advantageous for problem-solving. This algorithm offers three advantages: 1) it searches for minimum maximum solutions for multiple objective problems in the entire solution space; 2) the time required for optimization is much shorter compared with running a full design of experiments; and 3) it provides non-dominated solutions on the Pareto front. This allows the entire Pareto front to be explored in problem-solving.

The contributions of this chapter are, firstly, methods for providing experimental data with Pareto optimal solutions by data-driven simulation and multi-objective optimization and, secondly, methods to bridge between experimental data from optimization and model change by using the TRIZ concept of contradiction. Solving problems in design without these methods is time-consuming and has high requirements in terms of knowledge. The methods provide a systematic way to solve problems.

6-4 Summary

This section summarizes the contributions that were made and sets them in the context of the title of this work: **Redesigning production systems by their digital shadow**. The technical framework that was presented in Chapter 1 and is illustrated in Figure 65 serves to structure this section. In modelling and simulation (green) two topics were addressed: receiving simulation data from the digital shadow, and providing simulation models for problem-solving in the multi-domain space. The digital shadow provides event-logs with historical material flows. This data is available in any workflow-oriented information system, e.g. ERP. The methods of data-retrieval provide data for modelling and simulation. The methods of data-driven modelling provide models (in this work, material flows) in the multi-domain application space, i.e. the digital shadow environment. Data-driven modelling instantiates particular models of a specific system from partial models and simulation data. In the digital shadow environment, a variety of partial models can exist to analyse systems from different perspectives.

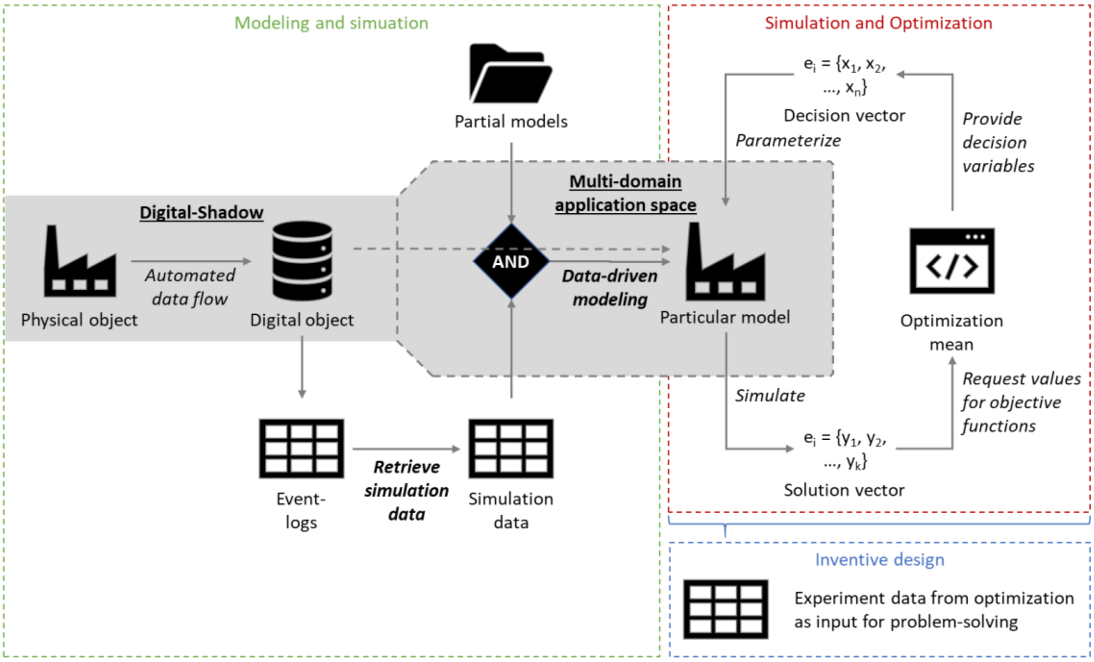


Figure 65: Technical framework

In inventive design, the digital shadow is used for optimization and model change. In simulation and optimization (red), optimization means provide decision vectors to be evaluated by simulation and request the corresponding objective vectors to minimize or maximize one or multiple objective functions. Data-driven simulation generates particular models from partial models and simulation data of the digital shadow and the decision vector of the optimization algorithm. Simulation of the material flows provides the objective vector for optimization means. The digital shadow becomes the enabler to optimize systems without the time-consuming and knowledge-intensive process of modelling, and can provide minimum and maximum solutions on request. Further optimization provides experimental data with a link between decision and objective vectors. When no satisfying solution is found in optimization, experimental data is the starting point for problem-solving with inventive design. Methods of data science close the gap between optimization and model change by providing technical and physical contradictions. However, changing the model remains a creative process. Providing contradictions is a systematic approach to provide model change, but the actual model change requires an expert.

6-5 Perspectives

Based on the summary in the previous section, this section provides directions for future research. Questions for future research exist mainly in modelling and simulation, and in inventive design. In modelling and simulation, added value can be achieved by enabling data-retrieval and data-driven simulation on request. In this scenario, the presented analysis and simulation results can be provided instantly from event-logs. However, the following challenges need to be overcome.

- Requirements regarding data-gathering from shop floor were provided. But they must be evaluated if these recommendations are to enable automation of the entire data-retrieval process.
- For simulation, partial models provide problem-specific behaviour. Partial models for job-shop systems were provided and validated in different domains. The degree to which these models can be used in further domains (e.g. shipyard, warehousing, flow-shop, etc.) should be tested.
- The bottleneck in data-driven simulation is the customization of partial models. The bottleneck is seen in the wide range of system specificities. It must be determined if these properties can be derived from the data and if there are general ways to provide them for simulation.

Assuming that the listed problems in modelling and simulation can be solved instantly for a wide range of cases and experimental data, inventive design can strongly benefit from availability of the data. However, the following open questions still remain in this domain.

- The concept of ideality (reference) and dispersion analysis strongly supports the analysis in the use-case and is recommended for systems with a high variance in products, such as emergency systems. It remains to be tested as to what degree this analysis can be applied in the remaining sectors.

- The methods for extracting contradictions in this work were executed manually. When dealing with experimental data with a large number of experiments, decision variables and objective functions, manual analysis is limited. Additional research is required to provide contradictions from big data.
- Changing the model to solve inventive design problems remains a creative process, executed by an expert. How the expert can be supported in inventive design beyond highlighting contractions should be explored. More research is required to discover alternative methods to solve design problems.

The entire approach of using simulation, optimization and inventive design strongly supported problem-solving in the case of the ED. To reliably understand the associated limitations, it is required to perform further case studies in additional domains.

Bibliography

1 A. Vieira, L. Dias, M. Santos, G. Pereira, and J. Oliveira, 'Setting an Industry 4.0 Research and Development Agenda for Simulation – a Literature Review', *International Journal of Simulation Modelling*, vol. 17, pp. 377–390, 2018, doi: 10.2507/IJSIMM17(3)429.

2 N. Bonnardel, 'Towards understanding and supporting creativity in design: analogies in a constrained cosnitive environment', 2000.

3 L. Lin, I. Rasovska, R. De Guio, and S. Dubois, 'Optimization Methods for Inventive Design', in *TRIZ – The Theory of Inventive Problem Solving*, D. Cavallucci, Ed., Cham: Springer International Publishing, 2017, pp. 151–185. doi: 10.1007/978-3-319-56593-4_7.

4 F. Z. Ben Moussa, R. De Guio, S. Dubois, I. Rasovska, and R. Benmoussa, 'Study of an innovative method based on complementarity between ARIZ, lean management and discrete event simulation for solving warehousing problems', *Computers & Industrial Engineering*, vol. 132, pp. 124–140, Jun. 2019, doi: 10.1016/j.cie.2019.04.024.

5 F. S. Hillier and G. J. Lieberman, *Introduction to operations research*, 6th ed. in McGraw-Hill series in industrial engineering and management science. New York: McGraw-Hill, 1995.

6 V. Goel and P. Pirolli, 'The structure of Design Problem Spaces', *Cognitive Science*, vol. 16, no. 3, pp. 395–429, Jul. 1992, doi: 10.1207/s15516709cog1603_3.

7 H. Chibane, S. Dubois, and R. De Guio, 'Innovation beyond optimization: Application to cutting tool design', *Computers & Industrial Engineering*, vol. 154, p. 107139, Apr. 2021, doi: 10.1016/j.cie.2021.107139.

8 S. Gao, J. Daaboul, and J. Le Duigou, 'Layout and scheduling optimization problem for a reconfigurable manufacturing system', *Int J Ind Eng Manag*, vol. 12, no. 3, pp. 174–186, Sep. 2021, doi: 10.24867/IJIEM-2021-3-286.

9 X. Luo, Q. Qian, and Y. F. Fu, 'Improved Genetic Algorithm for Solving Flexible Job Shop Scheduling Problem', *Procedia Computer Science*, vol. 166, pp. 480–485, 2020, doi: https://doi.org/10.1016/j.procs.2020.02.061.

10 G. S. Al'tšuller, L. Shulyak, and S. Rodman, *The innovation algorithm: TRIZ, systematic innovation and technical creativity*, 2. ed. Worcester, Mass: Technical Innovation Center, 2007.

11 M. A. Orloff, *Modern TRIZ Modeling in Master Programs: Introduction to TRIZ Basics at University and Industry*. Cham: Springer International Publishing, 2020. doi: 10.1007/978-3-030-37417-4.

12 V. Petrov, *TRIZ. Theory of Inventive Problem Solving: Level 1*. Cham: Springer International Publishing, 2019. doi: 10.1007/978-3-030-04254-7.

13 G. S. Al'tshuller, *Creativity as an exact science: the theory of the solution of inventive problems*. in Studies in cybernetics, no. v. 5. New York: Gordon and Breach Science Publishers, 1984.

14 S. Dubois, R. DE GUIO, and I. Rasovska, 'Resolution of Inventive Problems: Different Kind of Mechanisms', *Int. J. Systematic Innovation*, vol. 1, pp. 2–10, Jan. 2011.

15 F. Z. Ben Moussa, S. Dubois, R. De Guio, I. Rasovska, and R. Benmoussa, 'Integrating the Theory of Inventive Problem Solving with Discrete Event Simulation in Supply Chain Management', in *Automated Invention for Smart Industries*, D. Cavallucci, R. De Guio, and S. Koziołek, Eds., in IFIP Advances in Information and Communication Technology, vol. 541. Cham: Springer International Publishing, 2018, pp. 330–347. doi: 10.1007/978-3-030-02456-7_27.

16 N. Khomenko, R. D. Guio, L. Lelait, and I. Kaikov, 'A framework for OTSM TRIZ-based computer support to be used in complex problem management', *IJCAT*, vol. 30, no. 1/2, p. 88, 2007, doi: 10.1504/IJCAT.2007.015700.

17 F. Aggogeri, R. Faglia, M. Mazzola, and A. Merlo, 'Automating the Simulation of SME Processes through a Discrete Event Parametric Model', *International Journal of Engineering Business Management*, vol. 7, p. 4, Feb. 2015, doi: 10.5772/59958.

18 R. Al-Aomar, O. M. Ülgen, and E. Williams, *Process simulation using WITNESS*. Hoboken, New Jersey: John Wiley & Sons, Inc, 2015.

19 J. Wy *et al.*, 'A data-driven generic simulation model for logistics-embedded assembly manufacturing lines', *Computers & Industrial Engineering*, vol. 60, no. 1, pp. 138–147, Feb. 2011, doi: 10.1016/j.cie.2010.10.011.

20 M. Schlecht, R. de Guio, and J. Köbler, 'Automated generation of simulation model in context of industry 4.0', *International Journal of Modelling and Simulation*, pp. 1–13, May 2023, doi: 10.1080/02286203.2023.2206075.

21 A. Skoogh and B. Johansson, *TIME-CONSUMPTION ANALYSIS OF INPUT DATA ACTIVITIES IN DISCRETE EVENT SIMULATION PROJECTS*, vol. The Swedish Production Symposium. in The Swedish Production Symposium, no. 1, vol. The Swedish Production Symposium. Göteborg, 2007.

22 N. Robertson and T. Perera, 'Automated data collection for simulation?', *Simulation Practice and Theory*, vol. 9, no. 6, pp. 349–364, May 2002, doi: 10.1016/S0928-4869(01)00055-6.

23 A. Skoogh, T. Perera, and B. Johansson, ‘Input data management in simulation – Industrial practices and future trends’, *Simulation Modelling Practice and Theory*, vol. 29, pp. 181–192, Dec. 2012, doi: 10.1016/j.simpat.2012.07.009.

24 H. Reinhardt *et al.*, ‘Retrieving properties of manufacturing systems from traceability data for performance evaluation and material flow simulation’, *Procedia CIRP*, vol. 104, pp. 20–25, 2021, doi: 10.1016/j.procir.2021.11.002.

25 H. Reinhardt, M. Weber, and M. Putz, ‘A Survey on Automatic Model Generation for Material Flow Simulation in Discrete Manufacturing’, *Procedia CIRP*, vol. 81, pp. 121–126, Jan. 2019, doi: 10.1016/j.procir.2019.03.022.

26 R. Drath, *Datenaustausch in der Anlagenplanung mit AutomationML: Integration von CAEX, PLCopen XML und COLLADA*. Heidelberg; New York: Springer, 2010. Accessed: Nov. 25, 2019. [Online]. Available: http://public.eblib.com/choice/publicfullrecord.aspx?p=510658

27 AutomationML consortium, *Whitepaper AutomationML*. 2016.

28 Simulation Interoperability Standards Organization, *Standard for: Core Manufacturing Simulation Data*. 2010.

29 Y. Liu, H. Zhang, C. Li, and R. J. Jiao, ‘Workflow simulation for operational decision support using event graph through process mining’, *Decision Support Systems*, vol. 52, no. 3, pp. 685–697, Feb. 2012, doi: 10.1016/j.dss.2011.11.003.

30 J. Bokrantz, A. Skoogh, D. La¨mkull, A. Hanna, and T. Perera, ‘Data quality problems in discrete event simulation of manufacturing operations’, *Simulation: Transactions of the Society for Modeling and Simulation International 1–17*, vol. 94, pp. 1009–1025, Nov. 2018, doi: 10.1177/0037549717742954.

31 G. Ćwikła, ‘Methods of Manufacturing Data Acquisition for Production Management - A Review’, *AMR*, vol. 837, pp. 618–623, Nov. 2013, doi: 10.4028/www.scientific.net/AMR.837.618.

32 M. Grieves, ‘Origins of the Digital Twin Concept’, 2016, doi: 10.13140/RG.2.2.26367.61609.

33 Gartner, Inc., ‘Gartner’s Top 10 Strategic Technology Trends for 2017’, *Gartner*, 2017. //www.gartner.com/smarterwithgartner/gartners-top-10-technology-trends-2017/ (accessed Oct. 13, 2020).

34 Gartner, Inc., ‘Gartner’sTop 10 Strategic Technology Trends for 2019’, *Gartner*, 2019. //www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2019/ (accessed Oct. 13, 2020).

35 Gartner, Inc., 'Gartner's Top 10 Strategic Technology Trends for 2018', *Gartner*, 2018. //www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2018/ (accessed Oct. 13, 2020).

36 W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, 'Digital Twin in manufacturing: A categorical literature review and classification', *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1016–1022, 2018, doi: 10.1016/j.ifacol.2018.08.474.

37 R. Wagner, B. Schleich, B. Haefner, A. Kuhnle, S. Wartzack, and G. Lanza, 'Challenges and Potentials of Digital Twins and Industry 4.0 in Product Design and Production for High Performance Products', *Procedia CIRP*, vol. 84, pp. 88–93, 2019, doi: 10.1016/j.procir.2019.04.219.

38 ISO 19439, 'Enterprise integration — Framework for enterprise modelling', 2006.

39 B. Somekh, *Action research: a methodology for change and development*. in Doing qualitative research in educational settings. Maidenhead: Open University Press, 2006.

40 R. F. Singer, A. A. Infante, C. C. Oppenheimer, C. A. West, and B. Siegel, 'The Use of and Satisfaction with the Emergency Severity Index', *Journal of Emergency Nursing*, vol. 38, no. 2, pp. 120–126, Mar. 2012, doi: 10.1016/j.jen.2010.07.004.

41 Young Jun Son, A. T. Jones, and R. A. Wysk, 'Automatic generation of simulation models from neutral libraries: an example', in *2000 Winter Simulation Conference Proceedings (Cat. No.00CH37165)*, Orlando, FL, USA: IEEE, 2000, pp. 1558–1567. doi: 10.1109/WSC.2000.899140.

42 Y. J. Son and R. A. Wysk, 'Automatic simulation model generation for simulation-based, real-time shop floor control', *Computers in Industry*, vol. 45, no. 3, pp. 291–308, Jul. 2001, doi: 10.1016/S0166-3615(01)00086-0.

43 Y. J. Son, A. T. Jones, and R. A. Wysk, 'Component based simulation modeling from neutral component libraries', *Computers & Industrial Engineering*, vol. 45, no. 1, pp. 141–165, Jun. 2003, doi: 10.1016/S0360-8352(03)00023-8.

44 S. S. Choi, T. H. Yoon, and S. D. Noh, 'XML-based neutral file and PLM integrator for PPR information exchange between heterogeneous PLM systems', *International Journal of Computer Integrated Manufacturing*, vol. 23, no. 3, pp. 216–228, Mar. 2010, doi: 10.1080/09511920903443234.

45 M. Ahmad, B. R. Ferrer, B. Ahmad, D. Vera, J. L. Martinez Lastra, and R. Harrison, 'Knowledge-based PPR modelling for assembly automation', *CIRP Journal of Manufacturing Science and Technology*, vol. 21, pp. 33–46, May 2018, doi: 10.1016/j.cirpj.2018.01.001.

46 B. Ramis Ferrer, B. Ahmad, D. Vera, A. Lobov, R. Harrison, and J. L. Martínez Lastra, ‘Product, process and resource model coupling for knowledge-driven assembly automation’, *at - Automatisierungstechnik*, vol. 64, no. 3, Jan. 2016, doi: 10.1515/auto-2015-0073.

47 M. Lütjen, *Modellierungskonzept zur integrierten Planung und Simulation von Produktionsszenarien entwickelt am Beispiel der CFK-Serienfertigung*. 2014. [Online]. Available: https://elib.suub.uni-bremen.de/edocs/00103681-1.pdf

48 C. Johnsson, *ISA 95 - how and where can it be applied*. 2004.

49 L. Prades, F. Romero, A. Estruch, A. García-Dominguez, and J. Serrano, ‘Defining a Methodology to Design and Implement Business Process Models in BPMN According to the Standard ANSI/ISA-95 in a Manufacturing Enterprise’, *Procedia Engineering*, vol. 63, pp. 115–122, 2013, doi: 10.1016/j.proeng.2013.08.283.

50 G. Popovics and L. Monostori, ‘ISA Standard Simulation Model Generation Supported by Data Stored in Low Level Controllers’, *Procedia CIRP*, vol. 12, pp. 432–437, 2013, doi: 10.1016/j.procir.2013.09.074.

51 D. Weigert, P. Aurich, T. Reggelin, and O.-G.-U. Magdeburg, ‘Durchgehende Modellerstellung zwischen Simulations-, Visualisierungs- und Konstruktionswerkzeugen für die gesamtheitliche Planung von Produktions- und Intralogistiksystemen’, p. 10, 2017.

52 V. Rudtsch, F. Bauer, and J. Gausemeier, ‘Approach for the Conceptual Design Validation of Production Systems using Automated Simulation-Model Generation’, *Procedia Computer Science*, vol. 16, pp. 69–78, Jan. 2013, doi: 10.1016/j.procs.2013.01.008.

53 S. Bergmann, S. Stelzer, S. Wüstemann, and S. Strassburger, ‘Model generation in SLX using CMSD and XML stylesheet transformations’, in *Proceedings of the 2012 Winter Simulation Conference (WSC)*, Dec. 2012, pp. 1–11. doi: 10.1109/WSC.2012.6464981.

54 S. Bergmann and S. Straßburger, ‘On the use of the Core Manufacturing Simulation Data (CMSD) standard: experiences and recommendations’, in *Fall Simulation Interoperability Workshop 2015*, Orlando, USA, Nov. 2015. [Online]. Available: http://uri.gbv.de/document/gvk:ppn:839964773

55 T. Frank *et al.*, ‘Agent-based communication to map and exchange shop floor data between MES and material flow simulation based on the open standard CMSD’, *IFAC-PapersOnLine*, vol. 49, no. 12, pp. 1526–1531, Jan. 2016, doi: 10.1016/j.ifacol.2016.07.796.

56 N. Furian, M. O'Sullivan, C. Walker, S. Vössner, and D. Neubacher, 'A conceptual modeling framework for discrete event simulation using hierarchical control structures', *Simulation Modelling Practice and Theory*, vol. 56, pp. 82–96, Aug. 2015, doi: 10.1016/j.simpat.2015.04.004.

57 O. Schonherr and O. Rose, *Modelling of Production Systems with SysML*. 2010.

58 N. Brunner, *Entwicklung eines Plattformkonzeptes für die Planung von Montagesystemen*. Aachen: Shaker Verlag, 2016.

59 C. Jonas, *Konzept einer durchgängigen, rechnergestützten Planung von Montageanlagen*. in Forschungsberichte IWB, no. Bd. 145. München: Utz, 2000.

60 S. Klauke, *Methoden und Datenmodell der 'Offenen Virtuellen Fabrik' zur Optimierung simultaner Produktionsprozesse*, Als Ms. gedr. in Fortschritt-Berichte VDI Reihe 20, Rechnerunterstützte Verfahren, no. 360. Düssdeldorf: VDI-Verl, 2002.

61 P. Steinwasser, *Modulares Informationsmanagement in der integrierten Produkt- und Prozeßplanung*. in Fertigungstechnik - Erlangen, no. 63. Bamberg: Meisenbach, 1996.

62 G. Lugaresi and A. Matta, 'REAL-TIME SIMULATION IN MANUFACTURING SYSTEMS: CHALLENGES AND RESEARCH DIRECTIONS', in *2018 Winter Simulation Conference (WSC)*, Gothenburg, Sweden: IEEE, Dec. 2018, pp. 3319–3330. doi: 10.1109/WSC.2018.8632542.

63 G. Schuh *et al.*, 'Produktionsdaten als Enabler für Industrie 4.0/Production data as enabler for Industrie 4.0', *wt*, vol. 105, no. 04, pp. 200–203, 2015, doi: 10.37544/1436-4980-2015-04-42.

64 P. Goodall, R. Sharpe, and A. West, 'A data-driven simulation to support remanufacturing operations', *Computers in Industry*, vol. 105, pp. 48–60, Feb. 2019, doi: 10.1016/j.compind.2018.11.001.

65 R. G. Sharpe, P. A. Goodall, A. D. Neal, P. P. Conway, and A. A. West, 'Cyber-Physical Systems in the re-use, refurbishment and recycling of used Electrical and Electronic Equipment', *Journal of Cleaner Production*, vol. 170, pp. 351–361, Jan. 2018, doi: 10.1016/j.jclepro.2017.09.087.

66 W. M. P. Van Der Aalst, 'Object-Centric Process Mining: Unraveling the Fabric of Real Processes', *Mathematics*, vol. 11, no. 12, p. 2691, Jun. 2023, doi: 10.3390/math11122691.

67 G. Lugaresi and A. Matta, 'Discovery and digital model generation for manufacturing systems with assembly operations', in *2021 IEEE 17th International*

Conference on Automation Science and Engineering (CASE), Lyon, France: IEEE, Aug. 2021, pp. 752–757. doi: 10.1109/CASE49439.2021.9551479.

68 G. Lugaresi and A. Matta, 'Automated manufacturing system discovery and digital twin generation', *Journal of Manufacturing Systems*, vol. 59, pp. 51–66, Apr. 2021, doi: 10.1016/j.jmsy.2021.01.005.

69 G. Lugaresi and A. Matta, 'Automated digital twin generation of manufacturing systems with complex material flows: graph model completion', *Computers in Industry*, vol. 151, p. 103977, Oct. 2023, doi: 10.1016/j.compind.2023.103977.

70 G. Lugaresi and A. Matta, 'Generation and Tuning of Discrete Event Simulation Models for Manufacturing Applications', in *2020 Winter Simulation Conference (WSC)*, Orlando, FL, USA: IEEE, Dec. 2020, pp. 2707–2718. doi: 10.1109/WSC48552.2020.9383870.

71 G. Lugaresi and A. Matta, 'Automated Digital Twins Generation for Manufacturing Systems: a Case Study', *IFAC-PapersOnLine*, vol. 54, no. 1, pp. 749–754, 2021, doi: 10.1016/j.ifacol.2021.08.087.

72 S. Kursun-Bahadir and F. Kalaoğlu, 'Simulation of Production Line Balancing in Apparel Manufacturing', *Fibres & Textiles in Eastern Europe*, vol. Nr 4 (75), pp. 68–71, 2009.

73 A. M. Law and M. G. McComas, 'How the ExpertFit distribution-fitting software can make your simulation models more valid', in *Proceeding of the 2001 Winter Simulation Conference (Cat. No.01CH37304)*, Arlington, VA, USA: IEEE, 2001, pp. 256–261. doi: 10.1109/WSC.2001.977281.

74 D. Swanson, 'A Simulation-Based Process Model for Managing Drone Deployment to Minimize Total Delivery Time', *IEEE Eng. Manag. Rev.*, vol. 47, no. 3, pp. 154–167, Sep. 2019, doi: 10.1109/EMR.2019.2926245.

75 M. Kumbhar, A. H. C. Ng, and S. Bandaru, 'A digital twin based framework for detection, diagnosis, and improvement of throughput bottlenecks', *Journal of Manufacturing Systems*, vol. 66, pp. 92–106, Feb. 2023, doi: 10.1016/j.jmsy.2022.11.016.

76 J. L. Lay, J. Neveu, B. Dalmas, and V. Augusto, 'Automated Generation of Patient Population for Discrete-Event Simulation Using Process Mining', in *2022 Annual Modeling and Simulation Conference (ANNSIM)*, San Diego, CA, USA: IEEE, Jul. 2022, pp. 42–53. doi: 10.23919/ANNSIM55834.2022.9859406.

77 J. L. Lay *et al.*, 'COVID-19 Bed Management Using a Two-Step Process Mining and Discrete-Event Simulation Approach', *IEEE Trans. Automat. Sci. Eng.*, pp. 1–0, 2023, doi: 10.1109/TASE.2023.3274847.

78 M. Kumbhar, A. H. C. Ng, and S. Bandaru, 'Bottleneck Detection Through Data Integration, Process Mining and Factory Physics-Based Analytics', in *Advances in Transdisciplinary Engineering*, A. H. C. Ng, A. Syberfeldt, D. Högberg, and M. Holm, Eds., IOS Press, 2022. doi: 10.3233/ATDE220192.

79 Y. Wang, G. Zacharewicz, M. K. Traore, and D. Chen, 'A tool for mining discrete event simulation model', in *2017 Winter Simulation Conference (WSC)*, Las Vegas, NV: IEEE, Dec. 2017, pp. 3066–3077. doi: 10.1109/WSC.2017.8248027.

80 Y. Wang, G. Zacharewicz, M. K. Traoré, and D. Chen, 'Use of Fuzzy Clustering for Discrete Event Simulation Model Construction', Mar. 2018.

81 I. Cheaitou, A. A. E. Cadi, A. Bekrar, D. Duvivier, and A. Sahili, 'Modeling and simulation of patient flow at the emergency services : Case of Al-Zahraa Hospital University Medical Center in Lebanon', presented at the 10ème conférence Francophone en Gestion et Ingénierie des Systèmes Hospitaliers, GISEH2020, Oct. 2020. Accessed: Sep. 28, 2021. [Online]. Available: https://hal-uphf.archives-ouvertes.fr/hal-03199182

82 T. Cokelaer *et al.*, 'cokelaer/fitter: v1.5.2'. Zenodo, Jan. 01, 2023. doi: 10.5281/ZENODO.7497983.

83 T. Sauerbier and O. Mildenberger, 'Ablauf einer Simulationsstudie', in *Theorie und Praxis von Simulationssystemen*, O. Mildenberger, Ed., Wiesbaden: Vieweg+Teubner Verlag, 1999, pp. 7–16. doi: 10.1007/978-3-322-90773-8_3.

84 Verein deutscher Ingenieure, *VDI 3633 Blatt 1*. 2014.

85 K. Gutenschwager, M. Rabe, S. Spieckermann, and S. Wenzel, *Simulation in Produktion und Logistik*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017. doi: 10.1007/978-3-662-55745-7.

86 W. Kühn, *Digitale Fabrik: Fabriksimulation für Produktionsplaner*. München: Hanser, 2006.

87 C.-G. Grundig, *Fabrikplanung: Planungssystematik - Methoden - Anwendungen*. 2018.

88 C. McLean, A. Jones, T. Lee, and F. Riddick, 'An architecture for a generic data-driven machine shop simulator', in *Proceedings of the Winter Simulation Conference*, San Diego, CA, USA: IEEE, 2002, pp. 1108–1116. doi: 10.1109/WSC.2002.1166364.

89 G. van Hulzen, N. Martin, B. Depaire, and G. Souverijns, 'Supporting capacity management decisions in healthcare using data-driven process simulation', *Journal of Biomedical Informatics*, vol. 129, p. 104060, May 2022, doi: 10.1016/j.jbi.2022.104060.

90 D. Krenczyk, 'Automatic Generation Method of Simulation Model for Production Planning and Simulation Systems Integration', *AMR*, vol. 1036, pp. 825–829, Oct. 2014, doi: 10.4028/www.scientific.net/AMR.1036.825.

91 D. Krenczyk and G. Bocewicz, 'Data-Driven Simulation Model Generation for ERP and DES Systems Integration', in *Intelligent Data Engineering and Automated Learning – IDEAL 2015*, K. Jackowski, R. Burduk, K. Walkowiak, M. Wozniak, and H. Yin, Eds., in Lecture Notes in Computer Science, vol. 9375. Cham: Springer International Publishing, 2015, pp. 264–272. doi: 10.1007/978-3-319-24834-9_32.

92 N. Wu and M. Zhou, 'Process vs resource-oriented Petri net modeling of automated manufacturing systems', *Asian Journal of Control*, vol. 12, no. 3, pp. 267–280, Apr. 2010, doi: 10.1002/asjc.184.

93 A. Hubl, K. Altendorfer, H. Jodlbauer, M. Gansterer, and R. F. Hartl, 'Flexible model for analyzing production systems with discrete event simulation', in *Proceedings of the 2011 Winter Simulation Conference (WSC)*, Phoenix, AZ, USA: IEEE, Dec. 2011, pp. 1554–1565. doi: 10.1109/WSC.2011.6147873.

94 T. Felberbauer, K. Altendorfer, and A. Hubl, 'Using a scalable simulation model to evaluate the performance of production system segmentation in a combined MRP and kanban system', in *Proceedings Title: Proceedings of the 2012 Winter Simulation Conference (WSC)*, Berlin, Germany: IEEE, Dec. 2012, pp. 1–12. doi: 10.1109/WSC.2012.6465053.

95 A. J. Peirleitner, K. Altendorfer, and T. Felberbauer, 'Simulation based manufacturing system improvement focusing on capacity and mrp decisions — A practical case from mechanical engineering', in *2017 Winter Simulation Conference (WSC)*, Las Vegas, NV: IEEE, Dec. 2017, pp. 3876–3887. doi: 10.1109/WSC.2017.8248098.

96 T. Felberbauer, K. Altendorfer, and H. Jodlbauer, 'Linking deterministic aggregate production planning with a stochastic simulation model using a planned utilisation factor', in *ASIM Dedicated Conference on Simulation in Production and Logistics*, Paderborn: HNI-Verlagsschriftenreihe, 2013, pp. 621–630.

97 K. Altendorfer, 'Influence of lot size and planned lead time on service level and inventory for a single-stage production system with advance demand information and random required lead times', *International Journal of Production Economics*, vol. 170, pp. 478–488, Dec. 2015, doi: 10.1016/j.ijpe.2015.07.030.

98 K. Altendorfer and S. Minner, 'A comparison of make-to-stock and make-to-order in multi-product manufacturing systems with variable due dates', *IIE Transactions*, vol. 46, no. 3, pp. 197–212, Mar. 2014, doi: 10.1080/0740817X.2013.803638.

99 J. Karder, A. Beham, V. A. Hauder, K. Altendorfer, and M. Affenzeller, 'Simulation-based optimisation for worker cross-training', *IJSPM*, vol. 16, no. 3, p. 185, 2021, doi: 10.1504/IJSPM.2021.117309.

100 A. Thomas and P. Charpentier, 'Reducing simulation models for scheduling manufacturing facilities', *European Journal of Operational Research*, vol. 161, no. 1, pp. 111–125, Feb. 2005, doi: 10.1016/j.ejor.2003.08.042.

101 H. EL HAOUZI and A. Thomas, 'A Methodological Approach to Build Simulation Models of Manufacturing Systems with Distributed Control', in *International Conference on Industrial Engineering and Systems Management IESM 2005*, Marrakech (Morocco), Apr. 2019.

102 M. Schlecht, S. Himmiche, V. Goepp, R. De Guio, and J. Köbler, 'Data-driven decision process for robust scheduling of remanufacturing systems', *IFAC-PapersOnLine*, vol. 55, no. 10, pp. 755–760, 2022, doi: 10.1016/j.ifacol.2022.09.500.

103 H. de Swaan Arons, 'Knowledge-based modeling of discrete-event simulation systems', in *Proceedings of the 31st conference on Winter simulation Simulation---a bridge to the future - WSC '99*, Phoenix, Arizona, United States: ACM Press, 1999, pp. 591–597. doi: 10.1145/324138.324441.

104 H. Arons and C. Boer, 'Storage and retrieval of discrete-event simulation models', *Simulation Practice and Theory*, vol. 8, no. 8, pp. 555–576, Jul. 2001, doi: 10.1016/S0928-4869(01)00036-2.

105 H. Arons and C. Boer, 'RETRIEVING PARAMETERIZED SIMULATION MODELS FROM A DATABASE', in *PROCEEDINGS OF EUROSIM 2001*, Delft, Netherlands, 2001, p. 7.

106 C. Heavey *et al.*, 'Development of an open-source Discrete Event Simulation cloud enabled platform', in *Proceedings of the Winter Simulation Conference 2014*, Savanah, GA, USA: IEEE, Dec. 2014, pp. 2824–2835. doi: 10.1109/WSC.2014.7020124.

107 P. Barlas, C. Heavey, and G. Dagkakis, 'An Open Source Tool for Automated Input Data in Simulation', *Int. j. simul. model.*, vol. 14, no. 4, pp. 596–608, Dec. 2015, doi: 10.2507/IJSIMM14(4)3.306.

108 P. Barlas and C. Heavey, *Towards Automated Simulation Input Data*. 2014.

109 O. Olaitan *et al.*, 'Implementing ManPy, a Semantic-free Open-source Discrete Event Simulation Package, in a Job Shop', *Procedia CIRP*, vol. 25, pp. 253–260, 2014, doi: 10.1016/j.procir.2014.10.036.

110 S. Bergmann, *Automatische Generierung adaptiver Modelle zur Simulation von Produktionssystemen*. Ilmenau: Univ.-Verl, 2014. [Online]. Available: https://nbn-resolving.org/urn:nbn:de:gbv:ilm1-2013000530

111 S. Bergmann, A. Fiedler, and S. Straßburger, 'Generierung und Integration von Simulationsmodellen unter Verwendung des Core Manufacturing Simulation Data (CMSD) Information Model', p. 8, 2010.

112 S. Bergmann and S. Straßburger, 'Automatische Modellgenerierung – Stand, Klassifizierung und ein Anwendungsbeispiel', in *Ablaufsimulation in der Automobilindustrie*, G. Mayer, C. Pöge, S. Spieckermann, and S. Wenzel, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2020, pp. 333–347. doi: 10.1007/978-3-662-59388-2_23.

113 G. Zülch and W. Ernst, 'SIMULAST — ein personalbezogenes Simulationsverfahren zur Planung von Arbeitsstrukturen', in *Simulationstechnik*, W. Ameling, Ed., in Informatik-Fachberichte, vol. 179. Berlin, Heidelberg: Springer Berlin Heidelberg, 1988, pp. 470–475. doi: 10.1007/978-3-642-74051-0_69.

114 G. Zülch, 'Staff oriented simulation of new production systems', *International Journal of Production Economics*, vol. 24, no. 1–2, pp. 199–206, Nov. 1991, doi: 10.1016/0925-5273(91)90167-R.

115 G. Zülch, J. Krüger, H. Schindele, and S. Rottinger, 'Simulation-aided planning of quality-oriented personnel structures in production systems', *Applied Ergonomics*, vol. 34, no. 4, pp. 293–301, Jul. 2003, doi: 10.1016/S0003-6870(03)00053-X.

116 M. Leupold, M. van Hall, K. Höfer, and G. Zülch, 'Simulation Analysis of Alternative Personnel Structures in the Shipping Division of a Tinplate Manufacturer', *Procedia Manufacturing*, vol. 39, pp. 583–590, 2019, doi: 10.1016/j.promfg.2020.01.424.

117 L. Tiacci, 'Event and object oriented simulation to fast evaluate operational objectives of mixed model assembly lines problems', *Simulation Modelling Practice and Theory*, vol. 24, pp. 35–48, May 2012, doi: 10.1016/j.simpat.2012.01.004.

118 L. Tiacci, 'Simultaneous balancing and buffer allocation decisions for the design of mixed-model assembly lines with parallel workstations and stochastic task times', *International Journal of Production Economics*, vol. 162, pp. 201–215, Apr. 2015, doi: 10.1016/j.ijpe.2015.01.022.

119 L. Tiacci, 'Coupling a genetic algorithm approach and a discrete event simulator to design mixed-model un-paced assembly lines with parallel workstations and stochastic task times', *International Journal of Production Economics*, vol. 159, pp. 319–333, Jan. 2015, doi: 10.1016/j.ijpe.2014.05.005.

120 J. Wang, Q. Chang, G. Xiao, N. Wang, and S. Li, 'Data driven production modeling and simulation of complex automobile general assembly plant', *Computers in Industry*, vol. 62, no. 7, pp. 765–775, Sep. 2011, doi: 10.1016/j.compind.2011.05.004.

121 G. Popovics, A. Pfeiffer, B. Kádár, Z. Vén, L. Kemény, and L. Monostori, 'Automatic Simulation Model Generation Based on PLC Codes and MES Stored Data', *Procedia CIRP*, vol. 3, pp. 67–72, 2012, doi: 10.1016/j.procir.2012.07.013.

122 G. Popovics, A. Pfeiffer, and L. Monostori, 'Generic data structure and validation methodology for simulation of manufacturing systems', *International Journal of Computer Integrated Manufacturing*, vol. 29, no. 12, pp. 1272–1286, Dec. 2016, doi: 10.1080/0951192X.2016.1187296.

123 B. R. Ferrer, B. Ahmad, A. Lobov, D. A. Vera, J. L. M. Lastra, and R. Harrison, 'An approach for knowledge-driven product, process and resource mappings for assembly automation', in *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, Gothenburg, Sweden: IEEE, Aug. 2015, pp. 1104–1109. doi: 10.1109/CoASE.2015.7294245.

124 I. A. Friedewald and D.-I. L. Wagner, 'Verbesserte Planung durch Ablaufsimulation von Szenarien', in *Digital Engineering-Herausforderung für die Arbeits-und Betriebsorganisation*, GITO mbH Verlag, 2009, pp. 105–124.

125 H. Lödding, A. Friedewald, and L. Wagner, 'Szenariosimulation mit Simulationsbaukästen', *Integrationsaspekte der Simulation: Technik, Organisation und Personal*, p. 7, 2010.

126 A. F. und L. W. Christian Nedeß, 'Robuste Termin- und Kapazitätsplanung durch Szenariosimulation', *Industrie Management*, no. 2, pp. 45–48, 2010.

127 F. Kallat, C. Mieth, J. Rehof, and A. Meyer, 'Using Component-based Software Synthesis and Constraint Solving to generate Sets of Manufacturing Simulation Models', *Procedia CIRP*, vol. 93, pp. 556–561, 2020, doi: 10.1016/j.procir.2020.03.018.

128 F. Kallat, J. Pfrommer, J. Bessai, J. Rehof, and A. Meyer, 'Automatic Building of a Repository for Component-based Synthesis of Warehouse Simulation Models', *Procedia CIRP*, vol. 104, pp. 1440–1445, 2021, doi: 10.1016/j.procir.2021.11.243.

129 S. Butzer, J. Kemnitzer, S. Kunz, M. Pietzonka, and R. Steinhilper, 'Modular Simulation Model for Remanufacturing Operations', *Procedia CIRP*, vol. 62, pp. 170–174, 2017, doi: 10.1016/j.procir.2016.06.012.

130 K. M. Shawki, K. Kilani, and M. A. Gomaa, 'Analysis of earth-moving systems using discrete-event simulation', *Alexandria Engineering Journal*, vol. 54, no. 3, pp. 533–540, Sep. 2015, doi: 10.1016/j.aej.2015.03.034.

131 K. Evers, 'Simulationsgestützte Belegungsplanung in der Multiressourcen-Montage', 2002, doi: 10.15488/5982.

132 Z. X. Guo, W. K. Wong, Z. Li, and P. Ren, 'Modeling and Pareto optimization of multi-objective order scheduling problems in production planning', *Computers & Industrial Engineering*, vol. 64, no. 4, pp. 972–986, Apr. 2013, doi: 10.1016/j.cie.2013.01.006.

133 M. Junge, *Simulationsgestützte Entwicklung und Optimierung einer energieeffizienten Produktionssteuerung*. in Produktion & Energie, no. 1. Kassel: Kassel Univ. Press, 2007.

134 T. Sobottka, 'Eine anwendungsorientierte simulationsbasierte Methode, unter Berücksichtigung von Energieeffizienz, in der optimierenden Planung von Produktion und Logistik', TU Wien, 2017. doi: 10.34726/HSS.2017.50381.

135 A. McKendall and C. Li, 'A tabu search heuristic for a generalized quadratic assignment problem', *Journal of Industrial and Production Engineering*, vol. 34, no. 3, pp. 221–231, Apr. 2017, doi: 10.1080/21681015.2016.1253620.

136 T. Kanduc and B. Rodic, 'Optimisation of Machine Layout Using a Force Generated Graph Algorithm and Simulated Annealing', *Int. j. simul. model.*, vol. 15, no. 2, pp. 275–287, Jun. 2016, doi: 10.2507/IJSIMM15(2)7.335.

137 M. Lu, H.-C. Lam, and F. Dai, 'Resource-constrained critical path analysis based on discrete event simulation and particle swarm optimization', *Automation in Construction*, vol. 17, no. 6, pp. 670–681, Aug. 2008, doi: 10.1016/j.autcon.2007.11.004.

138 Teerapun Saeheaw, Nivit Charoenchai, and Wichai Chattinnawat, 'Application Of Ant Colony Optimization For Multi-Objective Production Problems', Dec. 2009, doi: 10.5281/ZENODO.1072417.

139 M. Schlecht, S. Berger, D. Wußler, M. Haun, and J. Köbler, 'Optimierung der Reihenfolgeplanung: Integration von maschinellem Lernen und generischen Materialflussmodellen', *Zeitschrift für wirtschaftlichen Fabrikbetrieb*, vol. 117, no. 1–2, pp. 9–13, Feb. 2022, doi: 10.1515/zwf-2022-1005.

140 Z. X. Guo, W. K. Wong, S. Y. S. Leung, J. T. Fan, and S. F. Chan, 'Genetic optimization of order scheduling with multiple uncertainties', *Expert Systems with Applications*, vol. 35, no. 4, pp. 1788–1801, Nov. 2008, doi: 10.1016/j.eswa.2007.08.058.

141 J. Blank and K. Deb, 'Pymoo: Multi-Objective Optimization in Python', *IEEE Access*, vol. 8, pp. 89497–89509, 2020, doi: 10.1109/ACCESS.2020.2990567.

142 J. Blank and K. Deb, 'A Running Performance Metric and Termination Criterion for Evaluating Evolutionary Multi- and Many-objective Optimization Algorithms', in *2020 IEEE Congress on Evolutionary Computation (CEC)*, Glasgow, United Kingdom: IEEE, Jul. 2020, pp. 1–8. doi: 10.1109/CEC48606.2020.9185546.

143 S. Gerbex, R. Cherkaoui, and A. J. Germond, 'Optimal location of facts devices to enhance power system security', in *2003 IEEE Bologna Power Tech Conference Proceedings,* Bologna, Italy: IEEE, 2003, pp. 61–67. doi: 10.1109/PTC.2003.1304363.

144 A. Biele and L. Monch, 'Using simulation to improve planning decisions in mixed-model assembly lines', in *2015 Winter Simulation Conference (WSC)*, Huntington Beach, CA, USA: IEEE, Dec. 2015, pp. 2148–2159. doi: 10.1109/WSC.2015.7408328.

145 A. Tighazoui, C. Sauvey, and N. Sauer, 'Minimizing the Total Weighted Waiting Times and Instability in a Rescheduling Problem with Dynamic Jobs Weight', *Applied Sciences*, vol. 11, no. 15, p. 7040, Jul. 2021, doi: 10.3390/app11157040.

146 R. Lostado Lorza, R. Escribano García, R. Fernandez Martinez, and M. Martínez Calvo, 'Using Genetic Algorithms with Multi-Objective Optimization to Adjust Finite Element Models of Welded Joints', *Metals*, vol. 8, no. 4, p. 230, Apr. 2018, doi: 10.3390/met8040230.

147 A. A. Freitas, T. M. Lima, and P. D. Gaspar, 'Meta-Heuristic Model for Optimization of Production Layouts Based on Occupational Risk Assessment: Application to the Portuguese Wine Sector', *ASI*, vol. 5, no. 2, p. 40, Mar. 2022, doi: 10.3390/asi5020040.

148 J. Xu, C.-C. Wu, Y. Yin, C. Zhao, Y.-T. Chiou, and W.-C. Lin, 'An order scheduling problem with position-based learning effect', *Computers & Operations Research*, vol. 74, pp. 175–186, Oct. 2016, doi: 10.1016/j.cor.2016.04.021.

149 W. Zhang, 'Modeling and Solving a Resource Allocation Problem with Soft Constraint Techniques', 2002, doi: 10.7936/K7668BH2.

150 S. Dubois, H. Chibane, R. De Guio, and I. Rasovska, 'From Simulation to Contradictions, Different Ways to Formulate Innovation Directions', in *Advances and Impacts of the Theory of Inventive Problem Solving*, S. Koziołek, L. Chechurin, and M. Collan, Eds., Cham: Springer International Publishing, 2018, pp. 83–91. doi: 10.1007/978-3-319-96532-1_8.

151 A. Homaifar, C. X. Qi, and S. H. Lai, 'Constrained Optimization Via Genetic Algorithms', *SIMULATION*, vol. 62, no. 4, pp. 242–253, Apr. 1994, doi: 10.1177/003754979406200405.

152 J. Wikarek and P. Sitek, 'A Data-Driven Approach to Constraint Optimization', in *Automation 2019*, R. Szewczyk, C. Zieliński, and M. Kaliczyńska, Eds., in Advances in Intelligent Systems and Computing, vol. 920. Cham: Springer International Publishing, 2020, pp. 135–144. doi: 10.1007/978-3-030-13273-6_14.

153 A. Tighazoui, M. Schlecht, R. de Guio, B. Rose, and J. Köbler, 'Mathematical model for the maintenance activities scheduling in the case of railway remanufacturing systems', in *International Conference on Information Technology & Smart Industrial Systems*, Paris, France, Jul. 2022. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03813860

154 S. Dubois, T. Eltzer, and R. DE GUIO, 'A dialectical based model coherent with inventive and optimization problems', *Computers in Industry*, vol. 60, pp. 575–583, 2009, doi: 10.1016/j.compind.2009.05.020.

155 S. Dubois and I. Rasovska, 'Interpretation of a General Model for Inventive Problems, the Generalized System of Contradictions', 2009, doi: hal-00395787.

156 L. Lin, S. Dubois, R. De Guio, and I. Rasovska, 'An exact algorithm to extract the generalized physical contradiction', *Int J Interact Des Manuf*, vol. 9, no. 3, pp. 185–191, Aug. 2015, doi: 10.1007/s12008-014-0250-3.

157 L. Lin, I. Rasovska, R. De Guio, and S. Dubois, 'Algorithm for identifying generalised technical contradictions in experiments', *Journal Européen des Systèmes Automatisés*, vol. 47, no. 4–8, pp. 563–588, Dec. 2013, doi: 10.3166/jesa.47.563-588.

158 M. Hanifi, H. Chibane, R. Houssin, D. Cavallucci, and N. Ghannad, 'Artificial intelligence methods for improving the inventive design process, application in lattice structure case study', *AIEDAM*, vol. 36, p. e20, 2022, doi: 10.1017/S0890060422000051.

159 M. Hanifi, 'Amélioration du processus de la conception inventive par l'utilisation de méthodes agiles et d'algorithmes d'apprentissage automatique.', PhD Thesis, 2021. [Online]. Available: http://www.theses.fr/2021STRAD002/document

160 N. Ghannad, R. De, and P. Parrend, 'One-Class Ant-Miner: Selection of Majority Class Rules for Binary Rule-based Classification', 2022.

161 N. Ghannad, R. De Guio, and P. Parrend, 'Feature Selection-Based Approach for Generalized Physical Contradiction Recognition', in *Systematic Complex Problem Solving in the Age of Digitalization and Open Innovation*, D. Cavallucci, S. Brad, and P. Livotov, Eds., in IFIP Advances in Information and Communication Technology, vol. 597. Cham: Springer International Publishing, 2020, pp. 321–339. doi: 10.1007/978-3-030-61295-5_26.

Appendix

Appendix 1: Optimization results for resource allocation problem with GA

	LOS(orange)		*LOS(blue)*		*LOS(mix)*	
exp	*Average*	*Standard deviation*	*Average*	*exp*	*Average*	*Standard deviation*
1	240,89	0,61	119,43	0,65	161,00	0,54

Appendix 2: Optimization results for resource allocation problem with NSGAII

	LOS(orange)		*LOS(blue)*		*LOS(mix)*	
exp	*Average*	*Standard deviation*	*Average*	*Standard deviation*	*Average*	*Standard deviation*
1	238,67	0,26	146,69	1,75	178,17	1,22
2	751,53	144,54	104,46	0,60	325,92	49,46
3	240,12	0,47	119,99	0,68	161,10	0,53
4	452,88	23,34	106,11	0,50	224,80	8,01
5	444,69	21,16	106,76	0,50	222,42	7,27
6	611,64	77,98	105,32	0,47	278,61	26,67
7	611,96	78,17	105,32	0,47	278,72	26,74
8	238,73	0,28	134,10	1,19	169,91	0,84
9	288,48	3,57	110,58	0,43	171,47	1,37
10	338,57	7,28	107,87	0,44	186,83	2,66
11	338,57	7,29	107,87	0,44	186,83	2,66
12	327,28	6,39	108,91	0,47	183,65	2,36
13	268,03	2,41	112,96	0,50	166,03	0,98
14	238,75	0,27	131,36	1,10	168,11	0,78
15	249,33	1,11	117,15	0,59	162,39	0,63
16	239,48	0,31	122,89	0,87	162,79	0,63
17	238,89	0,31	127,76	0,95	165,80	0,67
18	259,98	1,85	114,65	0,56	164,39	0,84
19	239,46	0,30	124,5	0,92	163,85	0,67
20	254,43	1,48	115,96	0,62	163,35	0,72

Appendix 3: Optimization results for constrained optimization with GA

	LOS(orange)		*LOS(blue)*		*LOS(mix)*	
exp	*Average*	*Standard deviation*	*Average*	*Standard deviation*	*Average*	*Standard deviation*
1	239,13	0,29	124,58	0,93	163,78	0,67

Appendix 4: Optimization results for constrained optimization with NSGAII

	LOS(orange)		*LOS(blue)*		*LOS(mix)*	
exp	*Average*	*Standard deviation*	*Average*	*Standard deviation*	*Average*	*Standard deviation*
1	238,86	0,31	126,31	1,03	164,83	0,73
2	239,84	0,36	123,81	0,91	163,52	0,68
3	238,88	0,31	122,91	0,87	162,60	0,63
4	239,41	0,36	121,11	0,81	161,60	0,6
5	239,56	0,33	124,44	0,93	163,84	0,68
6	239,15	0,30	126,27	1,01	164,91	0,73
7	239,41	0,37	123,68	0,92	163,29	0,68
8	239,49	0,31	126,34	1,01	165,06	0,72
9	239,37	0,44	125,94	1,00	164,76	0,73
10	239,73	0,37	123,73	0,92	163,43	0,68
11	238,85	0,28	128,54	0,97	166,29	0,68
12	239,38	0,45	124,16	0,91	163,60	0,68

Appendix 5: Optimization results for problem-solving loop

	LOS(orange)		*LOS(blue)*		*LOS(mix)*	
exp	*Average*	*Standard deviation*	*Average*	*Standard deviation*	*Average*	*Standard deviation*
Loop 1: Compare scenarios						
S1	257,04	0,00	108,26	0,00	157,41	0,00
S2	238,73	0,28	134,1	1,19	169,91	0,84
Loop 2: Change optimization parameters						
e11	279,45	2,18	104,42	0,31	164,32	0,87
e12	285,06	2,77	104,61	0,30	166,37	1,06
e14	257,92	0,61	115,53	0,74	164,26	0,60
e18	279,45	2,18	104,42	0,31	164,32	0,87
e3	257,63	0,57	115,78	0,76	164,33	0,60
e5	291,06	3,18	103,74	0,29	167,85	1,20
e8	258,29	0,62	114,90	0,69	163,98	0,58
Loop 3: Change bed assignment						
e1	259,58	0,70	107,42	0,38	159,50	0,47
e2	239,07	0,36	126,06	0,76	164,74	0,58
e4	239,69	0,35	119,50	0,60	160,64	0,47
e5	247,28	0,83	117,97	0,60	162,23	0,59
Loop 4: Change activities of nurses						
e10	248,45	1,08	95,46	0,20	147,82	0,46
e11	248,44	1,07	95,57	0,20	147,89	0,46
e12	248,59	1,07	95,45	0,20	147,86	0,46
e7	234,87	0,28	99,58	0,45	145,89	0,37
e8	235,05	0,28	99,58	0,45	145,94	0,37
Loop 5: Change priority strategy						
e2	246,05	0,90	96,09	0,22	147,41	0,42
e5	239,51	0,62	97,00	0,25	145,77	0,36

Résumé en français

Repenser les systèmes de production à l'aune de leur ombre numérique

Les organisations de production évoluent dans un environnement en mutation rapide. Elles ont à faire face à une concurrence accrue, des conditions de marché en constante évolution, des demandes imprévisibles, et une grande variété de produits. Les organisations sont obligées de faire évoluer en permanence leurs systèmes de production. La conception et l'amélioration des systèmes de production deviennent de plus en plus complexes. La simulation des flux physiques est un outil reconnu pour l'aide à la conception des flux de production. Les expériences de simulation génèrent de nouvelles connaissances sur le système et évaluent les scénarios de conception. Cependant, la modélisation et la simulation requièrent des connaissances spécialisées sur le système et sont très contraignantes pour les modélisateurs du système.

L'industrie 4.0 ouvre de nouvelles perspectives. Elle est marquée par un niveau élevé de connectivité et d'intelligence grâce à l'adoption de technologies de l'information et de communication omniprésentes. Les données sont mesurées automatiquement dans les systèmes physiques à l'aide de capteurs, de machines intelligentes et de technologies de transmission des données à distances. Le volume de données augmente et permet de mettre en œuvre l'ombre numérique qui représente le système physique dans l'espace virtuel. Elle peut fournir des données permettant de comprendre des systèmes complexes. Basée sur l'ombre numérique, la simulation pilotée par les données vise à mener des expériences et à générer de nouvelles connaissances.

Dans la conception et l'amélioration des systèmes de production, l'industrie 4.0 peut favoriser la résolution de problèmes. L'ombre numérique décrit le système de production à l'aide de données mesurées, tandis que la simulation pilotée par les données automatise l'expérimentation. L'ombre numérique et la simulation pilotée par les données permettent toutes deux de générer des données et des connaissances qui n'étaient pas disponibles auparavant. Toutefois, les stratégies, méthodes et outils

permettant d'appliquer l'ombre numérique et la simulation pilotée par les données à la résolution de problèmes n'ont pas encore été explorés et doivent être fournis. Ce travail présente les méthodes et les outils permettant d'utiliser l'ombre numérique et la simulation pilotée par les données des flux de matériaux pour résoudre les problèmes de conception des flux physiques des systèmes de production.

1. Le contexte

La résolution de problèmes est le processus systématique permettant d'atteindre un objectif en surmontant les obstacles. L'illustration 1 présente un processus systématique de résolution de problèmes : l'activité initiale de la résolution de problèmes consiste à énoncer les insatisfactions et/ou les objectifs du problème. L'activité « Réaliser un modèle du système » fournit un modèle de simulation du système à améliorer ou à modifier. Dans l'activité « simuler des expériences », le modèle est utilisé pour simuler le comportement du système et déterminer ses performances en explorant l'espace de conception à l'aide d'approches telles que les "scénarios de simulation", les plans d'expériences ou les algorithmes d'optimisation. L'objectif est de trouver des valeurs pour les variables de décision qui permettent de satisfaire les objectifs. Lorsque tous les objectifs sont atteints, le processus de résolution de problème prend fin et le décideur dispose d'une solution. Si les objectifs ne sont pas atteints, le décideur choisit d'accepter ou non un compromis. Accepter un compromis signifie choisir une solution parmi les résultats de l'activité de simulation, qui n'atteint pas tous les objectifs mais n'est pas inacceptable. Dans l'activité « changer de modèle », le décideur conçoit de nouveaux concepts de solution pour satisfaire tous les objectifs (au moins plus que le modèle précédent). Le changement de modèle fait référence à la génération de nouveaux concepts de solution par des changements structurels allant au-delà du simple changement des valeurs des variables de décision. De nouveaux modèles de simulation sont nécessaires pour valider les concepts de solution. La boucle se répète jusqu'à ce qu'une solution remplisse tous les objectifs ou offre un compromis acceptable. Les nouvelles technologies de l'industrie 4.0, en particulier l'ombre numérique et la simulation pilotée par les données, peuvent faciliter la résolution des problèmes.

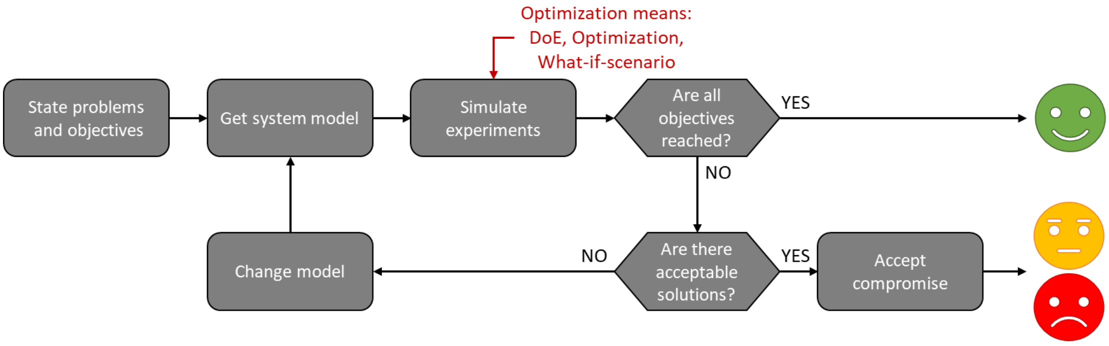


Illustration 1: Problem-solving loop

1.1. Optimisation versus invention

Il existe deux familles de méthodes pour résoudre les problèmes de conception : l'optimisation et la conception inventive. Tous deux nécessitent un modèle du système. L'optimisation recherche un ensemble de valeurs pour les paramètres d'action (décision) qui améliorent le système, sans remettre en question le modèle. Elle a prouvé son efficacité pour de nombreux problèmes, mais elle n'est pas capable de résoudre des problèmes en modifiant le système, par exemple en ajoutant de nouveaux paramètres d'action et en transformant les relations entre les paramètres d'action. Les problèmes qui ne peuvent être résolus par l'optimisation sont considérés comme des problèmes d'invention, qui nécessitent des modifications plus profondes du modèle. Dans la résolution des problèmes, la formulation des problèmes nécessite d'énoncer les paramètres du système ; l'optimisation des paramètres d'évaluation du modèle existant étant la première étape de la résolution des problèmes. Si l'optimisation ne peut pas fournir de solutions acceptables, le problème est considéré comme un problème inventif et nécessite un changement de modèle pour être résolu.

La conception inventive utilise des méthodes et des principes spécifiques pour proposer de nouvelles solutions. Parmi ces méthodes, la TRIZ, inventée par Genrich Altschuller, aide à l'analyse et à la résolution des problèmes d'invention techniques. Ainsi, la TRIZ utilise le concept de contradictions pour passer d'un problème à une solution. Les contradictions résultent de l'impossibilité de satisfaire des objectifs multiples et contradictoires, elles se manifestent par des valeurs conflictuelles entre des paramètres ou sur un paramètre du système technique. Pour résoudre ces contradictions, TRIZ propose trois niveaux de formulation. Tout d'abord, la contradiction administrative décrit la situation initiale, où un objectif ne peut être satisfait. Ensuite, la contradiction technique (CT) décrit une situation dans laquelle deux fonctions objectives sont en conflit. Pour différentes solutions, la première fonction objective s'améliore tandis que la seconde se dégrade et vice versa. Il n'est pas possible de satisfaire les deux simultanément. Les contradictions techniques contradictoires coexistent par paires. Troisièmement, la contradiction physique décrit la situation conflictuelle des contradictions techniques sur la base des variables de décision. Une variable de décision doit adopter simultanément des valeurs différentes, souvent opposées, pour satisfaire les deux objectifs. Dans le premier état, la variable de décision satisfait un objectif et dans le second, elle satisfait l'objectif complémentaire. Les paramètres impliqués dans les contradictions techniques sont appelés paramètres d'évaluation, puisqu'ils interviennent dans la définition de l'objectif du problème. Le paramètre caractérisant la contradiction physique est appelé paramètre d'action. En optimisation, ils correspondent aux variables de décision et aux valeurs de la fonction objective. Le système de contradictions de la TRIZ est illustré dans l'illustration 2. Le système de contradictions décrit le problème en combinant les deux niveaux de contradictions précédents. Pour Altschuller, la contradiction physique reflétait une contradiction plus profonde. TRIZ repose sur l'hypothèse que derrière chaque paire de contradictions techniques se cache une contradiction plus fondamentale, la contradiction physique.

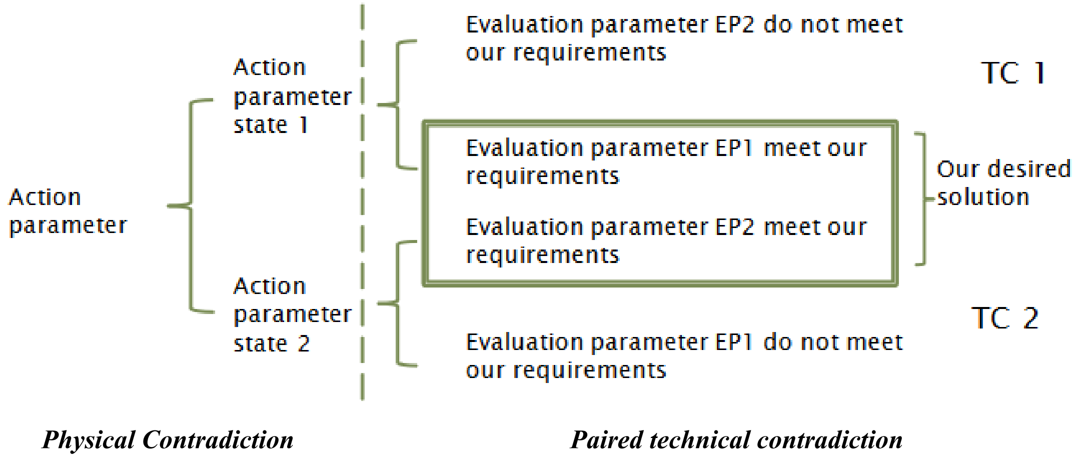


Illustration 2: System of contradictions

Dans la résolution de problèmes, le système de contradictions est le point de départ pour changer de modèle. La contradiction physique exprime la contradiction à résoudre. Pour la résoudre, le système doit être transformé qualitativement, ce qui est le résultat du changement de modèle. Pour changer de modèle, le décideur utilise des principes de conception inventive. Dans ce mémoire, les principes de séparation sont utilisés pour résoudre les contradictions rencontrées ; ils comprennent la séparation spatiale, temporelle et structurelle des variables de décision. La séparation permet aux variables de décision d'adopter des valeurs différentes au moment, à l'endroit ou pour fonction où elles sont requises. Lors de la résolution de problèmes, des contradictions peuvent apparaître dans les données expérimentales issues de la simulation. La simulation du système physique devient élémentaire.

1.2. Optimisation et conception inventive

Dans cet environnement, trois objectifs sont formulés pour aider à la résolution de problèmes. La simulation pilotée par les données à partir de l'ombre numérique permet d'évaluer des concepts de solution en fournissant des paramètres d'évaluation pour des ensembles de paramètres d'action (RQ1). Dans l'expérimentation, la simulation et l'optimisation pilotées par les données peuvent fournir des concepts de solution pour les problèmes d'optimisation et des données expérimentales avec des liens entre les paramètres d'action et d'évaluation pour la résolution de problèmes (RQ2). Les données expérimentales permettent d'extraire les contradictions techniques appairées et la contradiction physique sous-jacente. Trois questions de recherche sont abordées dans ce travail :

QR1 : Modélisation et simulation :	**Quelles sont les méthodes permettant de générer des modèles de simulation pour la résolution de problèmes à partir de l'ombre numérique ?**
QR2 : Simulation et optimization :	**Quelles sont les méthodes permettant d'optimiser les flux physiques grâce à l'ombre numérique et à la simulation basée sur les données ?**
RQ3 : Changement de modèle :	**Quelles sont les méthodes permettant de résoudre les problèmes de conception des flux physiques grâce à l'ombre numérique et à la simulation pilotée par les données ?**

En outre, trois hypothèses (H1 - H3) ont été définies et définissent le type de systèmes de fabrication, la disponibilité des données et l'étape du cycle de vie du système de production :

H1 : Type de système	**Systèmes de type job-shop.**
H2 : Disponibilité des données	**Les données sont disponibles numériquement et instantanément (données PPR et journaux d'événements historiques).**
H3 : Étape du cycle de vie	**Il est axé sur l'étape de la planification (conception et reconception) dans le cycle de vie des systèmes de flux de matériaux.**

Il existe plusieurs modèles et méthodes pour résoudre les problèmes liés aux différents types de systèmes de production. Il n'existe pas d'approche commune pour tous les systèmes. Ce travail se concentre sur les systèmes de production de type "job-shop". La simulation basée sur les données nécessite des données provenant du système physique. L'acquisition de données provenant de l'atelier prend beaucoup de temps. Les données sont supposées être disponibles instantanément et numériquement dans l'ombre numérique. Au cours du cycle de vie du système, de nombreux problèmes doivent être résolus. Pour limiter le champ d'application, ce travail se concentre sur les problèmes liés à la conception et à la reconception des systèmes.

1.3. Méthode de recherche et organisation de ce travail

Ce travail s'appuie sur la recherche-action. La recherche-action utilise une approche cyclique pour aborder les problèmes et générer des connaissances dans un nombre indéfini de boucles. Les phases de ces boucles sont la planification, l'action, le suivi et la réflexion : Le plan contient la formulation des objectifs et la définition de la méthode pour les atteindre. L'action consiste à exécuter le plan en mettant en œuvre de nouvelles méthodes. L'observation consiste à recueillir des données pour comprendre l'impact des interventions mises en œuvre. La réflexion comprend l'analyse des interventions, la réflexion sur la problématique de recherche et éventuellement la mise à jour des objectifs. Après chaque réflexion, un plan révisé peut être formulé et permet d'entrer dans un autre cycle. Pour l'observation, deux études de cas sont utilisées. Les études de cas décrivent la conception des flux de patients dans un service hospitalier d'accueil d'urgences et la conception des flux dans un système de remise à neuf de trains. Ce travail utilise quatre cycles pour répondre aux questions de recherche :

Le cycle 1 (chapitre 2) vise à fournir des méthodes d'extraction des données de simulation. Il s'agit d'extraire des données sur les flux de matières à partir de journaux d'événements contenant des données historiques. Dans la phase d'action, une revue de la littérature est fournie pour comprendre les méthodes proposées dans la littérature. Sur la base de cette étude, de nouvelles méthodes sont proposées et mises en œuvre. Deux études de cas sont présentées pour valider les méthodes et fournir des données de simulation. Dans la réflexion, les méthodes sont évaluées, les propriétés sont discutées et de nouveaux objectifs sont formulés.

Le cycle 2 (chapitre 3) vise à fournir des méthodes pour la simulation basée sur les données. Il s'agit de fournir des modèles et d'exécuter des expériences de simulation à partir de données. Les expériences évaluent des ensembles de paramètres d'action par simulation et fournissent des ensembles de paramètres d'évaluation. Dans la phase d'action, une revue de la littérature est fournie pour comprendre l'état de l'art. Sur la base de cette analyse, de nouvelles méthodes sont proposées et mises en œuvre pour générer et simuler des modèles. Deux études de cas sont présentées pour valider les méthodes et fournir des résultats de simulation. Dans la réflexion, les méthodes sont évaluées, les propriétés sont discutées et de nouveaux objectifs sont formulés.

Le cycle 3 (chapitre 4) vise à fournir des méthodes pour résoudre les problèmes d'optimisation par la simulation. Il s'agit de fournir des méthodes permettant d'utiliser la simulation guidée par les données pour résoudre divers problèmes d'optimisation. Dans la phase d'action, une revue de la littérature est réalisée pour cadrer la problématique. Sur la base de cet examen, de nouvelles méthodes sont proposées et mises en œuvre. Deux études de cas sont présentées pour valider les méthodes et trouver des optima pour trois problèmes d'optimisation : l'ordonnancement, l'allocation des ressources et l'optimisation des contraintes. Dans la réflexion, les méthodes sont évaluées, les propriétés sont discutées et de nouveaux objectifs sont formulés.

Le chapitre 4 (cycle 5) est consacré à la résolution de problèmes, sur la base de la simulation et de l'optimisation fondées sur les données. L'objectif est de fournir des méthodes pour résoudre les problèmes de conception, en se basant sur les résultats de la simulation et de l'optimisation basées sur les données. Au cours de la phase d'action, une étude est réalisée afin de comprendre les méthodes disponibles dans la littérature. Sur la base de cet examen, de nouvelles méthodes de résolution de problèmes sont proposées et mises en œuvre. Une étude de cas est présentée pour valider les méthodes et résoudre les problèmes de conception. L'étude de cas utilise le problème inventif qui n'a pas été résolu par la simulation et l'optimisation. Dans la réflexion, les méthodes sont évaluées et les propriétés sont discutées.

Après avoir décrit les quatre boucles de la recherche-action, le chapitre 6 (conclusion) résume les résultats des chapitres précédents, met en évidence les contributions et examine leurs avantages et leurs inconvénients. Des domaines de recherche future sont indiqués.

2. Acquisition des données.

La littérature montre que l'extraction des données de simulation consomme 40 à 50 % du temps d'une étude de simulation. Sur ce temps, 80 % sont consacrés à des activités manuelles. L'acquisition des données de simulation est cruciale pour l'exécution de l'étude de simulation. Pour résoudre le problème de la fourniture de données, ce travail établit une distinction entre les données de système et les données de flux. Les données de système sont principalement statiques et décrivent la configuration du système de production physique. La littérature fournit de nombreuses normes, modèles de référence et formats d'échange pour l'extraction des données de système. La situation est différente lorsqu'il s'agit d'extraire des données de flux. Les données de flux contiennent des informations sur les arrivées de produits et les schémas de processus. Les arrivées de produits et les diagrammes de processus décrivent tous deux le flux dans le système physique. Mais les méthodes d'extraction et de d'obtention des données de flux pour la simulation ne sont pas claires et doivent être fournies.

Les données d'entrée pour l'extraction des données de flux sont les journaux d'événements, qui sont fournis dans l'ombre numérique, selon l'hypothèse que les données sont disponibles numériquement et instantanément (AS2). Les journaux d'événements décrivent l'historique des flux de matières en fournissant pour chaque événement l'horodatage, la référence du cas, le type d'événement correspondant et l'horodatage de l'occurrence. Pour extraire les arrivées de produits, ce travail propose de filtrer le journal des événements pour les événements d'arrivée et d'extraire pour chaque événement le temps écoulé depuis l'événement précédent (le temps inter-arrivée) et le type de produit (routage). Pour extraire la carte des processus, ce travail propose de transformer les événements de chaque cas en activités et d'extraire leur séquence. La transformation des événements en activités nécessite de cartographier les paires d'événements correspondants. Le calcul du temps entre les événements permet d'extraire les durées des activités. Les cas présentant des séquences d'activités identiques sont regroupés en un seul itinéraire et mis en correspondance avec les cas d'arrivée des produits. La représentation graphique des itinéraires extraits permet de visualiser les itinéraires dans une carte de processus. Les résultats de cette procédure sont les arrivées et les acheminements des produits. Les arrivées de produits décrivent les séquences, les temps et les itinéraires des arrivées de produits et les itinéraires décrivent la séquence et la durée des processus de chaque itinéraire. Ce résultat est déterministe et basé sur l'historique des flux de matières.

La simulation utilise des modèles stochastiques dans un environnement dynamique. La détermination des données de simulation stochastique est nécessaire pour permettre la simulation avec des incertitudes. Trois groupes de méthodes permettent de créer des incertitudes : la méthode de Monte-Carlo pour randomiser les arrivées de produit, les chaînes de Markov pour randomiser le schéma du processus et l'ajustement de la distribution pour randomiser les durées d'activité. Pour randomiser les arrivées de produits par la méthode de Monte-Carlo, une base d'arrivées est générée avec les données historiques d'où sont tirés des échantillons aléatoires. Pour randomiser les itinéraires à l'aide de chaînes de Markov, la probabilité de passer d'une activité à l'autre est mesurée à partir des données historiques ou de données de simulation des données historiques et fournie dans un graphique de probabilité de transition. Pour randomiser les durées d'activité, les durées de toutes les activités du même type sont extraites et un histogramme des fréquences d'apparition est déterminé sur les données historiques. Le prélèvement d'échantillons aléatoires pour les classes de l'histogramme permet d'obtenir des durées aléatoires. En outre, un ajustement de la distribution à des lois statistiques analytiques permet d'exprimer l'incertitude à l'aide de fonctions analytiques. Les deux méthodes, la simulation Monte Carlo et les chaînes de Markov, visent à randomiser les chemins des pièces ou des patients. La méthode de Monte Carlo modifie les séquence des arrivées de produits (mais par leur chemin, leur gamme) et les chaînes de Markov modifient les séquences d'activités (gammes). La modification de la séquence des activités change directement les processus simulés et la modification de la séquence des arrivées change seulement le mix de produit (ou de patients). L'ajustement de la distribution fournit en outre la durée des activités.

Les méthodes ont été appliquées à l'étude de cas du service des urgences. Le service des urgences a fourni les journaux d'événements avec l'historique des flux de patients pour l'année 2021. Les journaux d'événements décrivent les flux de 35240 patients et 175386 traitements, qui ont été exécutés par 65 ressources (médecins, infirmières, laboratoires, etc.). Les arrivées de produits et les cartes de processus ont été extraites de l'historique des flux de patients. La simulation de Monte Carlo a été utilisée pour générer 20 variantes aléatoires pour les arrivées de patients (arrivées de produits) et une chaîne de Markov a été fournie pour décrire les parcours des patients (diagramme de processus). Ces deux approches ont donc pu être comparées. En outre, l'ajustement de la distribution a été appliqué pour fournir des lois aléatoires pour les durées des activités. Les données générées n'ont pas été utilisées uniquement pour la simulation, mais ont également permis d'effectuer une analyse statique et de déterminer la durée de séjour statique pour différents types de patients (itinéraires). La durée de séjour statique est basée sur la séquence des activités dans les circuits et sur la durée des activités, sans tenir compte des temps d'attente. Cela a permis d'analyser la charge de travail statique et de fournir une référence pour la durée de séjour, qui a été réutilisée dans les chapitres suivants.

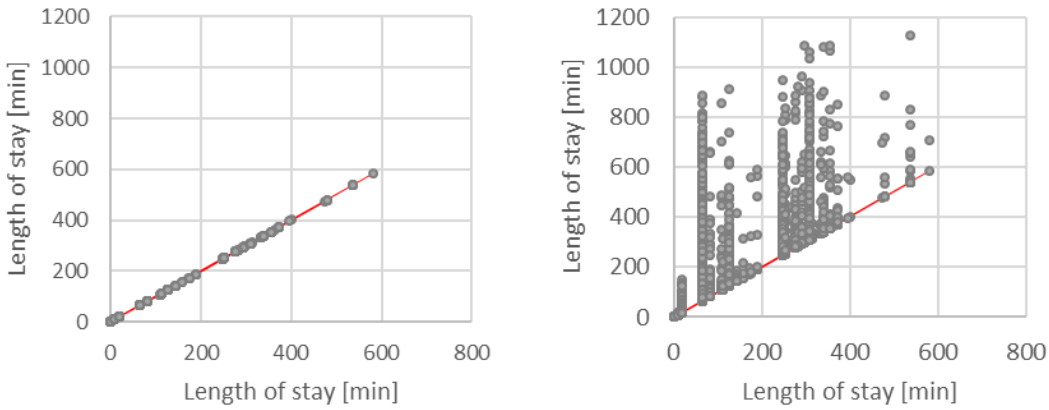
La discussion porte sur les avantages et les inconvénients des chaînes de Markov et de la simulation Monte Carlo pour la randomisation du modèle. Dans la littérature, certains articles extraient des chaînes de Markov du premier ordre à partir des journaux d'événements. Ces chaînes sont simples à extraire et à mettre en œuvre, mais notre étude de cas, les itinéraires obtenus à partir des chaînes de Markov simples ne convergent pas avec ceux des données historiques. L'analyse des résultats semble privilégier le constat que nos données ne suivent pas un processus de Markov aussi simple que celui qui a été extrait. Des études complémentaires sont à réaliser pour voir si un modèle plus sophistiqué mais encore simple à mesurer directement sur les données peut convenir. Avant de générer le modèle d'incertitude, il faut extraire les données du système physique. L'étude de cas a permis de définir les exigences en matière d'acquisition de données à partir de l'atelier. L'acquisition manuelle des données, comme les entrées des médecins dans l'étude de cas, n'est pas appropriée pour fournir des données de bonne qualité. Pour éviter les lacunes et les données corrompues, il est recommandé d'automatiser l'extraction des données. La définition d'événements pour le début et la fin des activités rend les activités de cartographie manuelle obsolètes. Au-delà de l'extraction automatisée, la structure des données a un impact sur l'extraction des arrivées de produits et des cartes de processus. Les liens manquants entre les événements et les produits nécessitent des interactions manuelles ; pour éviter cela, l'enregistrement de journaux d'événements centrés sur l'objet est nécessaire. En fournissant des données d'entrée de haute qualité, l'automatisation de la récupération des données permet de fournir des données de simulation sur demande.

3. Simulation basée sur les données

La littérature met en évidence les principales difficultés liées à la mise en place de modèles de simulation et à l'exécution d'expériences, à savoir les exigences élevées imposées au modélisateur du système en termes de temps, de connaissances et d'efforts de modélisation. Lorsque des modèles sont disponibles, la simulation peut fournir le lien entre les valeurs d'un ensemble de variables de décision et les valeurs des fonctions objectives, appelées respectivement paramètres d'action et d'évaluation dans la conception inventive. Les liens sont fournis par l'exécution d'expériences dans les modèles de simulation. La simulation pilotée par les données permet de surmonter les exigences élevées imposées au modélisateur du système en générant des modèles et en réalisant des expériences à partir de données de simulation et d'une bibliothèque de modèles réutilisables. Cependant, les "bonnes" méthodes de génération de modèles et d'exécution d'expériences ne sont pas explicitées. La littérature fournit une série de méthodes pour modéliser divers types de systèmes de production avec différents types de modèles. Des méthodes spécifiques sont disponibles en fonction de l'objectif poursuivi avec le modèle. Cette thèse fournit des méthodes des systèmes de production de type job-shop (AS1) :

Les méthodes de simulation guidée par les données utilisent le cadre de la modélisation d'entreprise. Pour l'exécution d'études de simulation, la norme de modélisation d'entreprise introduit des niveaux de généricité et des vues de modélisation. Les niveaux de généricité décrivent, par le biais de mécanismes de généralisation et de spécialisation, la transition entre les modèles génériques et les modèles réutilisables, appelés modèles partiels, pour finalement aboutir à des modèles particuliers représentant des systèmes spécifiques. Les vues de la modélisation d'entreprise décrivent les éléments modélisés qui interagissent pour décrire le comportement holistique du système : fonctions, informations, ressources et comportement. Pour la simulation pilotée par les données, ce travail utilise le concept de modèles partiels pour décrire les fonctions (processus), les informations (produits), les ressources (ressources) et l'organisation (comportement et logique de contrôle) des systèmes de production de type job-shop. La personnalisation des modèles partiels permet de spécifier des modèles partiels d'un système job-shop. La modélisation guidée par les données permet de générer des modèles particuliers à partir de modèles partiels (spécifiques) et de données de simulation. Au cours du processus de modélisation guidée par les données, un algorithme paramètre et instancie des modèles particuliers sur la base des données de simulation. Les données de simulation sont les arrivées de produits dans le modèle de simulation et les cartes de processus. Pour la randomisation, ce travail utilise les profils d'arrivée et les routages statiques issus des gammes ou des variantes de parcours dans le cas des patients. Des données supplémentaires sur les ressources et l'organisation sont supposées être disponibles pour chaque cas particulier.

Les méthodes ont été appliquées à deux cas portant sur un service hospitalier des urgences (ED) et un système de remise à neuf de trains (TRM). Les méthodes ont permis la modélisation et la simulation des deux systèmes. Dans les deux cas, la simulation a fourni des données permettant d'évaluer des scénarios. Les données fournies contenaient le lien entre les paramètres d'action et d'évaluation. Une nouvelle méthode d'analyse des données de simulation a été fournie, en particulier dans l'étude de cas du service des urgences (voir illustration 3). Le flux des patients du service des urgences se caractérise par un nombre élevé de patients ayant des parcours différents au sein des urgences. L'analyse des valeurs moyennes de traversée de l'ensemble de la population ne tient pas compte des différentes variantes de parcours et ne permet pas d'évaluer les performances et améliorations attendues pour une famille particulière de patients. Notons que les durées de parcours d'une famille de patients (équivalent des somme des temps gamme pour les produits) ne sont pas forcément homogènes comme le montre l'illustration 3a) ; ils constituent une référence pour le problème de conception ou d'amélioration. La simulation dynamique fournit la durée de séjour dans le système en tenant compte du temps d'attente pour l'accès aux ressources (illustration 3b) de chaque patient (produit). La distance entre la référence de la durée de séjour statique et la durée de séjour dynamique quantifie les temps d'attente. Une validation croisée dans le cas de la TRM serait intéressant.



a)Statique sans attente ***b)Dynamique***

Illustration 3: Analyse de la dispersion : en abscisse la durée des temps de parcours (temps gamme) des patients d'une famille

La discussion constitue la valeur ajoutée de ce chapitre. La simulation guidée par les données permet d'évaluer des solutions candidates sur demande, simplement en indiquant des variables de décision dans les données de simulation et en fournissant des modèles partiels. La personnalisation des modèles partiels a permis de mettre en œuvre le comportement spécifique des systèmes particuliers ED et TRM. Dans les deux cas, le comportement spécifique était lié à la logique de contrôle et n'avait d'impact que sur la vision organisationnelle de la modélisation d'entreprise. Des recherches supplémentaires sont nécessaires pour permettre l'extraction de la logique de contrôle des systèmes et la réutilisation des modèles disponibles. Outre la simulation de candidats à des solutions spécifiques, la simulation pilotée par les données permet de fournir la référence, qui a été calculée manuellement dans le chapitre précédent, en simulant le système "idéal". Le concept d'idéalité fait référence à un système idéal, où il n'y a pas de temps d'attente. Le système idéal est un concept virtuel qui ne peut exister dans la réalité. Pour simuler ce système, les capacités de production sont supposées infinies, ce qui signifie qu'elles sont modélisées avec une capacité suffisante pour éviter les temps d'attente. Le concept d'idéalité a été utilisé pour aider à l'analyse des résultats de la simulation. Une nouvelle méthode d'analyse a été fournie. En analysant la dispersion pour différents parcours, l'utilisation de la référence pour l'évaluation permet non seulement d'évaluer l'impact des changements de modèle en moyenne, mais aussi sur des parcours spécifiques au sein de l'ensemble de la population. Au-delà de l'évaluation de solutions spécifiques dans les scénarios de simulation, la simulation pilotée par les données permet d'évaluer les solutions candidates automatiquement et sur demande. Cela permet d'utiliser la simulation comme une fonction objective dans l'optimisation pour évaluer des ensembles de variables de décision.

4. Simulation et optimisation

Une revue de la littérature sur l'optimisation a permis d'identifier les défis et les approches disponibles pour résoudre le problème de conception (AS3) par l'optimisation mathématique des systèmes de production. Les principaux défis sont liés à la fourniture de modèles mathématiques et d'algorithmes pour l'optimisation. Le modèle mathématique doit évaluer les variables de décision à l'aide d'une fonction objective. Les modèles mathématiques sont spécifiques à chaque cas et à chaque problème. Chaque système de production ou chaque problème nécessite un modèle spécifique avec des variables de décision, des contraintes et des fonctions objectives spécifiques. En ce qui concerne les algorithmes, il existe une gamme d'algorithmes disponibles pour résoudre différents problèmes d'optimisation spécifiques.

La méthode proposée pour résoudre ces problèmes consiste à coupler la simulation pilotée par les données à une approche d'optimisation basée sur la simulation. Le principe est illustré dans l'illustration 4. Dans cette approche, un moyen d'optimisation, par exemple un plan d'expériences ou une optimisation algorithmique, fournit un ensemble de paramètres d'action à la simulation, la simulation évalue ces paramètres et fournit un ensemble de paramètres d'évaluation au moyen d'optimisation. L'échange entre les paramètres d'action (variables de décision) et les paramètres d'évaluation (fonctions objectives) se fait en boucle. Pour fournir la moyenne d'optimisation, le décideur choisit les algorithmes d'optimisation, définit les paramètres d'action et les paramètres d'évaluation ainsi que les contraintes entre les paramètres d'action et les paramètres d'évaluation. Le modèle de simulation contient toutes les contraintes du système physique. Pour évaluer la fonction objective, la simulation utilise la simulation guidée par les données. Un algorithme construit automatiquement un modèle à partir des données du système et des paramètres d'action de l'optimisation, utilise des expériences de simulation pour l'évaluation de la fonction objective et fournit de nouvelles valeurs des paramètres d'action jusqu'à l'atteinte d'un critère d'arrêt.

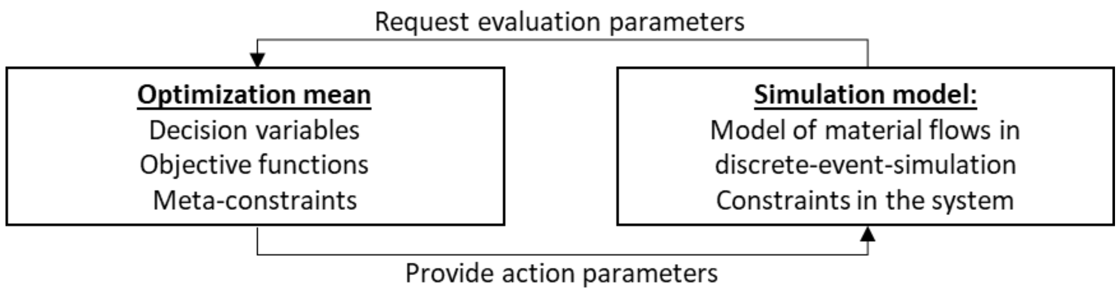


Illustration 4: Couplage simulation-optimisation

Les méthodes ont été appliquées dans les deux études de cas du service des urgences (ED) et du système de remise à neuf des trains (TRM) pour résoudre trois problèmes d'optimisation, en particulier un problème d'ordonnancement, un problème d'affectation des ressources et un problème d'optimisation sous contrainte. Pour résoudre ces problèmes, différents moyens d'optimisation ont été utilisés : plan d'expériences, algorithme génétique et optimisation de Pareto. Dans les études de cas, les résultats ont montré les différences entre l'optimisation mono-objectif avec l'algorithme génétique (AG) et l'optimisation multi-objectif avec l'algorithme de type (NSGAII). L'optimisation avec l'AG a été réalisée en pondérant les fonctions objectives et en fournissant un seul optimum, tandis que l'optimisation avec NSGAII fournit le front Pareto qui inclut les optimums de chaque fonction objective. La solution représente des solutions non dominées avec différents compromis entre les fonctions objectives. En particulier lorsqu'il s'agit de problèmes de conception et que tous les objectifs ne sont pas satisfaits, cela permet au décideur de choisir une solution parmi les solutions candidates qui lui offre un compromis acceptable.

La discussion constitue la valeur ajoutée de ce chapitre. L'utilisation de l'optimisation basée sur la simulation et guidée par les données permet de remplacer le problème d'optimisation, la moyenne d'optimisation et le cas de manière indépendante et permet d'aborder un large éventail de problèmes d'optimisation. Le défi lors du passage d'un problème d'optimisation à un autre est de définir la fonction objective et de reformuler les contraintes du problème d'optimisation. Dans la nouvelle approche, le modèle de simulation générique hérite des contraintes intrinsèques au système physique et représente la fonction objective. Il ne reste plus qu'à énoncer les paramètres d'action et d'évaluation et des contraintes supplémentaires éventuelles d’organisation. En outre, la simulation pilotée par les données permet de remplacer le cas en remplaçant simplement les données de simulation et d'élargir la gamme des paramètres d'action. Au-delà du paramétrage classique, la simulation pilotée par les données élargit l'espace de décision, par exemple en ajoutant ou en supprimant des produits et des machines ou en modifiant les processus. Il existe des algorithmes spécifiques pour résoudre différents types de problèmes. L'approche présentée permet de remplacer ces algorithmes. Au-delà des contributions énumérées, l'approche apporte des valeurs ajoutées en réduisant le temps et les efforts nécessaires à la modélisation des problèmes d'optimisation. Toutefois, il convient de souligner que l'utilisation de la simulation dynamique dans des modèles stochastiques pour évaluer la fonction objective augmente les exigences en matière de puissance de calcul.

5. Résolution de problèmes

Une revue de la littérature sur la conception inventive a fourni le contexte et les méthodes pour résoudre les problèmes de conception (AS3) à partir de données expérimentales. Elle présente les méthodes de collecte de données à partir d'expériences et introduit le concept de contradictions pour résoudre les problèmes. Sur la base des données expérimentales, elle décrit les méthodes pour extraire les systèmes de contradictions techniques et physiques qui sont le point de départ du changement de modèle. Pour modifier le modèle, le principe TRIZ des principes de séparation est retenu et présenté. Les méthodes fournies par la littérature constituent la base des méthodes de résolution des problèmes de ce travail.

La méthode proposée pour résoudre les problèmes utilise les données expérimentales de l'optimisation comme point de départ. Les solutions des données expérimentales sont regroupées en solutions dominées et non dominées, sur la base du principe de dominance de Pareto. La binarisation est appliquée aux solutions candidates non dominées. La binarisation évalue pour chaque solution la satisfaction des objectifs. En trouvant un ensemble de deux ou plusieurs solutions partielles dont la réunion des objectifs atteints recouvre tous les objectifs, le décideur met en évidence des contradictions techniques appariées. Les contradictions techniques appariées sont deux solutions candidates qui devraient exister au même moment pour résoudre tous les problèmes. Une analyse discriminante met en évidence les paramètres d'action qui sont à l'origine de la situation conflictuelle : pour satisfaire tous les objectifs le paramètre d'action doit prendre deux valeurs différentes. Cette situation caractérise la contradiction physique qui est le point de départ du changement de modèle et qui permet de résoudre le problème de conception. Les principes de séparation permettent de résoudre la contradiction physique en séparant les propriétés attendues du système dans le temps, l'espace, entre niveaux de système ou par combinaison.

Les méthodes de séparation dans le temps et l'espace ont été appliquées à l'étude de cas du service des urgences. Le point de départ de l'étude de cas est le problème inventif pour lequel aucune solution n'a été trouvée par optimisation. Le décideur souhaite améliorer la durée de séjour des patients dont l'indice de gravité est faible (priorité faible), sans dégrader la durée de séjour des autres patients et sans ajouter de nouvelles ressources. La boucle de résolution de problèmes, présentée dans l'illustration 1, a été appliquée et quatre boucles ont été réalisées. Dans chaque boucle, les paramètres d'action ont été optimisés, mais le problème n'a pas été résolu. Les résultats de l'optimisation (données expérimentales) ont servi de point de départ à la modification du modèle. Les solutions non dominées ont été extraites des données expérimentales, la satisfaction des objectifs a été mesurée et binarisée, les contradictions ont été extraites et des changements de modèle ont été proposés. Pendant quatre boucles, le système critique a été continuellement amélioré. Après quatre boucles, la durée de séjour des patients peu prioritaires a été améliorée sans dégrader la durée de séjour des autres patients et tout en respectant les restrictions de ressources.

La discussion porte sur les avantages et les inconvénients des méthodes proposées. Lorsqu'elle fournit des données expérimentales, l'optimisation de Pareto permet de réaliser expériences pertinentes pour la mise en évidence des contradictions. Sans l'optimisation de Pareto, l'utilisation des plans d'expériences et de l'optimisation à objectif unique pose des problèmes. Lors de l'exécution d'un plan d'expériences, l'évaluation d'un grand nombre de solutions et l'extraction des solutions non dominées peuvent prendre beaucoup de temps, en particulier lorsque le nombre de paramètres d'action est élevé. Lors de l'exécution d'une optimisation mono-objectif, la pondération des fonctions objectives permet d'éviter la recherche dans l'ensemble de l'espace de solution et la prise en compte d'objectifs multiples. L'optimisation de Pareto recherche efficacement dans l'ensemble de l'espace de solution et fournit des solutions non dominées, décrivant des compromis entre plusieurs fonctions objectives. Lorsque l'on dispose de données expérimentales, les méthodes fournissent un chemin systématique vers les contradictions et un guide pour le changement de modèle. Sans ces méthodes, la compréhension du problème et l'obtention de solutions pour le résoudre est un processus qui exige beaucoup de connaissances de la part du concepteur.

6. Discussion

La discussion fournit les réponses aux questions de recherche énoncées et énumère les contributions fournies dans chaque chapitre de cette thèse.

Le chapitre 2 présente les méthodes permettant d'extraire les données de simulation des journaux d'événements avec les flux de matières historiques disponibles. Des méthodes ont été fournies pour extraire les arrivées de produits et les cartes de processus. Pour permettre une simulation stochastique, des méthodes supplémentaires ont été fournies pour randomiser les arrivées de produits et les diagrammes de processus. La méthode de randomisation des arrivées de produits est une simulation de Monte Carlo, qui tire des arrivées aléatoires de l'ensemble des arrivées de produits historiques. La méthode permettant de rendre aléatoire la carte des processus est l'extraction de chaînes de Markov. Toutefois, la discussion a mis en évidence les avantages et les inconvénients des deux méthodes. Les inconvénients sont particulièrement visibles lors de l'utilisation de chaînes de Markov d'ordre 1 qui ne permettent pas de tenir compte des caractéristiques des familles de patient ou de produits.

Le chapitre 3 présente les méthodes permettant de générer des modèles et de réaliser des expériences de simulation à partir de données. Les méthodes sont la conception de modèles partiels pour fournir une bibliothèque de modélisation génériques pour les systèmes de job-shop et la simulation pilotée par les données. La simulation pilotée par les données utilise un algorithme pour instancier des modèles à partir de données de simulation et de modèles partiels. Ces méthodes permettent d'évaluer des solutions candidates sur demande en simulant des flux de matériaux dans un environnement dynamique et stochastique. La simulation pilotée par les données permet d'implémenter des paramètres d'action dans le modèle, de les évaluer par simulation et de fournir des paramètres d'évaluation. Cette capacité fait de la simulation pilotée par les données un outil d'optimisation basé sur la simulation.

Le chapitre 4 présente les méthodes permettant de coupler la simulation pilotée par les données et l'optimisation. Ces méthodes permettent d'aborder un large éventail de problèmes d'optimisation dans différents cas. L'approche sépare l'énoncé du problème d'optimisation et l’obtention d’un modèle mathématique. L'énoncé des problèmes d'optimisation se réduit à la définition des paramètres d'action, des paramètres d'évaluation et des contraintes entre paramètres. Les contraintes du système sont modélisées dans le modèle de simulation, qui est généré à partir des données. La simulation générée mémorise les fonctions objectives et fournit des paramètres d'évaluation pour un ensemble de paramètres d'action par simulation. La formulation de problèmes d'optimisation avec des modèles mathématiques représentant la complexité du système étudié n'est pas nécessaire. La valeur ajoutée de l'approche est la possibilité de sélectionner le problème d'optimisation, l'algorithme d'optimisation et le cas d'étude séparément.

Le chapitre 5 présente les méthodes d'utilisation de la simulation et de l'optimisation pilotées par les données pour la résolution de problèmes. La simulation et l'optimisation pilotées par les données facilitent la résolution de problèmes, d'abord en fournissant des solutions candidates pour résoudre les problèmes et ensuite, si elles ne peuvent pas résoudre le problème, en fournissant des données expérimentales pour la conception inventive. En conception inventive, il existe des méthodes pour orienter le changement de modèle. Elles visent à résoudre le problème en mettant en évidence les contradictions techniques et physiques, en énonçant le système de contradictions et en proposant un changement de modèle à l'aide des principes de séparation de la TRIZ. L'étude de cas a montré comment toutes les méthodes susmentionnées font partie d'une méthode globale de résolution de problèmes pour la conception de systèmes de production.

Michael Schlecht

Redesigning production systems by their digital shadow

Résumé

Cette thèse traite de la conception de systèmes de production par simulation et optimisation. La simulation des flux est un outil courant pour résoudre les problèmes de conception des systèmes. Les limites sont les exigences élevées en termes de temps et de connaissances pour exécuter des études de simulation, évaluer les résultats et résoudre les problèmes de conception. Les technologies de l'industrie 4.0 et de l'ombre numérique offrent de nouvelles possibilités en fournissant des données pour la simulation. Cependant, les méthodes permettant d'utiliser les données de production pour la reconception des systèmes de production ne sont pas encore disponibles. L'objectif de ce travail est de fournir des méthodes pour automatiser la simulation à partir de l'ombre numérique, utiliser la simulation pour optimiser, et résoudre des problèmes de conception. Deux cas d'étude sont utilisés comme support à la démarche de recherche action de ce travail. Le résultat de ce travail est un cadre pour l'application de l'ombre numérique dans l'optimisation et la résolution de problèmes.

Mots-clés : Ombre numérique, simulation, optimisation, conception inventive, résolution de problèmes.

Abstract

This thesis deals with the redesign of manufacturing systems by simulation and optimization. Material flow simulation is a common tool for solving problems in system design. Limitations are the high requirements in time and knowledge to execute simulation studies, evaluate results and solve design problems. New chances arrives with the technologies of industry 4.0 and the digital shadow, providing data for simulation. However, the methods to use production data for the redesign of production systems are not available yet. Purpose of this work is providing the methods to automate simulation from digital shadow, use simulation to optimize and solve problems in system design. Two case studies are used to support the action research approach of this work. The result of this work is a framework for the application of the digital shadow in optimization and problem-solving.

Keywords: Digital shadow, data-driven simulation, optimization, inventive design, problem-solving.