Inaugural-Dissertation

zur Erlangung der Doktorwürde

der Naturwissenschaftlich-Mathematischen Gesamtfakultät der

Ruprecht-Karls-Universität Heidelberg

vorgelegt von

Dipl.-Math. Michael Engelhart

aus Mannheim in Baden-Württemberg

Tag der mündlichen Prüfung

.....

Optimization-based Analysis and Training of Human Decision Making

DIPL.-MATH. MICHAEL ENGELHART

Interdisziplinäres Zentrum für Wissenschaftliches Rechnen
Ruprecht-Karls-Universität Heidelberg

Betreuer
PROF. DR. SEBASTIAN SAGER
PROF. DR. JOACHIM FUNKE

Michael Engelhart, Dipl.-Math.

Interdisziplinäres Zentrum für Wissenschaftliches Rechnen
Ruprecht-Karls-Universität Heidelberg
Im Neuenheimer Feld 368
69120 Heidelberg
Germany

michael.engelhart@iwr.uni-heidelberg.de

Mathematics Subject Classification (2010): 49M27; 90C26,30,90; 91E40,45

Zusammenfassung

Im psychologischen Forschungsgebiet *Komplexes Problemlösen* (engl.: *Complex Problem Solving*) werden computergestützte Tests eingesetzt, um die komplexe Entscheidungsfindung und das Lösen komplexer Probleme von Menschen zu analysieren. Dazu werden üblicherweise computerbasierte *Mikrowelten* verwendet, in denen die Leistung von Probanden ermittelt und mit bestimmten Eigenschaften verknüpft wird. Solche Testszenarien wurden bisher im Allgemeinen in iterativen, auf *Versuch und Irrtum* beruhenden Prozessen erarbeitet, bis sie bestimmte Charakteristika aufwiesen. Je komplexer solche Modelle werden, umso wahrscheinlicher ist es jedoch, dass unerwünschte Eigenschaften bei der Verwendung in Studien hervortreten.

In der vorliegenden Arbeit werden mathematische Optimierungsverfahren nicht nur als Analyse- und Trainingswerkzeug im Komplexen Problemlösen eingesetzt, sondern auch bereits in der Entwicklungsphase eines neuen komplexen Problemszenarios. Diese neuartige Mikrowelt, der *IWR Tailorshop*, wird in der Arbeit vorgestellt und besteht aus funktionalen Zusammenhängen, die auf Optimierungsergebnissen beruhen. Mit dem *IWR Tailorshop* wurde im Rahmen dieser Arbeit erstmals ein Testszenario für die Problemlöseforschung von Anfang an für den Einsatz mathematischer Optimierungsverfahren entwickelt. Als Basis diente hierbei das ökonomische *Framing* des *Tailorshop*s, eine weitverbreitete und häufig eingesetzte Mikrowelt.

Diese Arbeit beschreibt eine optimierungsbasierte Analysemethode von Probandendaten in solchen Mikrowelten, die um Methoden zur Berechnung eines optimierungsbasierten Feedbacks erweitert wird. Dabei werden sowohl für die Berechnung als auch für die Darstellung des Feedbacks verschiedene Ansätze diskutiert und implementiert. Darüber hinaus werden verschiedene differenzierbare Umformulierungen eines in der Formulierung des Testszenarios unvermeidlichen Minimumterms untersucht und Rechenergebnisse dazu präsentiert. Der Schwierigkeiten, für die sich aus dem Testszenario ergebenden nichtkonvexen gemischt-ganzzahligen Optimierungsprobleme global optimale Lösungen zu finden, wird in der vorliegenden Arbeit mit einem neuartigen Dekompositionsverfahren begegnet. Für diese Methode werden ebenfalls Rechenergebnisse vorgestellt. Die neue Mikrowelt wurde in einem webbasierten Interface implementiert, das von einer Analysesoftware für gesammelte Datensätze ergänzt wird. Diese Softwarepakete stehen als *Open-Source-Software* auch als Basis für andere Testszenarien zur Verfügung und sind aufgrund ihrer modularen Struktur gut für die Übertragung auf ebensolche geeignet.

Abschließend wird die entwickelte Methode in einer webbasierten Feedbackstudie unter Benutzung des *IWR Tailorshop* angewandt. Die Probanden werden hierbei beim Erlernen der Steuerung dieser Mikrowelt durch optimierungsbasiertes Feedback unterstützt. In dieser Studie mit 148 Teilnehmern wird gezeigt, dass das Feedback zu einer signifikanten Verbesserung der Probandenleistung führen kann, wenn eine günstige Darstellung hierfür gewählt wird. Je nach Berechnungs- und Darstellungsvariante ist die Verbesserung im Vergleich zu einer Kontrollgruppe gravierend. Die Arbeit enthält eine ausführliche Analyse der Studie und formuliert neue Erkenntnisse über die menschliche Entscheidungsfindung in komplexen Problemen, die erst durch den auf allen Ebenen optimierungsbasierten Ansatz ermöglicht wurden.

Abstract

In the research domain *Complex Problem Solving* (CPS) in psychology, computer-supported tests are used to analyze complex human decision making and problem solving. The approach is to use computer-based *microworlds* and to evaluate the performance of participants in such test-scenarios and correlate it to certain characteristics. However, these test-scenarios have usually been defined on a trial-and-error basis, until certain characteristics became apparent. The more complex models become, the more likely it is that unforeseen and unwanted characteristics emerge in studies.

In this thesis, we use mathematical optimization methods as an analysis and training tool for complex problems solving, but also in the design stage of a new complex problem scenario. We present the *IWR Tailorshop*, a novel test scenario with functional relations and model parameters that have been formulated based on optimization results. The *IWR Tailorshop* is the first CPS test-scenario designed for the application of optimization and is based on the economic framing of another famous microworld, the *Tailorshop*.

We describe an optimization-based analysis approach and extend it to optimization-based feedback with different approaches for both feedback computation and feedback presentation. Additionally, we investigate differentiable reformulations for an unavoidable minimum expression and show the according numerical results. To address the difficulties of computing globally optimal solutions for this test-scenario, which yields a nonconvex mixed-integer optimization problems, we present a decomposition approach for the *IWR Tailorshop*. The new test-scenario has been implemented in a web-based interface together with an analysis software for collected data, which both are available as open-source software and allow for an easy adaption to other test-scenarios.

In this work, we also apply our methodology in a web-based feedback study using the *IWR Tailorshop* in which participants are trained to control the microworld by optimization-based feedback. In this study with 148 participants, we show that such a feedback can significantly improve participants' performance in a complex microworld with a possibly huge difference to a control group. However, the performance improvement depends on the representation of the feedback. We give a detailed analysis of the study and report on new insights about human decision making which only have been possible through the *IWR Tailorshop* and our optimization-based analysis and training approach.

Danksagung

Ich danke der *Naturwissenschaftlich-Mathematischen Gesamtfakultät* und der *Fakultät für Mathematik und Informatik* der *Ruprecht-Karls-Universität Heidelberg* für die Möglichkeit, meine Dissertation dort zu erarbeiten.

Heidelberg, im Januar 2015 *Michael Engelhart*

Contents

CHAPTER 1

Introduction

1.1 Motivation

Modern life imposes daily decision making, often with important consequences. Illustrative examples are politicians who decide on actions to overcome a financial crisis, medical doctors who decide on complementary chemotherapy drug delivery strategies, or entrepreneurs who decide on long-term strategies for their company.

The process of human decision making is the subject of research in the field of *Complex Problem Solving* (CPS), which deals with *complex problems*. The complexity may result from one or several different characteristics, such as a coupling of subsystems, nonlinearities, dynamic changes, opaqueness, or others [42]. Such problems are considered to be similar to problems we encounter and solve in everyday life and thus investigation of CPS is claimed to yield more insight into real-world human decision making. Apparently, our introductory examples are complex problems and as such, they are ill-defined. More precisely, their problem space is open and a problem solver has to deal with lots of variables, dependencies and dynamics making them complex problems: which information is relevant? How is the data connected? What is the exact aim? How can contradictory aims be weighted?

The main intention in CPS is to understand how certain *exogenous variables* influence a solution process. In general, *personal and situational variables* are differentiated. The most typical and frequently analyzed personal variable is *intelligence*. It is an ongoing debate how intelligence influences complex problem solving [136]. Other interesting personal variables are *working memory* [106], *amount of knowledge* [77], and *emotion regulation* [96]. Situational variables like the impact of *goal specificity and observation* [95], *feedback* [29], and *time constraints* [62] attracted less attention. In a recent work [116], an abstract computer-simulated monopoly market is used to investigate dynamic decision making based on the choice of *goal systems*.

Doubts about the relevance of *simple problems* with a well-defined problem space, like the *Tower of Hanoi*, for real-world problems like the above-mentioned lead to the development of computer-based simulations of small parts of the real world, *microworlds*. These simulations resemble the properties of complex real-world problems, but offer researchers the possibility to conduct studies under controlled conditions. In CPS, the performance of *participants* in a clearly defined microworld is investigated, evaluated and correlated to certain characteristics, such as the participant's capacity to regulate emotions.

One microworld that comprises a variety of properties such as dynamics, complexity and interdependence, discrete choices, lack of transparency, and polytely in an economical framing is the *Tailorshop*. Participants have to make economic decisions to maximize the overall balance of a small company, specialized in the production and sales of shirts. The *Tailorshop* sometimes is referred to as the *Drosophila* for CPS researchers [55] and thus a prominent example for a computer-based microworld. It has been used in a large number of studies, e.g., [102, 78, 76, 87, 11, 12]. Comprehensive reviews on studies with *Tailorshop* have been published, e.g., [50, 54, 56, 55].

The calculation of *indicator functions* to measure performance of CPS participants is by no means trivial. To measure performance within the *Tailorshop* microworld, different indicator functions have been proposed in the literature, see [39] for a recent review. To use a comparison of the variable which the participants are requested to maximize between all participants was proposed in [72]. Such a

performance criterion seems natural. However, it cannot yield insight into the temporal process and is not objective in the sense that the performance depends on what other participants achieved. Analyzing the temporal evolution of other variables of this microworld has also been proposed (see, e.g., [101, 122, 51, 12]). An obvious drawback of comparing the development of variables which were not the actual objective for the participants is that a monotonic development does not necessarily indicate good or even optimal decisions. If we consider the variable *capital*, for instance, it may be better to invest into infrastructure at the beginning to have a higher pay-off towards the end of the time-scale in the test-scenario. Thus it might happen that decisions are analyzed to be bad by these approaches, while they are actually good ones and vice versa.

The availability of an objective performance indicator is an obstacle for analysis and it has often been argued that inconsistent findings are due to the fact that

> "...it is impossible to derive valid indicators of problem solving performance for tasks that are not formally tractable and thus do not possess a mathematically optimal solution. Indeed, when different dependent measures are used in studies using the same scenario (i.e., Tailorshop [51, 122, 101]), then the conclusions frequently differ."

as stated by Wenke and Frensch [131, p.95].

To overcome this problem, we propose to use indicator functions based on optimal solutions. In [111, 112] the question how to get a *reliable performance indicator* for the *Tailorshop* microworld has been addressed. Because all previously used indicators have unknown reliability and validity, decisions are compared to mathematically optimal solutions. For the first time, a complex microworld such as *Tailorshop* has been described in terms of a mathematical model. Thus, the assumption that the *fruit fly of complex problem solving* is not mathematically accessible has been disproved. The novel methodological approach has also been combined with experimental studies, [11, 12, 112].

So far, all CPS microworlds have been developed in a purely disciplinary trial-and-error approach. To our knowledge, a systematic development of CPS microworlds based on a mathematical model, sensitivity analysis, and eventually optimization methods to choose parameters that lead to a wanted behavior of the complex system has not yet been applied. An example for this necessity is the fact that the mathematical modeling of the *Tailorshop* microworld in [112] led to the discovery of unwanted and unrealistic winning strategies (e.g., the *vans bug* as described in Chapter 4).

With tasks for humans getting more complex in the real-world, there is an increasing need to train and assist participants in complex tasks. In [73], a framework for training engineering students in designing controllers for complex systems like chemical reactors is presented. In this approach, students can learn from the results of simulations depending on their inputs. In the context of CPS, an interesting approach would be to determine optimal solutions and corresponding controls for a microworld to compute a feedback for participants to support and train them. However, as [37] shows, the presentation of information in a dynamic context is crucial for the success of the participants. To the best of our knowledge, there have been no studies investigating the effects of an optimization-based feedback.

It turns out that the optimization problems that need to be solved in the context of the *Tailorshop* scenario are *mixed-integer nonlinear programs* with nonconvex continuous relaxations. Whenever optimization problems involve variables of continuous and discrete nature, the term *mixed-integer* is used. In this case they can be interpreted as discretized optimal control problems. See [109] for a recent review of algorithms to treat continuous-time mixed-integer optimal control problems. However, as the time grid is fixed, the applicability of such methods is limited, and we have to focus on combinatorial methods.

Progress in *mixed-integer linear program* (MILP) started with the fundamental work of DANTZIG and coworkers on the Traveling Salesman problem in the 1950s. Since then, enormous progress has

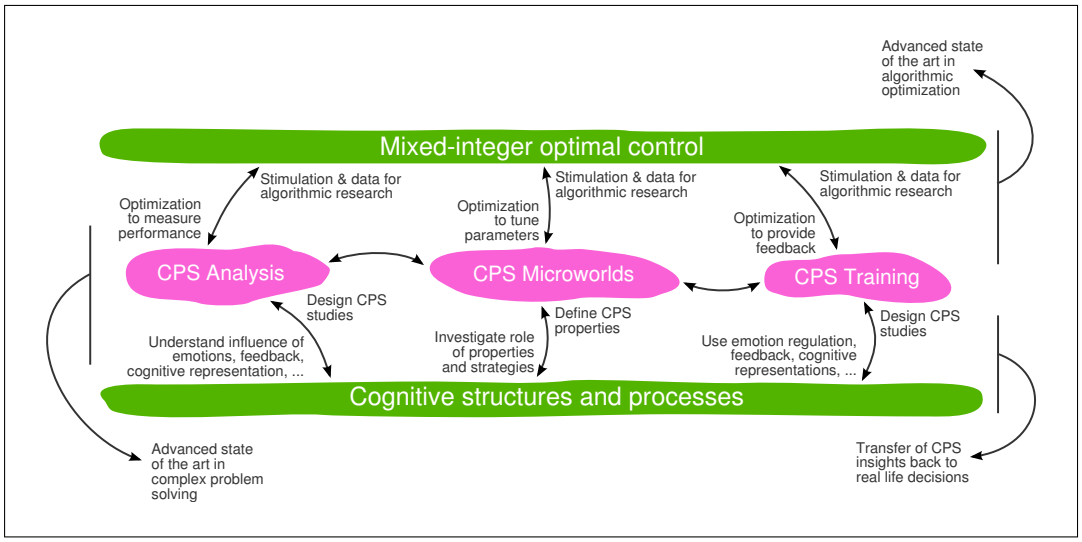


Figure 1.1: Relation between optimization methods, CPS, and cognitive processes: optimal solutions can be used for analysis, parameter optimization, and feedback in CPS test-scenarios. Problems arising from microworlds vice versa provide a stimulation and data for algorithmic research. CPS contributes to a better understanding of cognitive structures and processes.

been made in areas such as *linear programming* (and especially in the *dual simplex* method that is the core of almost all MILP solvers because of its restart capabilities), in the understanding of *branching rules* and more powerful selection criteria such as *strong branching*, the derivation of tight *cutting planes*, novel *preprocessing* and *bound tightening procedures*, and of course the computational advances roughly following MOORE's law. For specific problem classes problems with millions of integer variables can now be routinely solved [10]. Also generic problems can often be solved very efficiently in practice, despite the known exponential complexity from a theoretical point of view [20].

The situation is different in the field of *mixed-integer nonlinear program* (MINLP). Only at first sight many properties of MILP seem to carry over to the nonlinear case. Restarting nonlinear continuous relaxations within branching trees is essentially more difficult than restarting linear relaxations (which, e.g., global solvers like *BARON* and *Couenne* also use for nonlinear problems), as no dual algorithm comparable to the dual simplex is available in the general case. Nonconvexities lead to local minima and do not allow for easy calculation of subtrees, which is important to avoid an explicit enumeration. Additionally, nonlinear solvers are slower and less robust than LP solvers. However, the last decade saw great progress triggered by cross-disciplinary work of integer and nonlinear optimizers, resulting in generic MINLP solvers, e.g., [2, 25]. Most of them, however, still require the underlying functions to be convex. Comprehensive surveys on algorithms and software for convex MINLPs are given in [63, 26]. Recent progress in the solution of nonconvex MINLPs is in most cases based on methods from global optimization, in particular convex under- and overestimation. See, e.g., [123] for references on general under- and overestimation of functions and sets.

The connection of mathematical optimization methods and CPS seems promising for both disciplines, as illustrated in Figure 1.1. Optimal solutions can be used for analysis, parameter optimization, and feedback in CPS test-scenarios. Problems arising from microworlds vice versa provide a stimulation and data for algorithmic research. As discussed above, the problems arising from CPS microworlds usually are hard optimization problems which may trigger research in mathematical

optimization. With optimization-based analysis and feedback methods, CPS can contribute to a better understanding of cognitive structures and processes.

1.2 Contributions

The methodology *optimization* has a long record of successful improvements in many technological and scientific areas, being used for tasks such as design, scheduling, business control rules, process control, and the like. Optimization has also been successfully applied in the context of inverse problems, e.g., for the choice and calibration of mathematical models, or as a modeling paradigm for biological systems. In this work we propose to use numerical optimization as an analysis tool for the understanding of human problem solving, which to our knowledge has not yet received much attention.

Based on the experience with the original *Tailorshop*-microworld described in [112] with modeling oddities, bugs, and other undesirable properties, we decided to build a mathematical model for a CPS microworld from scratch. Therefore, in this work we present a new microworld with desirable (mathematical) properties based on the economical framing of *Tailorshop*, for which optimization methods have been considered already throughout the modeling phase, the *IWR Tailorshop*. To the best of our knowledge, the *IWR Tailorshop* is the first CPS test-scenario with functional relations and model parameters that have been formulated based on optimization results.

For the unavoidable minimum expression describing the variable *sales*, we investigate different reformulations and discuss numerical results. We describe the optimization-based analysis approach published in [112] and extend it to optimization-based feedback. We present different approaches for both feedback computation and feedback presentation. The new test-scenario has been implemented including the different optimization-based feedback methods in a web-based interface. For the analysis of data collected with this interface, optimization-based analysis methods have been implemented in the analysis software *Antils*. Both the web front end and the analysis back end are available as open-source software under the *GNU General Public License* (GPL).

To address the difficulties of computing globally optimal solutions for this test-scenario, which still yields nonconvex optimization problems, we present a decomposition approach tailored to the *IWR Tailorshop*. Mathematical model reduction techniques are quite common in other domains, see e.g., [18, 9, 115] for an overview. The basic idea of our new approach to solve the occurring *discretized mixed-integer optimal control problem* (dMIOCP) consists of a decomposition of the MINLP into a *master* and several smaller *subproblems*. This works if the objective function is separable. The idea is related to *Lagrangian relaxation*, one of the most used relaxation strategies for *MILPs*.

Finally, we proof the feasibility of our methodology in practice with a web-based feedback study using the *IWR Tailorshop*. In this study, we collected data from 148 participants and applied our optimization-based analysis and feedback approach. The participants were asked to play several rounds of the economic simulation via its web interface. We give a detailed description of study, hypotheses, analysis, and results. Both the analysis and feedback based on optimal solutions enabled insights on human decision making which else would not have been possible.

The thesis is organized as follows. We start with an overview of CPS in Chapter 2. Optimization problem classes and algorithms for the solution of dMIOCPs or MINLPs respectively are described in Chapter 3. Chapter 4 gives a description of the *Tailorshop* microworld and introduces the new test-scenario, *IWR Tailorshop*. In Chapter 5, the framework for our optimization-based analysis and feedback approach is specified. Additionally, we investigate different reformulations for a min-expression and explain the tailored decomposition. We present the results of the web-based feedback study using the *IWR Tailorshop* in Chapter 6 and conclude with an outlook in Chapter 7.

CHAPTER 2

Complex Problem Solving

Decision making is the cognitive process of selection between several alternatives [61]. Although many human decisions are made unconsciously, i.e., "without thinking much about the decision process" [93], *Human Decision Making* is closely connected with *problem solving*—a term which is used in different disciplines for different aspects. This chapter gives an introduction into problem solving with a focus on *Complex Problem Solving* (CPS) (CPS), a research domain in psychology, which is the field of application for the methods in this work.

2.1 Problems

2.1.1 What Is a Problem

Regarding the term *problem solving*, we first should spend some words on what is considered to be a problem. There are varying definitions of *problems*, classifications of problems, and origins of problems, but there is consensus on the following characteristics, see e.g., [54, 61]. A problem consists of an *initial state*, which describes an unsatisfactory situation at the beginning and a *goal state*, which is the state one strives to achieve by a set of *operations* which gradually transform the initial state into a goal state.

Imagine you are in Germany in autumn. It is getting colder and darker outside and you feel that lying at some beautiful beach in the Caribbean Sea is the only thing which could make your life bearable again. So, you have got a problem: an unsatisfactory initial state of you being at a place where you do not want to be and the goal state of you at a beach somewhere in the Caribbean. The set of possible operations is extensive: you could put on your swimwear and immediately start walking in the direction of Martinique (maybe not the best one, though). Or you start with booking a flight leaving Frankfurt Airport tomorrow. But do not forget the constraints: your employer will not be happy if you disappear without notice. And your PhD thesis will not be finished of its own volition either.

2.1.2 Ill- and Well-defined Problems

Both the initial and the goal state can be defined more or less accurate. Feasible operations can be described precisely or only given vaguely and a problem does not need to have a unique solution, as the example already showed. Depending on how accurately states and operations are defined, problems are called *well-defined* or *ill-defined*.

The task to successfully lead a company, for instance, is ill-defined compared to the task to maximize a single variable, say the company's capital or profit. Of course, there are problems which even more deserve the term ill-defined, like to find an answer to "the ultimate question of life, the universe, and everything" [4]. Nevertheless, even maximizing a company's profit may still be an ill-defined problem itself, as neither the actual initial state of the company nor the possible courses of action are given precisely by this description.

A prominent example for a well-defined problem is the *Tower of Hanoi*, see Figure 2.1, a puzzle probably invented by French mathematician ÉDOUARD LUCAS in the 19th century. The puzzle consists of a varying number of disks of different size and three rods on which the disks can be put. The

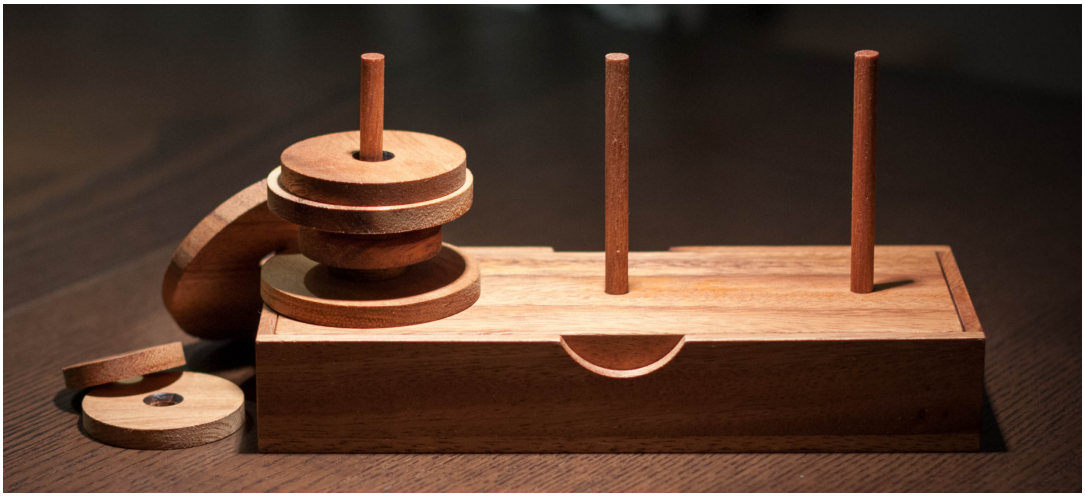


Figure 2.1: A disordered *Tower of Hanoi* set with three rods and eight disks. According to the rules of the task, no disk may be moved on top of a smaller one. Only one disk at a time and only the uppermost disk may be moved. The entire stack of disks has usually to be moved from the left rod to the right one.

task usually is to move the stack ordered by the size of the disks from one rod to another, e.g., from left to right. There are three rules: no disk may be moved on top of a smaller one, only one disk at a time, and only the uppermost disk may be moved. Obviously, both states and feasible operations are quite accurately defined in this puzzle making it a well-defined problem. Figure 2.5 shows all possible states and actions for a *Tower of Hanoi* with three disks.

2.2 Problem Solving

The higher-order cognitive processes related with the solution of a problem are usually called *problem solving* in psychology. According to [54], investigation of problem solving consists of the analysis of a series of decisions. In contrast, *decision theory* and *decision analysis* concentrate on the processes and circumstances that lead to a single decision. These fields are also related to the economic discipline *game theory*, which focuses on conflict and cooperation between rational decision-makers [89]. Further adjacent domains are *multi-criteria decision analysis* (MCDA), which considers multiple contradicting criteria in decision-making environments, and the development of *decision support systems* (DSS), which support a human decision maker in complex decision making settings.

The term *problem solving* is also used in computer science in the domain of *artificial intelligence*. Here, it refers to *algorithms* and *heuristics* in computer software, which are developed to solve problems in specified presentations. Actually, the development of artificial intelligence is connected to research in the *cognitive sciences*, as we will see below.

Finally, the term *problem solving* is used for methods like *Failure Mode Effects Analysis* (FMEA), *Fault tree analysis* (FTA), and *forensic engineering*. These are systematic techniques to analyze failures or to prevent problems, e.g., in *engineering*.

In the remainder of this chapter, we will have a closer look on the domain problem solving in psychology.

2.2.1 Cognitive Revolution

Although there were some earlier contributions, research in problem solving mainly evolved with the development nowadays called *cognitive revolution* starting in the 1950s, enforced by developments during the 1930s and 1940s. The preceding decades of psychological research were dominated by *behaviorism* (see e.g., [54]). A characterization by the famous behaviorist JOHN B. WATSON [130] shows that cognitive processes were not regarded as important by behaviorism:

> "Psychology as the behaviorist views it is a purely objective experimental branch of natural science. Its theoretical goal is the prediction and control of behavior. Introspection forms no essential part of its methods, nor is the scientific value of its data dependent upon the readiness with which they lend themselves to interpretation in terms of consciousness."

While this essentially was a North American phenomenon, psychological research in Europe and in Germany in particular suffered from the events related to World War II.

According to GARDNER [57], multiple developments during the 1930s and 1940s finally led to the emergence of the discipline *cognitive science*. First, in 1936, British mathematician ALAN TURING developed a theoretical machine, which could carry out every plan or program that could be expressed in a binary code [127]—the *Turing machine*, which was extended, e.g., by JOHN VON NEUMANN by the concept of storage. Furthermore, WARREN MCCULLOCH and WALTER PITTS introduced the concept of *neural networks*, i.e., modeling the operation of nerve cells and their connections by logic expressions.

Inspired by this concept, NORBERT WIENER and colleagues realized that a combined examination of problems from control engineering and communication engineering is useful for the investigation of both mechanical and human self-correcting and self-regulating systems. Thus, WIENER stated [134] that they

> "[...] have decided to call the entire field of control and communication theory, whether in the machine or in the animal, by the name *Cybernetics* [...]."

Also by this time, CLAUDE SHANNON and WARREN WEAVER developed the key notion of *information technology*, recognizing that the principles of logic can be used to describe the two states of a relay switch. This insight could be used to describe all kinds of information by the basic unit *binary digits—bits*.

Finally, findings achieved during World War I and World War II on mental pathology caused by injury to the brain could barely be explained by behaviorism. Research on deficits caused by brain damage like *aphasia* (language deficit) and *agnosia* (difficulty in recognition) also enforced indications on the information processing function of the brains of healthy individuals.

With the development of stored program computers around 1950 and the first high-level programming languages, including *LISP* in 1958, there was a basis for simulation and investigation of cognitive processes. This eventually led to the emergence of *cognitive science*, a discipline which GARDNER [57] defines as

> "...a contemporary, empirically based effort to answer long-standing epistemological questions—particularly those concerned with the nature of knowledge, its components, its sources, its development, and its deployment."

These developments were the basis for psychological research in problem solving.

2.3 Theory of Problem Solving

There have been multiple theories to explain how humans solve problems. [54] gives an extensive overview on the different approaches. In contrast to theories in mathematics, in a mainly empirical discipline, theories can often not be proved or disproved directly and thus may coexist, e.g., considering different aspects of a phenomenon. We will have a closer look at three important theories on problem solving: the rather historical *Gestalt psychology* or *gestaltism* perspective, the *functionalism* viewpoint of information processing, and symbolic and connectionist approaches of *cognitive modeling*. The following sections are not meant to be exhaustive and for details we refer to [54].

2.3.1 Gestaltism

Gestaltism is based on psychology of perception, characterized by the famous KOFFKA quotation "the whole is other than the sum of the parts". This refers to the fact that humans consider an arrangement of objects as a whole with a higher-level structure, *gestalt*, rather than as a sum of its parts. Some of the *principles of grouping* which lead to formation of a certain gestalt are shown in Figure 2.2: the *principle of proximity*—objects which are close to each other appear as a group, the *principle of similarity*—objects which are similar to each other, e.g., have the same color, appear as a group, the *principle of closure*—missing parts are filled by human perception, the *principle of symmetry*—objects which are symmetrical are perceived as a group, and the *principle of prägnanz*—the perception of easily memorable structures, e.g., an object which differs from all others.

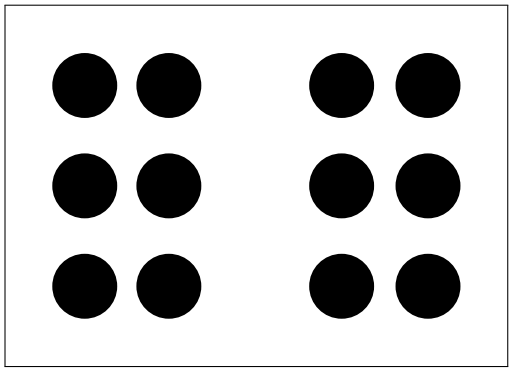
The connection of gestaltism to problem solving becomes apparent with a look at the *nine dots puzzle*, a problem in which the task is to connect nine dots with four straight lines or less without lifting the pen and without tracing one line more than once. The puzzle and possible solutions are depicted in Figure 2.3. The nine dots are arranged in a square, but a solution is only possible by drawing lines which go beyond this imaginary box. For most participants, it is difficult to overcome this boundary perceived because of the dots' proximity. This suggests that the principles of gestaltism are transferable to problem solving where a visualization is necessary and useful. However, the gestaltist description of problem solving by WALLAS [129], e.g., with the four stages *preparation*, *incubation*, *illumination*, and *verification* stays relatively vague on what actually happens in these stages. Another famous contributor was KARL DUNCKER, who did experiments in which participants had to "think aloud" during solving a problem and developed a theory of *psychology of productive thinking* [45] ("Psychologie des produktiven Denkens" in German).

2.3.2 Functionalism

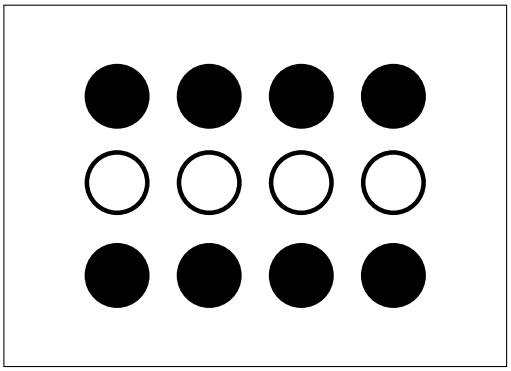
Functionalism is an approach concerned with the function of a system, e.g., a learning process, independently from its context, e.g., the learning content or the motivation of the learning person. Cognitive processes are considered as a functional organization of an information processing system [54].

In 1958, NEWELL, SHAW, and SIMON presented their *Logic Theorist* (LT) [91], a computer program able to draw logic conclusions. The software was designed to mimic human problem solving skills and thus can be considered the first *artificial intelligence* program. By the application to the problems in WHITEHEAD's and RUSSELL's *Principia Mathematica* [133], LT could not only proof several theorems, but also found some more elegant proofs.

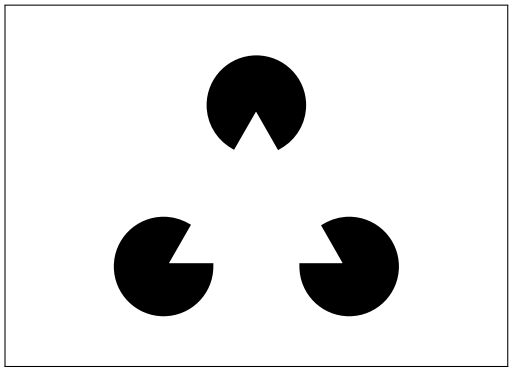
The LT was a basis for another computer program called *general problem solver* (GPS), also by NEWELL, SHAW, and SIMON, which was intended to solve a broad variety of problems by applying different generic problem solving methods. In fact, GPS could solve problems, which could be suf-



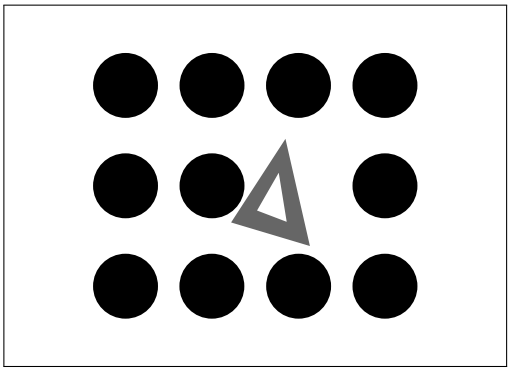
(a) Proximity



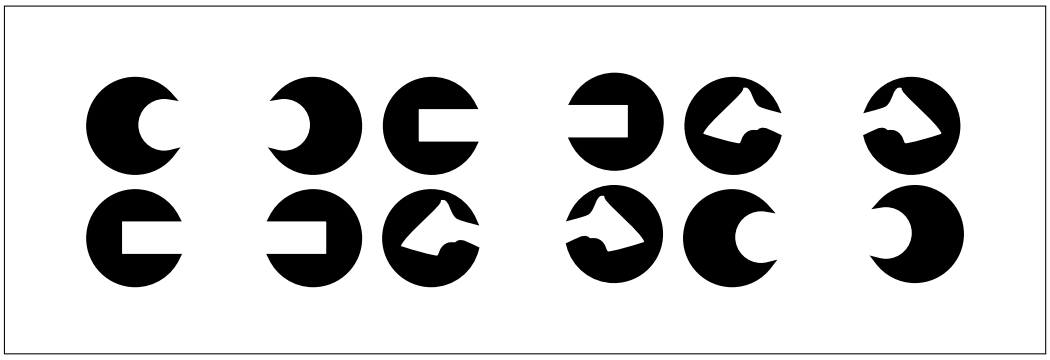
(b) Similarity



(c) Closure



(d) Prägnanz



(e) Symmetry

Figure 2.2: Principles of gestaltism: proximity—objects which are close to each other appear as a group, similarity—objects which are similar to each other appear as a group, closure—missing parts are filled by human perception, symmetry—objects which are symmetrical are perceived as a group, prägnanz—the perception of easily memorable structures, e.g., an object which differs from all others.

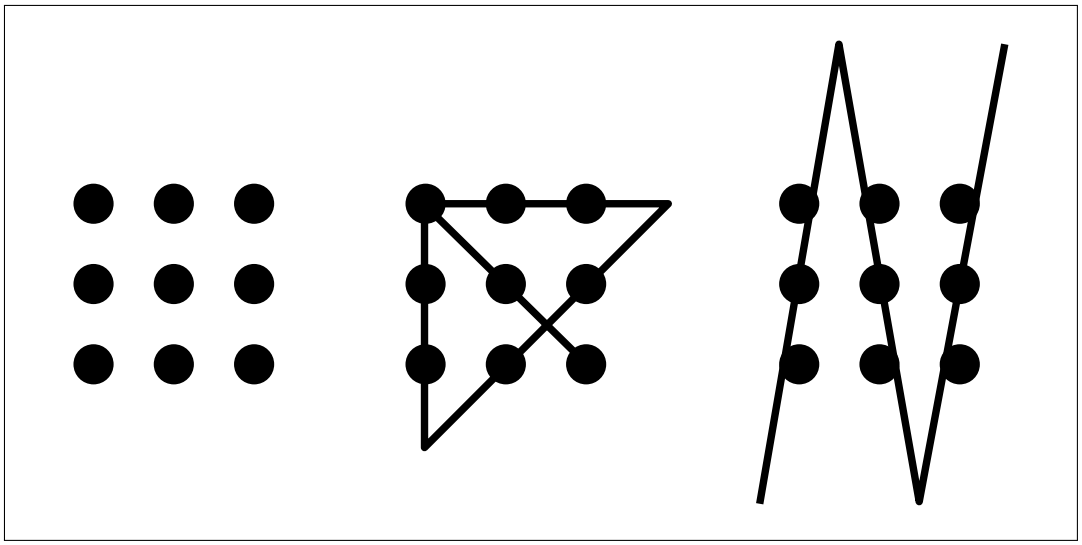


Figure 2.3: The *nine dots puzzle* and possible solutions thereof. The task is to connect nine points arranged in a square (left) with four (or less) straight lines without lifting the pen and without tracing the same line more than once. For a solution (mid and right) one needs to overcome the imaginary boundary induced by the square arrangement of the points.

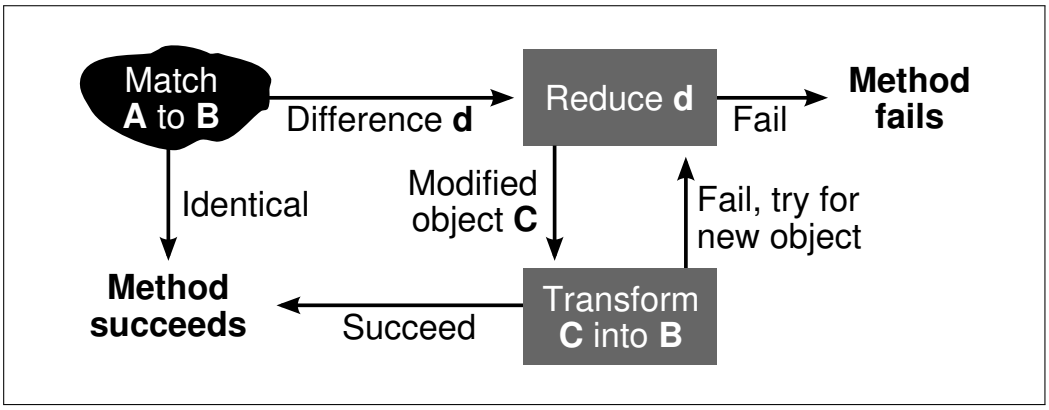
ficiently formalized with *objects* and *operators* being applied to them. The GPS separated its knowledge, i.e., input data, from its problem solving strategies.

GPS makes use of the division of a goal into subgoals or subproblems. Referring to the introductory example, if you want to get to Martinique, one subgoal possibly is to purchase a flight ticket and another could be to get to the airport. GPS can deal with three different types of subproblems corresponding to three different problem solving methods, see Figure 2.4. First type is *transformation* of an object A into another object B, which consists in computation and partial reduction of the difference between the two objects—another subproblem. The *application* of an operator to an object checks feasibility of an operator q and transforms an object into an appropriate input form if necessary. Finally, for a *reduction* of a difference, GPS searches for an operator which may reduce the difference and applies it if found. The way GPS chooses actions to solve problems is called *means-ends-analysis*.

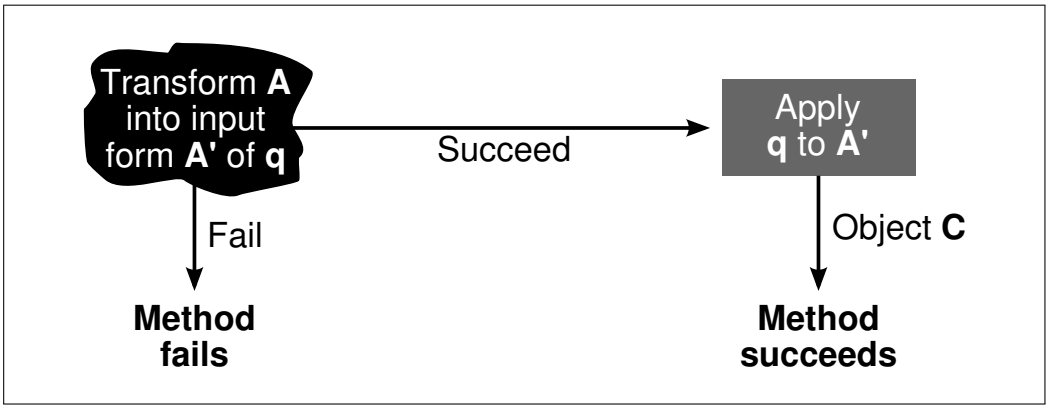
However, a fundamental problem of GPS is that it is only able to treat problems which can be sufficiently formalized and are simple enough, like the *Tower of Hanoi*. For more general problems, the division into subgoals suffers from combinatorial explosion and thus, GPS has not been able to solve real-world problems.

Later on, NEWELL and SIMON also published a theory on *Human Problem Solving* [90], which still is an important basis for the functionalist approach [54]. Their theory consists of the two processes of *understanding*, which creates the *problem space* as an inner representation of a problem, and *search* in that problem space. Such a problem space is characterized by the initial state, the available operators for transformation of states, and a criterion to evaluate if the goal has been achieved.

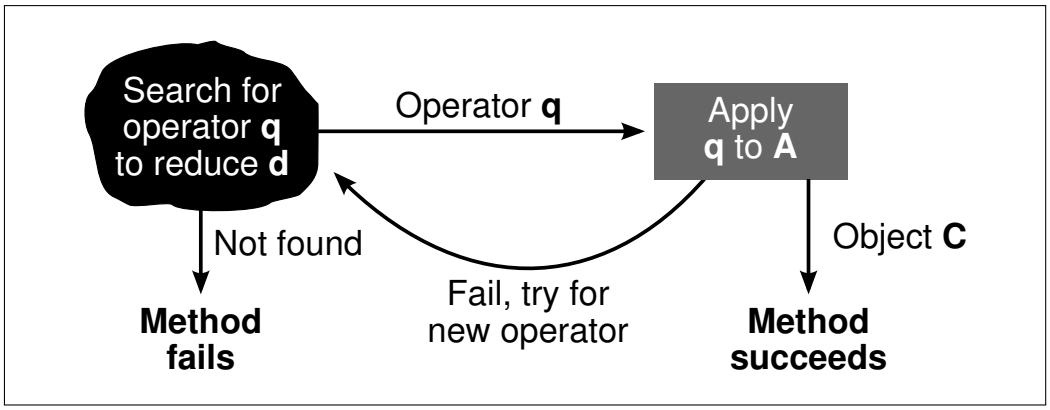
A problem solver searches for differences between the current state and the goal state and for operators which can be applied to achieve a change of the current state. Possible search procedures are, e.g., *generate and test*—generation of possible solutions, which then are tested, *forward* and *backward chaining*—the consecutive application of operators starting with the current or the goal state



(a) Transform object A into object B.



(b) Apply operator q on object A.



(c) Reduce difference d between objects A and B

Figure 2.4: Problem solving methods of the *General Problem Solver* according to [92]: transformation, operator application, difference reduction.

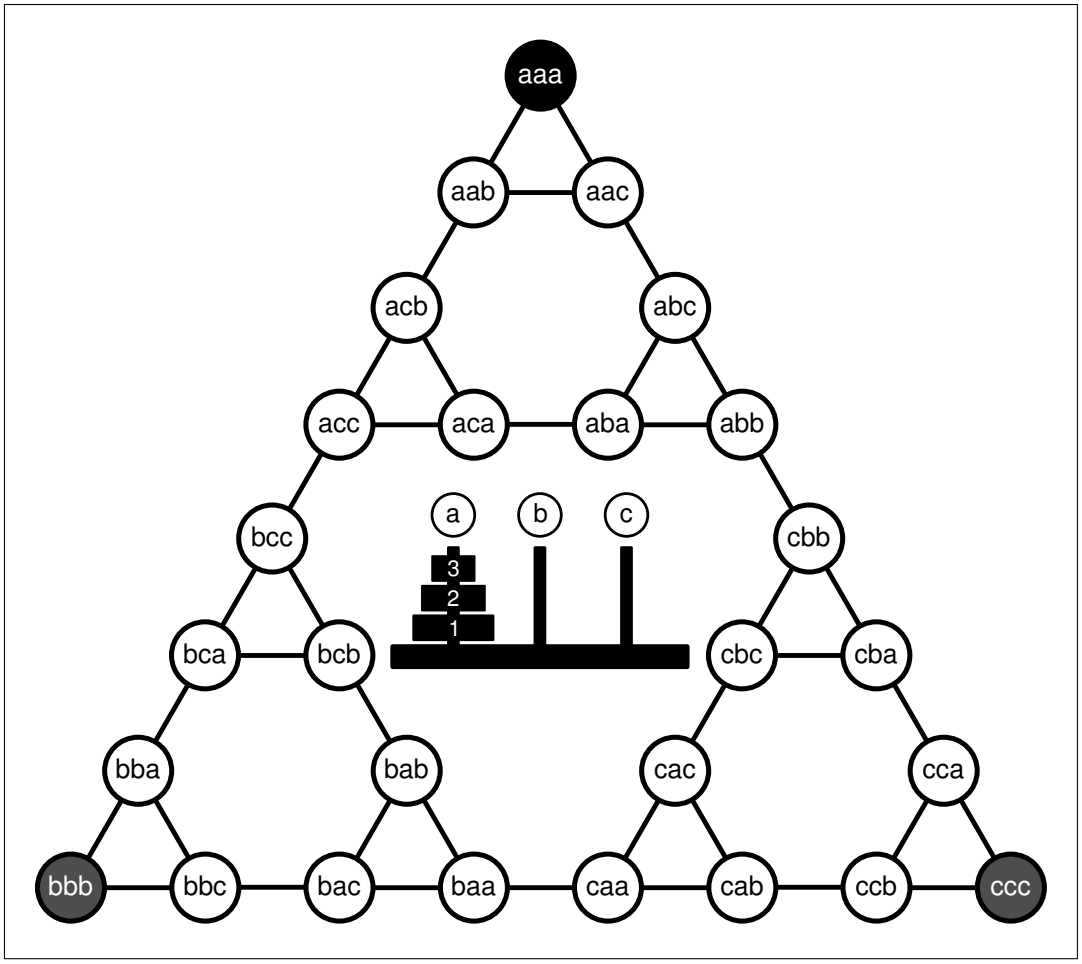


Figure 2.5: Possible states of *Tower of Hanoi* with three rods and three disks. Each circle represents one state with the first character referring to the position of the largest disk, the second character to the middle disk, and the third character to the smallest disk. Possible positions for each disk are the three rods labeled a, b, and c from left to right. The states resemble a *Sierpiński* [119] triangle. An optimal strategy obviously moves along the right hand edge of this triangle.

respectively, *subgoal decomposition*—the decomposition of the goal into subgoals which are easier to achieve, and *difference reduction*—the search for the operator which most reduces difference between current state and goal state. All these, however, are *weak methods* because of their generality. More specific methods may be stronger, but can be applied only to specific problems.

In this approach, problem solving starts with the creation of a problem space from the problem description. During the problem solving process, understanding and search in the problem space may be executed without a fixed sequence.

DÖRNER follows a similar approach, considering problem solving as information processing in relation to specific sections of reality which consist of *circumstances* and *operators* [41]. Circumstances are characterized by the five properties *complexity*, *interdependence*, *dynamics*, *opaqueness*, and *polytely*. We will explain these properties below in the context of complex problems as circumstances with a high level of all of them are considered to be complex.

Operators according to DÖRNER are characterized by the properties *reliability of impact*, *breadth of impact*, *reversibility*, and *conditions of application*. Reliability of impact means that at least for some operators there is only some probability that the desired impact is achieved, think e.g., of medicine. The number of properties influenced by an operator represents the breadth of impact. Reversibility refers to the possibility to undo the effect of an operator and of course, most operators can only be applied if certain conditions of application are fulfilled.

DÖRNER also developed a software to simulate psychological processes on computers, *Psi* (ψ), which includes a simulation of emotions. ψ is able to store experience and to draw conclusions, but also has needs which need to be fulfilled by certain activities. This is related to the approaches of *cognitive modeling* in the following section.

2.3.3 Cognitive Modeling

In [54], cognitive modeling is specified as the attempt to describe cognitive processes such that they can be executed on a machine. Cognitive modeling has the aim to reflect both processes and results of human problem solving. There are mainly two approaches to cognitive modeling, *symbolic* and *connectionist*. Both approaches make use of *cognitive architectures*, a term which ANDERSON [5] defines as

> "[...] a specification of the structure of the brain at a level of abstraction that explains how it achieves the function of the mind."

Symbolic models build upon a *production system*, which separates between production rules and data. Production rules consist of a precondition and an action which is executed if the condition is fulfilled, i.e., these rules are *if–then* statements. The approach is to *match* the production rules against the current state to check which rules apply. If this is true for more than one rule, then the system decides in a *conflict resolution* which rule to apply and finally executes respectively *fires* the corresponding rule.

One of the most famous cognitive architectures is *ACT-R* (*adaptive control of thought—rational*) mainly developed by JOHN R. ANDERSON. A detailed description of this architecture and the theory it is based on is given, e.g., in [6] and a schematic overview of ACT-R's modules and their interconnections is depicted in Figure 2.6. Knowledge is stored as *chunks* in the *declarative memory*, whereas *procedural knowledge* is stored as production rules in the production system. For different tasks, one has to create different *models* by programming production rules and defining the structure of chunks. The basic assumptions of the architecture are claimed to be based on research results on the human brain, e.g., from cognitive neuroscience.

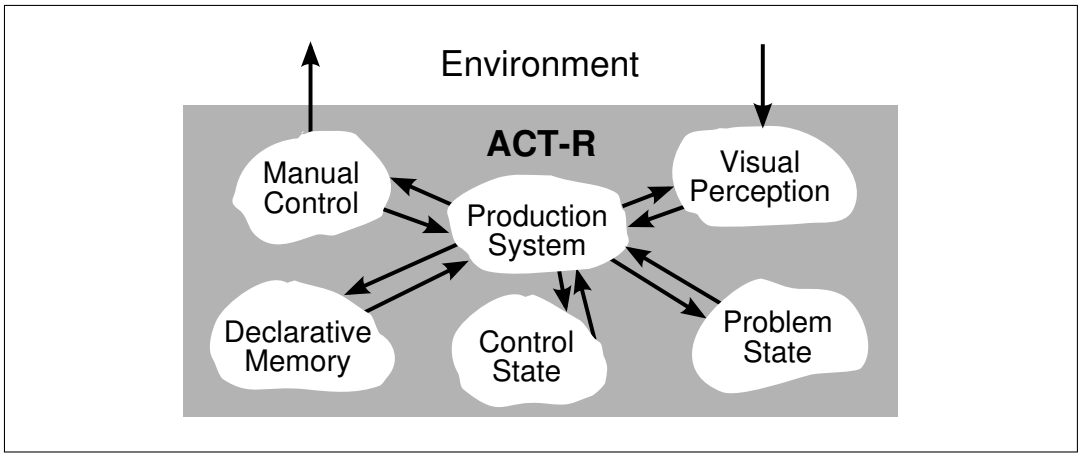


Figure 2.6: Interconnections among ACT-R modules according to [5].

The connectionist approach, on the other hand, is based on small and often uniform units and connections between them, i.e., *networks*, which change over time. This refers to the neurobiology of the human brain with units corresponding to neurons and connections corresponding to synapses. The most common connectionist models thus are *neural networks*.

In contrast to symbolic systems, a connectionist system does not need rules, but adapts to its environment by changing its network and thus is able to "learn". Therefore, connectionist models obviously are beneficial for *pattern recognition*.

2.4 Simple vs. Complex Problems

What is the difference between a problem like *Tower of Hanoi* or the *nine dots puzzle* and problems of daily life, like how to get to Martinique or to maximize a company's profit? Apparently, the latter are ill-defined. More precisely, their problem space is open and a problem solver has to deal with both knowledge and emotions.

Doubts about the relevance of *simple problems* like the above-mentioned for insight into the psychology and cognitive processes related with daily decision making created the research domain CPS starting in the 1970s, which deals with *complex problems*. Such problems are considered to be similar to problems we encounter and solve in everyday life and thus investigation of CPS is claimed to yield more insight into real world human decision making. FRENSCH and FUNKE [50] define CPS as follows.

> "CPS occurs to overcome barriers between a given state and a desired goal state by means of behavioral and/or cognitive, multistep activities. The given state, goal state and barriers between given state and goal state are complex, change dynamically during problem solving and are intransparent. The exact properties of the given state, goal state, and barriers are unknown to the solver at the outset. CPS implies the efficient interaction between a solver and the situational requirements of the task, and involves a solver's cognitive, emotional, personal, and social abilities and knowledge."

In [54], complex problems are characterized by the five properties *complexity*, *interdependence*, *opaqueness*, *dynamics*, and *polytely*. The word complex in CPS refers to the complexity of these prop-

erties. In fact, it is not clear that cognitive processes in CPS are more complex than those related to simple problems, although they probably differ.

Figure 2.7 visualizes the five properties of complex problems. Here, complexity often refers to the number of variables of a problem. However, this is not the only aspect of complexity and CASTI [34] states that

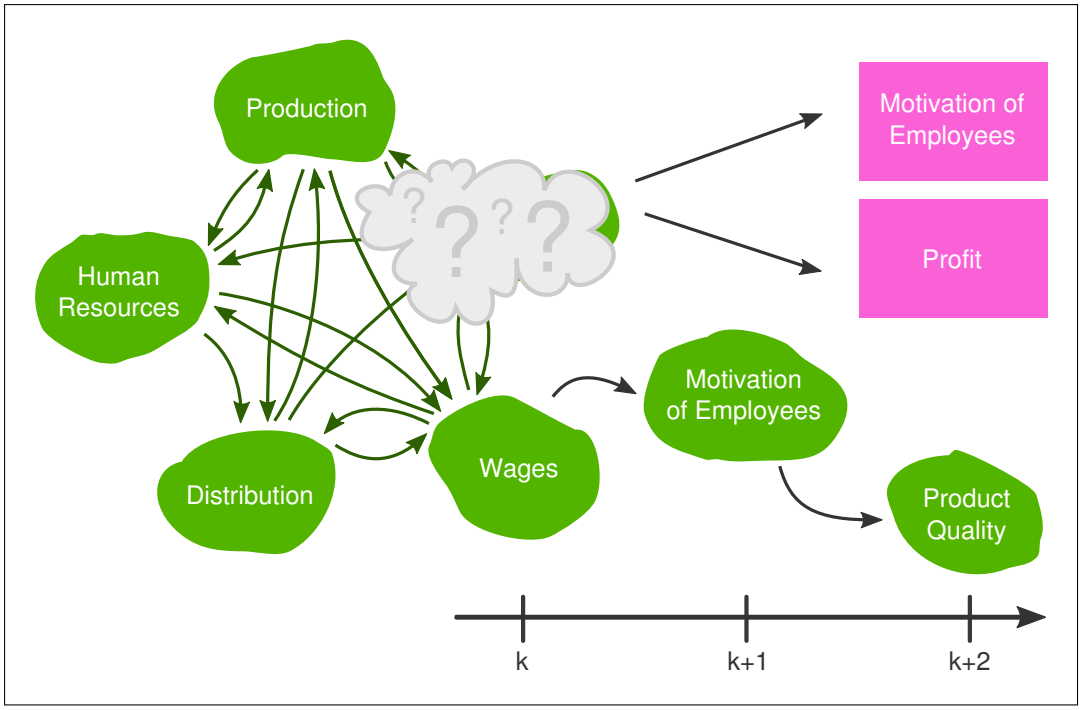


Figure 2.7: Properties of a complex problem: complexity and interdependencies, opaqueness, polytely, and dynamics.

> "Of all the adjectives in common use in the system analysis literature, there can be little doubt that the most overworked and least precise is the descriptor 'complex'. In a vague intuitive sense, a complex system refers to one whose static structure or dynamic behavior is 'unpredictable', 'counterintuitive', 'complicate', or the like. In short, a complex system is something quite complex [...]"

Complexity also incorporates the relevance of historical data, hierarchical organization, and system conditions like constraints and degrees of freedom [54]. Interdependence between variables is closely connected to complexity and is required for variables to build a system. Because of connections between variables, participants need to create a mental model of dependencies.

Dynamics refers to the question how a system develops over time. According to [54], it is the property human problem solvers have the most difficulties with. Dynamics require a prediction of future developments of the variables. In a company, for instance, the development of the variable *wages* may influence, e.g., the *motivation of employees* in the future. Depending on the sensitivity, a prediction may be difficult and can possibly be too demanding—at least for humans.

Opaqueness refers to incomplete information, e.g., on the connections between variables, which requires an active search for information. Polytely, finally, is characterized by multiple, possibly contradictory aims, e.g., to maximize a company's profit and to achieve a certain level of employees' motivation, which may have contradictory implications on the variable wages. Such aims can be set externally as a task, e.g., "maximize profit", but can also be set implicitly by the problem solvers themselves, e.g., as an interpretation of a vague aim like "solve the problem".

2.5 Microworlds

Complex problems are often implemented as computer-based scenarios which simulate a part of the real world, e.g., a shirt company. These simulation are also called *microworlds* and emerged in the 1980s, when computing time became available for psychological research.

Computer-simulated microworlds opened up a third type of research in CPS besides laboratory experiments and field research. While laboratory research has often been criticized for its lack of relevance for human behavior in the real world, field research often lacks control of the experimental conditions [30]. In laboratory experiments, the level of complexity is often too low to produce results which are relevant for the real world. Vice versa, in field research, there usually is too much complexity to make definite conclusions.

Microworlds, according to [30], yield the possibility to create complex settings for CPS research with controlled conditions and thus, to some extent, combine the advantages of laboratory and field research. In contrast to static problems, as [53] states, to control and explore computer-based scenarios, participants need to "continuously acquire and use knowledge about the internal structure of the system". Therefore, microworlds are widely used as computer-based test scenarios in CPS for some decades. Prominent examples are *Lohhausen*—in which participants have to take decisions as the (dictatorial) mayor of an imaginary city [43], *Moro*—in which participants advise an imaginary African tribe as a development worker [44], and *Tailorshop*—a business simulation.

2.6 The *Tailorshop*: a Complex Microworld

The *Tailorshop* (German name: *Schneiderwerkstatt*) is a microworld which has been developed and implemented as a computer-based test scenario in the 1980s by DÖRNER [42] and co-workers. There are numerous studies based on it, e.g., [102, 78, 76, 87, 11, 12]. Comprehensive reviews in which more information on the psychological background can be found have also been published, see e.g., [50, 52, 54, 56, 55]. Thus, *Tailorshop* is one of the most famous and most important test scenarios in CPS and was also referred to as the "Drosophila" for problem solving researchers [55].

In the *Tailorshop*, participants have to make economic decisions as the head of an imaginary company, which produces and sells shirts. Usually, the task is to maximize the overall balance of that company over twelve rounds, which are referred to as *months*. Participants can modify infrastructure (employees, machines, distribution vans), financial settings (wages, maintenance, prices), and logistical decisions (shop location, buying raw material) in each round. In total, the system consists of 31 variables. Differing numbers of variables in other publications, e.g., in [54], are due to different methods of counting, like the combination of states and their corresponding controls. These variables can be separated into 15 free *control* variables, which can be chosen by the participant within certain constraints, and 16 dependent *state* variables, which are computed depending on the participant's actions and historic state values. Table 2.1 gives an overview of the variables in *Tailorshop*.

The two types of machines refer to the amount of shirts they can produce in one month. Workers can be trained to one machine type and then have to work on either a 50 or a 100 shirt machine.

States	**Variable**	**Unit***	**Controls**	**Variable**	**Unit***
machines 50	$x^{M_{50}}$	machine(s)	advertisement	u^{AD}	MU
machines 100	$x^{M_{100}}$	machine(s)	shirt price	u^{SP}	MU
workers 50	$x^{W_{50}}$	worker(s)	buy raw material	$u^{\Delta MS}$	shirt(s)
workers 100	$x^{W_{100}}$	worker(s)	workers 50	$u^{\Delta W_{50}}$	worker(s)
demand	x^{DE}	shirt(s)	workers 100	$u^{\Delta W_{100}}$	worker(s)
vans	x^{VA}	van(s)	buy machines 50	$u^{\Delta M_{50}}$	machine(s)
shirts sales	x^{SS}	shirt(s)	buy machines 100	$u^{\Delta M_{100}}$	machine(s)
shirts stock	x^{ST}	shirt(s)	sell machines 50	$u^{\delta M_{50}}$	machine(s)
possible production	x^{PP}	shirt(s)	sell machines 100	$u^{\delta M_{100}}$	machine(s)
actual production	x^{AP}	shirt(s)	maintenance	u^{MA}	MU
material stock	x^{MS}	shirt(s)	wages	u^{WA}	MU
satisfaction	x^{SA}	—	social expenses	u^{SC}	MU
machine capacity	x^{MC}	shirt(s)	buy vans	$u^{\Delta VA}$	van(s)
base capital	x^{BC}	MU	sell vans	$u^{\delta VA}$	van(s)
capital after interest	x^{CA}	MU	choose site	u^{CS}	—
overall balance	x^{OB}	MU			

Table 2.1: Controls and states in the *Tailorshop* microworld. Note that units are only given implicitly in the test scenario. * MU means money units.

We will have a closer look on the mathematical model *Tailorshop* is based on and its equations in Chapter 4.

The *Tailorshop* was one of the first complex test scenarios available for direct control by participants on a computer. Earlier versions and microworlds were computed on a mainframe computer and thus had a delay from the participant's decision to the next round. A German *GW-BASIC* interface for *Tailorshop* is shown in Figure 2.9. The text-based interface shows values of states and controls (but not all of them) in the upper part, and possible actions in the lower part. Arrows next to the values indicate differences to the previous round.

Figure 2.8 shows the model structure of *Tailorshop*. The variables are interconnected with partly nonlinear relations and the microworld is dynamic with discrete time. In general, participants do not know the specific dependencies between the variables at the beginning which makes the problem opaque. However, participants can usually buy different levels of information on each variable. For most studies, polytely was not considered explicitly.

The usual approach in CPS is to evaluate the participant's performance in a test scenario like *Tailorshop* and to either correlate it to personal attributes or to analyze the influence of different experimental conditions for groups of participants. Thus, the measurement of performance is crucial.

Within the *Tailorshop* scenario different indicator functions have been proposed in the literature. For a review on *Tailorshop* success criteria, see e.g., [39]. To use a comparison of accumulated capital at the final month 12 between all participants was proposed in [72]. This criterion seems natural, as this is what the participants usually are requested to maximize. However, it cannot yield insight into the temporal process and is not objective in the sense that the performance depends on what other

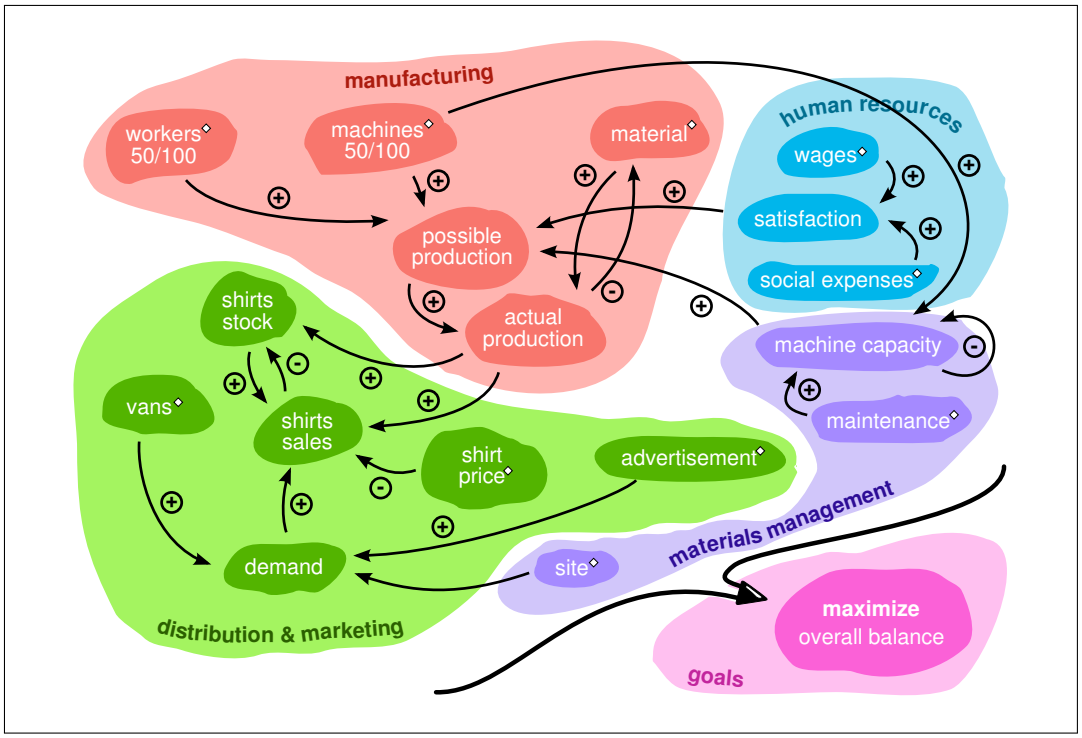


Figure 2.8: Schematic representation of the *Tailorshop* microworld: bubbles represent variables, white diamonds indicate participant's control possibilities, and arrows show dependencies with + and − for proportional and reciprocal influence respectively.

participants achieved.

Analyzing the temporal evolution of state variables has also been proposed. In [101, 122] the evolution of profit, equivalent to the evolution of capital after interest x^{CA}, was proposed. In [51, 12] the evolution of the overall worth of the *Tailorshop* x^{OB} was used.

An obvious drawback of comparing the results of several rounds with one another is that the main goal of the participant is to maximize the value at the end of the test, not necessarily in between. Thinking about the analogy of maximizing the amplitude of a pendulum with a hair dryer, in certain scenarios "going back" to gain momentum is obviously better than pushing it all the time in the desired direction. The same is true for the *Tailorshop* scenario. It may be better to invest into infrastructure at the beginning (which is actually decreasing the overall capital as infrastructure looses value over time) to have a higher pay-off towards the last rounds of the test. Hence it might happen that decisions are analyzed to be bad, while they are actually good ones and vice versa.

In Chapter 5, we present an approach to measure performance in microworlds like *Tailorshop* based on mathematically optimal solutions. Furthermore, we explain, how mathematical optimization can be used to train participants in controlling such microworlds.

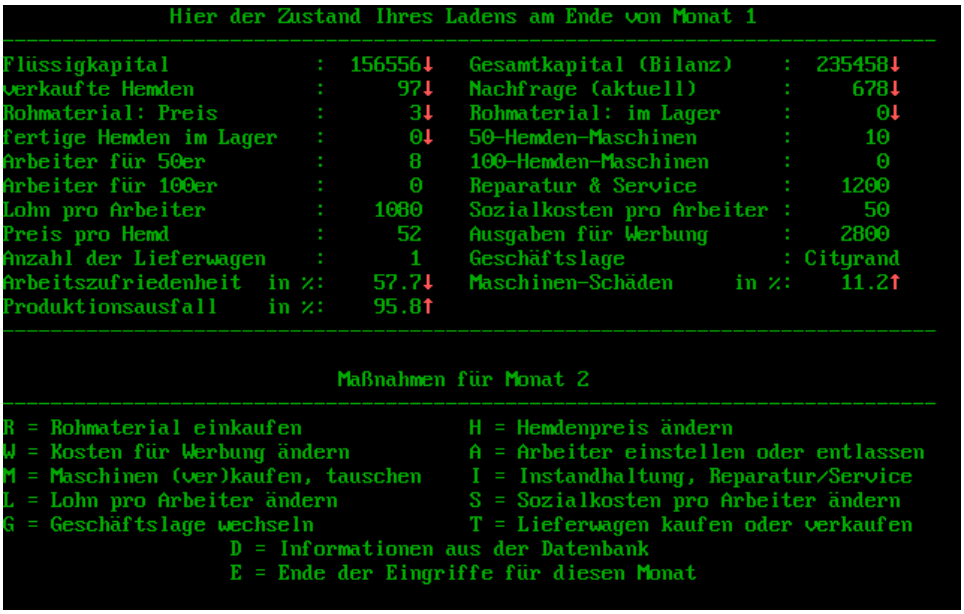


Figure 2.9: German *GW-BASIC* interface for *Tailorshop*. The text-based user interface shows current values of the model variables in the upper part, possible interventions in the lower part.

CHAPTER 3

Mathematical Optimization and Optimal Control

This chapter describes optimization problems and mathematical optimization algorithms, which are relevant for the analysis of human decision making in complex microworlds, as we will see in the following chapters. We start with a general introduction into mathematical optimization, then formulate the relevant problem classes, and eventually present algorithms for their solution.

3.1 Optimization

Mathematical *optimization* has a long record of successful improvements in many technological and scientific areas, for tasks as diverse as design, scheduling, business control rules, process control, and the like. More recently, optimization has also been successfully applied in the context of inverse problems, e.g., for the choice and calibration of mathematical models, or as a modeling paradigm for biological systems. In this thesis we use numerical optimization as a tool for both analysis and training of human decision making. For the modeling of our new test scenario, the *IWR Tailorshop*, optimization methods were considered from the beginning to ensure desired properties and model behavior. We start with a general introduction.

There are many different classes of optimization problems which are considered in mathematical optimization depending on the properties they have and the algorithms needed to solve them. A very general formulation of a mathematical optimization problem is the following.

Definition 3.1 An *optimization problem* consists of an *objective function* $F : X \to \mathbb{R}$, *equality constraints* $G : X \to \mathbb{R}^{n_e}$, and $H : X \to \mathbb{R}^{n_i}$ *inequality constraints* with $X \subset \mathbb{R}^{n_x}$, n_e the number of equality constraints, n_i the number of inequality constraints, and $x \in X$ the optimization variables. The problem then has the following form:

$$\begin{aligned} \min_x \quad & F(x) \\ \text{s.t.} \quad & G(x) = 0, \\ & H(x) \leq 0, \\ & x \in X \subset \mathbb{R}^{n_x}. \end{aligned} \tag{3.1}$$

Obviously, the restriction to minimization is no limitation of generality, since maximizing $F(x)$ is the same as minimizing $-F(x)$. Similar applies to $H(x) \leq 0$. If there are no constraints, i.e., $n_e = n_i = 0$, the problem is *unconstrained*. For most applications, however, there will be some kind of constraints.

Definition 3.2 A point $x \in X$ is said to be *feasible* if

$$G(x) = 0 \quad \text{and} \quad H(x) \leq 0. \tag{3.2}$$

The *feasible set* $\mathcal{F}$ is the set of feasible points,

$$\mathcal{F} = \{x \in X | G(x) = 0; H(x) \leq 0\}. \tag{3.3}$$

Definition 3.3 The *active set* $\mathcal{A}(x)$ for any feasible point x consists of the indices of all active inequality constraints, i.e.,

$$\mathcal{A}(x) = \{i | H_i(x) = 0; 1 \leq i \leq n_i\}. \tag{3.4}$$

The active set is in particular important for *active set optimization methods* like *sequential quadratic programming* (SQP), which will be discussed below. In optimization, one seeks for minima or maxima respectively. Therefore, it is important to recall the following definitions.

Definition 3.4 A point $x^* \in X$ is said to be a *local minimizer* of F if there is a neighborhood U of x^*, such that

$$F(x^*) \leq F(x) \quad \forall x \in U \cap X. \tag{3.5}$$

Definition 3.5 A point $x^* \in X$ is said to be a *global minimizer* of F if

$$F(x^*) \leq F(x) \quad \forall x \in X. \tag{3.6}$$

In general, one would like to find a global minimizer for Problem (3.1) and for some problems, e.g., *convex* problems, every local minimizer is a global minimizer. However, for non-convex problems, one often is satisfied to find a local optimum. The field of *global optimization* deals with the development of algorithms which find a global optimum for non-convex problems.

Note that all these general definitions contain no additional assumptions on F, G, H, and X. Thus, optimization problems according to Definition 3.1 comprise problems as diverse as *linear programs*, *nonlinear programs*, problems with integral or binary variables, problems with stochastic components, problems with nondifferentiabilities, and the like. This variety of optimization problem classes requires different approaches for an efficient solution. In the following, we concentrate on *nonlinear* and *mixed-integer nonlinear programs* and algorithms for these classes.

3.2 Nonlinear and Mixed-Integer Nonlinear Programming

Definition 3.6 A *nonlinear program* (NLP) is an optimization problem of the form

$$\begin{aligned} \min_{x} \quad & F(x) \\ \text{s.t.} \quad & G(x) = 0 \\ & H(x) \leq 0 \\ & x \in X \subset \mathbb{R}^{n_x} \end{aligned} \tag{3.7}$$

with nonlinear, smooth, twice continuously differentiable functions $F : X \to \mathbb{R}$, $G : X \to \mathbb{R}^{n_e}$, and $H : X \to \mathbb{R}^{n_i}$.

Definition 3.7 For a point x, we say that the *linear independence constraint qualification* (LICQ) holds if the gradients of the equality constraints $\nabla G_i(x) \forall i \in \{1, \ldots, n_e\}$ and the gradients of all active inequality constraints $\nabla H_i(x) \forall i \in \mathcal{A}(x)$ are linearly independent, where ∇ means the derivative with respect to x.

We now have a look at the necessary and sufficient conditions for optimality. For these conditions, LICQ is important, although there are other (i.e., weaker) constraint qualifications which can be used as well. For the ease of notation, we first introduce the Lagrangian.

Definition 3.8 The *Lagrangian function* for problem (3.7) is defined as

$$\mathcal{L}(x,\lambda,\mu) = F(x) + \lambda^T G(x) + \mu^T H(x) \tag{3.8}$$

with the LAGRANGE *multiplier* vectors $\lambda \in \mathbb{R}^{n_e}$ and $\mu \in \mathbb{R}^{n_i}$.

Now we can formulate the *first order necessary conditions* for minimizers.

Theorem 3.1 Let x^* be a local minimizer of problem (3.7) and assume that LICQ holds at x^*. Then there are unique LAGRANGE multipliers $\lambda^* \in \mathbb{R}^{n_e}$ and $\mu^* \in \mathbb{R}^{n_i}$ such that the following conditions hold at (x^*, λ^*, μ^*).

$$\nabla_x \mathcal{L}(x^*,\lambda^*,\mu^*) = 0 \tag{3.9a}$$
$$G(x^*) = 0 \tag{3.9b}$$
$$H(x^*) \leq 0 \tag{3.9c}$$
$$\mu^* \geq 0 \tag{3.9d}$$
$$\mu^{*T} H(x^*) = 0. \tag{3.9e}$$

The conditions (3.9) are known as the KARUSH-KUHN-TUCKER *conditions* and a point (x, λ, μ) which fulfills them is often called a KARUSH-KUHN-TUCKER or *KKT point. Second order necessary conditions*, where *second order* refers to the second derivative of the Lagrangian, are the following.

Theorem 3.2 Let x^* be a local minimizer of problem (3.7) and assume that LICQ holds at x^*. Furthermore, let λ^* and μ^* be LAGRANGE multiplier vectors, such that the conditions (3.9) are satisfied. For all Δx with

$$\nabla G_i(x^*)\,\Delta x = 0 \qquad \forall i \in \{1,\dots,n_e\}, \tag{3.10a}$$
$$\nabla H_i(x^*)\,\Delta x = 0 \qquad \forall i \in \mathcal{A}(x^*), \tag{3.10b}$$

we then have

$$\Delta x^T\, \nabla^2_{xx}\mathcal{L}(x^*,\lambda^*,\mu^*)\, \Delta x \geq 0. \tag{3.11}$$

Finally, *(second order) sufficient conditions* for minimizers, i.e., conditions that guarantee that a feasible x^* is a (local) minimizer, are given by the following theorem.

Theorem 3.3 Let x^* be a feasible point for problem (3.7) and λ^* and μ^* LAGRANGE multiplier vectors, such that the conditions (3.9) are satisfied. Assume that

$$\Delta x^T\, \nabla^2_{xx}\mathcal{L}(x^*,\lambda^*,\mu^*)\, \Delta x > 0. \tag{3.12}$$

for all $\Delta x \neq 0$ with

$$\nabla G_i(x^*)\,\Delta x = 0 \qquad \forall i \in \{1,\dots,n_e\}, \tag{3.13a}$$
$$\nabla H_i(x^*)\,\Delta x = 0 \qquad \forall i \in \mathcal{A}(x^*) \text{ and } \mu_i > 0, \tag{3.13b}$$
$$\nabla H_i(x^*)\,\Delta x \geq 0 \qquad \forall i \in \mathcal{A}(x^*) \text{ and } \mu_i = 0. \tag{3.13c}$$

Then x^* is a local minimizer for problem (3.7).

Proofs for these three theorems can be found, e.g., in [94].

Some components of optimization problems may need to be formulated with *integer* or *binary* variables. In complex microworlds, for instance, impartible resources like employees, machines, and production sites are represented by an integer variable. Problems which only consist of integer variables are called *integer programs*. In the context of complex microworlds, however, such variables are often combined with continuous components like an amount of money spent, e.g., for wages or advertising. Together with nonlinear dependencies between the variables, this leads to the following problem class.

Definition 3.9 A *mixed-integer nonlinear program* (MINLP) is an optimization problem of the form

$$\begin{aligned} \min_{x,y} \quad & F(x,y) \\ \text{s.t.} \quad & G(x,y) = 0 \\ & H(x,y) \leq 0 \\ & x \in X \subset \mathbb{R}^{n_x} \\ & y \in Y \subset \mathbb{R}^{n_y} \cap \mathbb{Z}^{n_y} \end{aligned} \tag{3.14}$$

with $F : X \times Y \to \mathbb{R}$, $G : X \times Y \to \mathbb{R}^{n_e}$, and $H : X \times Y \to \mathbb{R}^{n_i}$ nonlinear and twice continuously differentiable.

Such problems with linear F, G, and H are called *mixed-integer linear programs*. This problem class is known to be at least *NP-hard* (see, e.g., [58]), and as it is contained in the class of MINLP, the MINLP problem class is at least NP-hard, too. Note that in problem (3.14), the notation of the inequality constraints changed to $H(x,y) \leq 0$, which is useful for the description of MINLP algorithms in Section 3.4.

3.3 Optimal Control

Optimal control is a field related to (nonlinear) optimization. In optimal control, one strives for a *control function* $u(t)$, which minimizes an *objective functional* subject to some kind of dynamics determining the state function $x(t)$. These dynamics model a (nonlinear) process and usually are based on some kind of differential equations, e.g., *ordinary differential equations* (ODE), *differential algebraic equations* (DAE), or *partial differential equations* (PDE). Considering ODE dynamics, an optimal control problem can be defined as follows.

Definition 3.10 An *optimal control problem* (OCP) is an optimization problem of the form

$$\begin{aligned} \min_{x,u,p} \quad & F[x,u,p] \\ \text{s.t.} \quad & \dot{x}(t) && = G(x(t),u(t),p), && t \in [t_0,t_f], \\ & 0 && \geq H(x(t),u(t),p), && t \in [t_0,t_f], \\ & u(t) && \in U \subset \mathbb{R}^{n_u}, && t \in [t_0,t_f], \\ & x(t_0) && = x_0. \end{aligned} \tag{3.15}$$

with F the objective functional, $G : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_x}$ the ODE right hand side, and $H : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_i}$ additional control and path constraints. t is the time, $x : [t_0,t_f] \to \mathbb{R}^{n_x}$ the state function, $u : [t_0,t_f] \to \mathbb{R}^{n_u}$ the control function, and $p \in \mathbb{R}^{n_p}$ parameters with their corresponding numbers n_x, n_u, and n_p. x_0 is called initial value, t_0 is the start time, and t_f the end time.

Optimal control problems can be formulated for a variety of processes, e.g., in physics, chemistry, and engineering. For instance, for a car traveling on a road, an optimal control problem may be to minimize the traveling time to get from a starting point to some end point elsewhere on the road. The car can be controlled by acceleration, braking, and some other controls. An ODE may be used to model the physics and there are several constraints, e.g., the car has to stay above the road (and its wheels should touch the road, at least most of the time).

By the notation of Problem (3.15), an optimal control problem is an infinite-dimensional problem, as the optimization variables are functions. For the solution of these problems, there are two basic approaches. *Indirect methods* are based on PONTRYAGIN's *maximum principle* [121], which formulates necessary optimality conditions in the infinite-dimensional function space. These conditions are used to transform the optimal control problem into a boundary value problem, which can be solved by an appropriate discretization, e.g., by *single* or *multiple shooting* (*first-optimize-then-discretize*, [32, 21]). *Direct methods*—including *direct single shooting*, *direct multiple shooting* [24], and *collocation* [125, 19]—discretize the optimal control problem by an appropriate state and control discretization to transform it into a finite-dimensional nonlinear program and the optimize the NLP (*first-discretize-then-optimize*). Another approach is *dynamic programming*, which is based on the HAMILTON-JACOBI-BELLMAN equation [14].

3.3.1 Mixed-Integer Optimal Control Problems (MIOCP)

In the case of the optimal car control example from the previous section, in reality, the choice of gear will also be a control function. In contrast to braking and acceleration, however, gears require discrete values as it usually is not possible to select gear 3.141592, but gear 3 or 4 (see [75] for this application). This example illustrates that some processes require a mixed-integer control function, which leads to the following modification of Problem (3.15), comparable to the relation of MINLPs to NLPs.

Definition 3.11 A *mixed-integer optimal control problem* (MIOCP) is an optimization problem of the form

$$\begin{aligned} \min_{x,u,v,p} \quad & F[x,u,v,p] \\ \text{s.t.} \quad & \dot{x}(t) && = G(x(t),u(t),v(t),p), && t\in[t_0,t_f], \\ & 0 && \geq H(x(t),u(t),v(t),p), && t\in[t_0,t_f], \\ & u(t) && \in U\subset\mathbb{R}^{n_u}, && t\in[t_0,t_f], \\ & v(t) && \in \Omega\subset\mathbb{Z}^{n_v}, && t\in[t_0,t_f], \\ & x(t_0) && = x_0 \end{aligned} \tag{3.16}$$

with F the objective functional, $G:\mathbb{R}^{n_x}\times\mathbb{R}^{n_u}\times\mathbb{R}^{n_v}\times\mathbb{R}^{n_p}\to\mathbb{R}^{n_x}$ the ODE right hand side, and $H:\mathbb{R}^{n_x}\times\mathbb{R}^{n_u}\times\mathbb{R}^{n_v}\times\mathbb{R}^{n_p}\to\mathbb{R}^{n_i}$ additional control and path constraints. t, x, u, p, x_0, t_0, t_f are defined as in Problem (3.15). $v:[t_0,t_f]\to\mathbb{Z}^{n_v}$ is the integer control function.

Thus, a MIOCP is an OCP with additional integrality constraints on some controls. Indirect methods have been applied to such problems (see, e.g., [22]) but are not capable of large scale MIOCPs. [108, 110, 113] present direct methods for this problem class.

3.3.2 Discretized Mixed-Integer Optimal Control Problems (dMIOCP)

Tasks in complex microworlds can also be seen as optimal control problems. The participant has to make decisions, comparable to the controls of an OCP, which influence some dependent variables, like the states. Furthermore, the task usually contains some kind of optimization problem, for

instance, to maximize a company's profit. However, most microworlds are round-based, i.e., they require the participant to make decisions at discrete time points. Together with mixed-integer decisions, such a task can be considered as a discretized version of Problem (3.16):

Definition 3.12 A *discretized mixed-integer optimal control problem* (dMIOCP) is an optimization problem of the form

$$\begin{aligned} \min_{x,u} \quad & F(x,u,p) \\ \text{s.t.} \quad & x_{k+1} = G(x_k,u_k,p), && k = t_0,\dots,t_f-1, \\ & 0 \geq H(x_k,u_k,p), && k = t_0,\dots,t_f, \\ & u_k \in \Omega_k, && k = t_0,\dots,t_f-1, \\ & x_{t_0} = x_0, \end{aligned} \tag{3.17}$$

with the objective function F, the state progression law $G : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_x}$, additional constraints $H : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_i}$, and $\Omega_k \in \mathbb{R}^{n_u}$ the feasible region for the controls at time k.

A dMIOCP therefore is both a special case of a MINLP and a *mixed-integer optimal control problem* (MIOCP). In the following section, we describe algorithms for the solution of the dMIOCP as a MINLP.

3.4 Optimization Algorithms

This section gives an overview of algorithms for mixed-integer nonlinear optimization and presents two general methods for nonlinear optimization, which build the basis for MINLP algorithms. A recent overview of methods for (convex) nonlinear programs is given in [80] and especially in [26]. An extensive introduction into nonlinear programming can be found in [94]. For the underlying methods from (mixed-integer) linear programming, we refer to standard textbooks, like [36], [94], and [118]. For this section, we assume that the NLP is convex and the MINLP has a convex relaxation, i.e., if integrality constraints are relaxed, the resulting NLP is convex and F, G, and H are convex functions (also called *convex MINLP* in the following).

3.4.1 Interior Point Methods

Interior point methods are *barrier methods* which have been studied for nonlinear optimization in the 1960s. The following decades, barrier methods fell out of research interest, until KARMARKAR presented his famous interior point algorithm for linear programs [74]. This development also triggered research on interior point methods for NLPs and today, interior point methods together with *sequential quadratic programming* (see Section 3.4.2) are considered the most powerful approaches for the solution of NLPs. The *interior* in the name refers to the method's property to approach the boundary of a linear problem's feasible set only in the limit with iterates in the interior of the feasible set—in contrast to the *simplex method*, which moves along the vertices of the feasible polytope.

For this section, we assume the NLP to have the form

$$\begin{aligned} \min_{x,s} \quad & F(x) \\ \text{s.t.} \quad & G(x) = 0 \\ & H(x) - s = 0 \\ & s \geq 0, \end{aligned} \tag{3.18}$$

where s are *slack variables* used to transform inequality constraints into equality constraints and all

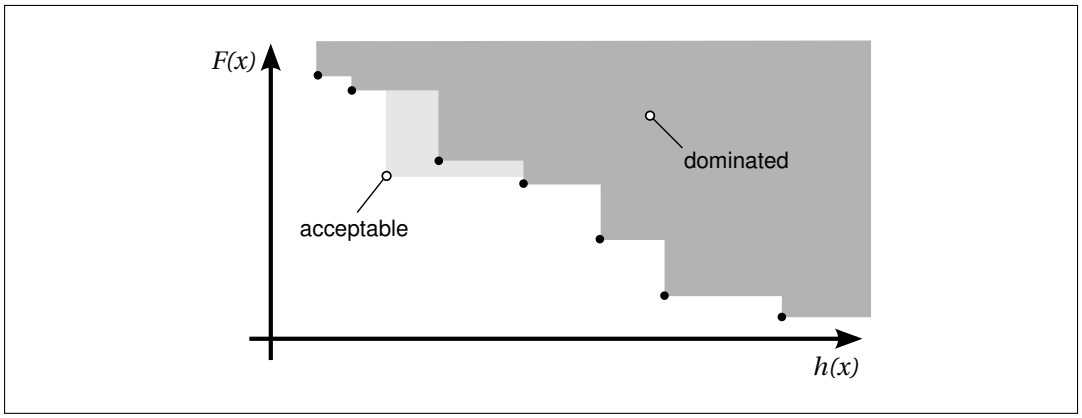


Figure 3.1: Scheme of a filter in nonlinear optimization algorithms: points right and above the filter points are dominated, acceptable points need at least either a lower feasibility gap or a higher objective. Points which are dominated by a point added to the filter will be removed.

other symbols as in Definition 3.6. The KKT conditions for this problem can be written as

$$\nabla F(x) - {J_G}^T(x)\lambda - {J_H}^T(x)\mu = 0 \tag{3.19a}$$
$$S\mu - \beta e = 0 \tag{3.19b}$$
$$G(x) = 0 \tag{3.19c}$$
$$H(x) - s = 0 \tag{3.19d}$$

with $\beta = 0$, $s \geq 0$, $\mu \geq 0$, and e the all-ones vector. J_G and J_H are the Jacobians of the constraint functions, λ and μ their LAGRANGE multipliers, and S a diagonal matrix whose diagonal entries are given by s. For $\beta \neq 0$, these conditions are called *perturbed KKT conditions* and β is the *barrier parameter*. Interior points method can be seen as *homotopy* or *continuation methods* solving these conditions for a sequence $\{\beta_k\}$ with $\beta_k \to 0$. With M a matrix with diagonal μ, we can apply NEWTON's method to (3.19) and obtain the system

$$\begin{pmatrix} \nabla^2_{xx}\mathcal{L} & 0 & -{J_G}^T(x_k) & -{J_H}^T(x_k) \\ 0 & M & 0 & S \\ J_G(x_k) & 0 & 0 & 0 \\ J_H(x_k) & -I & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta s \\ \Delta\lambda \\ \Delta\mu \end{pmatrix} = - \begin{pmatrix} \nabla F(x_k) - {J_G}^T(x_k)\lambda_k - {J_H}^T(x_k)\mu_k \\ S\mu_k - \beta e \\ G(x_k) \\ H(x_k) - s_k \end{pmatrix}, \tag{3.20}$$

with I the identity matrix and $\mathcal{L}$ the Lagrangian,

$$\mathcal{L}(x, s, \lambda, \mu) = F(x) - \lambda^T\, G(x) - \mu^T\, (H(x) - s). \tag{3.21}$$

For a step $(\Delta x, \Delta s, \Delta\lambda, \Delta\mu)$, new iterates can then be computed, e.g., by a line search or a trust region approach.

An exemplary interior point algorithm is presented in Algorithm 3.1. *Ipopt 3.10* is an open source interior point solver by WÄCHTER and BIEGLER [128] available for the use with *Bonmin 1.5*.

1. **Initialize:**
   Choose x_0 and $s_0 > 0$;
   Determine initial values for multipliers λ_0 and $\mu_0 > 0$;
   Select initial parameter β_0;
   Choose parameters $\sigma, \tau \in (0,1)$;
   $k := 0$;

2. **Check convergence:**
   If x_k, e.g., fulfills the KKT conditions for problem (3.18), stop;

3. **Compute search direction:**
   Determine search direction $(\Delta x, \Delta s, \Delta\lambda, \Delta\mu)$ by solution of

$$\begin{pmatrix} \nabla^2_{xx}\mathcal{L} & 0 & -{J_G}^T(x_k) & -{J_H}^T(x_k) \\ 0 & M & 0 & S \\ J_G(x_k) & 0 & 0 & 0 \\ J_H(x_k) & -I & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta s \\ \Delta\lambda \\ \Delta\mu \end{pmatrix} = - \begin{pmatrix} \nabla F(x_k) - {J_G}^T(x_k)\lambda_k - {J_H}^T(x_k)\mu_k \\ S\mu_k - \beta e \\ G(x_k) \\ H(x_k) - s_k \end{pmatrix}$$

4. **Compute step:**
   Compute step, e.g., by

$$\alpha_s^{\max} = \max_{\alpha\in(0,1]} (s + \alpha\, \Delta s \geq (1-\tau)s)$$
$$\alpha_\mu^{\max} = \max_{\alpha\in(0,1]} (z + \alpha\, \Delta z \geq (1-\tau)z)$$

5. **Update:**
   Set $(x_{k+1}, s_{k+1}, \lambda_{k+1}, \mu_{k+1})$ by

$$x_{k+1} = x_k + \alpha_s^{\max}\, \Delta x \qquad s_{k+1} = s_k + \alpha_s^{\max}\, \Delta s$$
$$\lambda_{k+1} = \lambda_k + \alpha_\mu^{\max}\, \Delta\lambda \qquad \mu_{k+1} = \mu_k + \alpha_\mu^{\max}\, \Delta\mu$$

6. **Determine barrier parameter:**
   Determine β_{k+1}, e.g., $\beta_{k+1} \in (0, \sigma\, \beta_k)$;
   $k := k+1$;
   go to 2;

Algorithm 3.1: An exemplary interior point algorithm.

3.4.2 Sequential Quadratic Programming

Sequential Quadratic Programming (SQP) is an algorithm for the solution of NLPs originally developed by WILSON [135], HAN [68], and POWELL [100]. The idea is to iteratively solve quadratic approximations of the NLP until some convergence criterion is fulfilled. The quadratic approximation for some iteration k is given by the problem

$$\begin{aligned} \min_{\Delta x} \quad & \nabla F(x_k)^T \, \Delta x + \tfrac{1}{2} \Delta x^T \, \hat{H}_k \, \Delta x \\ \text{s.t.} \quad & G(x_k) + \nabla G(x_k)^T \, \Delta x = 0 \\ & H(x_k) + \nabla H(x_k)^T \, \Delta x \leq 0 \end{aligned} \tag{3.22}$$

where $\hat{H}_k$ is the Hessian or an approximation of the Hessian of the Lagrangian of the NLP,

$$\hat{H}_k \approx \nabla^2_{xx} \mathcal{L}(x, \lambda, \mu). \tag{3.23}$$

In Problem (3.22), $\hat{H}_k$ is chosen such that the Lagrangian of the NLP and the Lagrangian of the QP are identical up to second order if $\hat{H}_k$ is the exact Hessian, which guarantees every minimizer of the NLP to also be a minimizer of the QP. With a search direction Δx computed from (3.22), the iteration's step α usually is computed by a *line search* or a *trust region* method or variations of these, i.e.,

$$x_{k+1} = x_k + \alpha \, \Delta x \tag{3.24}$$

Algorithm 3.2 describes an exemplary SQP algorithm.

It can be shown that SQP is equivalent to NEWTON's method on the KKT conditions of the NLP. Therefore, convergence properties are the same as for the corresponding NEWTON's method. An implementation for the use with *Bonmin 1.5* is available in *FilterSQP* by FLETCHER and LEYFFER [49].

3.4.3 Filter Methods

Both in the interior point and the sequential quadratic programming approach, modern solvers often—and especially *Ipopt 3.10* and *FilterSQP* do—implement *filter methods*, to decide on the acceptance of a step. This concept considers nonlinear programming as a multi-objective problem with the minimization of the problem's objective function F on the one hand and the minimization of constraint violation on the other hand, i.e.,

$$\min_x \; h(x) \qquad \text{with} \quad h(x) = \sum_{i=1}^{n_e} |G_i(x)| - \sum_{i=1}^{n_i} \min(0, H_i(x)). \tag{3.25}$$

In this context, a pair $(f^{(k)}, h^{(k)})$ is said to be *dominated* by some other pair $(f^{(j)}, h^{(j)})$ if $f^{(k)} \leq f^{(j)}$ and $h^{(k)} \leq h^{(j)}$. A *filter* consists of a set of pairs $(f^{(k)}, h^{(k)})$ which do not dominate each other, Figure 3.1 gives an illustration. A new iterate x_k is *acceptable* to the filter if it is not dominated by the points in the filter. Usually, if a step is accepted, the corresponding pair $(f^{(k)}, h^{(k)})$ will be added to the filter and dominated pairs will be removed. Filter methods can be combined with both line search and trust region methods and are enhanced in practice by several modifications, e.g., to ensure global convergence.

1. **Initialize:**
   Determine initial values for x_0, λ_0, μ_0;
   $k := 0$;
2. **Check convergence:**
   If terminal condition is fulfilled by x_k, stop;
3. **Evaluate derivatives:**
   Evaluate $F(x_k)$, $G(x_k)$, $H(x_k)$ and derivatives $\nabla F(x_k)$, $\nabla G(x_k)$, $\nabla H(x_k)$;
   Compute the Hessian or an approximation of the Hessian of the Lagrangian,
   $$\hat{H}_k \approx \nabla^2_{xx} \mathcal{L}(x, \lambda, \mu)$$
4. **Solve QP:**
   Determine δx, $\tilde{\lambda}$, and $\tilde{\mu}$ as solution and corresponding multipliers from the QP
   $$\begin{aligned} \min_{\Delta x} \quad & \nabla F(x_k)^T \, \Delta x + \tfrac{1}{2} \Delta x^T \, \hat{H}_k \, \Delta x \\ \text{s.t.} \quad & G(x_k) + \nabla G(x_k)^T \, \Delta x = 0 \\ & H(x_k) + \nabla H(x_k)^T \, \Delta x \leq 0 \end{aligned}$$
5. **Compute step:**
   Compute step size α, e.g., by line search
6. **Update:**
   Set $(x_{k+1}, \lambda_{k+1}, \mu_{k+1})$ by
   $$\begin{aligned} x_{k+1} &= x_k + \alpha \, \Delta x \\ \lambda_{k+1} &= \lambda_k + \alpha \, \left(\tilde{\lambda} - \lambda_k \right) \\ \mu_{k+1} &= \mu_k + \alpha \, \left(\tilde{\mu} - \mu_k \right) \end{aligned}$$
7. **Repeat:**
   $k := k + 1$;
   go to 2;

Algorithm 3.2: An exemplary sequential quadratic programming algorithm.

3.4.4 Branch and Bound

Branch and bound is an algorithmic framework first proposed by LAND and DOIG [79] for the solution of integer and combinatorial problems. It was originally developed for *mixed-integer linear program* (MILP)s and later was extended for the solution of MINLPs by DAKIN [38]. A branch and bound algorithm performs a systematic enumeration of all possible solutions by doing a tree search.

The single root node P_0 of the tree often is a relaxation of the problem to solve. In the case of MINLPs, the relaxation usually consists in dropping the integrality constraints. The two essential elements of the approach are *branching* and *bounding*, which is where the method's name comes from.

Branching divides the problem into smaller subproblems P_i by adding additional inequalities. The minimum of the solutions of the children of a node, which have been derived by branching, is the same as the minimum of the parent node. Based on this principle, a tree with smaller and smaller subproblems is generated. In the case of binary variables, for instance, branching on such a variable would fix it to 0 or 1.

Bounding is used to eliminate nodes and subtrees from the tree to avoid a complete enumeration. During the processing of the tree, the algorithm computes an upper bound UB for the problem and lower bounds for subproblems $LB(P_i)$ (in the case of a minimization problem). Every solution which is feasible for the original problem, obviously yields an upper bound for the optimal solution. Vice versa, a solution for a relaxation of a problem yields a lower bound for the problem's mixed-integer solution. As by adding constraints in the branching process, the feasible set gets smaller, a node's lower bound is a valid lower bound for all its child nodes, too. Apparently, if a node's lower bound is higher than the problem's upper bound, it cannot contain the optimal solution and so do its (potential) child nodes. Thus, if the solution of the current yields a better upper bound UB is found, all nodes P_i with $LB(P_i) > UB$ can be fathomed from the tree. Furthermore, if the current node is infeasible or its lower bound is higher than the current upper bound, it can be removed from the tree without branching on it.

The algorithm stops if there are no more nodes to be considered for branching on the tree with the current mixed-integer solution as the optimal solution for the problem. An illustration of a problem tree in a branch and bound algorithm is shown in Figure 3.2. Algorithm 3.3 describes a generic branch and bound algorithm.

Of course, this general description stays vague on several critical aspects of the algorithm. First, the selection of the next node to be processed is crucial. The algorithm may follow a *breadth-first*, i.e., nodes on the highest level are processed first, or a *depth-first* approach, i.e., newly created subproblems are processed first, for instance. Other aspects are the choice of the variable to branch on and the number of branches to create. Additionally, For MILPs, details on these algorithmic decisions are given in [3]. [16] states that "most observations generalize from the MILP to the MINLP case". In [27], branch and bound strategies for MINLPs are investigated.

3.4.5 Outer Approximation

The *outer approximation* algorithm for MINLPs was introduced by DURAN and GROSSMANN [46]. It is based on a linear approximation of the objective function and the constraints and iteratively solves MILPs and NLPs yielding lower and upper bounds for the MINLP.

For the description of the algorithm, we assume that the MINLP has the following form, i.e., equal-

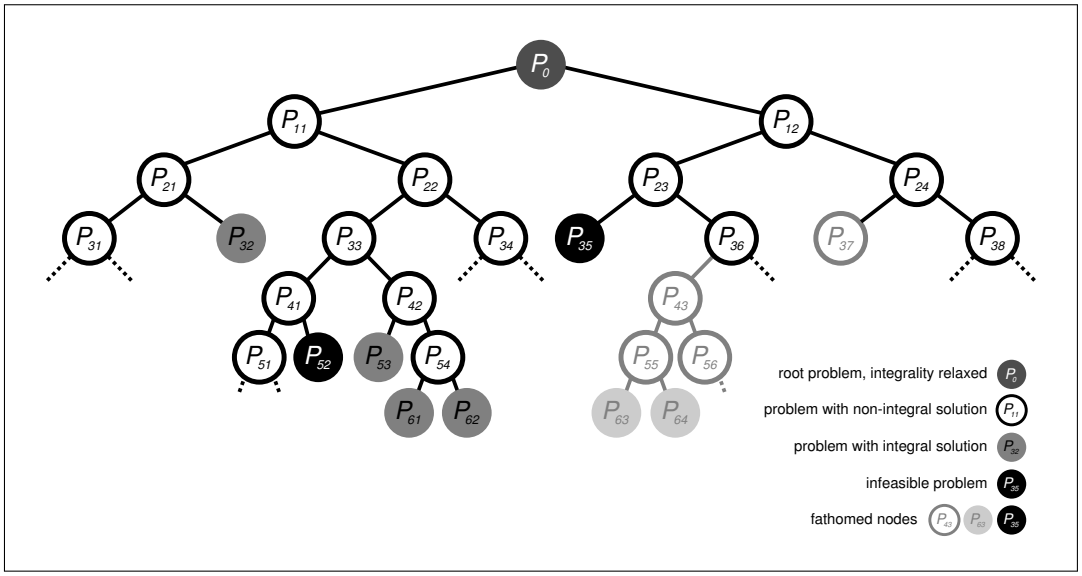


Figure 3.2: Scheme of a branch and bound problem tree: in the root node at the top, integrality is relaxed, but the solution typically is non-integral. By recursive branching, smaller child problems are generated and one will eventually find integral solutions. Nodes can be fathomed if they are infeasible or their objective value is below the lower bound.

1. **Initialize:**
   Relax integrality in MINLP to get root problem P_0;
   $UB := \infty$; $LB(P_0) := -\infty$;
   $L := \{P_0\}$;
2. **Select:**
   If $L \neq \emptyset$, select $P \in L$, $L := L \setminus P$;
   Else stop, S is a solution;
3. **Solve:**
   Solve P;
   If P infeasible, go to 2;
   Else let F_P be the objective value;
4. **Prune/Bound:**
   If $F_P > UB$, go to 2;
   Else if solution is integral, $UB := F_P$; $S := \text{sol}(P)$;
   Remove nodes with $LB(\cdot) > UB$, $L := \{p | p \in L, LB(P) < UB\}$;
5. **Branch:**
   Branch on P, i.e., create new nodes $P_1, \ldots, P_k$;
   Determine lower bounds for P_i, e.g., $LB(P_1) = \cdots = LB(P_k) = F_P$;
   If $LB(P_i) \leq UB$, add P_i to L, $L := L \cup \{P_i\}$;

Algorithm 3.3: An exemplary branch and bound algorithm for MINLPs.

ity constraints are transformed into corresponding inequality constraints,

$$\begin{aligned} \min_{x,y} \quad & F(x,y) \\ \text{s.t.} \quad & H(x,y) \leq 0 \\ & x \in X \subset \mathbb{R}^{n_x} \\ & y \in X \subset \mathbb{R}^{n_y} \cap \mathbb{Z}^{n_y}. \end{aligned} \tag{3.26}$$

Given such an MINLP, FLETCHER and LEYFFER [48] showed for convex and twice continuously differentiable F and H and bounded sets X and Y that solutions of the following MILP—a linearized version of the MINLP—and solutions for the MINLP are identical,

$$\begin{aligned} \min_{x,y} \quad & \eta \\ \text{s.t.} \quad & \eta \geq F(x^{(j)},y^{(j)}) + \nabla F(x^{(j)},y^{(j)})^T \begin{pmatrix} x-x^{(j)} \\ y-y^{(j)} \end{pmatrix} \qquad \forall (x^{(j)},y^{(j)}) \in \mathcal{K} \\ & 0 \geq H(x^{(j)},y^{(j)}) + \nabla H(x^{(j)},y^{(j)})^T \begin{pmatrix} x-x^{(j)} \\ y-y^{(j)} \end{pmatrix} \qquad \forall (x^{(j)},y^{(j)}) \in \mathcal{K} \\ & x \in X \subset \mathbb{R}^{n_x} \\ & y \in Y \subset \mathbb{R}^{n_y} \cap \mathbb{Z}^{n_y} \end{aligned} \tag{3.27}$$

if $\mathcal{K}$ contains all feasible solutions $(x^{(j)},y^{(j)})$ for all possible integer assignments of the problem

$$\begin{aligned} \min_{x} \quad & F(x,y^{(k)}) \\ \text{s.t.} \quad & H(x,y^{(k)}) \leq 0 \\ & x \in X \subset \mathbb{R}^{n_x} \end{aligned} \tag{3.28}$$

or the solutions of a given feasibility problem, in case Problem (3.28) is infeasible for some integer assignment.

In the algorithm, Problem (3.27) is called *master problem* and $\mathcal{K}$ contains a much smaller number of linearization points, usually starting only with the solution of the relaxed MINLP. With a smaller $\mathcal{K}$, however, Problem (3.27) still yields lower bounds for the MINLP (3.26) and an integer assignment $y^{(k)}$. As more and more points get added, the approximation and the lower bounds get better. On the other hand, optimal solutions of Problem (3.28) with integer variables fixed to an integer assignment $y^{(k)}$ give upper bounds for the MINLP and new linearization points for the master problem. Such a new linearization point cuts off the current solution of the master problem, unless it is optimal for the MINLP.

Thus, the algorithm iterates between the master problem and the NLP with fixed integer variables while the solutions of the master problem give a nondecreasing sequence of lower bounds. If the difference between lower and upper bound is within a certain tolerance, the algorithm stops with the best mixed-integer solution found. A generic description of an outer approximation algorithm is given in Algorithm 3.4.

3.4.6 LP/NLP-based Branch and Bound

The LP/NLP-based branch and bound algorithm was introduced by QUESADA and GROSSMAN [103] and is an extension of the outer approximation from the previous section. The idea is to avoid solving a large number of MILPs and instead evaluate a single MILP with a branch and bound method being dynamically updated with new linearization points.

1. **Initialize:**
   $UB := \infty$; $LB := -\infty$; $i := 1$;
   Choose convergence tolerance ϵ;
   Determine $x^{(0)}$, e.g., by solving the NLP relaxation ;
   $\mathcal{K} := \{x^{(0)}\}$; $S := \emptyset$;

2. **Check convergence:**
   If $UB - LB < \epsilon$ or the master problem,

$$
\begin{aligned}
\min_{x,y,\eta} \quad & \eta \\
\text{s.t.} \quad & \eta \geq F(x^{(j)}, y^{(j)}) + \nabla F(x^{(j)}, y^{(j)})^T \begin{pmatrix} x - x^{(j)} \\ y - y^{(j)} \end{pmatrix} && \forall (x^{(j)}, y^{(j)}) \in \mathcal{K} \\
& 0 \geq H(x^{(j)}, y^{(j)}) + \nabla H(x^{(j)}, y^{(j)})^T \begin{pmatrix} x - x^{(j)} \\ y - y^{(j)} \end{pmatrix} && \forall (x^{(j)}, y^{(j)}) \in \mathcal{K} \\
& x \in X \subset \mathbb{R}^{n_x} \\
& y \in Y \subset \mathbb{R}^{n_y} \cap \mathbb{Z}^{n_y},
\end{aligned}
$$

   is infeasible, stop. S is a solution;

3. **Lower Bound:**
   Let η_{MP} be the optimal value and $(x_{MP}, y_{MP}, \eta_{MP})$ the corresponding solution of the master problem;
   $LB := \eta_{MP}$;

4. **Solve NLP:**
   Fix integer variables $y := y_{MP}$ in the original problem and solve the resulting NLP;
   Let $(x^{(i)}, y^{(i)})$ be the solution;

5. **Upper Bound:**
   If $(x^{(i)}, y^{(i)})$ feasible for the original problem and $F(x^{(i)}, y^{(i)}) < UB$, set $S := (x^{(i)}, y^{(i)})$ and $UB := F(x^{(i)}, y^{(i)})$;

6. **Refine:**
   $\mathcal{K} := \mathcal{K} \cup \{(x^{(i)}, y^{(i)})\}$;
   $i := i + 1$;
   go to 1;

Algorithm 3.4: An exemplary outer approximation algorithm for MINLPs.

1. **Initialize:**
   $UB := \infty$;
   Let P_0 be the problem's NLP relaxation;
   Let $x^{(0)}$ be the solution of P_0;
   $\mathcal{K} := \{x^{(0)}\}$; $L := \{P_0\}$; $S := \emptyset$;
2. **Select:**
   If $L \neq \emptyset$, select $P \in L$, $L := L \backslash P$;
   Else stop, S is a solution;
3. **Evaluate:**
   Solve the linear master program

$$\begin{aligned} \min_{x,y,\eta} \quad & \eta \\ \text{s.t.} \quad & \eta \geq F(x^{(j)}, y^{(j)}) + \nabla F(x^{(j)}, y^{(j)})^T \begin{pmatrix} x - x^{(j)} \\ y - y^{(j)} \end{pmatrix} && \forall (x^{(j)}, y^{(j)}) \in \mathcal{K} \\ & 0 \geq H(x^{(j)}, y^{(j)}) + \nabla H(x^{(j)}, y^{(j)})^T \begin{pmatrix} x - x^{(j)} \\ y - y^{(j)} \end{pmatrix} && \forall (x^{(j)}, y^{(j)}) \in \mathcal{K} \\ & x \in X \subset \mathbb{R}^{n_x} \\ & y \in Y \subset \mathbb{R}^{n_y}. \end{aligned}$$

   If the linear master program is infeasible, go to 2;
   Else let η_{LMP} be the optimal value and $(x_{LMP}, y_{LMP}, \eta_{LMP})$ the corresponding solution of the master problem;
4. **Prune:**
   If $\eta_{LMP} > UB$, go to 1;
5. **Solve NLP:**
   If y_{LMP} integer, fix variables $y := y_{MP}$ in the original problem and solve the resulting NLP;
   Let $(x^{(P)}, y^{(P)})$ be the solution;
   Else go to 8;
6. **Upper Bound:**
   If $(x^{(P)}, y^{(P)})$ feasible for the original problem and $F(x^{(P)}, y^{(P)}) < UB$,
   set $S := (x^{(P)}, y^{(P)})$ and $UB := F(x^{(P)}, y^{(P)})$;
7. **Refine:**
   $\mathcal{K} := \mathcal{K} \cup \{(x^{(P)}, y^{(P)})\}$;
   go to 3;
8. **Branch:**
   Branch on P, i.e., create new nodes $P_1, \ldots, P_k$;
   Determine lower bounds for P_i, e.g., $LB(P_1) = \cdots = LB(P_k) = \eta_{LMP}$;
   go to 2;

Algorithm 3.5: An exemplary LP/NLP-based branch and bound for MINLPs.

Therefore, relaxations of the master program (3.27) are solved in a branch and bound scheme for a solution of the master program. Whenever an integer solution (x_{LMP}, y_{LMP}) is found (i.e., integral y_{LMP}), the algorithm is stopped and Problem (3.28) with y fixed to y_{LMP} is solved, which yields a new linearization point $(x^{(P)}, y^{(P)})$ (and an upper bound for the MINLP, of course). The master problem is updated with the new linearization point and the branch and bound continues. Thus, a major advantage over the outer approximation algorithm is that the LP/NLP-based branch and bound avoids a restart of the tree search at this point. A generic description is given in Algorithm 3.5.

3.4.7 Further MINLP Algorithms

The algorithms from the sections above are all implemented in the open source MINLP solver *Bonmin 1.5* [25], which has been used within this thesis for the (local) solution of MINLPs. Further algorithms for MINLPs include the *extended cutting planes* (ECP) algorithm [132] and *generalized* BENDERS *decomposition* (GBD) [17, 59]. ECP is an outer approximation variant which does not solve any NLP, but linearizes objective and constraints at the solution of the master problem and alternates between the solution of the master problem and the generation of new linearizations. GBD is an outer approximation variant, too, and uses a different master problem. In GBD's master problem, linearizations of objective and constraints are summed up into a single constrained using duality theory. However, the resulting BENDERS cuts are weaker than outer approximation cuts.

3.5 Nonconvex Problems and Global Optimization

In the previous section, we assumed convex NLPs and MINLPs and the algorithms presented so far find local solutions, i.e., if the problem is nonconvex, there is no guarantee that these algorithms find the global optimum. Indeed, a dMIOCP is nonconvex if the dependencies between its variables are nonlinear and thus, the state progression function G in Problem (3.17) is nonlinear. Because of the equality constraint,

$$x_{k+1} = G(x_k, u_k, p), \qquad k = t_0, \dots, t_f - 1, \tag{3.29}$$

nonconvexities can only be avoided with linear dMIOCPs. A restriction to linear dMIOCPs, however, is a severe limit for the possible models and tasks for complex problem solving. Many aspects of real world problems have to be modeled with nonlinear functions, a demand function, for instance, which often is assumed to be exponentially decreasing in price (e.g., [120]).

The field of *global optimization* is concerned with algorithms for nonconvex problems, which often but not necessarily include nonconvex MINLPs. Extensive introductions into global optimization can, e.g., be found in [63, 123, 83]. *Couenne* [15] is a global solver for nonconvex MINLPs based on *Bonmin 1.5* and *Ipopt 3.10*, which has been used within this thesis.

There are a lot of heuristics considered for global optimization, like *genetic algorithms*, *simulated annealing*, *particle swarm* algorithms and the like. Although such methods may find a good solution, there typically is no guarantee to find an optimum at all. Modern deterministic solvers for global optimization of MINLPs including *Couenne* mostly perform a *spatial branch and bound* (sBB) method, see, e.g., [16]. sBB is a branch and bound method, comparable to the one in Section 3.4.4, which partitions the search domain for smaller subproblems, not only for integer variables but also for continuous variables. The central aspect of sBB is then to find lower bounds for sub-regions of the feasible set by convex relaxations of the problem. Often, *bounds tightening* techniques are used to reduce the search domain. Obviously, the quality of the convex relaxation is crucial to the performance of the algorithm. In general, it is required that the problem is *factorable* for many reformulations, such as the famous MCCORMICK relaxation for products of variables [86].

Although there has been much progress in recent years, optimization algorithms, which guarantee global optimality for a nonconvex MINLP, are rather slow and are only capable of small-scale systems.

CHAPTER 4

The *IWR Tailorshop*—a New Complex Microworld

This chapter presents the new microworld developed within this thesis, the *IWR Tailorshop*. We will describe the assumptions for variables and equations in this model and give the parameter set, e.g., used in Chapter 6. We start with a short analysis of the original *Tailorshop* test-scenario to see why a new microworld was necessary.

4.1 Modeling the *Tailorshop* Microworld

In Section 2.6, we introduced the *Tailorshop* microworld, which has been developed since the 1980s by DÖRNER and others. Originally created for a *TI59* programmable calculator [54], this microworld later was implemented as a *GW-BASIC* application, which was the basis for the first thorough mathematical analysis [112]. This implementation is still in use in multiple variations, although there now are modern reimplementations. A snippet from the *GW-BASIC* code is shown in Figure 4.1. From a modern programming viewpoint, this typical early *BASIC* code style with line numbers and many `GOTO`s makes it hard to debug and maintain the code. Indeed, this implementation included several bugs which—at least partly—only became apparent through the mathematical analysis.

As described in Section 2.6, the *Tailorshop* is an economic microworld, in which a participant has to take economic decisions as the head of a company which produces and sells shirts. For most of the existing implementations, the scenario comprises $t_f = 12$ rounds, which are called months. The scenario consists of 15 free *control* variables u_k, which can be chosen by the participant within certain constraints, and 16 dependent *state* variables x_k, i.e. 31 time-discrete variables in total.

In the *Tailorshop* microworld, there are two different kinds of machines to produce either 50 or 100 shirts per month. Workers can only work on either one of them. Participants can choose to buy and sell machines as well as to hire and dismiss employees. The machines need to be maintained to preserve their capacity and equipped with raw material to actually produce something. The possible production depends furthermore on the satisfaction of the workers, linked to the controls wages and social expenses. The number of vans (in reimplementations sometimes renamed to *stores*) influences the demand in a positive way, which itself is a limit to the shirts sales. Shirts which cannot be sold remain in the stock. Furthermore, advertisement, location of the sales shop, and shirt price decisions can be used to maximize profit. All the decisions influence the base capital, and thus the capital after interest. Table 4.1 repeats the overview of all the states and controls (note that units of control and state variables are only given implicitly depending on how they enter the model equations and constraints) in the *Tailorshop* microworld and Figure 2.8 illustrates again the dependencies between the model variables.

4.1.1 Model Equations

In [112], a mathematical model of the *Tailorshop* microworld has been extracted from the *GW-BASIC* implementation. In general, the model has the form of a *discretized mixed-integer optimal control problem* (dMIOCP) (3.17), but with several min / max-expressions in the state progression function G. *Slack variables* s_k can be used to reformulate these expressions by standard techniques using

```
2650 ZA=.5+((LO-850)/550)+SM/800:IF ZA>ZM THEN:ZA=ZM
2660 SK=SM*(N1+N2):KA=KA-SK
2670 X=A1:IF N1<X THEN:X=N1
2680 Y=A2:IF N2<Y THEN:Y=N2
2690 PM=X*(MA+RND*4-2)+Y*(MA*2+RND*6-3):PM=PM*(ABS(ZA)^.5)
2700 X=PM:IF RL<X THEN:X=RL
2710 PA=X:HL=HL+PA:RL=RL-PA:KA=KA-(PA*1)-(RL*.5)
2720 NA=(NA/2+280)*1.25*2.7181^(-(PH^2)/4250):KA=KA-HL
2730 X=NA:IF HL<X THEN:X=HL
2740 VH=X:HL=HL-VH:KA=KA+VH*PH
2750 KA=KA-WE
2760 X1=WE/5:IF X1>NM THEN:X1=NM
2770 KA=KA-LW*500:X1=X1+LW*100
2780 KA=KA-GL*2000
2790 X=0:IF GL=.5 THEN:X=.1:ELSE IF GL=1 THEN:X=.2
2800 X1=X1+X1*X
2810 NA=X1+(RND*100-50)
2820 RP=2+(RND*6.5)
2830 MA=MA-.1*MA+(RS/(A1+A2*1E-08))*.017
2840 IF MA>MM THEN:MA=MM
2850 KA=KA-RS
2860 KA=KA-(N1+N2)*LO
2870 IF KA>0 THEN:KA=KA+KA*GZ:ELSE KA=KA+KA*SZ
```

Figure 4.1: Excerpt of the *GW-BASIC* implementation of the original *Tailorshop*. Special care is necessary to separate already updated variables x_{k+1} from the values x_k, compare the role of $x_k^{MS} \approx$ `RL` and $x_k^{PP} \approx$ `PM` in lines 2690 to 2710.

several constraints, which are given below. We define

$$(x^{\mathrm{P}}, u^{\mathrm{P}}) = (x_0^{\mathrm{P}}, \ldots, x_N^{\mathrm{P}}, u_0^{\mathrm{P}}, \ldots, u_{N-1}^{\mathrm{P}}) \tag{4.1}$$

to be the vector of decisions and state variables for all months of a participant. Common initial values x_0 are given together with fixed parameters p in Table 4.2. Pseudo-random values ξ appear in the equations, e.g., line 2810 in Figure 4.1. However, the analysis of the compiled code revealed that the random values are only dependent on a fixed initialization (seed) within the *GW-BASIC* code, hence they are identical for all participants and can be considered as fixed parameter vectors.

The number of machines for 50 and 100 shirts per month depends on buying and selling of machines. Note that there is a difference between buying and selling in the base capital equation so that two independent controls are needed here:

$$x_{k+1}^{M_{50}} = x_k^{M_{50}} + u_k^{\Delta M_{50}} - u_k^{\delta M_{50}}, \tag{4.2}$$

$$x_{k+1}^{M_{100}} = x_k^{M_{100}} + u_k^{\Delta M_{100}} - u_k^{\delta M_{100}}. \tag{4.3}$$

For the workers, a single control which stands for hiring and firing workers is sufficient since there is no such difference (one might even avoid the state variable if the control was the current number

States	**Variable**	**Unit***	**Controls**	**Variable**	**Unit***
machines 50	$x^{M_{50}}$	machine(s)	advertisement	u^{AD}	MU
machines 100	$x^{M_{100}}$	machine(s)	shirt price	u^{SP}	MU
workers 50	$x^{W_{50}}$	worker(s)	buy raw material	$u^{\Delta MS}$	shirt(s)
workers 100	$x^{W_{100}}$	worker(s)	workers 50	$u^{\Delta W_{50}}$	worker(s)
demand	x^{DE}	shirt(s)	workers 100	$u^{\Delta W_{100}}$	worker(s)
vans	x^{VA}	van(s)	buy machines 50	$u^{\Delta M_{50}}$	machine(s)
shirts sales	x^{SS}	shirt(s)	buy machines 100	$u^{\Delta M_{100}}$	machine(s)
shirts stock	x^{ST}	shirt(s)	sell machines 50	$u^{\delta M_{50}}$	machine(s)
possible production	x^{PP}	shirt(s)	sell machines 100	$u^{\delta M_{100}}$	machine(s)
actual production	x^{AP}	shirt(s)	maintenance	u^{MA}	MU
material stock	x^{MS}	shirt(s)	wages	u^{WA}	MU
satisfaction (motiv.)	x^{MO}	—	social expenses	u^{SC}	MU
machine capacity	x^{MC}	shirt(s)	buy vans	$u^{\Delta VA}$	van(s)
base capital	x^{BC}	MU	sell vans	$u^{\delta VA}$	van(s)
capital after interest	x^{CA}	MU	choose site	u^{CS}	—
overall balance	x^{OB}	MU			

Table 4.1: Controls and states in the *Tailorshop* microworld. Note that units are only given implicitly in the test scenario. * MU means money units.

of workers, but we stick to the hiring control for better comparability with the microworld):

$$x_{k+1}^{W_{50}} = x_k^{W_{50}} + u_k^{\Delta W_{50}}, \tag{4.4}$$

$$x_{k+1}^{W_{100}} = x_k^{W_{100}} + u_k^{\Delta W_{100}}. \tag{4.5}$$

Demand depends on a time-dependent pseudorandom parameter p_k^{DE} as well as on the advertisement expenses and the number of vans multiplied by a factor depending on the site, $f^1\left(u_k^{CS}\right)$,

$$x_{k+1}^{DE} = 100 \cdot p_k^{DE} - 50 + \left(\frac{u_k^{AD}}{5} + 100 \cdot x_{k+1}^{VA} \right) \cdot f^1\left(u_k^{CS}\right), \quad \text{with} \quad f^1\left(u_k^{CS}\right) = 1 + \frac{u_k^{CS}}{10}. \tag{4.6}$$

While the influence of advertisement is bounded, see (4.20), the effect of vans is unbounded. This will be discussed in Section 4.1.3.

For the vans again, two controls for buying and selling are needed due to differences in the base capital. Shirt sales are determined by the slack variable s_k^{SS} and shirts in stock depend on the slack

State	**Unit***	**Variable**	**Value**
machines 50	machines	$x_0^{M_{50}}$	10
machines 100	machines	$x_0^{M_{100}}$	0
workers 50	workers	$x_0^{W_{50}}$	8
workers 100	workers	$x_0^{W_{100}}$	0
demand	shirts	x_0^{DE}	766.636
vans	vans	x_0^{VA}	1
shirts sales	shirts	x_0^{SS}	407.2157
shirts stock	shirts	x_0^{ST}	80.7164
possible production	shirts	x_0^{PP}	403.93
actual production	shirts	x_0^{AP}	403.93
material stock	shirts	x_0^{MS}	16.06787
satisfaction (motiv.)	—	x_0^{MO}	0.9807
machine capacity	shirts	x_0^{MC}	47.04
capital after interest	MU	x_0^{CA}	165774.66
overall balance	MU	x_0^{OB}	250690.66

Parameter	**Unit**	**Variable**	**Value**
max. demand	shirts	p^{MD}	900
max. machine capacity	shirts	p^{MM}	50
max. satisfaction	—	p^{MS}	1.7
interest rate	—	p^{IR}	0.0025

Table 4.2: Fixed initial values x_0 and parameters p for *Tailorshop*. Note that some initial values are not needed, as they do not enter the right-hand-side function $G(\cdot)$. Note also that units are only implicitly given in the test scenario. * MU means money units.

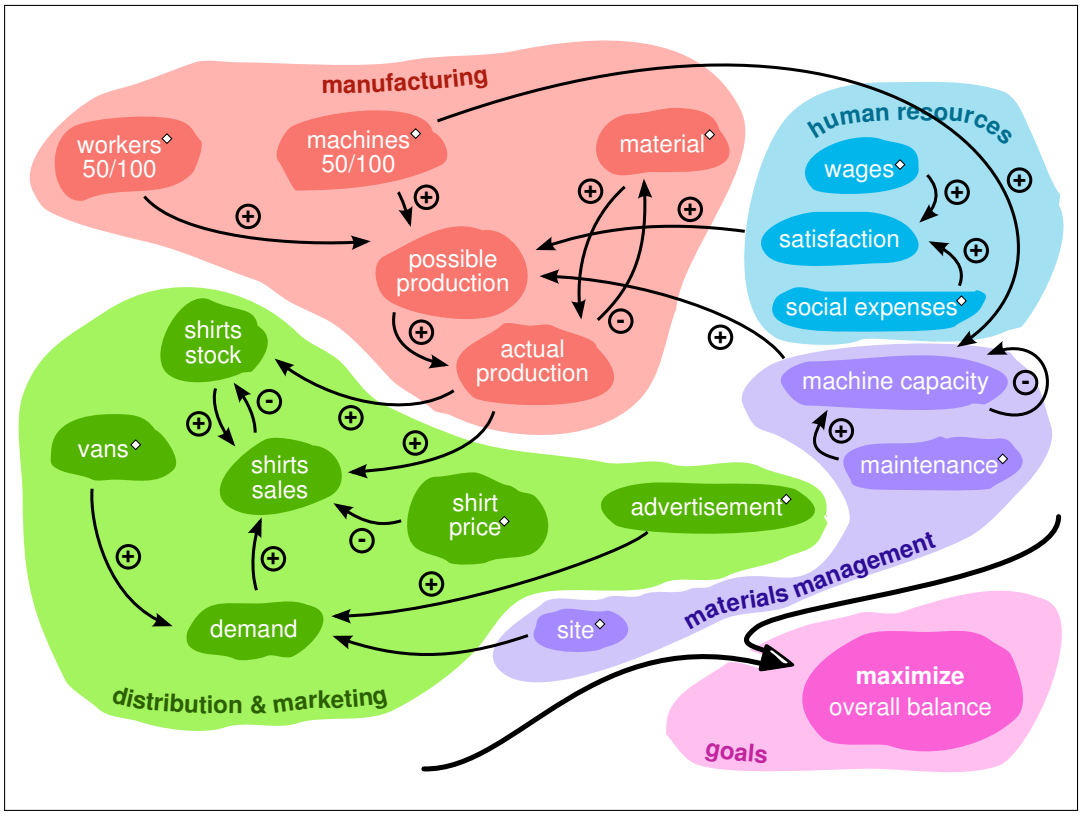


Figure 4.2: Original *Tailorshop* model. Arrows show proportional/reciprocal dependencies, diamond indicates participants' control influence.

variables for actual production s_k^{PP} and shirt sales s_k^{SS},

$$x_{k+1}^{VA} = x_k^{VA} + u_k^{\Delta VA} - u_k^{\delta VA}, \tag{4.7}$$

$$x_{k+1}^{SS} = s_k^{SS}, \tag{4.8}$$

$$x_{k+1}^{ST} = x_k^{ST} + s_k^{PP} - s_k^{SS}. \tag{4.9}$$

In the possible production equation, the part representing machine and worker dependence consists of a term for each machine type with slack variables $s_k^{M_{50}}$ and $s_k^{M_{100}}$, which are used to replace min expressions of workers and machines, multiplied by a machine capacity term (machines for 100 shirts have double machine capacity). This part is multiplied by the square root of workers' satisfaction. The actual production is determined by a slack variable:

$$x_{k+1}^{PP} = \left(s_k^{M_{50}} \cdot \left(x_k^{MC} + 4 \cdot p_k^{P_{50}} - 2\right) + s_k^{M_{100}} \cdot \left(2 \cdot x_k^{MC} + 6 \cdot p_k^{P_{100}} - 3\right)\right) \cdot \left(x_{k+1}^{MO}\right)^{\frac{1}{2}}, \tag{4.10}$$

$$x_{k+1}^{AP} = s_k^{PP}. \tag{4.11}$$

Raw material in stock depends on the use of material represented by the slack variable for actual production and the purchase of new material. Wages and social expenses influence satisfaction and

the machine capacity is determined by a slack variable:

$$x_{k+1}^{MS} = x_k^{MS} + u_k^{\Delta MS} - s_k^{PP}, \tag{4.12}$$

$$x_{k+1}^{MO} = \frac{1}{2} + \frac{u_k^{WA} - 850}{550} + \frac{u_k^{SC}}{800}, \tag{4.13}$$

$$x_{k+1}^{MC} = s_k^{MC}. \tag{4.14}$$

The equation for base capital,

$$\begin{aligned} x_{k+1}^{BC} = x_k^{CA} &+ s_k^{SS} \cdot u_k^{SP} - p_k^{RP} \cdot u_k^{\Delta MS} - 10000\, u_k^{\Delta M_{50}} - f^2\left(u_k^{CS}\right) \\ &+ 8000\, \frac{x_k^{MC}}{p^{MM}}\, u_k^{\delta M_{50}} - 20000\, u_k^{\Delta M_{100}} + 16000\, \frac{x_k^{MC}}{p^{MM}}\, u_k^{\delta M_{100}} \\ &- u_k^{AD} - u_k^{MA} - \left(x_{k+1}^{W_{50}} + x_{k+1}^{W_{100}}\right) \cdot \left(u_k^{WA} + u_k^{SC}\right) - 2\, s_k^{PP} - \frac{1}{2}\, x_{k+1}^{MS} \\ &- x_k^{ST} - 10000\, u_k^{\Delta VA} + (8000 - 100\, k) \cdot u_k^{\delta VA} - 500\, x_{k+1}^{VA}, \end{aligned} \tag{4.15}$$

contains all income and expenses during a round added to the capital after interest from the previous round. The income consists of the amount of shirts sold times the shirt price $s_k^{SS} \cdot u_k^{SP}$, the sale of machines $8000 \cdot \left(x_k^{MC} / p^{MM}\right) \cdot u_k^{\delta M_{50}}$ and $16000 \cdot \left(x_k^{MC} / p^{MM}\right) \cdot u_k^{\delta M_{100}}$ (depending on the current machine capacity), and the sale of vans $(8000 - 100k) \cdot u_k^{\delta VA}$.

Money is spent for the raw material bought times the price of a raw material unit $-p_k^{RP} \cdot u_k^{\Delta MS}$, the purchase of machines $-10000\, u_k^{\Delta M_{50}}$ and $-20000\, u_k^{\Delta M_{100}}$, the purchase of vans $-10000\, u_k^{\Delta VA}$, advertisement and maintenance $-u_k^{AD} - u_k^{MA}$, and the number of workers times wages plus social expenses $-(x_{k+1}^{W_{50}} + x_{k+1}^{W_{100}}) \cdot (u_k^{WA} + u_k^{SC})$. Additionally, each unit of material in stock at the end of a round costs half a *monetary unit* (MU) $-\frac{1}{2} \cdot x_{k+1}^{MS}$, the production of a shirt costs two MU $-2 s_k^{PP}$, each shirt in stock costs one MU, and each van costs 500 MU per round $-500 x_{k+1}^{VA}$. There is another amount of money to be paid, which depends on the site,

$$f^2\left(u_k^{CS}\right) = 500 + 250\, u_k^{CS} + 250\, u_k^{CS} \cdot u_k^{CS}. \tag{4.16}$$

From the base capital the capital after interest is computed by multiplying it with an interest rate factor $(1 + p^{IR})$. Overall balance, the objective function, besides capital after interest contains terms for material and shirts in stock, for machines, and for vans. However, machines are worth less in the overall balance than if they were sold:

$$x_{k+1}^{CA} = x_{k+1}^{BC} \cdot \left(1 + p^{IR}\right) \tag{4.17}$$

$$\begin{aligned} x_{k+1}^{OB} = &\frac{x_k^{MC}}{p^{MM}} \left(8000\, x_{k+1}^{M_{50}} + 16000\, x_{k+1}^{M_{100}}\right) + (8000 - 100\, k) \cdot x_{k+1}^{VA} \\ &+ 2\, x_{k+1}^{MS} + 20\, x_{k+1}^{ST} + x_k^{CA} \end{aligned} \tag{4.18}$$

This leads to end time effects, as discussed in [112].

4.1.2 Reformulations

The *GW-BASIC* code which was the basis for the mathematical model formulated above, contains several min-expressions. Two expressions which originally enter the motivation or the demand equa-

tion respectively,

$$\min\left(p^{MS}, \frac{1}{2} + \frac{u_k^{WA} - 850}{550} + \frac{u_k^{SC}}{800}\right) \qquad \text{and} \qquad \min\left(\frac{u_k^{AD}}{5}, p^{MD}\right) \tag{4.19}$$

could be directly replaced by introducing additional constraints,

$$\frac{1}{2} + \frac{u_k^{WA} - 850}{550} + \frac{u_k^{SC}}{800} \leq p^{MS}, \qquad \frac{u_k^{AD}}{5} \leq p^{MD}. \tag{4.20}$$

The remaining expressions,

$$s_k^{PP} \approx \min\left(x_{k+1}^{PP}, x_k^{MS} + u_k^{\Delta MS}\right), \qquad s_k^{M_{50}} \approx \min\left(x_{k+1}^{W_{50}}, x_{k+1}^{M_{50}}\right), \tag{4.21}$$

$$s_k^{MC} \approx \min\left(p^{MM}, 0.9 x_k^{MC} + 0.017 \frac{u_k^{MA}}{x_{k+1}^{M_{50}} + 10^{-8}\, x_{k+1}^{M_{100}} + 10^{-8}}\right), \qquad s_k^{M_{100}} \approx \min\left(x_{k+1}^{W_{100}}, x_{k+1}^{M_{100}}\right), \tag{4.22}$$

$$s_k^{SS} \approx \min\left(x_k^{ST} + x_{k+1}^{AP}, \frac{5}{4}\left(\frac{x_k^{DE}}{2} + 280\right) \cdot 2.7181^{-\frac{\left(u_k^{SP}\right)^2}{4250}}\right), \tag{4.23}$$

could be reformulated using slack variables with corresponding constraints,

$$s_k^{PP} \leq x_k^{MS} + u_k^{\Delta MS}, \qquad s_k^{PP} \leq x_{k+1}^{PP}, \tag{4.24}$$

$$s_k^{MC} \leq p^{MM}, \qquad s_k^{MC} \leq 0.9 x_k^{MC} + 0.017 \frac{u_k^{MA}}{x_{k+1}^{M_{50}} + 10^{-8}\, x_{k+1}^{M_{100}} + 10^{-8}}, \tag{4.25}$$

$$s_k^{SS} \leq x_k^{ST} + x_{k+1}^{AP}, \qquad s_k^{SS} \leq \frac{5}{4}\left(\frac{x_k^{DE}}{2} + 280\right) \cdot 2.7181^{-\frac{\left(u_k^{SP}\right)^2}{4250}}, \tag{4.26}$$

$$s_k^{M_{50}} \leq x_{k+1}^{W_{50}}, \qquad s_k^{M_{50}} \leq x_{k+1}^{M_{50}}, \tag{4.27}$$

$$s_k^{M_{100}} \leq x_{k+1}^{W_{100}}, \qquad s_k^{M_{100}} \leq x_{k+1}^{M_{100}}, \tag{4.28}$$

for all $k \in \{0, \ldots, 11\}$. s_k^{PP} is used for the minimum of possible production and material in stock. With s_k^{MC}, the minimum of maximum machine capacity p^{MM} and the machine capacity determined by loss of capacity over time and the recovery by maintenance is described. Finally, s_k^{SS} is used to reformulate the minimum of shirts available for sale $x_k^{ST} + x_{k+1}^{AP}$ and a nonlinear term depending on the demand and the shirt price. For equation (4.26), note that 2.7181 has been used in the *GW-BASIC* code instead of exp. These reformulations are valid, because the corresponding variable have only positive effects in the objective and thus will be at the limit in a solution.

As we have seen, including these reformulations, the functions $G(\cdot)$ and $H(\cdot)$ are smooth, nonlinear functions of the unknown variables x, u and s. The nonlinearities are often bilinear, but sometimes also include denominators and exponentials.

4.1.3 Model Shortcomings

In the *GW-BASIC* implementation, there are few bounds on the controls and only little checks on reasonable values. For instance, the participant is only allowed to *sell* as much machines $x^{M_{50}}$ and $x^{M_{100}}$ as the *Tailorshop* currently owns, but there is neither a limit nor a check for *buying* machines,

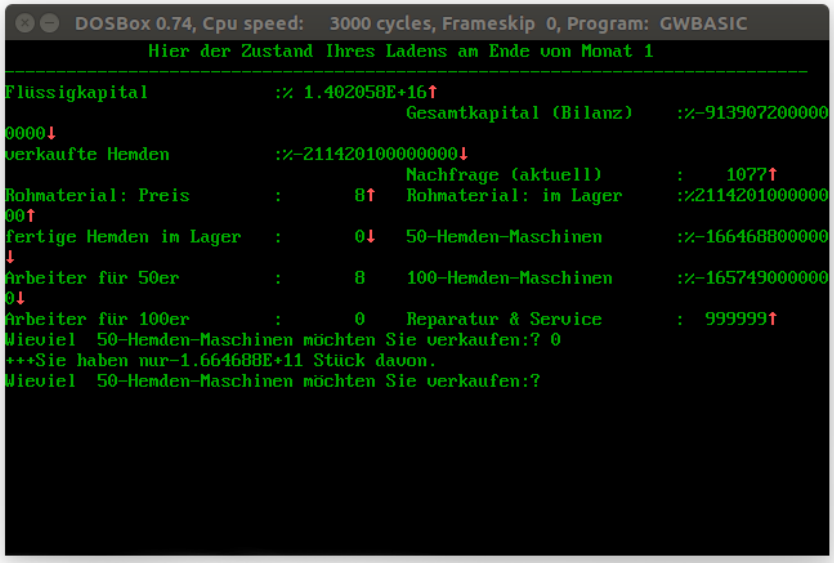


Figure 4.3: *Tailorshop* interface with unbounded decisions: as only few bounds are included, a participant, e.g., may "buy" an infinitely low negative number of machines. Compare also Figure 2.9.

such that a participant may "buy" a (infinitely low) negative number of machines, see Figure 4.3. The bounds included in the code are the following.

$$u_k^{AD} \in [0,\infty] \qquad u_k^{\Delta M_{50}} \in [-\infty,\infty] \qquad u_k^{WA} \in [850,\infty] \tag{4.29}$$

$$u_k^{SP} \in [10,100] \qquad u_k^{\Delta M_{100}} \in [-\infty,\infty] \qquad u_k^{SC} \in [0,\infty] \tag{4.30}$$

$$u_k^{\Delta MS} \in [0,\infty] \qquad u_k^{\delta M_{50}} \in [0, x_k^{M_{50}}] \qquad u_k^{\Delta VA} \in [0,\infty] \tag{4.31}$$

$$u_k^{\Delta W_{50}} \in [-x_k^{W_{50}},\infty] \qquad u_k^{\delta M_{100}} \in [0, x_k^{M_{100}}] \qquad u_k^{\delta VA} \in [0, x_k^{VA}] \tag{4.32}$$

$$u_k^{\Delta W_{100}} \in [-x_k^{W_{100}},\infty] \qquad u_k^{MA} \in [0,\infty] \tag{4.33}$$

However, in studies using the *Tailorshop*, participants did not do this and even restricted themselves to integer values in general, although only for u^{AD}, u^{SP}, and $u^{\Delta MS}$, inputs are converted into integer numbers. Obviously, with these bounds on the controls, the problem is unbounded, although one can assume reasonable bounds, e.g., $u_k^{\Delta M_{50}}, u_k^{\Delta M_{100}} \geq 0$ for existing data.

But even with these bounds, there are further shortcomings in the model. First, the effect of the variable *vans* x_k^{VA} on the demand, in contrast to *advertising*, is not limited. Probably, *demand* was meant to be

$$x_{k+1}^{DE} = 100 \cdot p_k^{DE} - 50 + \min\left(\frac{u_k^{AD}}{5} + 100 \cdot x_{k+1}^{VA}, p^{MD}\right) \cdot f^1\left(u_k^{CS}\right), \tag{4.34}$$

but in the code actually is determined as

$$x_{k+1}^{DE} = 100 \cdot p_k^{DE} - 50 + \left(\min\left(\frac{u_k^{AD}}{5}, p^{MD}\right) + 100 \cdot x_{k+1}^{VA}\right) \cdot f^1\left(u_k^{CS}\right). \tag{4.35}$$

Thus, the effect of advertising is limited to 900 by p^{MD}, but vans can increase demand without any limit. For $x_k^{VA} \gg 9$, vans become the dominating factor in this equation, and computations in [112] showed that this effect also dominates the development of the whole model, even if a lower bound on the capital is introduced to make the problem bounded (see Figure 4.4). Furthermore, it seems rather unrealistic that *demand* is influenced by the number of vans. This also applies if the variable is renamed to *stores* which could plausibly influence the *sales*, but might not have much effect on the *demand*.

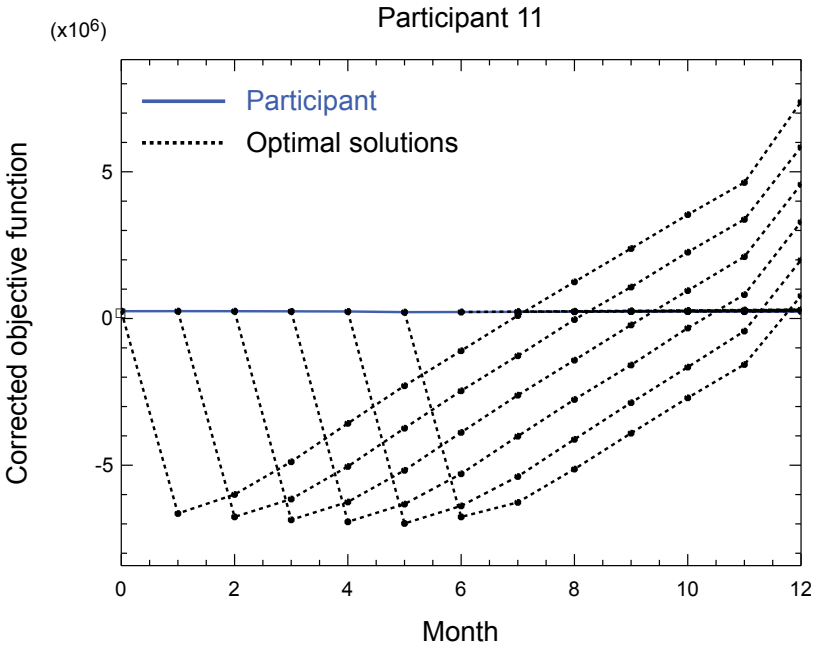


Figure 4.4: *Tailorshop* objective with vans bug: the effect of vans on the model is so strong that it dominates the model. Even with an artificial lower bound on the capital to make the problem bounded, optimal solutions starting at different months increase so fast that the participant's solution seems to be constantly 0 in this example.

Finally, in equation (4.25) for *machine capacity*, a typing error might have lead to another oddity in the *Tailorshop* model. The first 10^{-8} in the denominator was probably meant as a *safe guard* and should be another summand instead of a factor (the second 10^{-8} actually was not in the code, but added during the modeling as such a safe guard). By the multiplication of *100 shirt machines* with 10^{-8}, however,

$$\cdots + 0.017 \frac{u_k^{MA}}{x_{k+1}^{M_{50}} + 10^{-8}\, x_{k+1}^{M_{100}} + 10^{-8}}, \tag{4.36}$$

an infinitesimal u_k^{MA} is sufficient to achieve a high level of machine capacity if only *100 shirt machines* are used (which turns out to be optimal).

All these shortcomings together with the necessary reformulations illustrate the need for a new complex microworld with controlled properties—not only, but especially for the application of optimization methods as an analysis and training tool.

4.2 The *IWR Tailorshop* Microworld

For this work, we systematically built a new microworld based on the economical framing of *Tailorshop* with controlled properties. Besides the reformulation and shortcomings discussed above, the decision variables in the *Tailorshop* microworld are a mixture of operational (e.g., decide how much employees work on which machine class) and strategical decisions (e.g., shirt price, amount of advertising). First, depending on the size of the company of course, these decisions will rarely all be made by the same person, i.e., by the head of the company. Second, it seems not quite realistic, for instance, that employees who are trained to operate a 50 shirt machine are by no means able to operate a 100 shirt machine and will also not report to the person who (accidentally) trained them for the wrong machine type but instead sit around and do nothing all the month. This illustrates that there are some (implicit) assumptions in the *Tailorshop* which are not very plausible and a simple smoothening of min-expressions and correction of modeling bugs would still leave some questions about the model and its assumptions. Altogether, these problems lead to the development of a new test-scenario, the *IWR Tailorshop* microworld.

Compared to the *Tailorshop*, the variety of variables has been shifted towards a more abstract level. For example, the participants have no longer the task to buy or sell *machines*, but instead have to take care of the number of *production sites* x^{PS} of their company. The rather concrete variable *vans* has been replaced by more abstract *distribution sites* x^{DS}, and so on. We chose to set up *IWR Tailorshop* on such an abstract level because this yields a more realistic position of a *decision maker* for the participants. Table 4.3 lists all states and controls the *IWR Tailorshop* contains together with corresponding units. The final model consists of 14 state variables and 10 control variables including 5 integer controls.

4.3 Variable Assumptions and Equations

In the following, we will discuss the assumptions for each variable in this microworld and develop the model equations. The starting point for the modeling was a concept for a model structure shown in Figure 4.5 which contains possible variables and influences without precise formulations. Based on this concept, possible model assumptions and variables have been developed. The approach is for each variable to consider by which other variables it is influenced. Control or decision variables are discussed together with their corresponding state variables if there are any.

4.3.1 Employees

For employees x_k^{EM}, a basic assumption is that their number always is greater or equal 1, i.e., there has to be at least one employee in the company. We do not differentiate between employees and assume that they are distributed to different tasks appropriately. The participant is allowed to recruit (u_k^{DEM}) and dismiss (u_k^{dEM}) employees,

$$x_{k+1}^{EM} = x_k^{EM} - u_k^{dEM} + u_k^{DEM} \tag{4.37a}$$

$$x_k^{EM} \geq 1 \tag{4.37b}$$

but only an integer number of employees, i.e., the controls for recruiting and dismissal are required to be integer,

$$u_k^{DEM} \in \mathbb{Z}_0^+, \qquad u_k^{dEM} \in \mathbb{Z}_0^+. \tag{4.38}$$

States	**Variable**	**Unit***	**Controls**	**Variable**	**Unit***
employees	x^{EM}	person(s)	shirt price	u^{SP}	MU/shirt
production sites	x^{PS}	site(s)	advertising	u^{AD}	MU
distribution sites	x^{DS}	site(s)	wages	u^{WA}	MU/person
shirts in stock	x^{SH}	shirt(s)	working conditions**	u^{WC}	MU
resources in stock	x^{RS}	shirt(s)	maintenance	u^{MA}	MU
production	x^{PR}	shirt(s)	buy resources**	u^{DRS}	shirt(s)
sales	x^{SA}	shirt(s)	sell resources**	u^{dRS}	shirt(s)
demand	x^{DE}	shirt(s)	resources quality	u^{RQ}	—
reputation	x^{RE}	—	recruit/dismiss empl.	u^{dEM} / u^{DEM}	person(s)
shirts quality	x^{SQ}	—		or u^{EM}	
machine quality	x^{MQ}	—	create production site	u^{DPS}	site(s)
resources quality	x^{RQ}	—	close production site	u^{dPS}	site(s)
motivation of empl.	x^{MO}	—	create distribution site	u^{DDS}	site(s)
resources price**	x^{RP}	MU/shirt	close distribution site	u^{dDS}	site(s)
capital	x^{CA}	MU			

Table 4.3: States and controls in the *IWR Tailorshop* microworld (* MU means monetary units, ** not part of the final model for the web-based study)

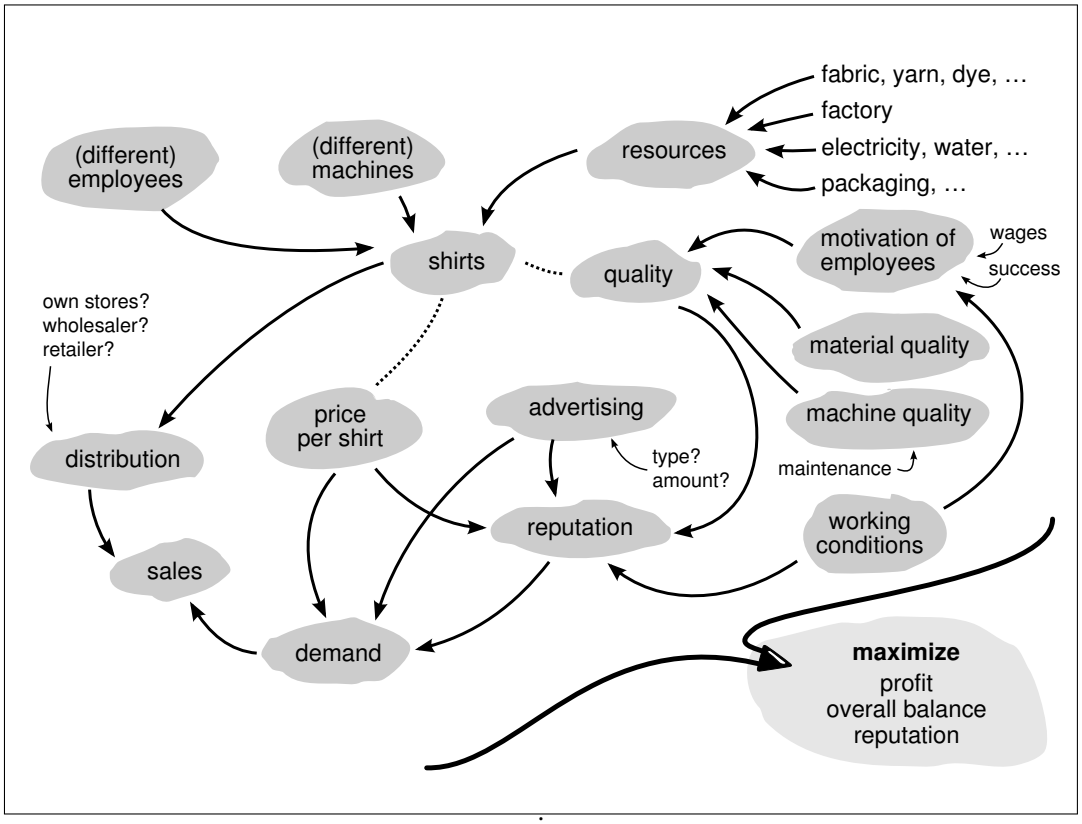


Figure 4.5: Concept for *IWR Tailorshop* with possible variables and dependencies. Arrows indicate possible dependencies in general, *not* positive, negative, or linear influence.

One could argue that a fractional amount of employees can be understood, e.g., as part-time employees, but usually employees will not work at an arbitrary full-time fraction (e.g., for 13.37 min per month) but at certain rates from a finite set (e.g., 25 %, 50 %, and 75 %). This can easily be modeled using integer variables as well. Thus, the integer constraints on u_k^{DEM} and u_k^{dEM} are reasonable.

At the beginning, *two* controls for recruiting and dismissal have been used to be able to include separate effects (i.e., motivation and discouragement) in the motivation equation. In the final model, however, these effects were rather weak so that a formulation with two separate controls was close to singularity with respect to these variables and thus, solvers had problems to deal with this. So, the two decision variables have been reduced to the variable u_k^{EM} with the corresponding equation

$$x_{k+1}^{EM} = x_k^{EM} + u_k^{EM}. \tag{4.39}$$

This simple equation follows directly from the assumptions and there are no reasonable alternatives. For the decisions on the number of employees, we assume that the amount of employees available for recruitment is limited due to the job market. Each production and distribution site yields access to a different job market,

$$u_k^{EM} \leq p^{DEM,0} \cdot x_k^{PS} + p^{DEM,1} \cdot x_k^{DS}. \tag{4.40}$$

The dismissal is also limited to a fixed number of employees per round under the assumption that higher dismissal would not be compatible, e.g., with unions,

$$u_k^{EM} \geq -p^{dEM}. \tag{4.41}$$

Parameter values for these equations are

$$p^{DEM,0} = 5\,\text{persons/site}, \qquad p^{DEM,1} = 10\,\text{persons/site}, \qquad p^{dEM} = 10\,\text{persons}. \tag{4.42}$$

4.3.2 Production Sites

The different types of machines from the *Tailorshop* microworld are represented by production sites in our new model. The number of production sites has to be greater or equal 1 and the participant is allowed to create (u_k^{dPS}) and close (u_k^{DPS}) production sites,

$$x_{k+1}^{PS} = x_k^{PS} - u_k^{dPS} + u_k^{DPS}, \tag{4.43a}$$

$$x_k^{PS} \geq 1. \tag{4.43b}$$

Again, there are no other options for this equation. One can only create or close whole production sites, i.e., these controls also have to be integer,

$$u_k^{DPS} \in \mathbb{Z}_0^+, \qquad u_k^{dPS} \in \mathbb{Z}_0^+. \tag{4.44a}$$

Here, two different controls are needed because of a different treatment in both the motivation of employees and the capital. There is a maximum amount for the creation of production sites in a month due to logistical and technical constraints and it is not possible to close more than one production site in two months (compare dismissal of employees),

$$u_k^{DPS} \leq p^{DPS} \tag{4.45a}$$

$$u_k^{dPS} + u_{k-1}^{dPS} \leq p^{dPS} \qquad \text{with} \quad p^{dPS} = 1\,\text{site} \tag{4.45b}$$

In this work, we used $p^{DPS} = 1$.

4.3.3 Distribution Sites

Distribution sites are introduced as a replacement for vans or stores and modeled quite symmetric to production sites. The assumption is again that the amount of distribution sites is ≥ 1 and the participant is allowed to create (u_k^{DDS}) and close (u_k^{dDS}) distribution sites,

$$x_{k+1}^{DS} = x_k^{DS} - u_k^{dDS} + u_k^{DDS}, \tag{4.46a}$$

$$x_k^{DS} \geq 1. \tag{4.46b}$$

Again, creating and closing distribution sites is treated differently in motivation of employees and capital equations and thus, two separate control variables which are required to be integer,

$$u_k^{DDS} \in \mathbb{Z}_0^+, \qquad u_k^{dDS} \in \mathbb{Z}_0^+, \tag{4.47}$$

are needed. There also are maximum numbers of distribution sites which may be created and closed in one month for the same reasons as given above,

$$u_k^{DDS} \leq p^{DDS}, \qquad u_k^{dDS} \leq p^{dDS}, \tag{4.48}$$

with the parameter values

$$p^{DDS} = 2\,\text{sites} \qquad p^{dDS} = 1\,\text{site}. \tag{4.49}$$

4.3.4 Shirts in Stock

The amount of shirts in stock is assumed to be greater or equal 0. Shirts are sold from the stock and in each month, newly produced shirts get in stock and can be sold in the same month. Thus, shirts in stock are computed from the old number of shirts in stock added the number of produced shirts minus the number of shirts sold,

$$x_{k+1}^{SH} = x_k^{SH} - x_{k+1}^{SA} + x_{k+1}^{PR} \tag{4.50a}$$

$$x_k^{SH} \geq 0 \tag{4.50b}$$

This requires, of course, shirt sales to be lower than shirts in stock and produced shirts together. Note that shirts in stock are not required to be integer which actually would be difficult to realize as the variable only depends on other states. A fractional amount of shirts can be considered to represent unfinished shirts. Here, a fractional value makes a lot more sense than for employees.

Another possible assumption is that shirt storage capacity is limited and that for each distribution site there is a certain storage capacity,

$$x_k^{SH} \leq p^{SH,0} \cdot x_k^{DS}, \tag{4.51}$$

with, e.g., $p^{SH,0} = 2000\,\text{shirts/site}$. However, such a constraint introduces a lot of additional complexity. Production, for instance, would then require a min-expression or an equivalent reformulation. In a reasonably configured model it should be optimal to sell the produced goods anyway and therefore, we dropped this assumption and the corresponding constraint in the final model.

4.3.5 Resources

All types of resources are considered in aggregated form. The assumption is that one resource unit is needed to produce one shirt. If a resource stock should be modeled, it would additionally be assumed that the amount of resources in stock x_k^{RS} is ≥ 0. Then, a participant would be allowed to buy u_k^{DRS} and possibly sell u_k^{dRS} (a limited amount of) resources,

$$u_k^{dRS} \leq p^{dRS,max}, \tag{4.52a}$$

$$u_k^{dRS} \geq 0, \tag{4.52b}$$

$$u_k^{DRS} \geq 0, \tag{4.52c}$$

e.g., with $p^{dRS,max} = 350$ shirts. A limitation of resource storage capacity can be realized much easier than a shirt storage limitation, as it only concerns the potential decision *buy resources*. If realized, for each production site there would be a certain storage capacity,

$$x_k^{RS} \leq p^{RS,0} \cdot x_k^{PS}, \tag{4.53a}$$

$$x_k^{RS} \geq 0, \tag{4.53b}$$

e.g., with $p^{RS,0} = 2000\,\text{shirts/site}$. The resulting equation would then consist of the old amount of resources in stock added the resources bought minus resources sold and resources consumed in production of shirts,

$$x_{k+1}^{RS} = x_k^{RS} - x_{k+1}^{PR} - u_k^{dRS} + u_k^{DRS}. \tag{4.54}$$

However, as mentioned above, buying and selling resources at an (rather) undynamic price is an operational, not a strategical decision and therefore both have been dropped for the final model. Additionally, a limitation of available resources to resources in stock again requires a min-expression or a corresponding reformulation for production. An alternative assumption is that resources can always be bought from the market for a fixed price per shirt.

We further assume that the participant can choose the resource quality u_k^{RQ} for each round either for the resources consumed by production or for the resources bought, depending on whether buying resources is included in the model. There is a finite number of different resource qualities available, identified by values between 0 and 1,

$$u_k^{RQ} \in \left\{ p^{RQ,1}, \ldots, p^{RQ,n^{RQ}} \right\}, \tag{4.55a}$$

$$\text{e.g., with} \quad n^{RQ} = 2, \qquad p^{RQ,1} = 0.5, \qquad p^{RQ,2} = 1.0. \tag{4.55b}$$

Low quality resources are cheaper than high quality ones, but also result in lower quality products. If a resource storage is modeled, quality may be interpolated linearly if different qualities get mixed, e.g.,

$$x_{k+1}^{RQ} = \frac{x_k^{RQ}\left(x_k^{RS} - u_k^{dRS}\right) + u_k^{RQ} \cdot u_k^{DRS}}{x_k^{RS} - u_k^{dRS} + u_k^{RS} + p^{RQ,zeroOffset}}, \tag{4.56a}$$

$$x_k^{RQ} \in [0,1]\,. \tag{4.56b}$$

In this case, of course, the resource quality decision has no impact if no resources are bought in a

round.

4.3.6 Production

For the shirt production, two basic assumptions are that neither without production sites nor without employees, the company can produce any shirts. The more production sites and employees there are, the more can be produced. We assume that there are saturation effects for employees per site. For instance, if there are already 783 employees at one production site, one additional employee will increase productivity less than at a site of the same size (e.g., think of a fixed number of machines) with only 2 employees. This assumption is modeled with a logarithmic term corresponding to a uniform distribution of employees to all production and distribution sites which is adjusted to be well-defined for all possible variable values,

$$\log\left(p^{PR,1}\cdot\frac{x_{k+1}^{EM}}{x_{k+1}^{PS}+x_{k+1}^{DS}+p^{PR,2}}+1\right). \tag{4.57}$$

This factor for the employees is multiplied with the number of production sites,

$$x_{k+1}^{PR}=p^{PR,0}\cdot\log\left(p^{PR,1}\cdot\frac{x_{k+1}^{EM}}{x_{k+1}^{PS}+x_{k+1}^{DS}+p^{PR,2}}+1\right)\cdot x_{k+1}^{PS}. \tag{4.58}$$

By the design of this equation, with appropriate parameter values, e.g.,

$$p^{PR,0}=99.9\,{}^{\text{shirts}}/_{\text{sites}},\qquad p^{PR,1}=2.0\,{}^{\text{sites}}/_{\text{persons}},\qquad p^{PR,2}=10^{-6}\,\text{sites}, \tag{4.59}$$

we also fulfill the assumption that production is always greater or equal 0,

$$x_k^{PR}\geq 0. \tag{4.60}$$

If a resource storage is modeled as discussed in the previous section, production requires enough resources in stock. In this case, production would be modeled as

$$x_{k+1}^{PR}=\min\left\{p^{PR,0}\cdot\log\left(p^{PR,1}\cdot\frac{x_{k+1}^{EM}}{x_{k+1}^{PS}+x_{k+1}^{DS}+p^{PR,2}}+1\right)\cdot x_{k+1}^{PS};x_k^{RS}+u_k^{DRS}-u_k^{dRS}\right\}, \tag{4.61}$$

which requires an equivalent reformulation for optimization.

4.3.7 Sales

The modeling of the variable *sales* is analog to production with respect to distribution sites instead of production sites. Without distribution sites and employees, the company cannot sell any shirts and the more distribution sites and employees there are, the more can be sold. We again assume that employees are uniformly distributed on production and distribution sites and that there is a saturation effect for the number of employees per site. As for production sites, this is modeled with a logarithmic term,

$$\log\left(p^{SA,1}\cdot\frac{x_{k+1}^{EM}}{x_{k+1}^{PS}+x_{k+1}^{DS}+p^{SA,2}}+1\right). \tag{4.62}$$

This term corresponds to the productivity of the employees per site and thus is multiplied by the number of distribution sites. However, for the sales there are restrictions which require a min-expression. First, sales cannot exceed the demand, see also the following section. Furthermore, the company can only sell what is in stock (except if its name is *FlowTex*, perhaps).

$$x_{k+1}^{SA} = \min\left\{p^{SA,0} \cdot \log\left(p^{SA,1} \cdot \frac{x_{k+1}^{EM}}{x_{k+1}^{PS} + x_{k+1}^{DS} + p^{SA,2}} + 1\right) \cdot x_{k+1}^{DS}; x_k^{SH} + x_{k+1}^{PR}; p^{SA,3} \cdot x_{k+1}^{DE}\right\} \tag{4.63}$$

This equation also ensures that sales are ≥ 0,

$$x_k^{SA} \geq 0. \tag{4.64}$$

Parameter values used in this work are

$$p^{SA,0} = 99.9\,\text{shirts/sites}, \qquad p^{SA,1} = 2.0\,\text{sites/persons}, \qquad p^{SA,2} = 10^{-6}\,\text{sites}, \qquad p^{SA,3} = 1.0. \tag{4.65}$$

4.3.8 Demand

The variable *demand* refers to the demand for the single company which the participant controls and not to the demand for the whole market. For goods like shirts, it is reasonable to assume that in competition the demand at a single company will fall if this company raises the price of shirts and vice versa. For the price component, we use a negative exponential term which is well-defined for all (positive) prices,

$$\exp\left(-p^{DE,1} \cdot u_k^{SP}\right). \tag{4.66}$$

Compared to, e.g., linear terms, the advantage of this formulation is that it does not need to be adapted to the maximum feasible shirt price.

We further assume that advertising raises demand with a saturation effect, modeled with a logarithmic term,

$$\log\left(p^{DE,2} \cdot u_k^{AD} + 1\right). \tag{4.67}$$

Finally, in *IWR Tailorshop*, the reputation of the company influences the demand as a factor with an offset. With these three components, we have

$$x_{k+1}^{DE} = p^{DE,0} \cdot \exp\left(-p^{DE,1} \cdot u_k^{SP}\right) \cdot \log\left(p^{DE,2} \cdot u_k^{AD} + 1\right) \cdot \left(x_k^{RE} + p^{DE,3}\right), \tag{4.68}$$

and this equation also ensures

$$x_k^{DE} \geq 0, \tag{4.69}$$

with the parameter values

$$p^{DE,0} = 2200.0\,\text{shirts}, \qquad p^{DE,1} = 2 \cdot 10^{-2}\,\text{shirts/M.U.}, \tag{4.70a}$$

$$p^{DE,2} = 2 \cdot 10^{-2}\,\text{1/M.U.}, \qquad p^{DE,3} = 0.5. \tag{4.70b}$$

4.3.9 Reputation

For the company's reputation, we assume that there is a memory effect, i.e., the reputation depends partly on the previous reputation. Further assumptions are that both high shirt quality and price, as well as advertising raise the reputation. Considering the shirt price, we additionally assume that there is an interaction with the shirt quality. A luxury product will not keep its luxury reputation if it

is sold at a low budget price. On the other hand, it will not be possible to achieve a good reputation by a high price when the product quality is low. These effects are modeled with the nonlinear term

$$p^{RE,3} \cdot u_k^{SP} \cdot (x_k^{SQ})^2, \tag{4.71}$$

where shirt quality enters quadratically. Working conditions, represented here by the variable *wages*, will also influence reputation.

All influences on reputation are assumed to be subject to saturation effects and thus modeled with a logarithmic term, as we only consider positive effects. The whole equation for reputation is

$$x_{k+1}^{RE} = p^{RE,0} \cdot x_k^{RE} + p^{RE,1} \log\left(\left(p^{RE,2} \cdot u_k^{AD} + p^{RE,3} \cdot u_k^{SP} \cdot (x_k^{SQ})^2 + p^{RE,4} \cdot u_k^{WA} \right) \cdot p^{RE,5} \right). \tag{4.72}$$

The corresponding parameter values,

$$p^{RE,0} = 0.5, \qquad p^{RE,1} = 0.627, \qquad p^{RE,2} = 2.5 \cdot 10^{-5}, \tag{4.73a}$$

$$p^{RE,3} = 10^{-4}\,\text{shirts}, \qquad p^{RE,4} = 6 \cdot 10^{-5}\,\text{persons}, \qquad x_5^{RE} = 12.0, \tag{4.73b}$$

are determined such that the log-term is always positive considering the lower (and upper) bounds on the controls. With a positive initial reputation, we then also have

$$x_k^{RE} \geq 0. \tag{4.74}$$

Additionally, these parameter values keep reputation between 0 and 1 for the majority of decisions. Realizing a true normalization, i.e., $x_k^{RE} \in [0,1]$, would require more complexity (min/max-expressions in the worst case) without yielding any advantage for the modeling.

4.3.10 Shirts Quality

Shirts quality is assumed to depend on the motivation of employees, the quality of machines, and the quality of resources. Highly motivated employees will produce better shirts and of course, high machine and material quality will improve the shirt quality as well. Thus, shirt quality is a linear combination of these factors which is positive because all the summands are positive,

$$x_{k+1}^{SQ} = p^{SQ,0} \cdot x_k^{MO} + p^{SQ,1} \cdot x_k^{MQ} + p^{SQ,2} \cdot u_k^{RQ}, \tag{4.75a}$$

$$x_k^{SQ} \geq 0, \tag{4.75b}$$

with the parameters

$$p^{SQ,0} = 0.2, \qquad p^{SQ,1} = 0.3, \qquad p^{SQ,2} = 0.5. \tag{4.76}$$

4.3.11 Machine Quality

We assume that the quality of machines for production of shirts decreases when machines are used. This decrease consists of a loss due to the machine load and is complemented by a temporal loss. The participant may decide on the amount of money spent for machine maintenance u_k^{MA} in each month. Maintenance spendings increase the machine quality and are greater or equal 10 M.U. Unless otherwise stated, maintenance is limited to 5000 M.U. Machine quality is determined in *IWR*

Tailorshop by

$$x_{k+1}^{MQ} = x_k^{MQ} \cdot p^{MQ,0} \cdot \exp\left(-p^{MQ,1} \frac{x_k^{PR}}{x_k^{PS} + p^{MQ,2}}\right) + p^{MQ,3} \cdot \log\left(u_k^{MA} \cdot p^{MQ,4} + 1\right), \tag{4.77a}$$

$$u_k^{MA} \in [10\,\text{M.U.}, 5000\,\text{M.U.}]. \tag{4.77b}$$

With a positive initial value, we have

$$x_k^{MQ} \geq 0, \tag{4.78}$$

and with the corresponding parameter values,

$$p^{MQ,0} = 0.8, \qquad p^{MQ,1} = 6 \cdot 10^{-3}\,\text{sites/shirts}, \qquad p^{MQ,2} = 10^{-6}\,\text{sites}, \tag{4.79a}$$

$$p^{MQ,3} = 0.13, \qquad p^{MQ,4} = 0.2\,\text{M.U.}^{-1}, \tag{4.79b}$$

machine quality mostly stays between 0 and 1, but for the same reasons as reputation it is not limited to $[0,1]$.

4.3.12 Motivation of Employees

The basic assumption for the motivation of employees is that there are factors which increase and factors which decrease the motivation. Factors which are assumed to motivate employees comprise the growth of the company (i.e., recruitment of employees and an increasing number of production or distribution sites), good working conditions (i.e., a high level of wages), and a good company reputation,

$$\log\left(p^{MO,1} \cdot (u_k^{EM} + p^{dEM}) + p^{MO,2} \cdot u_k^{DPS} + p^{MO,3} \cdot u_k^{DDS} + p^{MO,4} \cdot u_k^{WA} + p^{MO,5} \cdot x_k^{RE} + p^{MO,6}\right). \tag{4.80}$$

On the other hand, dismissal of employees and site closures are sources of discouragement,

$$\exp\left(-(p^{MO,7} \cdot u_k^{dPS} + p^{MO,8} \cdot u_k^{dDS}) + p^{MO,9}\right). \tag{4.81}$$

The product of these two effects determines the new level of motivation. Motivation of employees is assumed to be a convex combination of the old motivation and the new level of motivation,

$$\begin{aligned} x_{k+1}^{MO} = \left(1 - p^{MO,0}\right) \cdot x_k^{MO} + p^{MO,0} \cdot \log\Big(p^{MO,1} \cdot (u_k^{EM} + p^{dEM}) + p^{MO,2} \cdot u_k^{DPS} + p^{MO,3} \cdot u_k^{DDS} \\ + p^{MO,4} \cdot u_k^{WA} + p^{MO,5} \cdot x_k^{RE} + p^{MO,6}\Big) \cdot \exp\left(-(p^{MO,7} \cdot u_k^{dPS} + p^{MO,8} \cdot u_k^{dDS}) + p^{MO,9}\right) \\ \cdot p^{MO,10}. \end{aligned} \tag{4.82}$$

The corresponding parameter values,

$$p^{MO,0} = 0.5, \qquad p^{MO,1} = 2 \cdot 10^{-2}\,\text{persons}^{-1}, \qquad p^{MO,2} = 0.5\,\text{sites}^{-1}, \tag{4.83a}$$

$$p^{MO,3} = 0.25\,\text{sites}^{-1}, \qquad p^{MO,4} = 2.0 \cdot 10^{-4}\,\text{persons/M.U.}, \qquad p^{MO,5} = 0.3, \tag{4.83b}$$

$$p^{MO,6} = 1.0, \qquad p^{MO,7} = 2.5\,\text{sites}^{-1}, \qquad p^{MO,8} = 2.0\,\text{sites}^{-1}, \tag{4.83c}$$

$$p^{MO,9} = 1.0, \qquad p^{MO,10} = 0.5, \tag{4.83d}$$

are again chosen such that the motivation of employees is always ≥ 0 and between 0 and 1 for most cases, but it is not explicitly limited to $[0,1]$.

4.3.13 Resources Price

If the variables *buy* and *sell resources* are included in the model, the resources price may also be determined using the following equation instead of using a fixed value. If the resources price is modeled, we assume that it depends on supply and demand, i.e., on the amount of resources bought or sold by the company, as this is the only data available on supply and demand. It is assumed that there are always enough resources available on the market. Resources price should be ≥ 0 and will not immediately adapt to demand,

$$x_{k+1}^{RP} = \left(1 - p^{RP,0}\right) x_k^{RP} + p^{RP,0} \cdot \left(p^{RP,1}\left(u_k^{DRS} - u_k^{dRS}\right) + p^{RP,2}\right), \tag{4.84}$$

$$x_k^{RP} \geq 0. \tag{4.85}$$

Possible parameter values are

$$p^{RP,0} = 0.3, \qquad p^{RP,1} = \frac{7}{1080}\,\text{M.U.}/\text{shirt}^2, \qquad p^{RP,2} = \frac{101}{27}\,\text{M.U.}/\text{shirt}. \tag{4.86}$$

However, in the final model, the resources' price is not included as the variables for the amount of resources in stock have been removed.

4.3.14 Shirt Price, Advertising, Wages, and Working Conditions

The participant is allowed to set the price per shirt u_k^{SP}, the amount of money spent for advertising per month u_k^{AD}, and the wages per employee and month u_k^{WA}. All these controls are required to be greater or equal zero in general, and are subject to reasonable bounds, both for the *IWR Tailorshop* context and for the optimization,

$$u_k^{SP} \in [35\,\text{M.U.}, 55\,\text{M.U.}], \quad u_k^{AD} \in [1000\,\text{M.U.}, 2000\,\text{M.U.}], \quad u_k^{WA} \in [1000\,\text{M.U.}, 2000\,\text{M.U.}]. \tag{4.87}$$

In the early modeling phase, a variable for *working conditions* u_k^{WC} was considered to model monthly expenses for all workers, similar to social expenses in the *Tailorshop* microworld. The only equation this variable could reasonably influence seems to be the *motivation of employees*. However, the influence would have been quite low without dominating the other effects and thus, this imprecise variable has been dropped.

4.3.15 Capital

For the *capital* equation, components mostly follow directly from all other equations and assumptions. The capital is determined from the old capital plus revenues minus expenses, multiplied with an interest rate. We assume that the interest rate $p^{CA,0}$ is the same for both assets and debts. Although reality differs from this assumption, it is still close enough and the assumption of identical interest rates makes the model easier (for different interest rates, a smoothened *if/else* expression would be necessary). The capital itself remains unbounded, i.e., the participant is allowed to get into debt arbitrarily high.

Revenue is assumed to consist of the following components in each month. Each shirt sold yields an income in the amount of the shirt price. Closing a production or distribution site brings a fixed amount of money per site, $p^{CA,1}$ or $p^{CA,2}$. Furthermore, if *sell resources* is modeled, each resource unit sold earns a part of the resources price depending on the resource quality.

Expenses are assumed to contain the following components in each month. Each employee is payed wages. Production and distribution sites produce monthly fix costs, $p^{CA,4}$ and $p^{CA,5}$, per site. Furthermore, there are monthly expenses for machine maintenance and advertising. Each shirt in stock produces storage costs $p^{CA,6}$. If production or distribution sites are created, a fixed amount of money $p^{CA,7}$ or $p^{CA,8}$ per new site is spent. Finally, if resources are modeled, each resource unit bought costs the resource price multiplied by the chosen resource quality and resources in stock produces costs. Else, each shirt produced causes expenses of $p^{CA,3}$ multiplied by the resource quality.

Thus, we have the following equation for capital,

$$\begin{aligned} x_{k+1}^{CA} = p^{CA,0} \cdot \Big(& x_k^{CA} + \left(x_{k+1}^{SA} \cdot u_k^{SP}\right) + \left(u_k^{dPS} \cdot p^{CA,1}\right) + \left(u_k^{dDS} \cdot p^{CA,2}\right) - \left(x_{k+1}^{EM} \cdot u_k^{WA}\right) \\ & - \left(x_{k+1}^{PR} \cdot u_k^{RQ} \cdot p^{CA,3}\right) - \left(x_k^{PS} \cdot p^{CA,4}\right) - \left(x_k^{DS} \cdot p^{CA,5}\right) - u_k^{MA} - u_k^{AD} - \left(x_{k+1}^{SH} \cdot p^{CA,6}\right) \\ & - \left(u^{DPS} \cdot p^{CA,7}\right) - \left(u^{DDS} \cdot p^{CA,8}\right)\Big), \end{aligned} \tag{4.88}$$

with the corresponding parameters,

$$p^{CA,0} = 1.03, \qquad p^{CA,7} = 10000\,\text{M.U.}/\text{site}, \qquad p^{CA,1} = 5000\,\text{M.U.}/\text{site}, \tag{4.89a}$$

$$p^{CA,4} = 1000\,\text{M.U.}/\text{site}, \qquad p^{CA,8} = 7000\,\text{M.U.}/\text{site}, \qquad p^{CA,2} = 3500\,\text{M.U.}/\text{site}, \tag{4.89b}$$

$$p^{CA,5} = 700\,\text{M.U.}/\text{site}, \qquad p^{CA,6} = 1.5\,\text{M.U.}/\text{shirt}. \tag{4.89c}$$

4.4 Model Overview

With the assumptions and equations from the previous section, the mathematical representation of the *IWR Tailorshop* consists of the following set of equations for $k = t_0, \ldots, t_f$.

$$x_{k+1}^{EM} = x_k^{EM} + u_k^{EM} \tag{4.90a}$$

$$x_{k+1}^{PS} = x_k^{PS} - u_k^{dPS} + u_k^{DPS} \tag{4.90b}$$

$$x_{k+1}^{DS} = x_k^{DS} - u_k^{dDS} + u_k^{DDS} \tag{4.90c}$$

$$x_{k+1}^{DE} = p^{DE,0} \cdot \exp\left(-p^{DE,1} \cdot u_k^{SP}\right) \cdot \log\left(p^{DE,2} \cdot u_k^{AD} + 1\right) \cdot \left(x_k^{RE} + p^{DE,3}\right) \tag{4.90d}$$

$$x_{k+1}^{RE} = p^{RE,0} \cdot x_k^{RE} + p^{RE,1} \log\left(\left(p^{RE,2} \cdot u_k^{AD} + p^{RE,3} \cdot u_k^{SP} \cdot (x_k^{SQ})^2 + p^{RE,4} \cdot u_k^{WA}\right) \cdot p^{RE,5}\right) \tag{4.90e}$$

$$x_{k+1}^{PR} = p^{PR,0} \cdot x_{k+1}^{PS} \cdot \log\left(\frac{p^{PR,1} \cdot x_{k+1}^{EM}}{x_{k+1}^{PS} + x_{k+1}^{DS} + p^{PR,2}} + 1\right) \tag{4.90f}$$

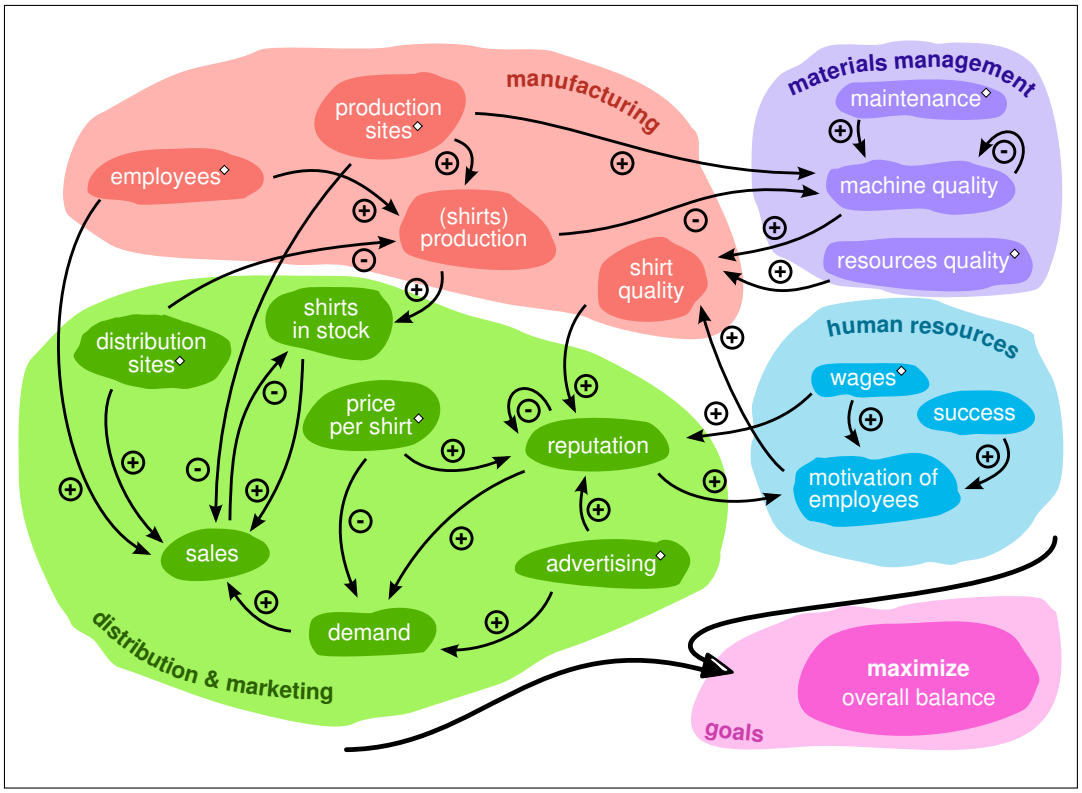


Figure 4.6: *IWR Tailorshop* model. Arrows show proportional/reciprocal dependencies, diamond indicates participants' control influence.

$$x_{k+1}^{SA} = \min\left\{ p^{SA,0} \cdot x_{k+1}^{DS} \cdot \log\left(\frac{p^{SA,1} \cdot x_{k+1}^{EM}}{x_{k+1}^{PS} + x_{k+1}^{DS} + p^{SA,2}} + 1 \right); x_k^{SH} + x_{k+1}^{PR}; p^{SA,3} \cdot x_{k+1}^{DE} \right\} \tag{4.90g}$$

$$x_{k+1}^{SH} = x_k^{SH} - x_{k+1}^{SA} + x_{k+1}^{PR} \tag{4.90h}$$

$$x_{k+1}^{SQ} = p^{SQ,0} \cdot x_k^{MO} + p^{SQ,1} \cdot x_k^{MQ} + p^{SQ,2} \cdot u_k^{RQ} \tag{4.90i}$$

$$x_{k+1}^{MQ} = x_k^{MQ} \cdot p^{MQ,0} \cdot \exp\left(-p^{MQ,1} \frac{x_k^{PR}}{x_k^{PS} + p^{MQ,2}} \right) + p^{MQ,3} \cdot \log\left(u_k^{MA} \cdot p^{MQ,4} + 1 \right) \tag{4.90j}$$

$$\begin{aligned} x_{k+1}^{MO} = {} & \left(1 - p^{MO,0}\right) \cdot x_k^{MO} + p^{MO,0} \cdot \log\left(p^{MO,1} \cdot (u_k^{EM} + p^{dEM}) + p^{MO,2} \cdot u_k^{DPS} + p^{MO,3} \cdot u_k^{DDS} \right. \\ & \left. + p^{MO,4} \cdot u_k^{WA} + p^{MO,5} \cdot x_k^{RE} + p^{MO,6} \right) \cdot \exp\left(-(p^{MO,7} \cdot u_k^{dPS} + p^{MO,8} \cdot u_k^{dDS}) + p^{MO,9} \right) \\ & \cdot p^{MO,10} \end{aligned} \tag{4.90k}$$

$$
\begin{aligned}
x_{k+1}^{CA} = p^{CA,0} \cdot \Big(& x_k^{CA} + \left(x_{k+1}^{SA} \cdot u_k^{SP}\right) + \left(u_k^{dPS} \cdot p^{CA,1}\right) + \left(u_k^{dDS} \cdot p^{CA,2}\right) - \left(x_{k+1}^{EM} \cdot u_k^{WA}\right) \\
& - \left(x_{k+1}^{PR} \cdot u_k^{RQ} \cdot p^{CA,3}\right) - \left(x_k^{PS} \cdot p^{CA,4}\right) - \left(x_k^{DS} \cdot p^{CA,5}\right) - u_k^{MA} - u_k^{AD} - \left(x_{k+1}^{SH} \cdot p^{CA,6}\right) \\
& - \left(u^{DPS} \cdot p^{CA,7}\right) - \left(u^{DDS} \cdot p^{CA,8}\right) \Big)
\end{aligned} \tag{4.90l}
$$

Additional constraints are given by the inequalities

$$u_k^{dPS} + u_{k-1}^{dPS} \leq p^{dPS}, \tag{4.91a}$$

$$u_k^{EM} \leq p^{DEM,0} \cdot x_k^{PS} + p^{DEM,1} \cdot x_k^{DS}, \tag{4.91b}$$

$$x_k^{EM}, x_k^{PS}, x_k^{DS} \geq 1, \tag{4.91c}$$

$$x_k^{SH}, x_k^{PR}, x_k^{SA}, x_k^{DE}, x_k^{RE}, x_k^{SQ}, x_k^{MQ}, x_k^{MO} \geq 0, \tag{4.91d}$$

and the simple bounds on the constraints,

$$u_k^{SP} \in [35\,\text{M.U.}, 55\,\text{M.U.}], \qquad u_k^{AD} \in [1000\,\text{M.U.}, 2000\,\text{M.U.}], \tag{4.92a}$$

$$u_k^{WA} \in [1000\,\text{M.U.}, 2000\,\text{M.U.}], \qquad u_k^{MA} \in [10\,\text{M.U.}, 5000\,\text{M.U.}], \tag{4.92b}$$

$$u_k^{RQ} \in \left\{p^{RQ,1}, p^{RQ,2}\right\}, \qquad u_k^{EM} \in \left[-p^{dEM}, \infty\right] \cap \mathbb{Z}_0^+, \tag{4.92c}$$

$$u_k^{DPS} \in \left[0, p^{DPS}\right] \cap \mathbb{Z}_0^+, \qquad u_k^{dPS} \in \mathbb{Z}_0^+, \tag{4.92d}$$

$$u_k^{DDS} \in \left[0, p^{DDS}\right] \cap \mathbb{Z}_0^+, \qquad u_k^{dDS} \in \left[0, p^{dDS}\right] \cap \mathbb{Z}_0^+. \tag{4.92e}$$

Compared to equation (3.17), these equations and inequalities together with the reformulation of the sales equation, see Chapter 3, form the functions G and H. For the objective function F, one could think of different options, e.g., a weighted combination of maximizing profit, reputation, and some other factors. We decided to use the capital at the end of the discrete time-scale in this work, which effectively avoids end-time effects. Hence, we use the following objective:

$$\max_{x,u,p} \quad x_N^{CA} \tag{4.93}$$

Of course, the set of parameters has a significant influence on the model behavior. One could, e.g., think of applying derivative-free optimization methods with a subset of the parameters to determine an appropriate parameter set for a microworld like *IWR Tailorshop*. For this work, however, we set up a parameter set manually such that the model fulfills a certain desired behavior, as described in Section 4.3. The chosen parameters also yield a model behavior that makes sense for the optimization, i.e. there are feasible solutions and the optimization problem is not unbounded. The parameter values used throughout this work unless otherwise stated are listed in Tables 4.4 and 4.5.

Parameter	Value	Parameter	Value
$p^{DE,0}$	2200.0 shirts	$p^{MQ,3}$	0.13
$p^{DE,1}$	$2 \cdot 10^{-2}$ shirts/MU	$p^{MQ,4}$	0.2 MU^{-1}
$p^{DE,2}$	$2 \cdot 10^{-2}$ 1/MU	$p^{MO,0}$	0.5
$p^{DE,3}$	0.5	$p^{MO,1}$	$2 \cdot 10^{-2}$ persons^{-1}
$p^{RE,0}$	0.5	$p^{MO,2}$	0.5 sites^{-1}
$p^{RE,1}$	0.672	$p^{MO,3}$	0.25 sites^{-1}
$p^{RE,2}$	$2.5 \cdot 10^{-5}$ 1/MU	$p^{MO,4}$	$2.0 \cdot 10^{-4}$ persons/MU
$p^{RE,3}$	10^{-4} shirts/MU	$p^{MO,5}$	0.3
$p^{RE,4}$	$6 \cdot 10^{-5}$ persons/MU	$p^{MO,6}$	1.0
$p^{RE,5}$	12.0	$p^{MO,7}$	2.5 sites^{-1}
$p^{PR,0}$	99.9 shirts/sites	$p^{MO,8}$	2.0 sites^{-1}
$p^{PR,1}$	2.0 sites/persons	$p^{MO,9}$	1.0
$p^{PR,2}$	10^{-6} sites	$p^{MO,10}$	0.5
$p^{SA,0}$	99.9 shirts/sites	$p^{CA,0}$	1.03
$p^{SA,1}$	2.0 sites/persons	$p^{CA,1}$	5000 MU/site
$p^{SA,2}$	10^{-6} sites	$p^{CA,2}$	3500 MU/site
$p^{SA,3}$	1.0	$p^{CA,3}$	5.0 MU/shirt
$p^{SQ,0}$	0.2	$p^{CA,4}$	1000 MU/site
$p^{SQ,1}$	0.3	$p^{CA,5}$	700 MU/site
$p^{SQ,2}$	0.5	$p^{CA,6}$	1.5 MU/shirt
$p^{MQ,0}$	0.8	$p^{CA,7}$	10000 MU/site
$p^{MQ,1}$	$6 \cdot 10^{-3}$ sites/shirts	$p^{CA,8}$	7000 MU/site
$p^{MQ,2}$	10^{-6} sites		

Table 4.4: Parameter set for states used with *IWR Tailorshop*. MU means monetary units.

Parameter	Value	Parameter	Value
n^{RQ}	2	p^{dEM}	10 persons
$p^{RQ,1}$	0.5	p^{DPS}	1 site
$p^{RQ,2}$	1.0	p^{dPS}	1 site
$p^{DEM,0}$	5 persons/site	p^{DDS}	2 sites
$p^{DEM,1}$	10 persons/site	p^{dDS}	1 site

Table 4.5: Parameter set for controls used with *IWR Tailorshop*.

CHAPTER 5

Methods for Analysis and Training of Human Decision Making

In this chapter, we describe methods for an optimization-based analysis and training of human decision making. We also investigate different reformulations of the *IWR Tailorshop* model which are necessary for the application of optimization methods. We present a tailored decomposition approach for the computation of valid upper bounds for optimal solutions of *IWR Tailorshop* and discuss an approach for model parameter optimization.

5.1 Optimization-based Analysis of Human Decision Making

In Section 2.6, we have seen that complex microworlds are used as test-scenarios in CPS to analyze human decision making. Remember that in CPS, researchers need an indicator for the participant's performance, as the performance in such test-scenarios is usually correlated with some other variables (e.g., personal attributes or experimental conditions). Common approaches—including the analysis of the evolution of variables like *capital* in the *Tailorshop*—have severe drawbacks, see also Section 2.6.

We recall that many microworlds and especially the *IWR Tailorshop* presented in the previous chapter can be formulated as a *discretized mixed-integer optimal control problem* (dMIOCP), and thus, we can formulate the following definition.

Definition 5.1 The *IWR Tailorshop optimization problem* (ITOP) is the task to solve the problem

$$\begin{aligned} \min_{x,u} \quad & F(x,u,p) \\ \text{s.t.} \quad & x_{k+1} = G(x_k,u_k,p), \quad k = t_0,\dots,t_f-1, \\ & 0 \le H(x_k,u_k,p), \quad k = t_0,\dots,t_f, \\ & u_k \in \Omega, \quad k = t_0,\dots,t_f-1, \\ & x_{t_0} = x_0, \end{aligned} \tag{5.1}$$

with F, G, H, Ω, x, p, and u defined by the description of *IWR Tailorshop* in Chapter 4. Unless otherwise stated, we have

$$t_0 = 0 \quad \text{and} \quad t_f = 10. \tag{5.2}$$

Reasonable initial values x_0 are given, e.g., in Table 6.1.

The ITOP is the task for a potential participant in a study using the *IWR Tailorshop*. For the analysis of decisions made by a participant, we suggest to use optimal solutions of the ITOP. A comparison of the objective function values, i.e., the end time capital, achieved by the participants with the one of the optimal solution for ITOP also gives an objective indicator. However, this approach does not yield any information to also determine *when* significant performance deviations occurred or even which decisions were particularly good or bad ones with respect to the overall outcome.

Note that a comparison with the *controls* of the optimal solution for starting month $t_0 = 0$ would not yield a good indicator function, as there might be multiple ways to perform well. If, for instance, due to his previous actions, a participant has many shirts on his stock, good decisions may differ

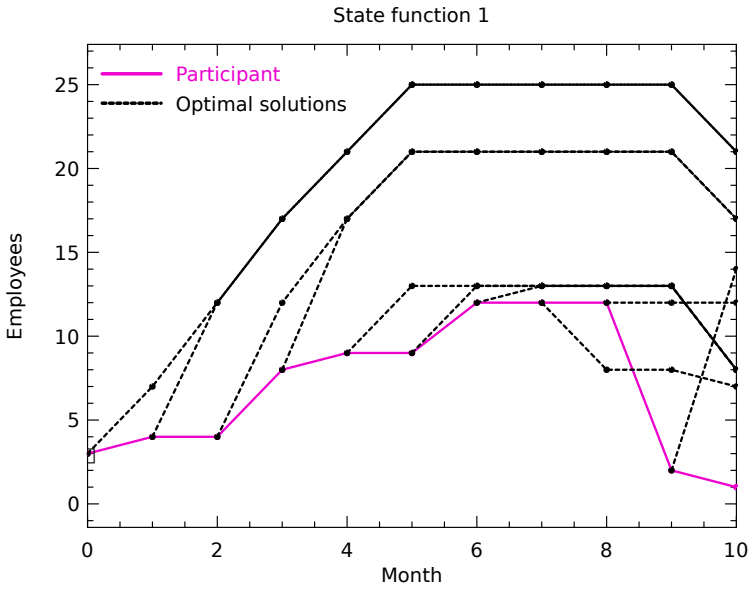


Figure 5.1: Variable *employees* as derived by a participant (solid line) together with optimal solutions (dashed lines). The optimal strategy changes several times (months 1 to 2, 3 to 4, and the last three times) depending on the remaining time and the state the *IWR Tailorshop* is in.

drastically from a situation in which the stock is empty and demand is much higher than current shirt sales (see Figure 5.1). Therefore, we do not only consider the ITOP, but also a series of optimization problems:

Definition 5.2 The *IWR Tailorshop analysis problem* (ITAP) is a series of optimization problems for

$$t_s \in \{0, 1, \ldots, t_f - 1\} \tag{5.3}$$

with objective values $\Phi_H(t_s)$ based on the ITOP,

$$\begin{aligned} \Phi_H(t_s) := \min_{x,u} \quad & F(x,u,p) \\ \text{s.t.} \quad x_{k+1} \quad & = G(x_k,u_k,p), \quad && k = t_s,\ldots,t_f-1, \\ 0 \quad & \leq H(x_k,u_k,p), \quad && k = t_s,\ldots,t_f, \\ u_k \quad & \in \Omega, && k = t_s,\ldots,t_f-1, \\ x_{t_s} \quad & = x^{\mathrm{P}}_{t_s}, \end{aligned} \tag{5.4}$$

with F, G, H, Ω, x, p, and u defined by the description of *IWR Tailorshop* in Chapter 4. The initial values for the problem series, $x^{\mathrm{P}}_{t_s}$, are the states derived by a participant's decisions until month t_s. Unless otherwise stated, in this work we have again $t_f = 10$.

In contrast to the ITOP, the problem series in ITAP starts with initial states derived by a participant. Thus, in the ITAP, we solve the ITOP for every round of the participant's data, starting with exactly the same conditions as the participant. In a certain analogy to the *cost-to-go*-function in dynamic programming, the optimal objective function values for *all* months yield a monotonically decreasing

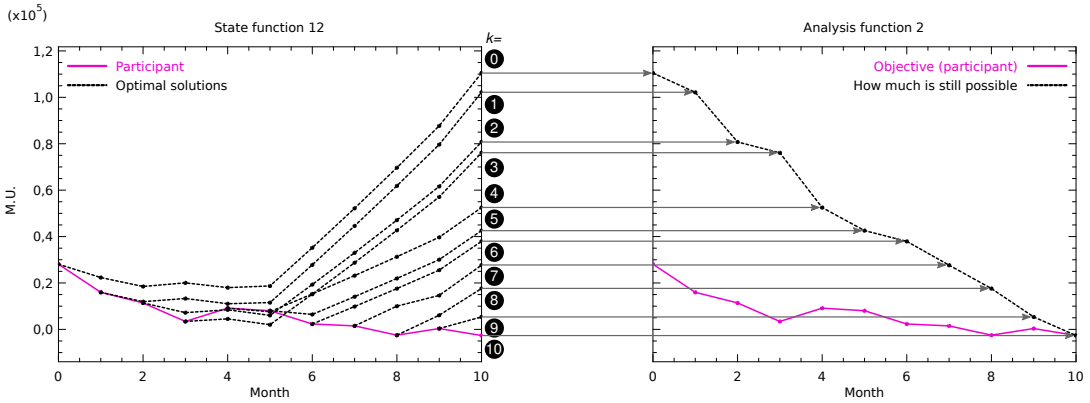


Figure 5.2: Illustration of the definition of the *How much is still possible*-function. The left plot contains the variable *capital* (magenta) as derived by the decisions of a participant together with all optimal solutions for the variable *capital* (dashed black) with the optimization starting in the states derived by the participant in all the months. The final values of the optimal solutions can be seen as a function over the months again, see the plot on the right hand side, which is the *How much is still possible*-function.

function (if we found global optima or, at least, if the participants did not find better solutions for any month). Therefore, we suggest to use the values $\Phi_H(t_s)$ as an indicator function for the performance of a participant.

Definition 5.3 We call $\Phi_H : [t_0, t_f] \cap \mathbb{Z}_0^+ \longrightarrow \mathbb{R} : t_s \mapsto \Phi_H(t_s)$ the *How much is still possible*-function for *IWR Tailorshop*.

The values of this function indicate—corresponding to its name—how much still would be possible to achieve if all future decisions were optimal. Figure 5.2 shows the solutions of ITAP together with the variable *capital* for a single participant and illustrates how the *How much is still possible*-function is derived. With this indicator function, we can analyze in which months potential for a higher objective function value has been lost by suboptimal decisions. In Figure 5.3, two examples are analyzed. Using the evolution of the variable *capital* as an indicator, the analysis coincides with the *How much is still possible*-function for the first example. Here, performance is not so good in the first two months, but significantly better from month 3 on. For the second plot, the evolution of *capital* would mislead, as decisions in month 3, for instance would be analyzed as good (*capital* increases), although the participant actually loses much potential in this round, as the *How much is still possible*-function shows. Decisions for month 2 were quite good but would vice versa be analyzed as bad considering the *capital*.

By comparing $\Phi_H(t_s)$ with $\Phi_H(t_s+1)$, we obtain the exact value of how much less the participant is able to obtain.

Definition 5.4 We call

$$\Phi_P : [t_0, t_f - 1] \cap \mathbb{Z}_0^+ \longrightarrow \mathbb{R} : t_s \mapsto \Phi_P(t_s) := \Phi_H(t_s + 1) - \Phi_H(t_s) \tag{5.5}$$

the *Use of potential*-function.

Under the same assumptions as for the *How much is still possible*-function, this is a non-positive function. The *Use of potential* indicates, as explained above, how much potential a participant lost in

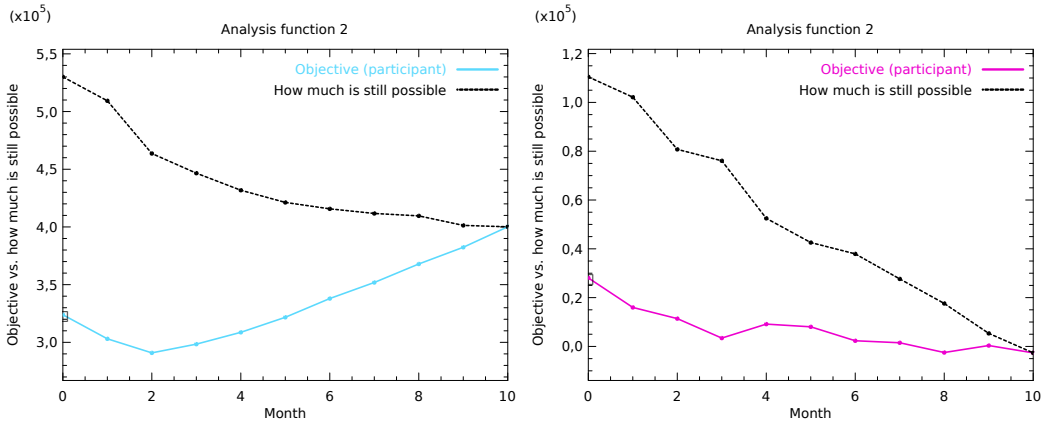


Figure 5.3: *How much is still possible*-function and variable *capital*: the solid lines show the evolution of *capital* derived by the participants' decisions, the dashed lines show the *How much is still possible*-function which is composed of objective function values of optimal solutions with optimization starting in the corresponding months. Using *capital*, which is evaluated as the objective at the end of the discrete time scale, as an indicator, the analysis coincides with the *How much is still possible*-function for the left plot: performance is not so good in the first two months, but significantly better from month 3 on. For the right plot, the evolution of *capital* would mislead, as decisions in month 3, for instance would be analyzed as good (*capital* increases), while they actually are quite bad, as the *How much is still possible*-function reveals.

a month, i.e., it is 0 for optimal decisions. An increase of *Use of potential* can thus also be considered as learning: when a participant learns how to control the microworld, he is able to better use the potential.

By its definition, *Use of potential* is kind of a discrete derivative of *How much is still possible*. Figure 5.4 illustrates this dependency between these two functions. Note that in general also a relative loss given as a percentage could be used, but this does only make sense in the presence of reasonable bounds on F or Φ_H respectively, which is not the case for *IWR Tailorshop*.

The approach described in this section is generic and should also be used for other test scenarios in complex problem solving in the future. In [112], this methodology has been applied to the original *Tailorshop* microworld. Due to the drawbacks of that test-scenario described in Section 4.1.3, the variable *vans*, for instance, was not part of the optimization. In Section 6.3, we present results obtained by using the *How much is still possible*- and *Use of potential*-functions in a web-based feedback study with *IWR Tailorshop*. Once the performance of all participants has been determined, an aggregation and further statistical analysis can be performed. The proposed methodology is more reliable than non–optimization-based indicator functions and generally applicable to test-scenarios in complex problem solving.

5.2 Computing Optimization-based Feedback

Depending on the computing time for one problem in the ITAP series, the approach from the previous section can also be used to compute an optimization-based feedback while a participant solves the task. Computing times have to be sufficiently low then, of course—see Section 5.3 for the efforts taken in *IWR Tailorshop* regarding this issue.

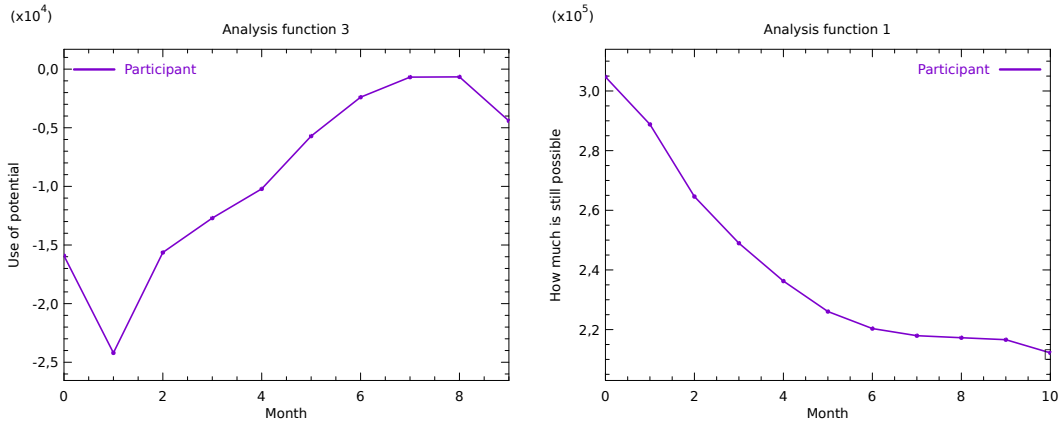


Figure 5.4: Dependency between *Use of potential*- and *How much is still possible*-functions: *Use of potential* (left) is kind of a discrete derivative of *How much is still possible* (right). In this example, the participant seems to learn how to control the microworld as *Use of potential* is increasing.

For an optimization-based feedback, we distinguish to aspects: on the one hand, there is the way the feedback is computed and on the other hand there is the presentation of such a feedback. Figure 5.5 gives an overview of the optimization-based feedback methods discussed in this section and implemented in the *IWR Tailorshop* web interface.

Considering the way a feedback is computed, there are basically two approaches. The first one is to compute an optimal solution and to use the optimal controls for the feedback.

Definition 5.5 The *IWR Tailorshop feedback problem* (ITFP) is the problem to determine optimal controls $u^*(t_s)$ starting in the state $x^{\mathrm{P}}_{t_s}$ derived by a participant until month t_s,

$$
\begin{aligned}
u^*(t_s) := \arg\min_{u} \quad & F(x,u,p) \\
\text{s.t.} \quad x_{k+1} \quad & = G(x_k,u_k,p), \quad && k = t_s,\dots,t_f-1, \\
0 \quad & \le H(x_k,u_k,p), && k = t_s,\dots,t_f, \\
u_k \quad & \in \Omega, && k = t_s,\dots,t_f-1, \\
x_{t_s} \quad & = x^{\mathrm{P}}_{t_s}, &&
\end{aligned}
\tag{5.6}
$$

with F, G, H, Ω, x, p, u, t_f defined as before.

Let the microworld be at month $k+1$, i.e., the participant has to determine controls u_{k+1}. Then there are two reasonable approaches: first, to compute $u^*(k+1)$ to give the participant a hint what would be optimal in the current state—approach A in Figure 5.5; and second, to compute $u^*(k)$ to give the participant information about the previously made decisions—approach C.

The second method for optimization-based feedback can be considered sensitivity-based. We introduce artificial constraints to fix the controls chosen by the participant for month k and—at least in the continuous case—use the LAGRANGE *multipliers* of these constraints as a sensitivity feedback.

Definition 5.6 The *IWR Tailorshop sensitivity feedback problem* (ITSFP) is the problem to determine

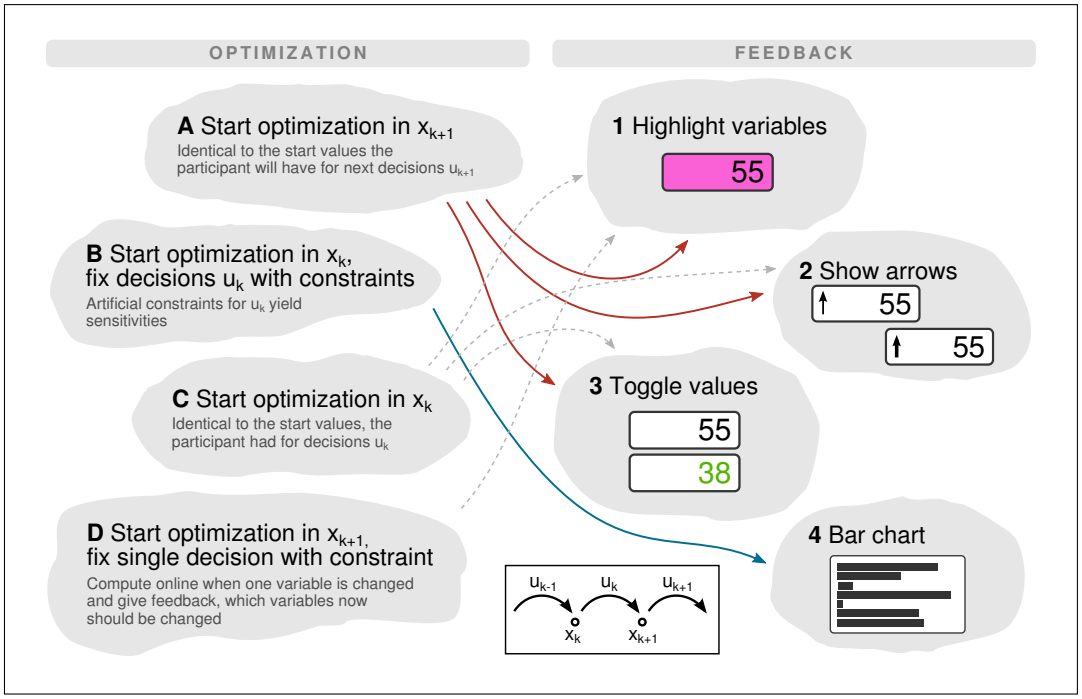


Figure 5.5: Optimization-based feedback at month $k+1$: on the left hand side, there are different methods to compute a feedback and on the right hand side there are different types of feedback presentation. Optimization method A is used with feedback presentation 1, 2, and 3 and optimization method B is used with feedback presentation 4 in the *IWR Tailorshop* web interface. Further optimization methods C and D have not been implemented for the use with *IWR Tailorshop*.

sensitivities for the controls $u^{\mathrm{P}}_{t_s}$ determined by a participant for month t_s from

$$\begin{aligned}
\min_{x,u} \quad & F(x,u,p) \\
\text{s.t.} \quad & x_{k+1} = G(x_k,u_k,p), && k = t_s,\dots,t_f-1, \\
& 0 \le H(x_k,u_k,p), && k = t_s,\dots,t_f, \\
& u_k \in \Omega, && k = t_s,\dots,t_f-1, \\
& u_{t_s} = u^{\mathrm{P}}_{t_s}, \\
& x_{t_s} = x^{\mathrm{P}}_{t_s},
\end{aligned} \tag{5.7}$$

by determining LAGRANGE multipliers $\lambda^*_{t_s}$ for the constraints

$$u_{t_s} = u^{\mathrm{P}}_{t_s} \tag{5.8}$$

with F, G, H, Ω, x, p, u, t_f, and $x^{\mathrm{P}}_{t_s}$ defined as before.

With the ITSFP formulation, an optimization code will automatically calculate the LAGRANGE multipliers or dual variables for the constraints (5.8). It is well known that the LAGRANGE multipliers indicate the shadow prices, i.e., how much the objective function will vary if the corresponding constraints were relaxed. However, this is a local information for the point $(x^{\mathrm{P}}_{t_s},\dots,x^{\mathrm{P}}_{t_f},u^{\mathrm{P}}_{t_s},\dots,u^{\mathrm{P}}_{t_f-1})$ and assumes that the active set of inequality constraints does not change.

Mixed-integer variables need additional care in the ITSFP, as for integer variables this is not a feasible concept. For *IWR Tailorshop*, we first compute a mixed-integer solution then and in a second step fix this solution with constraints in a relaxed (i.e., continuous) formulation of the problem. These multipliers can then be used as an approximation of the desired sensitivity information. This approach is realized in the *IWR Tailorshop* web interface as approach B.

Both from a training and a decision support aspect, one could think of a kind of a combination of these different methods—approach D. Whenever the participant changes a control value in the interface, an ITSFP with the fixation of only that single control value could be solved. The resulting optimal solution gives a feedback which controls need to be changed. This could possibly also be done based on sensitivities. Unfortunately, this is far beyond the current capabilities of MINLP solvers and would lead to an unusable interface as the participant would have to wait far too long after changing one single control (at least for high $t_f - t_0$).

Regarding the presentation of the feedback, four options are implemented in the *IWR Tailorshop* web interface, see Figure 5.5. Control variables can be highlighted, up and down arrows in different thickness can be shown next to them, and values can be toggled. Another option is to display a bar chart for all control variables when controls have been submitted.

For *IWR Tailorshop*, we use the combination of optimization approach A with highlighting, arrows, and toggling. Approach B is used together with the bar chart. Variables are highlighted if they differ from the optimal value more than a given threshold, e.g., 30 % of the difference δ between lower and upper bound of the variable. Arrows indicate the direction of the optimal control: if the optimal control is greater, the arrow points up and vice versa. Arrow thickness is also determined by thresholds on δ. For the value toggling, the exact optimal control values are shown. In the bar chart, LAGRANGE multipliers are displayed scaled according to δ. We will refer to these four options as *highlight*, *trend*, *value*, and *chart*.

5.3 Model Reformulations

5.3.1 Shirt Sales

We recall the state progression law for the variable *sales*, equation (4.90g),

$$x_{k+1}^{SA} = \min\left\{ p^{SA,0} \cdot \log\left(p^{SA,1} \cdot \frac{x_{k+1}^{EM}}{x_{k+1}^{PS} + x_{k+1}^{DS} + p^{SA,2}} + 1 \right) \cdot x_{k+1}^{DS}; x_k^{SH} + x_{k+1}^{PR}; p^{SA,3} \cdot x_{k+1}^{DE} \right\}, \tag{5.9}$$

which consists of the minimum of three expressions. For the solution of ITOP and the like, the problem is that a min-expression is not differentiable. However, we want to apply the derivative-based optimization methods described for dMIOCP in Chapter 3. Hence we need a differentiable reformulation of *IWR Tailorshop*. In the following, we will use abbreviations T_k^i, $i \in \{1,2,3\}$, for the min-terms,

$$T_{k+1}^1 := p^{SA,0} \cdot \log\left(p^{SA,1} \cdot \frac{x_{k+1}^{EM}}{x_{k+1}^{PS} + x_{k+1}^{DS} + p^{SA,2}} + 1 \right) \cdot x_{k+1}^{DS}, \tag{5.10a}$$

$$T_{k+1}^2 := x_k^{SH} + x_{k+1}^{PR}, \tag{5.10b}$$

$$T_{k+1}^3 := p^{SA,3} \cdot x_{k+1}^{DE}, \tag{5.10c}$$

so that we have

$$x_{k+1}^{SA} = \min\left\{ T_{k+1}^1; T_{k+1}^2; T_{k+1}^3 \right\}. \tag{5.11}$$

5.3.2 Reformulations of Shirt Sales

An obvious reformulation of the minimum would be to use inequality conditions,

$$x_{k+1}^{SA} \le T_{k+1}^1, \qquad x_{k+1}^{SA} \le T_{k+1}^2, \qquad x_{k+1}^{SA} \le T_{k+1}^3. \tag{5.12}$$

However, this only guarantees that x_{k+1}^{SA} is *less* than all the parts of the minimum. In our case, though the influence of *sales* is positive in the objective,

$$x_{k+1}^{CA} = p^{CA,0} \cdot \left(\cdots + \left(x_{k+1}^{SA} \cdot u_k^{SP} \right) + \ldots \right), \tag{5.13}$$

because of *shirts in stock's* equation,

$$x_{k+1}^{SH} = x_k^{SH} - x_{k+1}^{SA} + x_{k+1}^{PR}, \tag{5.14}$$

it could be beneficial to keep shirts in stock to sell them later at some higher price if possibly the demand also is higher due to a better reputation. Of course, this can easily be avoided by choosing storage costs to be higher than the maximum shirt price, since

$$x_{k+1}^{CA} = p^{CA,0} \cdot \left(\cdots - \left(x_{k+1}^{SH} \cdot p^{CA,6} \right) - \ldots \right), \tag{5.15}$$

but it would be an unrealistic assumption that storage costs of an item for one month are as high as the value of the item itself. Nevertheless, for the most cases, *sales* will be equal to the minimum as the trade-off to generate a higher demand at some future time point is not profitable. We refer to this

reformulation as *INEQ*.

The INEQ reformulation can be extended such that it is an equivalent formulation to the min-expression. This is done by adding binary variables which are required to sum up to 1,

$$y_1 + y_2 + y_3 = 1, \tag{5.16a}$$
$$y_1, y_2, y_3 \in \{0,1\}. \tag{5.16b}$$

The shirt sales can then be determined as a *linear combination* of the y_i multiplied by the T_i,

$$x_{k+1}^{SA} = y_1 \cdot T_{k+1}^1 + y_2 \cdot T_{k+1}^2 + y_3 \cdot T_{k+1}^3, \tag{5.17}$$

and together with the inequality constraints on x_{k+1}^{SA}, we have

$$x_{k+1}^{SA} = y_1 \cdot T_{k+1}^1 + y_2 \cdot T_{k+1}^2 + y_3 \cdot T_{k+1}^3, \tag{5.18a}$$
$$x_{k+1}^{SA} \leq T_{k+1}^1, \tag{5.18b}$$
$$x_{k+1}^{SA} \leq T_{k+1}^2, \tag{5.18c}$$
$$x_{k+1}^{SA} \leq T_{k+1}^3, \tag{5.18d}$$
$$y_1 + y_2 + y_3 = 1, \tag{5.18e}$$
$$y_1, y_2, y_3 \in \{0,1\}, \tag{5.18f}$$

which is called the *LC* reformulation in the remainder of this section. Unfortunately, this reformulation violates the LICQ as for any feasible (integer) solution, two of the newly introduced constraints coincide. However, this is no problem in practice as we will see below.

An alternative formulation for the inequality constraints is

$$y_1 \cdot T_{k+1}^1 \leq (1 - y_2) \cdot T_{k+1}^2 \qquad y_1 \cdot T_{k+1}^1 \leq (1 - y_3) \cdot T_{k+1}^3 \tag{5.19a}$$
$$y_2 \cdot T_{k+1}^2 \leq (1 - y_1) \cdot T_{k+1}^1 \qquad y_2 \cdot T_{k+1}^2 \leq (1 - y_3) \cdot T_{k+1}^3 \tag{5.19b}$$
$$y_3 \cdot T_{k+1}^3 \leq (1 - y_1) \cdot T_{k+1}^1 \qquad y_3 \cdot T_{k+1}^3 \leq (1 - y_2) \cdot T_{k+1}^2 \tag{5.19c}$$

which together with the other constraints,

$$x_{k+1}^{SA} = y_1 \cdot T_{k+1}^1 + y_2 \cdot T_{k+1}^2 + y_3 \cdot T_{k+1}^3 \tag{5.20a}$$
$$y_1 \cdot T_{k+1}^1 \leq (1 - y_2) \cdot T_{k+1}^2 \qquad y_1 \cdot T_{k+1}^1 \leq (1 - y_3) \cdot T_{k+1}^3 \tag{5.20b}$$
$$y_2 \cdot T_{k+1}^2 \leq (1 - y_1) \cdot T_{k+1}^1 \qquad y_2 \cdot T_{k+1}^2 \leq (1 - y_3) \cdot T_{k+1}^3 \tag{5.20c}$$
$$y_3 \cdot T_{k+1}^3 \leq (1 - y_1) \cdot T_{k+1}^1 \qquad y_3 \cdot T_{k+1}^3 \leq (1 - y_2) \cdot T_{k+1}^2 \tag{5.20d}$$
$$y_1 + y_2 + y_3 = 1 \tag{5.20e}$$
$$y_1, y_2, y_3 \in \{0,1\} \tag{5.20f}$$

will be called *LCV*. Note that this variation uses vanishing constraints which are also known to violate LICQ.

A different approach is the application of *generalized disjunctive programming* (GDP) which was introduced by GROSSMANN et al., see [64, 65]. GDP describes an optimization problem class contain-

k	0	1	2	3	4	5	6	7	8	9	10
x_k^{RE}	0.79	0.53	0.42	0.37	0.35	0.34	0.34	0.34	0.34	0.34	0.34
x_k^{SQ}	0.75	1.996	2.274	2.412	2.48	2.52	2.532	2.54	2.544	2.546	2.548
x_k^{MO}	0.73	2.12	2.81	3.15	3.32	3.41	3.45	3.47	3.48	3.49	3.49

Table 5.1: Computation of upper bounds for variables x_k^{RE}, x_k^{SQ}, and x_k^{MO}.

ing disjunctions:

$$\begin{aligned} \min_x \quad & F(x) + \sum_k c_k \\ \text{s.t.} \quad & G(x) \le 0 \\ & \bigvee_{j \in J_k} \begin{bmatrix} Y_{jk} \\ G_{jk}(x) \le 0 \\ c_k = \gamma_{jk} \end{bmatrix}, k \in K \\ & \Gamma(Y) = \mathit{true}, Y_{jk} \in = \{\mathit{true}, \mathit{false}\} \\ & x \in \mathbb{R}^{n_x}, c_k \in \mathbb{R} \end{aligned} \tag{5.21}$$

Each option j of a disjunction k may contain different constraints G_{jk} and the boolean variables Y_{jk} indicate if the corresponding disjunction option is active, i.e., if the constraints have to be fulfilled. Disjunctions can be used to describe the minimum expression. As we have three terms of which one has to equal the *sales* (the others may equal also but must not, so we can assume w.l.o.g. that one option has to be active), we have one disjunction with three options,

$$\begin{bmatrix} Y_1 \\ T_{k+1}^1 \le T_{k+1}^2 \\ T_{k+1}^1 \le T_{k+1}^3 \\ x_{k+1}^{SA} = T_{k+1}^1 \end{bmatrix} \vee \begin{bmatrix} Y_2 \\ T_{k+1}^2 \le T_{k+1}^1 \\ T_{k+1}^2 \le T_{k+1}^3 \\ x_{k+1}^{SA} = T_{k+1}^2 \end{bmatrix} \vee \begin{bmatrix} Y_3 \\ T_{k+1}^3 \le T_{k+1}^1 \\ T_{k+1}^3 \le T_{k+1}^2 \\ x_{k+1}^{SA} = T_{k+1}^3 \end{bmatrix}, \tag{5.22}$$

with $Y_i \in \{\mathit{true}, \mathit{false}\}$ for $i = 1,2,3$.

5.3.3 Big M Relaxation of the GDP Reformulation

To be able to treat the GDP reformulation with the same MINLP solvers, we apply the *Big M* relaxation to transform the GDP part above in an MINLP:

$$T_{k+1}^1 - T_{k+1}^2 \le M \cdot (1 - y_1) \qquad T_{k+1}^1 - T_{k+1}^3 \le M \cdot (1 - y_1) \tag{5.23a}$$
$$T_{k+1}^2 - T_{k+1}^1 \le M \cdot (1 - y_2) \qquad T_{k+1}^2 - T_{k+1}^3 \le M \cdot (1 - y_2) \tag{5.23b}$$
$$T_{k+1}^3 - T_{k+1}^1 \le M \cdot (1 - y_3) \qquad T_{k+1}^3 - T_{k+1}^2 \le M \cdot (1 - y_3) \tag{5.23c}$$
$$x_{k+1}^{SA} - T_{k+1}^1 \le M \cdot (1 - y_1) \qquad -x_{k+1}^{SA} + T_{k+1}^1 \le M \cdot (1 - y_1) \tag{5.23d}$$
$$x_{k+1}^{SA} - T_{k+1}^2 \le M \cdot (1 - y_2) \qquad -x_{k+1}^{SA} + T_{k+1}^2 \le M \cdot (1 - y_2) \tag{5.23e}$$
$$x_{k+1}^{SA} - T_{k+1}^3 \le M \cdot (1 - y_3) \qquad -x_{k+1}^{SA} + T_{k+1}^3 \le M \cdot (1 - y_3) \tag{5.23f}$$
$$y_1 + y_2 + y_3 = 1 \tag{5.23g}$$
$$y_1, y_2, y_3 \in \{0,1\} \tag{5.23h}$$

Note that equality constraints need to be formulated as two inequalities in this context,

$$\left\{ \begin{array}{r} x_{k+1}^{SA} - T_{k+1}^{1} \leq M \cdot (1 - y_1) \\ -x_{k+1}^{SA} + T_{k+1}^{1} \leq M \cdot (1 - y_1) \end{array} \right\} \Leftrightarrow \left\{ x_{k+1}^{SA} - T_{k+1}^{1} = M \cdot (1 - y_1) \right\}. \tag{5.24}$$

Written as a vector, we have

$$g = \begin{pmatrix} T_{k+1}^{1} - T_{k+1}^{2} - M \cdot (1 - y_1) \\ T_{k+1}^{1} - T_{k+1}^{3} - M \cdot (1 - y_1) \\ T_{k+1}^{2} - T_{k+1}^{1} - M \cdot (1 - y_2) \\ T_{k+1}^{2} - T_{k+1}^{3} - M \cdot (1 - y_2) \\ T_{k+1}^{3} - T_{k+1}^{1} - M \cdot (1 - y_3) \\ T_{k+1}^{3} - T_{k+1}^{2} - M \cdot (1 - y_3) \\ x_{k+1}^{SA} - T_{k+1}^{1} - M \cdot (1 - y_1) \\ -x_{k+1}^{SA} + T_{k+1}^{1} - M \cdot (1 - y_1) \\ x_{k+1}^{SA} - T_{k+1}^{2} - M \cdot (1 - y_2) \\ -x_{k+1}^{SA} + T_{k+1}^{2} - M \cdot (1 - y_2) \\ x_{k+1}^{SA} - T_{k+1}^{3} - M \cdot (1 - y_3) \\ -x_{k+1}^{SA} + T_{k+1}^{3} - M \cdot (1 - y_3) \end{pmatrix}, \qquad \frac{\partial g}{\partial z} = \begin{pmatrix} M & 0 & 0 & 1 & -1 & 0 \\ M & 0 & 0 & 1 & 0 & -1 \\ 0 & M & 0 & -1 & 1 & 0 \\ 0 & M & 0 & 0 & 1 & -1 \\ 0 & 0 & M & -1 & 0 & 1 \\ 0 & 0 & M & 0 & -1 & 1 \\ M & 0 & 0 & -1 & 0 & 0 \\ M & 0 & 0 & 1 & 0 & 0 \\ 0 & M & 0 & 0 & -1 & 0 \\ 0 & M & 0 & 0 & 1 & 0 \\ 0 & 0 & M & 0 & 0 & -1 \\ 0 & 0 & M & 0 & 0 & 1 \end{pmatrix} \tag{5.25}$$

with $z = (y_1, y_2, y_3, T_{k+1}^{1}, T_{k+1}^{2}, T_{k+1}^{3})$ and thus, the Big M relaxation of the GDP fulfills the LICQ. We will refer to this reformulation as *GDP*. For optimization, we need a numerical value for M. Detailed computations are given in Appendix C. For $n_x = t_f - t_0 = 10$, we eventually get

$$T_{k+1}^{1} \in [6, 16545], \qquad T_{k+1}^{1} - T_{k+1}^{2} \in [-50661, 16539], \tag{5.26a}$$

$$T_{k+1}^{2} \in [6, 50667], \qquad T_{k+1}^{1} - T_{k+1}^{3} \in [-1654, 16241], \tag{5.26b}$$

$$T_{k+1}^{3} \in [304, 1660], \qquad T_{k+1}^{2} - T_{k+1}^{3} \in [-1654, 50363], \tag{5.26c}$$

and for the remaining constraints, with $x_k^{SA} \in [6, 1660]$,

$$T_{k+1}^{1} - x_{k+1}^{SA} \in [0, 16539], \tag{5.27a}$$

$$T_{k+1}^{2} - x_{k+1}^{SA} \in [0, 50661], \tag{5.27b}$$

$$T_{k+1}^{3} - x_{k+1}^{SA} \in [0, 1654], \tag{5.27c}$$

which means that we can chose, e.g., $M = 50661$.

5.3.4 Numerical Results

With these different reformulations, we conducted several computations. The initial values used for all computations with *IWR Tailorshop* in this chapter are given in Table 5.7. We always used $t_0 = 0$ and $t_f = 10$ unless otherwise stated and all computations were done using *Bonmin* via the AMPL interface of *IWR Tailorshop*.

In a first step, the *IWR Tailorshop* model has been used with separate controls for recruiting and dismissing employees, u^{dEM} and u^{DEM}. Parameters in this model differ from the ones given in Tables 4.4 and 4.5 only in the way that both models behave equivalent, but these changes can be neglected in this case. Computations using the u^{dEM}/u^{DEM}-model of *IWR Tailorshop* revealed that the different effects of u^{dEM} and u^{DEM} were too weak so that the model was almost singular with respect to

t_0	**GDP**		**LC**		**LCV**		**INEQ**	
9	0.6	181939.84	0.3	181939.82	0.8	181939.82	0.3	181939.82
8	10.2	189087.89	3.7	189087.83	18.2	189087.83	2.9	189087.83
7	134.1	196450.37	46.7	196450.28	179.3	196450.28	39.0	196450.28
6	2223.2	204033.73	760.8	204033.60	3051.8	204033.60	620.5	204033.60
5	>120h	213277.82	> 20h	213277.65	>20h	213277.65	>20h	213277.65
4	?	?	?	?	?	?	?	?
3	?	?	?	?	?	?	?	?
2	?	?	?	?	?	?	?	?
1	?	?	?	?	?	?	?	?
0	?	?	?	?	?	?	?	?
Σ	>720 h		>120 h		>120 h		>120 h	

Table 5.2: Computation times for different min-reformulations of *IWR Tailorshop* using the u^{dEM}/u^{DEM}-model: all approaches are far beyond the feasible scope. All approaches returned similar results. LCV seems to be the slowest, LC and INEQ are the fastest.

this variables (i.e., $u_k^{dEM} = 0$ and $u_k^{DEM} = 5$ was almost the same as $u_k^{dEM} = -5$ and $u_k^{DEM} = 10$, for instance). In Table 5.2, computation times for the reformulations and different t_0 are displayed. For $t_0 = 5$, computation times are far beyond the feasible scope for all approaches. LCV seems to be the slowest, followed by GDP, whereas LC and INEQ are dramatically faster and at about the same level. All approaches returned similar (often even the same) objective values, which is also the case for the remaining computations as we will see below.

In the next step, we introduced a small penalty on u^{dEM} and u^{DEM} in the *capital* equation. Computation times for this modification are given in Table 5.3. For all approaches, the computation times are drastically lower than before and it is possible to solve the ITOP with this model. INEQ is the fastest approach, closely followed by LC. GDP and LCV are far behind and because of the theoretical disadvantages of LCV, this approach has been discarded. Times for LC are already in an acceptable region for optimization-based feedback computation, i.e., ITFP.

A shift towards a single control u^{EM} for the *employees* in the model yielded even lower computation times, see Table 5.4. INEQ is the fastest approach again, closely followed by LC. Both INEQ and LC are fast enough for optimization-based feedback. GDP falls apart and is too slow for online feedback computation.

In August 2012, within the scope of the *International Symposium on Mathematical Programming*, the *Mathematical Optimization Society* hosted a *Klaus Tschira Workshop* on mathematical optimization for high school students and teachers organized by ARMIN FÜGENSCHUH. In one of the practical exercises, students were asked to solve the *IWR Tailorshop* via an early version of its web interface (optimization-based feedback was not yet implemented). Most of the participants played several rounds of 10 months each. This data has later been used together with results obtained during software testing as a test set for the optimization according to ITAP: the states derived by the participants were used as initial values for the optimization.

Table 5.5 shows average computation times for each month for this test set together with the percentage of problems which could be solved using the different reformulations. Both GDP and INEQ were able to solve all but one of the 1593 problems. The average computation times confirm the previous results: INEQ is the fastest, LC closely behind, and GDP far behind. LC is also fast enough for

t_0	**GDP**		**LC**		**LCV**		**INEQ**	
9	0.4	181939.94	0.2	181908.92	0.2	181908.92	0.1	181908.92
8	2.5	189056.06	0.4	189056.00	2.4	189056.00	0.3	189056.00
7	6.5	196417.59	2.2	196417.49	5.5	196417.49	0.8	196417.49
6	23.9	203999.96	4.2	203999.83	18.1	203999.83	2.2	203999.83
5	1194.6	213245.62	16.8	213245.46	64.9	213245.46	9.1	213245.46
4	434.5	224031.37	40.0	224031.17	3502.6	224031.17	22.8	224031.17
3	787.6	237150.46	62.6	237150.46	255.9	237133.07	11.9	237150.46
2	1423.0	251604.12	66.9	251604.12	805.0	251585.46	10.9	251604.12
1	3231.2	269315.38	79.4	269315.30	2153.8	269044.70	19.2	269315.30
0	7358.1	290367.51	75.2	290367.51	10222.6	290088.79	28.0	290367.51
Σ	ca. 14500 s		ca. 350 s		ca. 17000 s		ca. 110 s	

Table 5.3: Computation times for different min-reformulations of *IWR Tailorshop* with a small penalty on u^{dEM}/u^{DEM}: dramatically decreased times for all approaches. INEQ is the fastest, closely followed by LC. GDP and LCV are far behind.

t_0	**GDP**		**LC**		**INEQ**	
9	0.3	181939.84	0.1	181939.82	0.1	181939.82
8	2.8	189087.89	0.3	189087.83	0.2	189087.83
7	7.0	196450.37	1.4	196450.28	0.5	196450.28
6	30.8	204033.73	3.1	204033.60	1.3	204033.60
5	66.2	213277.82	8.8	213277.65	4.3	213277.65
4	153.4	224063.71	16.4	224063.71	8.2	224063.71
3	434.9	237243.32	26.6	237243.32	7.2	237243.32
2	398.7	251465.31	21.5	251746.64	8.0	251746.64
1	1805.6	269491.15	53.8	269491.15	14.0	269491.15
0	7279.3	290583.98	103.1	290583.98	21.1	290583.98
Σ	ca. 10200 s		ca. 240 s		ca. 70 s	

Table 5.4: Computation times for different min-reformulations of *IWR Tailorshop* using the u^{EM}-model: times are even lower. INEQ is the fastest again, closely followed by LC. GDP falls apart and is too slow for optimization-based feedback.

t_0	GDP			LC			INEQ		
	Time	Problems solved		Time	Problems solved		Time	Problems solved	
1	2461.82	177	100.0%	65.41	173	97.7%	13.43	177	100.0%
2	1013.53	176	99.4%	47.13	173	97.7%	8.55	176	99.4%
3	478.91	177	100.0%	27.27	174	98.3%	6.70	177	100.0%
4	206.85	177	100.0%	15.90	173	97.7%	6.24	177	100.0%
5	50.82	177	100.0%	7.39	176	99.4%	2.88	177	100.0%
6	17.99	177	100.0%	3.17	175	98.9%	1.30	177	100.0%
7	5.48	177	100.0%	1.41	176	99.4%	0.60	177	100.0%
8	1.75	177	100.0%	0.50	177	100.0%	0.22	177	100.0%
9	0.24	177	100.0%	0.11	177	100.0%	0.06	177	100.0%
Total	4234.33	1592	99.9%	166.81	1574	98.8%	39.97	1592	99.9%

Table 5.5: Average computation times for each month for a set of 177 datasets collected via an early version of the *IWR Tailorshop* web interface.

optimization-based feedback. However, LC ran into infeasibilities for some problems which could be prevented by setting the iteration limit for the NLP solver in the *branch and bound* algorithm to a lower number, see Table 5.6.

From Table 5.8, one can determine how much problems (i.e., optimization starting at some month) and datasets (i.e., all optimization problems for $t_s = 0, \ldots, t_f$) could be processed successfully by each approach in a given time limit. This is important in particular, because we consider a waiting time of more than 180 s per month as unacceptable for participants. If participants have to wait too long, they probably will not finish the task and thus produce incomplete datasets which are almost useless for the optimization-based analysis. We can conclude that LC seems to be the method of choice in this context and thus has been implemented for the web-based study described in Chapter 6.

5.4 A Tailored Decomposition Approach

The systematically built microworld *IWR Tailorshop* with desirable properties could now be used for studies, evaluating participants' performance based on optimal solutions as explained in Section 5.1. For the computation of an indicator function, however, one would want to use guaranteed *globally* optimal solutions because then we can be sure to get an objective indicator function. Fortunately, the optima found by *Bonmin* are good enough such that participants are not able to find better solutions for *IWR Tailorshop* and therefore the solutions found by *Bonmin* were used for analysis and training in the study in Chapter 6. However, globally optimal solutions or at least upper bounds on the global solutions would be even better. But—as already mentioned above—the *IWR Tailorshop* yields a nonconvex problem. This property is unavoidable as long as we are interested in turn-based scenarios with nonlinear model equations. Hence, it is difficult to compute global solutions for such test-scenarios.

And indeed, the computation times with *Couenne* on a Intel Core i7 machine with 12 GB RAM look bad: for $n_x = 1$ it takes less than 1 sec, for $n_x = 2$ already 3 sec, and for $n_x = 3$ by far more than 10 min (see also Table 5.10). For higher values of n_x, we cannot hope for a solution at all before the machine runs out of memory.

t_0	LC			LC, iter. limit		
	Time	Problems solved		Time	Problems solved	
1	65.41	173	97.7%	75.01	177	100.0%
2	47.13	173	97.7%	54.99	177	100.0%
3	27.27	174	98.3%	30.97	177	100.0%
4	15.90	173	97.7%	15.98	177	100.0%
5	7.39	176	99.4%	7.67	177	100.0%
6	3.17	175	98.9%	3.23	177	100.0%
7	1.41	176	99.4%	1.49	177	100.0%
8	0.50	177	100.0%	0.52	177	100.0%
9	0.11	177	100.0%	0.11	177	100.0%
Total	166.81	1574	98.8%	189.99	1593	100.0%

Table 5.6: Average computation times for each month for a set of 177 datasets collected via an early version of the *IWR Tailorshop* web interface: LC shows some problems with infeasibilities which can be solved by setting a lower iteration limit for the NLP solver of the *branch and bound* algorithm.

State	Value
x^{EM}	10
x^{PS}	1
x^{DS}	1
x^{SH}	67.00
x^{PR}	200.00
x^{SA}	200.00
x^{DE}	700.00
x^{RE}	0.79
x^{SQ}	0.75
x^{MQ}	0.81
x^{MO}	0.73
x^{CA}	175,000.00

Table 5.7: Initial values used for computations with *IWR Tailorshop* in this chapter.

Mathematical model reduction techniques are quite common in other domains, see e.g., [18, 9, 115] for an overview. The basic idea of our new approach to solve problem (3.17) consists of a decomposition of the MINLP into a *master* and several smaller *subproblems*. This works if the objective function is separable. The idea is related to *Lagrangian relaxation*, one of the most used relaxation strategies for *MILPs*. Its first application was the one-tree relaxation of the traveling salesman problem in the famous *Held-Karp algorithm* in [69, 70]. The traditional application fields are variants of the knapsack problem like, e.g., facility location and capacity planning [99], general assignment, network flow and the unit commitment problem [88]. The general approach is thoroughly explained in [60] and in [81]. A problem-specific decomposition approach has been proposed in [33]. The authors reformulate the MIOCP as a large-scale, structured nonlinear program (NLP) and solve a small scale linear integer program on a second level to approximate the calculated continuous aggregated output of all pumps in a water works. To obtain objective performance measures, we need guaranteed upper bounds for the maximum. Hence the mentioned techniques can not be applied in a straightforward way.

In the following, we will develop a decomposition approach tailored to the *IWR Tailorshop* microworld (but not made from fabric, though). The idea of the decomposition approach is to exploit the structure of the problem—especially the separability of the objective function or the variable *capital* respectively, see (4.93)—to create a relaxation of the original problem where parts of the problem are replaced by free variables (i.e., free within some simple bounds), for which costs are computed in *decoupled programs*, which contain the complexity from the original program. A schematic representation of this decomposition can be found in Figure 5.9. The decoupling of certain parts of the original problem obviously makes the remaining *master problem* smaller and therefore easier to handle. This approach has been published in [47].

Such a decomposition is not unique. We chose one with few overlapping variables. A schematic

Time limit	**GDP**		**LC**		**LC, iter. limit**		**INEQ**	
	Prob.	Datas.	Prob.	Datas.	Prob.	Datas.	Prob.	Datas.
60 s	760	177	159	118	144	99	1	1
90 s	684	177	40	31	28	22	1	1
120 s	655	177	35	28	18	14	1	1
180 s	576	177	28	23	13	11	1	1
240 s	532	177	26	21	10	8	1	1
300 s	496	177	25	21	9	7	1	1
600 s	356	176	20	17	4	4	1	1
1200 s	229	166	19	16	0	0	1	1
3600 s	45	6	19	16	0	0	1	1

Table 5.8: Number of problems (i.e., months) and corresponding datasets in the test set, which could not be solved within a given time limit by the different reformulation approaches.

representation of the resulting master problem is shown in Figure 5.6. The costs computation via the decoupled problems is done *offline* on a discretized grid. The decoupled problems themselves yield an optimization problem of the type

$$\begin{aligned} \min \quad & \text{costs} && \text{(5.28a)} \\ \text{s.t.} \quad & \text{achieve desired value of free variable} \\ & \text{(as in master problem)} && \text{(5.28b)} \end{aligned}$$

The optimal solutions on the grid points can be used to fit some model, which underestimates the costs, details can be found below. This cost model is now plugged into the objective function of the master problem representing *costs* for the newly introduced free variables. We then can compute a globally optimal solution for the reduced master problem. If the relaxation is valid, this approach yields a valid upper bound for the original problem. The upper bound determined by the decomposition can then be used as an indicator how far a local solution for the original problem is away at the most from a global one.

To compare the problem sizes, we consider the time-discrete state and control variables as single variables. Then, by the decomposition, the problem size has been reduced from $12 \cdot n_x$ state variables and $11 \cdot (n_x - 1)$ control variables to $4 \cdot n_x + 3 \cdot (n_x - 1)$ free variables and $5 \cdot n_x$ states with 2 decoupled problems.

The master problem in our decomposition consists of the following equations, which form a relaxation of the original problem (4.90) by underestimating negative and overestimating positive effects:

$$x_{k+1}^{DE} = p^{DE,0} \cdot \exp\left(-p^{DE,1} \cdot u_k^{SP}\right) \cdot \log\left(p^{DE,2} \cdot u_k^{AD} + 1\right) \cdot \left(x_k^{RE} + p^{DE,3}\right) \tag{5.29a}$$

$$x_{k+1}^{RE} = p^{RE,0} \cdot x_k^{RE} + p^{RE,1} \log\left(p^{RE,2} \cdot u_k^{AD} + p^{RE,3} \cdot u_k^{SP} \cdot (u_k^{SQ})^2 + p^{RE,4} \cdot u_k^{WA} + 1\right) \tag{5.29b}$$

$$x_{k+1}^{SA} = \min\left\{p^{SA,0} \cdot u_{k+1}^{sites} \cdot \log\left(\frac{p^{SA,1} \cdot u_{k+1}^{EM}}{u_{k+1}^{sites} + p^{SA,2}} + 1\right); x_k^{SH} + u_{k+1}^{PR}; p^{SA,3} \cdot x_{k+1}^{DE}\right\} \tag{5.29c}$$

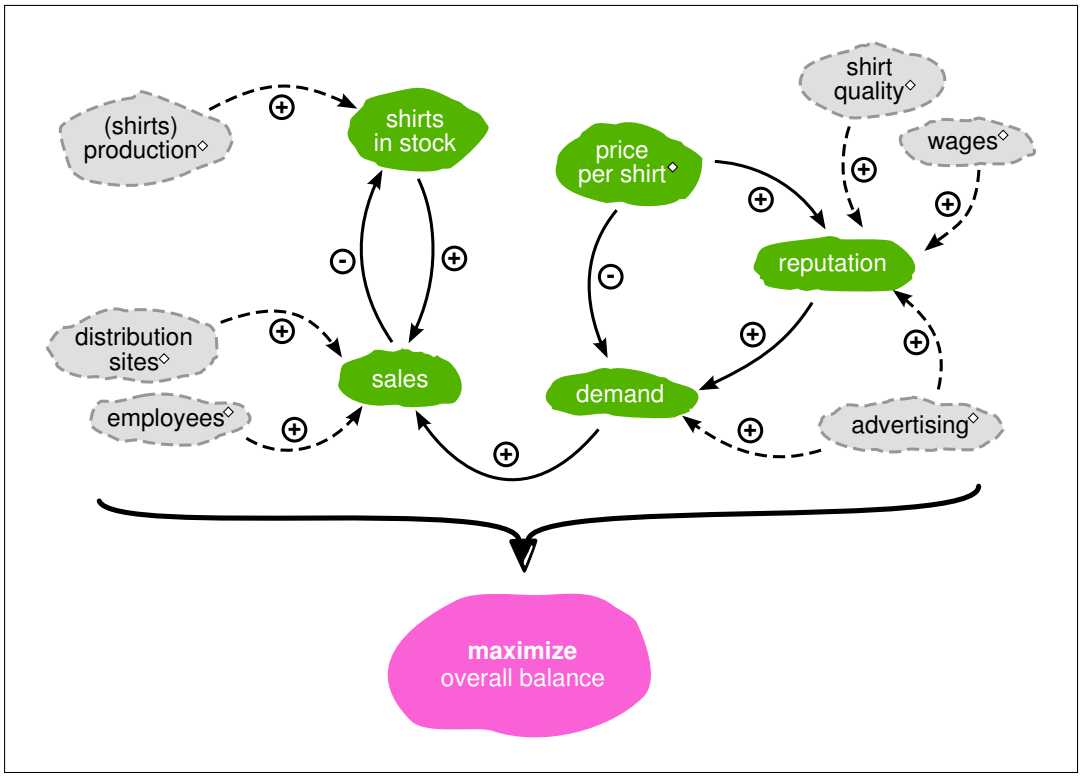


Figure 5.6: *IWR Tailorshop* reduced *master problem* with dependencies and proportional/reciprocal influences. Diamonds indicate free variables.

$$x_{k+1}^{SH} = x_k^{SH} - x_{k+1}^{SA} + u_{k+1}^{PR} \tag{5.29d}$$

$$\begin{aligned} x_{k+1}^{CA} = p^{CA,0} \cdot \Big(& x_k^{CA} + \left(x_{k+1}^{SA} \cdot u_k^{SP} \right) - u_k^{AD} - u_{k+1}^{EM} \cdot u_k^{WA} - \left(x_{k+1}^{SH} \cdot p^{CA,6} \right) \\ & - f_1(u_k^{sites}; u_k^{PR}, u_k^{EM}) - f_2(u_k^{SQ}; u_k^{PR}) \Big) \end{aligned} \tag{5.29e}$$

$$u_k^{SP} \in \left[lb^{SP}, ub^{SP} \right] \qquad u_k^{SQ} \in \left[lb^{SQ}, ub^{SQ} \right] \tag{5.29f}$$

$$u_k^{PR} \in \left[lb^{PR}, ub^{PR} \right] \qquad u_k^{WA} \in \left[lb^{WA}, ub^{WA} \right] \tag{5.29g}$$

$$u_k^{sites} \in \left[lb^{sites}, ub^{sites} \right] \cap \mathbb{Z}_0^+ \qquad u_k^{AD} \in \left[lb^{AD}, ub^{AD} \right] \tag{5.29h}$$

$$u_k^{EM} \in \left[lb^{EM}, ub^{EM} \right] \cap \mathbb{Z}_0^+ \tag{5.29i}$$

Here, we denote lower and upper bounds on the free variables by lb and ub, respectively which can be found in Table 5.9 together with initial values for the decomposition. The functions f_1 and f_2 return the costs to be determined in the decoupled problems. We choose the objective again as

$$\max_{x,u,p} \quad x_N^{CA}. \tag{5.30}$$

Original model	**Decomposition**
$x_0^{EM} = 10$	$u_0^{EM} = 10$
$x_0^{PS} = 1$	$u_0^{sites} = 2$
$x_0^{DS} = 1$	
$x_0^{SH} = 67$	$x_0^{SH} = 67$
$x_0^{PR} = 200$	$u_0^{PR} = 200$
$x_0^{SA} = 200$	$x_0^{SA} = 200$
$x_0^{DE} = 700$	$x_0^{DE} = 700$
$x_0^{RE} = 0.79$	$x_0^{RE} = 0.79$
$x_0^{SQ} = 0.75$	$u_0^{SQ} = 0.75$
$x_0^{MQ} = 0.81$	—
$x_0^{MO} = 0.73$	—
$x_0^{CA} = 175000$	$x_0^{CA} = 175000$

(a) Initial values

Original model	**Decomposition**
$u_k^{SP} \in [35,55]$	$u_k^{SP} \in [35,55]$
$u_k^{AD} \in [1000,2000]$	$u_k^{AD} \in [1000,2000]$
$u_k^{WA} \in [1000,1500]$	$u_k^{WA} \in [1000,1500]$
$u_k^{MA} \in [0,5000]$	$u_k^{MA} \in [0,5000]$
$x_k^{EM} \in [8,16]$	$u_k^{EM} \in [8,16]$
$x_k^{PS}, x_k^{DS} \in [1,6]$	$u_k^{sites} \in [2,6]$
$x_k^{PR} \in [0,1000]$	$u_k^{PR} \in [0,1000]$
$x_k^{SQ} \in [0.25,0.75]$	$u_k^{SQ} \in [0.25,0.75]$
$x_k^{SH}, x_k^{DE}, x_k^{RE}, x_k^{SA} \geq 0$	$x_k^{SH}, x_k^{DE}, x_k^{RE}, x_k^{SA} \geq 0$
$x_k^{MO}, x_k^{MQ} \geq 0$	—

(b) Simple bounds

Table 5.9: Initial values and simple bounds used for computations with original full problem and decomposition.

The first decoupled program, which determines the costs for a given *shirt quality*, is

$$\begin{aligned}
\min \quad & u_k^{RQ} \cdot \widehat{u_{k+1}^{PR}} \cdot p^{PR,cost} + u_{k-1}^{MA} && \text{(5.31a)}\\
\text{s.t.} \quad & \widehat{u_k^{SQ}} = p^{SQ,1} \cdot x_k^{MQ} + p^{SQ,2} \cdot u_k^{RQ} && \text{(5.31b)}\\
& x_k^{MQ} = p^{MQ,3} \cdot \log\left(p^{MQ,4} \cdot u_{k-1}^{MA} + 1\right) && \text{(5.31c)}\\
& u_k^{RQ} \in \left\{p^{RQ,1}, \ldots, p^{RQ,n_{RQ}}\right\} && \text{(5.31d)}\\
& u_{k-1}^{MA} \in \left[lb^{MA}, ub^{MA}\right] && \text{(5.31e)}
\end{aligned}$$

Here, the variables with a *hat* are considered to be given, e.g., from the *free variables* in the master problem. In the following, we call them *input variables* in this context. The second subproblem determines the costs for a given total number of *sites* and consists of the following equations.

$$\begin{aligned}
\min \quad & u_{k+1}^{DS} \cdot p^{CA,5} + u_{k+1}^{PS} \cdot p^{CA,4} && \text{(5.32a)}\\
\text{s.t.} \quad & \widehat{u_{k+1}^{sites}} = u_{k+1}^{PS} + u_{k+1}^{DS} && \text{(5.32b)}\\
& \widehat{u_{k+1}^{PR}} = p^{PR,0} \cdot \log\left(u_{k+1}^{PS} \cdot \frac{p^{PR,1} \cdot \widehat{u_{k+1}^{EM}}}{u_{k+1}^{PS} + u_{k+1}^{DS} + p^{PR,2}} + 1\right) && \text{(5.32c)}\\
& u_{k+1}^{DS} \in \left[lb^{DS}, ub^{DS}\right] \cap \mathbb{Z}_0^+ && \text{(5.32d)}\\
& u_{k+1}^{PS} \in \left[lb^{PS}, ub^{PS}\right] \cap \mathbb{Z}_0^+ && \text{(5.32e)}
\end{aligned}$$

We evaluate these decoupled programs on a grid, i.e., on a discretization of the feasible interval for each *input variable*. For $u_k^{sites} \in [2,16]$, e.g., we could choose the grid $2,4,8,10,12,14,16$. With more than one discretized variable, this leads to multidimensional grids. For each grid point, we compute

an optimal solution for the corresponding decoupled program. With the solutions for all grid points, we can fit, e.g., a quadratic model, like

$$f(u_k^{SQ}; u_k^{PR}) = a_0 + a_1 \cdot u_k^{PR} + a_2 \cdot u_k^{SQ} + a_3 \cdot u_k^{PR} \cdot u_k^{SQ} + a_4 \cdot (u_k^{PR})^2 + a_5 \cdot (u_k^{SQ})^2. \tag{5.33}$$

Of course, we could as well use a linear or a cubic model or something completely different. The fit can then be done by solving a simple least squares problem, with X being the set of grid points and $h(x)$ a function, which returns the optimal objective value for each grid point $x \in X$:

$$\min_{a,x} \quad \sum_{x \in X} \| f(x) - h(x) \|_2^2 \tag{5.34a}$$

$$\text{s.t.} \quad f(x) \leq h(x) \quad \forall x \in X. \tag{5.34b}$$

Especially when considering the integrality conditions, equality constraints are unlikely to be fulfilled exactly. Therefore the following reformulation is introduced for each equality constraint.

$$\widehat{u_k} = \ldots \quad \longrightarrow \quad \widehat{u_k} + \epsilon = \ldots \qquad \text{with} \quad \epsilon \in [-\rho, \rho] \tag{5.35}$$

Here, ρ should be chosen reasonably small, such that the decoupled program is feasible for almost all of the grid points.

t_0	Original model			Decomposition
	Ipopt	**Bonmin**	**Couenne**	**Couenne**
9	$\ll$ 1 s	< 1 s	< 1 s	<1 s
8	$\ll$ 1 s	4 s	3 s	1 s
7	< 1 s	45 s	> 10 min	2 s
6	< 1 s	537 s	> 10 min	3 s
5	< 1 s	> 10 min	> 10 min	5 s
4	< 1 s	> 10 min	> 10 min	10 s
3	1 s	> 10 min	> 10 min	17 s
2	< 1 s	> 10 min	> 10 min	27 s
1	< 1 s	> 10 min	> 10 min	52 s
0	1 s	> 10 min	> 10 min	88 s

Table 5.10: Comparison of computation times between *Ipopt*, *Bonmin*, and *Couenne* for the original problem, as well as *Couenne* for the decomposition.

We present results of our decomposition approach for the *IWR Tailorshop*. All computations have been done on an Intel Core i7 machine with 12 GB RAM running *Ubuntu 11.10 (64-bit)* with the *COIN-OR* solvers *Ipopt*, *Bonmin*, and *Couenne*. Remember that *Ipopt* is not able to treat integer constraints and has only been used for reference. For the computations in this article, an NLP-based *branch-and-bound* algorithm, has been used in *Bonmin*. Initial values and simple bounds on states and controls used in all computations can be found in Tables 5.9a and 5.9b. All computations for the original model refer to the two control version for employees. The parameter sets used are mainly the ones from Tables 4.4 and 4.5, but adapted to the u^{dEM}/u^{DEM} and with $p^{DE,0} = 600.0$ shirts. This is the reason for the differences compared to Section 5.3.

For the decomposition, in a first step the cost functions f_1 and f_2 for the new free variables u_k^{SQ} and u_k^{sites} have been computed. Therefore the subproblems (5.31) and (5.32) have been solved on the grids

$$u_k^{SQ} \in \{0.25, 0.26, 0.27, \ldots, 0.74, 0.75\}, \tag{5.36a}$$
$$u_k^{PR} \in \{100, 200, 300, \ldots, 900, 1000\}, \tag{5.36b}$$

respectively

$$u_k^{sites} \in \{2, 3, 4, 5, 6\}, \tag{5.37a}$$
$$u_k^{EM} \in \{8, 9, 10, \ldots, 15, 16\}, \tag{5.37b}$$
$$u_k^{PR} \in \{100, 200, 300, \ldots, 900, 1000\}. \tag{5.37c}$$

By solving the corresponding problems of type (5.34) with this data, we received the following underestimators for the costs:

$$\begin{aligned} f_1(u_k^{sites}; u_k^{EM}, u_k^{PR}) = {} & 21.6754 - 944.6455 \cdot u_k^{sites} + 1.4968 \cdot u_k^{PR} - 28.9341 \cdot u_k^{EM} \\ & + 0.1338 \cdot u_k^{sites} \cdot u_k^{PR} - 3.3626 \cdot u_k^{sites} \cdot u_k^{EM} - 0.0586 \cdot u_k^{PR} \cdot u_k^{EM} \\ & - 1.3478 \cdot (u_k^{sites})^2 + 1.8831 \cdot (u_k^{EM})^2 \end{aligned} \tag{5.38a}$$

$$\begin{aligned} f_2(u_k^{SQ}; u_k^{PR}) = {} & -898.0761 + 0.1991 \cdot u_{k+1}^{PR} + 4726.3749 \cdot u_{k+1}^{SQ} - 8.5390 \cdot u_{k+1}^{PR} \cdot u_{k+1}^{SQ} \\ & + 0.0004 \cdot (u_{k+1}^{PR})^2 - 5501.7182 \cdot (u_{k+1}^{SQ})^2 \end{aligned} \tag{5.38b}$$

The problems for all grid points of one subproblem could be solved in less than 1 min including the fit of the quadratic model. A plot of the resulting cost function for the u^{SQ}-subproblem can be found in Figure 5.7. However, it was necessary to use the *global* solver *Couenne* at least in this subproblem, as we got different solutions with *Ipopt* for a relaxed version of this subproblem which obviously are not globally optimal as one can observe from the comparison to the solutions of *Couenne* in Figure 5.8. For the u^{sites}-subproblem a plot of the cost function is not possible due to its dimensions.

When comparing solutions and objective function values, three effects need to be distinguished: integrality, local vs. global solutions, and full versus overestimating reduced model. We investigated two scenarios. First, the variables u_k^{sites} respectively u_k^{PS} and u_k^{DS} have been fixed to their lower bounds 2 respectively 1. The results are listed in table 5.11. Here, *Ipopt* and *Bonmin* found the same solutions for the original problem, which is due to the fact that the solutions determined by *Ipopt* are already integral. Thus, there is no difference between these solvers. In this special case, *Couenne* also finds the same solutions for the original problem in an acceptable time (< 1 min). This setting allows us to focus exclusively on the third effect, the gap between our reduced and the full model. The gap determined by *Couenne* in both cases reaches from 4.0% to 16.3%.

If we let u_k^{sites} free within their simple bounds as shown in Table 5.9b, the gaps between local solution to the full model and global solution to the reduced model alternate from 4.0% to 8.1%. Note that the gap relating to *Ipopt* is only for reference, since *Ipopt* cannot handle integer constraints and thus solves a relaxed version of the problem. One observes that the gap first increases, but then decreases, seeming to converge to some $c > 0$. This behavior can be explained by the fact that the mentioned effect leads to an increase in cost (due to storage of not-sold shirts) that is about linear in the number of turns. The possible winnings making use of a free choice of u_k^{sites} outperform these additional costs if the time scale for the optimization is long enough. Thus, the gap first increases and than

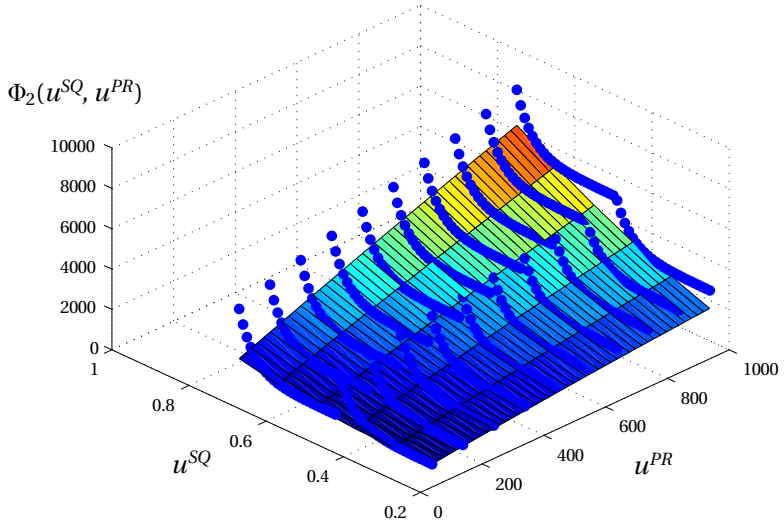


Figure 5.7: Cost values Φ_2 (blue dots) for solutions by *Couenne* for the decoupled problem for u^{SQ} with $p^{RQ,n_{RQ}} = 2$ on the grid $u_k^{SQ} \in \{0.25, 0.26, \ldots, 0.75\}$, $u_k^{PR} \in \{100, 200, \ldots, 1000\}$ together with the underestimating cost function (colored surface).

again decreases.

For this general case, *Couenne* is not able anymore to find a solution for the original problem in less than 10 min for $t_0 \leq 7$. All computation times can be found in Table 5.10. Obviously, the decomposition can be solved faster by orders of magnitude. Even for $t_0 = 0$, it takes less than 2 min with *Couenne*, while *Bonmin* is not able to compute a local solution for the u^{dEM}/u^{DEM} version of the original problem in less than 10 min for $t_0 \leq 5$.

Summing up, we could estimate the gap between reduced and full model to be in the range of a few percent. For longer time horizons and more freedom of variable choice, however, our approximation becomes better and better. The computational gains are dramatic and allow to calculate global solutions even on the full length of the time horizon.

5.5 Model Parameter Optimization

The *IWR Tailorshop* model parameters presented in Chapter 4 have been chosen by hand such that the model exhibits a desired behavior and solutions of ITOP and ITAP contain necessary interventions, i.e., not all controls are on their lower or upper bound in all the months. In Section 4.4, we already discussed the importance of the parameter set. It is well known for models of, e.g., chemical reactions (e.g., [23]) that derivative-based optimization methods should be used to estimate model parameters to significantly improve the model's performance in representing the real world process (and if there is no feasible parameter set to explain the data, the model might be wrong). However, in the case of complex microworlds, there usually is no real world process to model and thus there is no data the parameters can be estimated from. Nevertheless, model parameter optimization may be done for complex microworlds following a different approach.

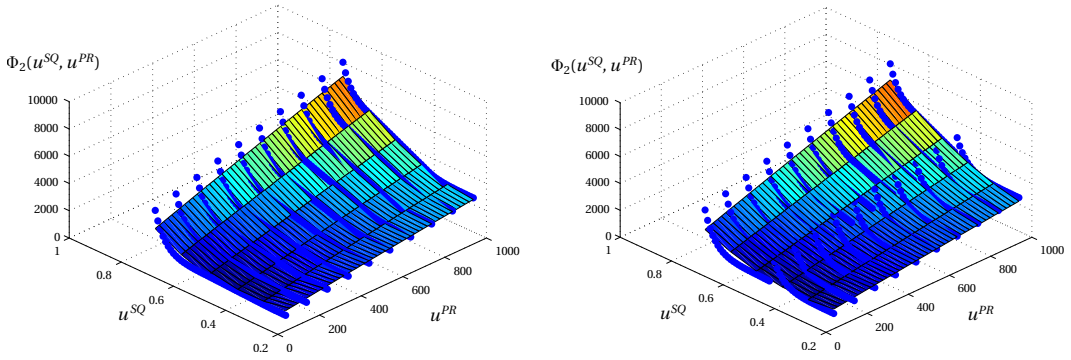
For a subset of the model parameters, a two-level optimization problem can be solved with a problems similar to the ITOP on the lower level and an upper level optimization of the parameters subject to the lower level objective function values. The aim of this parameter optimization could be, e.g.,

t_0	Original model		Decomposition	Gap in %
	Ipopt	Bonmin	Couenne	
9	180995.1	180995.1	188495.0	4.0 %
8	187170.0	187170.0	198599.3	5.8 %
7	193530.2	193530.2	209006.8	7.4 %
6	200081.2	200081.2	219726.5	8.9 %
5	206828.8	206828.8	230767.7	10.4 %
4	213778.7	213778.7	242140.2	11.7 %
3	220937.2	220937.2	253853.9	13.0 %
2	228310.4	228310.4	265919.0	14.1 %
1	235904.8	235904.8	278346.0	15.2 %
0	243727.0	243727.0	291145.9	16.3 %

Table 5.11: Solutions using the full problem with fixed number of sites compared to the decomposition approach. Note that the solutions by *Ipopt* are already integer, so that there is no difference between *Bonmin* and *Ipopt*.

t_0	Original model				Decomposition
	Ipopt	Gap in %	Bonmin	Gap in %	Couenne
9	181835.6	3.5%	180995.1	4.0%	188495.0
8	189161.4	4.8%	187170.0	5.8%	198599.3
7	196180.0	6.1%	193530.2	7.4%	209006.8
6	204760.9	6.8%	201860.5	8.1%	219726.5
5	215097.9	6.8%	212332.9*	8.0%	230767.7
4	226408.7	6.5%	223118.0*	7.9%	242140.2
3	239011.7	5.8%	236196.6*	7.0%	253853.9
2	252536.7	5.0%	250100.3*	6.0%	265919.0
1	266817.6	4.1%	264399.8*	5.0%	278346.0
0	281619.2	3.3%	279119.3*	4.1%	291145.9

Table 5.12: Solutions using the full problem compared to the decomposition approach. For solutions with a *, *Bonmin* did not find an optimal solution within 10 min. However, the gap between lower and upper bound was in all cases significantly below 1%.



(a) Couenne **(b)** Ipopt

Figure 5.8: Cost values Φ_2 (blue dots) for solutions by *Couenne* and *Ipopt* for the decoupled problem for u^{SQ} with $p^{RQ,n_{RQ}} = 2$ and relaxed u^{RQ} on the grid $u_k^{SQ} \in \{0.25, 0.26, \ldots, 0.75\}$, $u_k^{PR} \in \{100, 200, \ldots, 1000\}$ together with the underestimating cost function (colored surface). From the differences between *Couenne* (global solver) and *Ipopt* (local solver) one can determine that it is necessary here to use a global solver for the decoupled problem.

to guarantee a comparable result for the application of different strategies in the microworld—for instance, a *low* and a *high price* strategy, see Figure 5.10. In the lower level problem, some of the control variables will then be fixed to a given strategy while others remain free. In the case of two strategies, one could think of minimizing the difference between the two lower level problems while parameters are required to be within reasonable bounds.

Definition 5.7 The *IWR Tailorshop parameter optimization problem* (ITPOP) is the multilevel problem

$$\begin{aligned} \min_{p} \quad & \Psi(p) \\ \text{s.t.} \quad & p \in \Pi \subset \mathbb{R}^{n_p} \end{aligned} \tag{5.39}$$

with, e.g., the objective function

$$\Psi(p) = \sum_i \alpha_i \cdot \Phi_i(p) \tag{5.40}$$

which contains objective values $\Phi_i(p)$ for $i \in \{0, 1\}$ of an underlying optimization problem similar to the ITOP,

$$\begin{aligned} \Phi_i(p) := \min_{x,u} \quad & F(x, u, p) \\ \text{s.t.} \quad & x_{k+1} = G(x_k, u_k, p), && k = t_0, \ldots, t_f - 1, \\ & 0 \leq H(x_k, u_k, p), && k = t_0, \ldots, t_f, \\ & u_k \in \Omega, && k = t_0, \ldots, t_f - 1, \\ & u_k^{(j)} = \hat{u}_k^{(j)}, && j \in U_i, k = t_0, \ldots, t_f - 1, \\ & x_{t_0} = x_0, \end{aligned} \tag{5.41}$$

with F, G, H, Ω, x, p, u, t_0, and t_f defined as before.

Note that in (5.41), we fix some of the controls with the constraint

$$u_k^{(j)} = \hat{u}_k^{(j)}, \quad j \in U_i, k = t_0, \ldots, t_f - 1. \tag{5.42}$$

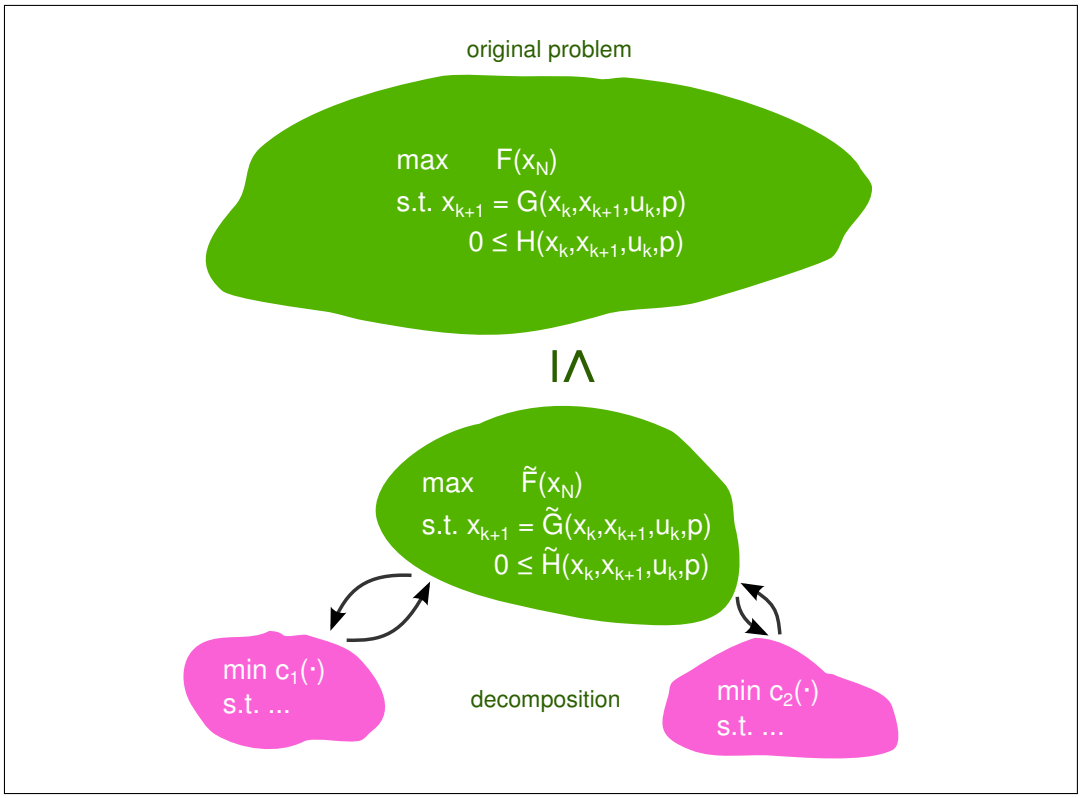


Figure 5.9: Schematic representation of the tailored decomposition approach

For two competing strategies, one might choose the upper level objective function

$$\Psi(p) = \Phi_0(p) - \Phi_1(p), \tag{5.43}$$

but another weighting might be applicable as well. One might even think of choosing other objective functions than linear combinations of the lower level objectives. However, without implementing advanced techniques, the lower level problems have to be considered as black boxes in the upper level problem. Thus, the upper level problem needs to be solved by derivative-free optimization methods. As the capabilities of these methods are very limited, the amount of parameters to optimize should be rather low. The same is true for the number of lower level problems, as they all have to be solved for a single evaluation of $\Psi(p)$.

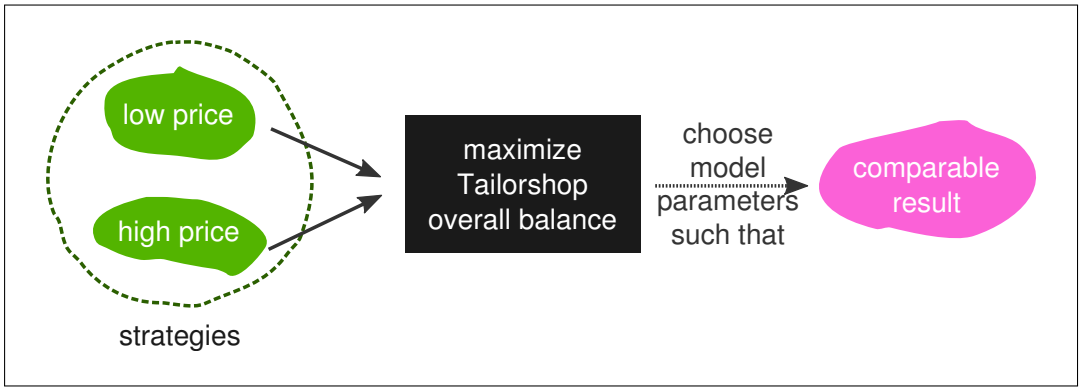


Figure 5.10: Parameter optimization for different strategies: some of the control variables are fixed to a given strategy and, e.g., the difference between these strategies are minimized in an upper level optimization.

CHAPTER 6

A Web-based Feedback Study

From November to December 2013, we conducted a feedback study with the *IWR Tailorshop* microworld. 148 participants were asked to play the economic simulation via its web interface. This chapter gives a description of study, hypotheses, analysis, and results including an optimization-based analysis.

6.1 Study Design

6.1.1 Task

Participants had to play four rounds of the *IWR Tailorshop* microworld of 10 months each via its web interface, see Figure 6.1. They were allowed to interrupt the process at any time. For the four rounds, different initial values were used, see Table 6.1, but the same for all participants. Rounds 1 and 3 started with the same values, whereas in rounds 2 and 4 pairwise different values were used. Control values for recruitment and dismissal of employees and creation and closing of sites were always reset to 0 in order to avoid accidental execution.

Participants were divided randomly into six groups based on pseudorandom numbers generated by a *Mersenne twister* [85]. Four of the groups—*indicate* group, *trend* group, *value* group, and *chart* group—received optimization-based feedback as described in Section 5.2 during the first two rounds. These rounds therefore will be called *feedback* or *training rounds* in the following. One group, *high-score* group, received a feedback based on the results of previous participants during training rounds, giving a ratio of participants who performed better and worse of the kind "Until now x% of participants performed better and y% performed worse than you." The sixth group was a *control* group without any feedback at all. In the last two rounds, however, no group received any feedback. These rounds will be referred to as *performance rounds*.

As an incentive, there was a competition with chances weighted according to success in which participants could win one of six 20 euro *Amazon* gift cards. For this, only the results of performance rounds were considered.

Additional information on the participants was collected via three questionnaires. The first survey was on general properties including, e.g., gender and interest in economics and is described in Table 6.2. This survey had to be answered before participants could start the *main task*, i.e., the four *IWR Tailorshop* rounds. The other two surveys were carried out after the main task. The second survey was targeted on participants' model knowledge. Participants were shown five claims about the *IWR Tailorshop* microworld and had to decide if they were right or wrong. A detailed description is given in Table 6.3. Final survey was the 10-item short version of the *Big Five Inventory* test proposed by [105] to measure the *Big Five dimensions* of personality [40], i.e., *agreeableness*, *conscientiousness*, *extraversion*, *neuroticism*, and *openness*.

For the main task, the control of the *IWR Tailorshop* microworld, the participants received guidance by the following introduction:

> *Thank you! Now you can start into the **IWR Tailorshop microworld**. Please note, that you need to finish **4 rounds of 10 "months" each** to participate in the competition.*

Variable		**Round 1**	**Round 2**	**Round 3**	**Round 4**
Employees	x_0^{EM}	14	3	14	42
Production sites	x_0^{PS}	1	1	1	2
Distribution sites	x_0^{DS}	1	5	1	7
Shirts in stock	x_0^{SH}	319	0	319	0
Production	x_0^{PR}	270	69	270	467
Sales	x_0^{SA}	270	69	270	467
Demand	x_0^{DE}	3877	2399	3877	3065
Reputation	x_0^{RE}	0.7934	0.1805	0.7934	0.4711
Shirts quality	x_0^{SQ}	0.7500	0.6558	0.7500	0.8136
Machine quality	x_0^{MQ}	0.8125	0.9998	0.8125	0.7712
Motivation of employees	x_0^{MO}	0.7403	0.4032	0.7403	0.5108
Capital	x_0^{CA}	175226	28075	175226	323907
Shirt price	u_0^{SP}	50	39	50	42
Advertising	u_0^{AD}	2000	1599	2000	1337
Wages	u_0^{WA}	1500	1750	1500	1451
Maintenance	u_0^{MA}	500	3000	500	267
Resources quality	u_0^{RQ}	2	1	2	2
Recruit employees	u_0^{DEM}	0	0	0	0
Dismiss employees	u_0^{dEM}	0	0	0	0
Create production site	u_0^{DPS}	0	0	0	0
Close production site	u_0^{dPS}	0	0	0	0
Create distribution site	u_0^{DDS}	0	0	0	0
Close distribution site	u_0^{dDS}	0	0	0	0

Table 6.1: Initial values for each round used in *IWR Tailorshop* feedback study. Note that values for controls (lower part) were only preset values and could still be changed by the participant. The last six controls, starting from *recruit employees*, were always set to the value in the table after each month to avoid accidental recruitment and dismissal as well as site creation and closing. Round 1 and 3 had the same initial values.

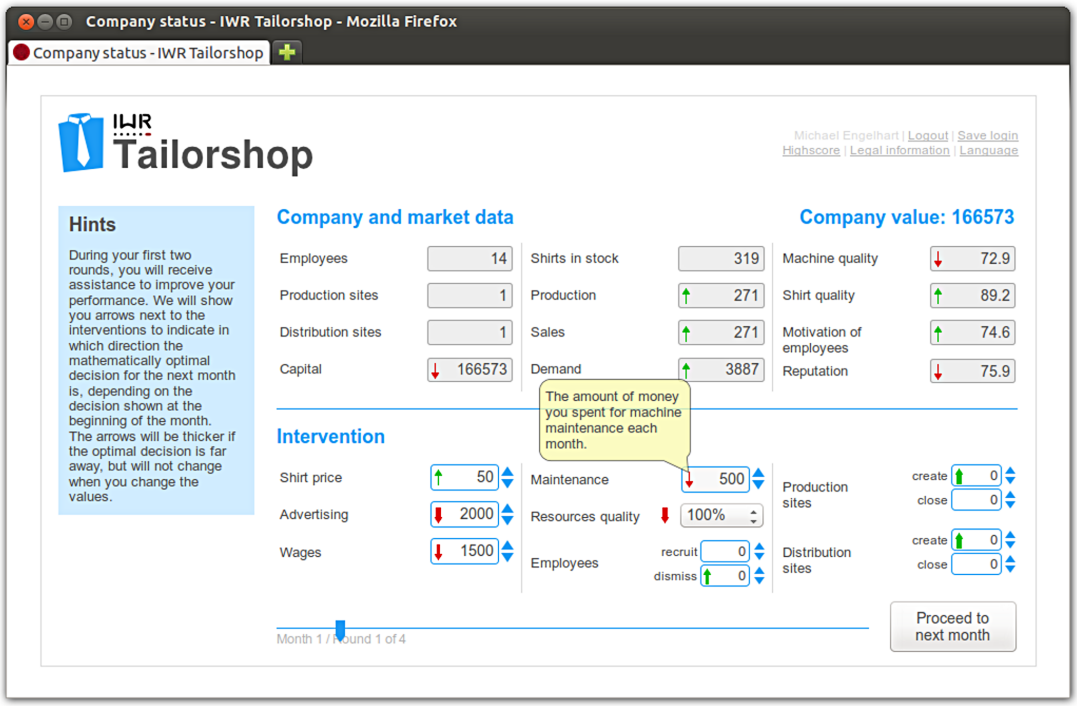


Figure 6.1: The *IWR Tailorshop* web interface with *trend* feedback and a hint for *maintenance* control.

Abbreviation	Claim	Possible answers
Computer Games	I play computer games regularly (i.e., at least once per week)	{yes, no}
Economics	I am interested in economic connections	{yes, no}
Problem Solving	I solve problems systematically in general	{yes, no}
Gender	I am ...	{female, male}
Age	My age is in the range ...	{<18, 18-24, 25-29, 30-39, 40-49, 50-59, 60-69, 69+}

Table 6.2: Survey on general properties at the beginning of task. This survey had to be answered before participants could start the main task. The participants were told that *"First, we would like to ask you to answer the following five questions truthfully. This data will be used exclusively for research purposes after being anonymized and will not be given to third party under no circumstances. Your answers do not affect your chances to win."* The content of the five items can be found in the *claim* column, *possible answers* are shown in the corresponding column. *Abbreviation* lists the terms used in this chapter to refer to these items.

Claim	Answer	Correct	Wrong	Don't know
Motivation of employees plays an important role.	false	56%	28%	16%
Maintenance is an important intervention possibility.	false	55%	26%	19%
The higher the shirt price is, the lower is the demand.	true	41%	45%	14%
Opening and Closing sites are important intervention possibilities.	true	90%	3%	6%
It is wise to dismiss employees at the end.	true	31%	33%	36%

Table 6.3: Survey on model properties at the end of task. The participants were told that *"We would like to ask you a few questions once again. Your answers will help us very much and it only takes two minutes. [...] Please decide if the following propositions are correct or wrong according to your experience from all four rounds."* Participants could always choose between *true*, *false*, and *don't know*. The content of the five items can be found in the *claim* column, the correct *answer* is shown in the corresponding column. The remaining columns show the ratio of correct, wrong and *don't know* answers. Differences to 100% are due to rounding.

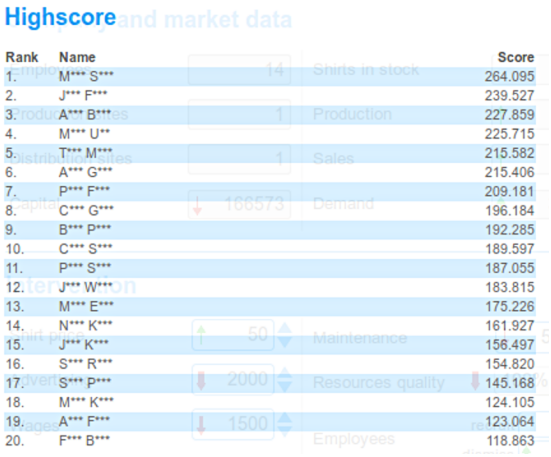
> *All in all it will take you about **30–45 minutes**. You ideally play all 4 rounds at a stretch, but you may interrupt after each "month" and continue at a later date. The first two rounds are **training rounds**, only your points (not your rank) in the last two rounds are considered for the drawing.*
>
> *Now, please imagine you are the **head of a company, which produces shirts**. Your aim is to **maximize** the **company's capital** at the end of each round, i.e. in month 10. For this there are several possibilities of **intervention** available, which will be located in the lower part. In the upper part you will find **important figures** of your company.*
>
> *However, your intervention possibilities are subject to certain **constraints**, e.g. you are not allowed to close all company sites. At the end of each round, you will find a **highscore table** and after the last round the table, which is important for the competition. In the blue **hint box** you can find assistance and useful hints during your game.*
>
> *Good luck!*

The *hint box* the introduction refers to was displayed at the border and contained hints corresponding to the situation and the feedback group the participant was in, e.g.,

> *During your first two rounds, you will receive assistance to improve your performance. We will show you arrows next to the interventions to indicate in which direction the mathematically optimal decision for the next month is, depending on the decision shown at the beginning of the month. The arrows will be thicker if the optimal decision is far away, but will not change when you change the values.*

Hints on each state and control, e.g., "*the wages for each employee per month in money units*" for control *wages*, were available as a tooltip on mouse rollover.

After each round, participants were shown an anonymized highscore list with the top 20 participants in their group (Figure 6.2).



Highscore

Rank	Name	Score
1.	M*** S***	264.095
2.	J*** F***	239.527
3.	A*** B***	227.859
4.	M*** U**	225.715
5.	T*** M***	215.582
6.	A*** G***	215.406
7.	P*** F***	209.181
8.	C*** G***	196.184
9.	B*** P***	192.285
10.	C*** S***	189.597
11.	P*** S***	187.055
12.	J*** W***	183.815
13.	M*** E***	175.226
14.	N*** K***	161.927
15.	J*** K***	156.497
16.	S*** R***	154.820
17.	S*** P***	145.168
18.	M*** K***	124.105
19.	A*** F***	123.064
20.	F*** B***	118.863

Figure 6.2: Anonymized highscore list with top 20 of participants in one feedback group in *IWR Tailorshop* web interface.

6.1.2 Prestudy

In October 2013, 18 participants recruited directedly via e-mail took part in a prestudy. The aim was twofold: on the one hand, this was a test under realistic conditions for the main study and an opportunity to eliminate bugs in the interface. On the other hand, the data were used for *highscore* feedback in the main study. This was particularly necessary to avoid a feedback like "0% performed better and 0% worse than you" for the first participant in that group. However, the data were considered neither in our statistical nor in our optimization-based analysis.

6.1.3 Data Collection

Starting from November 15, 2013, the study was announced in several first and third term lectures for mathematics, physics, computer science, engineering, and psychology students at *Heidelberg University* and *Otto von Guericke University Magdeburg* in Germany. These announcements were complemented by public announcements in the *social networks Google+* and *Facebook* as well as selective announcements via e-mail.

Potential participants were informed that they would have to play four rounds of the economic simulation *IWR Tailorshop* via a device of their choice with a web browser (e.g., PC, tablet, or smartphone) which in total would take approximately 30–45 minutes. It was advertised as an incentive that there will be a competition with chances weighted according to success where participants can win one of six 20 euro *Amazon* gift cards. The deadline for participation was December 15, 2013.

Participants had to create an account with an e-mail address, which they needed to confirm in order to avoid multiplicate participation. Creating multiple accounts was also prohibited by terms of participation leading to exclusion from the competition.

Until the end of data collection, 157 accounts were registered for participation. Two accounts have not been activated, maybe because of erroneous e-mail addresses or the like. Furthermore, seven participants did not answer the first survey and therefore could not start the main task, i.e., no data was recorded for them at all. Thus, we received data from 149 participants, of which 101 provided complete datasets, i.e., they played four full rounds and answered all three surveys. One account was identified as a duplicate participation and was excluded from the analysis. The first account of the corresponding participant was part of the analysis, but was not considered in the competition. This results in 100 complete datasets and 148 datasets in total for our statistical analysis.

Platform	Absolute	Relative
Desktop	141	80%
Mobile	36	20%
Total	177	100%

(a) Platform

System	Absolute	Relative
Windows	106	60%
Android	24	14%
Mac OS	19	11%
Linux	16	9%
iOS	12	7%
Total	177	100%

(b) Operating system

Browser	Absolute	Relative
Firefox	72	41%
Version <25	*9*	*5%*
Version 25	*57*	*32%*
Version 26	*5*	*3%*
Version 27	*1*	*1%*
Chrome	54	31%
Version <30	*6*	*3%*
Version 30	*12*	*7%*
Version 31	*33*	*19%*
Version 32	*3*	*2%*
Safari	22	12%
Android	10	6%
Internet Explorer	10	6%
Opera	9	5%
Total	177	100%

(c) Browser

Table 6.4: Usage of platforms, operating systems and browsers by all participants based on user logins. Note that participants using more than one device were counted more than once. Differences to 100% are due to rounding.

6.1.4 Technical Implementation

For data collection, the *IWR Tailorshop* web interface was used, which is implemented using *XHTML* and *JavaScript* with *jQuery 1.10* and usage of *AJAX* client-side, complemented by a server-side *PHP* code. For the online optimization, *AMPL Version 20131012* together with *Bonmin 1.5* and *Ipopt 3.10* was used via *IWR Tailorshop*'s AMPL interface. The web server for the study was an *Intel Core i7 920* machine with 12 GB RAM running *PHP 5.5* and *MySQL 5.5* with an *Apache 2.4 HTTP server* on *Ubuntu 13.10 64-bit*. More details on the technical implementation can be found in Appendix A.

The web interface implemented a so-called *responsive grid*, which allowed participants to use both mobile devices and desktop PCs conveniently. Usage statistics based on user logins (Table 6.4) show that approximately 20% of participants used mobile devices.

6.1.5 Hypotheses

Before the beginning of the study, 28 hypotheses were formulated. One of the more obvious expectations is e.g., Hypothesis (A), participants with optimization-based feedback perform better overall than those without. The 18 hypotheses concerning the statistical analysis are listed in Table 6.5, the 10 hypotheses for the optimization-based analysis are listed in Table 6.28. They are checked in Sections 6.2 and 6.3 respectively, and Table 7.1 gives an overview of the results.

(A) participants with **optimization-based feedback** perform better overall than those without

(B) participants with **optimization-based feedback** perform better in feedback rounds than those without

(C) participants with **optimization-based feedback** perform better in performance rounds than those without

(D) **control group** performs worst

(E) **control group** performs worse in performance rounds than groups with optimization-based feedback

(F) **trend group** performs best overall

(G) **trend group** performs best in performance rounds

(H) **value group** performs best in feedback rounds

(I) **value group** will perform better in *feedback rounds*, but worse in *performance rounds* (compared to other feedback groups)

(J) participants with high BFI-10 values perform worse/better than those with low values

(K) participants who play computer games regularly perform better than those who don't

(L) participants who claim to be interested in economic connections perform better than those who don't

(M) participants who claim to solve problems systematically in general perform better than those who don't

(N) **control group** needs more time than **optimization-based feedback groups**

(O) participants who performed well in performance rounds know more about the model

(P) participants who know much about the model perform well in performance rounds

(Q) **value group** knows less about the model than other groups, **trend group** knows most about the model, i.e., participants choose correct answers more and *don't know* less often

Table 6.5: Hypotheses on results of the web-based feedback study

6.2 Statistical Analysis

Statistical analysis of the data was done using the open source package *R Version 3.0.1* [104]. 148 datasets have been considered, 100 of them were complete. This corresponds to all participants for whom any data has been recorded. For all statistical tests, p-values of < 0.05 were considered statistically significant (i.e., $\alpha = 0.05$). All such values are printed in bold in tables.

6.2.1 Incomplete

An analysis of the 48 dropouts revealed that more than half of them aborted during the first round, as one can determine from Table 6.6. Indeed, 11 of 27 dropouts during round 1 did not play a single month after having answered the first survey.

Overall, 48 dropouts out of 148 participants is a dropout rate of approximately one third. By feedback groups, control group shows a slightly lower dropout rate of about 17%, whereas all other groups are approximately at the same level of about 40%. However, these differences are not significant. For gender, the differences between complete and incomplete datasets are only marginal.

By score, incomplete datasets exhibit lower means during the first two rounds. For performance rounds, there are too few datasets to draw any conclusions, but for the sake of completeness, it should be mentioned that in round 3 incomplete datasets have a higher mean than complete datasets.

The aim of this analysis was to detect systematic differences between complete and incomplete datasets, e.g., if participants who did not complete the task were demotivated by their poor performance. However, the differences in score means are not significant and thus it can only be speculated about the reasons. Finally, we can summarize that incomplete datasets do not show any systematic differences compared to complete datasets.

6.2.2 Outliers

A score boxplot for all rounds of all complete datasets, Figure 6.3, reveals that there are some severe outliers, especially in rounds 2 and 4. The extremest outlier in round 2 is approximately 16 times the *Interquartile Range* (*IQR*, difference between upper and lower quartile) below the 25% quartile. Although it is often stated that no data should be rejected, given the severity of the outliers and our size of 10 to 27 datasets per feedback group, retention of all datasets would cause huge biases and thus hardly seems sensible in our case.

On the other hand, excluding datasets from the analysis has to be done carefully and such that only a small portion of the data is considered to be an outlier, of course. We compared three different approaches for outlier detection, applied to all complete datasets in total and in groups.

First, we determined the best and worst datasets of all complete datasets in groups and globally, based on the score sum. Doing this round-based would detect about one third of the datasets as outliers and thus makes no sense. Nevertheless, among the 12 worst and best datasets (12%) are also some one would barely call an outlier. This is especially due to the fact that almost all datasets detected as outliers by the other approaches are near the corresponding lower bound. Therefore, this approach has been discarded.

A second approach was GRUBBS' test for outliers, which additionally was applied to the score sum over all rounds in groups and in total. GRUBBS' test is a statistical test proposed by FRANK E. GRUBBS [66, 67], which detects one outlier at a time in a normally distributed population. We used the implementation of GRUBBS' test available in the R package *outliers*. For significance level $\alpha = 0.05$, GRUBBS' test recursively identifies 12 datasets (12%) as outliers applied in groups (Table 6.7) and 9 datasets (9%) applied globally (Table 6.8). Applied to the score sum 4 and 2 datasets (4% and 2%) are detected as outliers in groups and globally respectively.

Round	Participants	Total	Ratio	Dropouts
1	27	27	56.25%	18.24%
2	14	41	29.17%	9.46%
3	6	47	12.50%	4.05%
4	1	48	2.08%	0.68%
Total	48	48	100.00%	32.43%

(a) Dropouts per round: more than half of the dropouts aborted during round 1. Overall dropout ratio was about one third.

Group	Total	Complete	Incomplete	Dropouts
Control	35	29	6	17.14%
Highscore	25	15	10	40.00%
Indicate	17	10	7	41.18%
Trend	35	22	13	37.14%
Value	19	12	7	36.84%
Chart	17	12	5	29.41%
Total	148	100	48	32.43%

(b) Absolute and relative dropouts per group: *control* group shows the lowest relative dropout. All other groups are on a similar level.

Gender	Total	Complete	Incomplete	Dropouts
Female	42	28	14	33.33%
in %	*28.38%*	*28.00%*	*29.17%*	
Male	106	72	34	32.08%
in %	*71.62%*	*72.00%*	*70.83%*	
Total	148	100	48	32.43%

(c) Gender distribution for dropouts: only marginal differences between complete and incomplete datasets.

Round	Total	Complete	Incomplete	t-Test
1	46156.2	50196.1	26918.4	0.2785
2	-74465.6	-59074.5	-294339.6	0.1976
3	147677.7	147574.2	158022.3	—

(d) Means per round for dropouts: no significant differences.

Table 6.6: Incomplete datasets: no significant deviations.

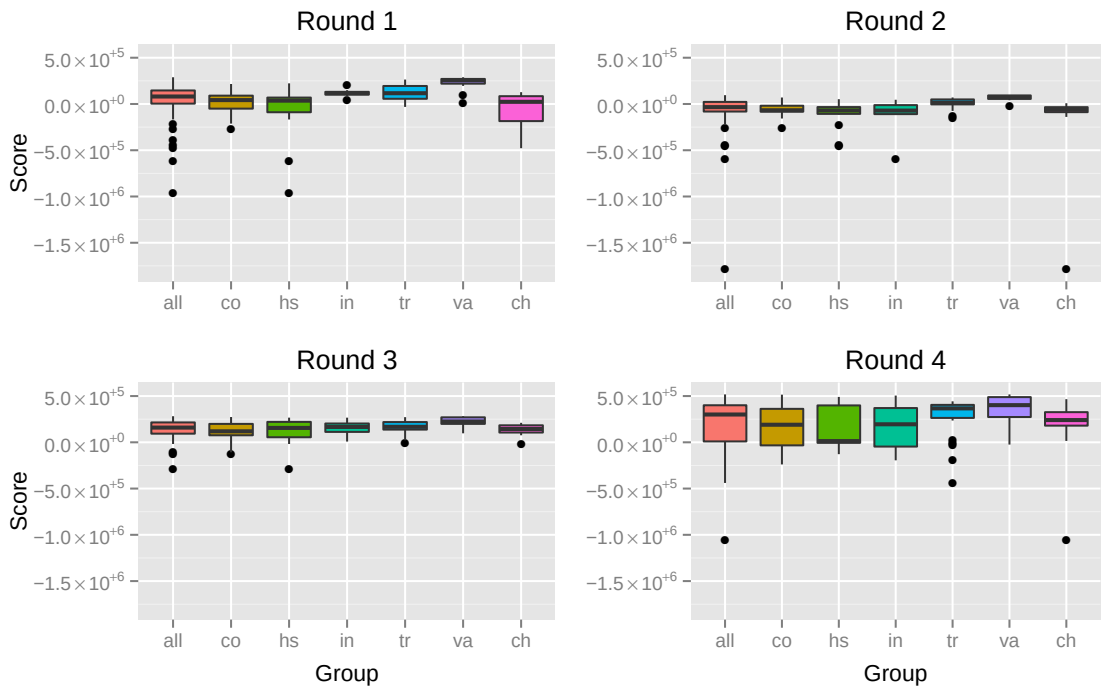


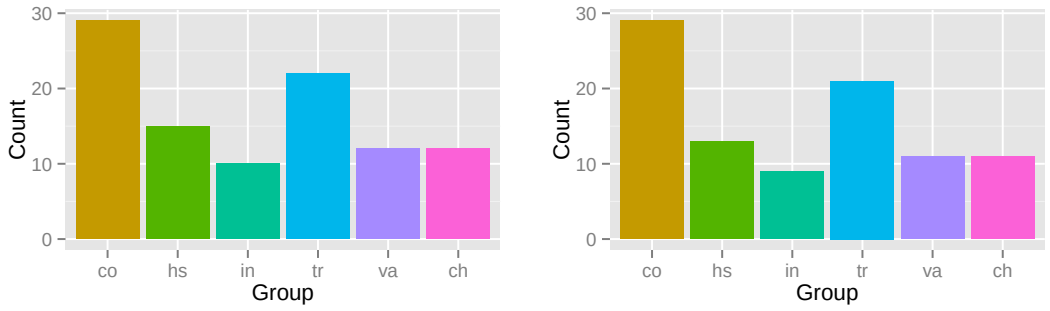
Figure 6.3: Score boxplot of all feedback groups (co: control, hs: highscore, in: indicate, tr: trend, va: value, ch: chart) for all rounds and all complete datasets ($N = 100$). Round 1, round 4, and especially round 2 show some severe outliers. It seems quite obvious from the boxplot that *value* group is better than the other groups. *Chart* group could not profit, indeed rather suffered from the feedback.

Group	R	p-val.	Score	P
Control	1	0.0913	-268972.3	90
		0.1355	-212152.1	205
	2	**0.0127**	-255313.9	73
		0.2769	70124.4	129
	3	0.0571	-131941.7	88
		0.0207	-107466.2	86
		0.8806	273425.1	129
	4	0.9085	-239364.6	76
Highscore	1	**0.0037**	-966712.5	85
		0.0004	-617007.9	122
		0.2581	-167571.4	144
	2	0.0881	-454850.5	85
		0.0012	-441338.3	122
		0.0314	-227081.0	144
		0.1342	51796.2	193
	3	**0.0057**	-288026.4	144
		0.5360	-18202.1	85
	4	1.0000	490396.3	193
Indicate	1	0.1135	206425.5	175
	2	**0.0001**	-593318.0	131
		0.3853	45760.5	179
	3	0.2244	8022.8	214
	4	0.7710	-194799.7	116
Trend	1	0.8375	-31777.7	108
	2	**0.0359**	-155986.8	108
		0.0057	-135683.9	216
		0.0388	-74903.0	83
		0.4149	-30902.5	119
	3	0.2634	-5213.9	83
	4	**0.0086**	-439288.0	83
		0.0336	-188878.4	216
		0.1077	-27816.9	123
Value	1	**0.0161**	14885.7	207
		0.0031	98571.8	165
		0.0764	195469.9	200
		0.3132	227910.6	208
	2	**0.0038**	-25724.4	207
		0.3320	40111.0	200
	3	0.1047	97178.2	207
	4	**0.0234**	-24511.4	207
		0.5141	226552.2	134
Chart	1	0.3641	-478441.6	101
	2	**0.0000**	-1783737.6	218
		0.1940	-139539.2	126
	3	**0.0395**	-17715.5	99
		0.6088	76031.2	218
	4	**0.0000**	-1054995.1	218
		0.3355	14819.6	101

Table 6.7: Results of recursive GRUBBS' test for outliers separately for each feedback group (R: round, P: participant, p-val.: p-value). With $\alpha = 0.05$, participants 73, 83, 85, 99, 108, 122, 131, 144, 165, 207, 216 and 218 are identified as outliers by this approach. This is a total of 12% of all complete datasets.

Round	p-value	Score	Participant
1	**0.0000**	-966712.5	85
	0.0004	-617007.9	122
	0.0041	-478441.6	101
	0.0028	-442934.0	96
	0.0038	-389675.3	218
	0.0732	-268972.3	90
	0.2337	-212152.1	205
2	**0.0000**	-1783737.6	218
	0.0000	-593318.0	131
	0.0001	-454850.5	85
	0.0000	-441338.3	122
	0.0608	-255313.9	73
	0.1304	-227081.0	144
3	**0.0001**	-288026.4	144
	0.0342	-131941.7	88
	0.0462	-107466.2	86
	1.0000	-18202.1	85
4	**0.0000**	-1054995.1	218
	0.1213	-439288.0	83

Table 6.8: Results of recursive GRUBBS' test for outliers on all complete datasets, i.e., without respect for feedback groups. With $\alpha = 0.05$, participants 85, 86, 88, 96, 101, 122, 131, 144 and 218 are identified as outliers by this approach. This is a total of 9% of all complete datasets.



(a) all complete datasets

(b) complete datasets without outliers

Figure 6.4: Histogram for feedback groups (co: control, hs: highscore, in: indicate, tr: trend, va: value, ch: chart). Left: all complete datasets ($N = 100$), right: complete datasets without 6 outliers ($N = 94$). Distributions are qualitatively similar.

The last approach were the *outer fences* for boxplots as described by JOHN W. TUKEY. The outer fences are defined in [126] as

$$\begin{aligned} \text{Lower outer fence} &:= Q_1 - 3 \cdot \text{IQR}, \\ \text{Upper outer fence} &:= Q_3 + 3 \cdot \text{IQR}, \end{aligned}$$

where Q_3 is the upper quartile, Q_1 the lower quartile, and IQR the interquartile range, the difference between the upper and lower quartiles. Note that the whiskers in boxplots in this thesis represent the *inner* fences, i.e., upper and lower quartile $\pm\, 1.5 \cdot \text{IQR}$. [126] considers "values between an inner fence and its neighboring outer fence [...] *outside*" and "values beyond outer fences [...] *far out*". Hence, we checked in groups and globally for each round, which datasets are *far out*. The result is shown in Table 6.9: each variant identified 7 (different) datasets (7%) as outliers.

An overview of the datasets identified as outliers by the different approaches is given in Table 6.10. Two datasets, participants 85 and 218, are considered to be outliers by all approaches. GRUBBS' test applied globally on score sum does not detect any further outliers. Worst and best datasets and GRUBBS' test for each round applied in groups identify more than 10% as outliers and therefore have been discarded. GRUBBS' test applied globally detects the same outliers as global outer fences except of participants 86 and 88, which no other approach considers to be outliers. From the remaining three, outer fences in groups seemed to yield the best trade-off between a small total number of outliers and group-specific outlier detection with the exception of participant 216, which is not considered an outlier by almost all other approaches. Thus, in the following analysis participants 83, 85, 122, 131, 207, and 218, i.e., 6% of all complete datasets, will be excluded as outliers. The analysis in the remainder is therefore based on 94 datasets. The histograms in Figure 6.4 show that no qualitative difference is caused by this rejection.

6.2.3 Normality and Variance Homogeneity

In the following analysis, we want to test the statistical significance of differences between means of scores and other variables. Statistical tests used in the remainder like STUDENT's t-test and WELCH's t-test require normality of the population—although these two are known to be relatively robust against non-normality (e.g., [114]).

Group	R	Min	Max	Lower	Upper	Participants
Control	1	-268972.3	216017.6	-471166.5	511195.3	—
	2	-255313.9	70124.4	-274254.7	172544.4	—
	3	-131941.7	273425.1	-292962.5	568022.8	—
	4	-239364.6	515249.8	-1216884.3	1545476.0	—
Highscore	1	**-966712.5**	223102.9	**-564521.7**	544564.6	85, 122
	2	**-454850.5**	51796.2	**-327944.7**	187664.0	85, 122
	3	-288026.4	266810.8	-443324.0	718864.3	—
	4	-128185.7	490396.3	-1219911.3	1612883.7	—
Indicate	1	36950.1	206425.5	10267.8	221926.6	—
	2	**-593318.0**	45760.5	**-400647.2**	278866.0	131
	3	8022.8	266534.6	-160097.0	475882.0	—
	4	-194799.7	505103.2	-1295279.5	1619710.2	—
Trend	1	-31777.7	264094.7	-358952.2	609654.6	—
	2	-155986.8	70603.8	-165307.7	210189.5	—
	3	-5213.9	271909.4	-106402.0	462778.8	—
	4	**-439288.0**	442941.5	**-158903.8**	827099.4	83, 216
Value	1	**14885.7**	288819.6	**67071.6**	423438.8	207
	2	-25724.4	95189.3	-56897.8	204007.4	—
	3	97178.2	281928.9	-13303.3	482540.8	—
	4	-24511.4	517939.7	-372493.1	1134446.6	—
Chart	1	-478441.6	127763.6	-992831.6	892192.1	—
	2	**-1783737.6**	8063.8	**-250258.0**	127622.3	218
	3	-17715.5	210841.4	-128407.9	417987.5	—
	4	**-1054995.1**	466244.6	**-256996.7**	763162.2	218
All	1	**-966712.5**	288819.6	**-423424.1**	573410.9	85, 96, 101, 122
	2	**-1783737.7**	95189.3	**-393507.4**	334401.6	85, 122, 131, 218
	3	**-288026.4**	281928.9	**-271070.9**	579494.2	144
	4	-1054995.1	517939.7	-1163753.7	1573760.8	—

Table 6.9: Detection of outliers by outer fences according to TUKEY [126], i.e., 1st/3rd quartile ±3·IQR (Lower: lower outer fence, Upper: upper outer fence, R: round, Min: minimal value, Max: maximal value). By groups, participants 83, 85, 122, 131, 207, 216, and 218 are identified as outliers by this approach. This is a total of 7% of all complete datasets. For all complete datasets together, participants 85, 96, 101, 122, 131, 144, and 218 are considered as outliers, which is also 7% of all complete datasets.

Participant	Grubbs All	Grubbs Groups	Fences All	Fences Groups	Grubbs Sum All	Grubbs Sum Groups	Worst & Best	Sum
73	—	✓	—	—	—	—	—	1
76	—	—	—	—	—	—	✓	1
83	—	✓	—	✓	—	—	✓	3
85	✓	✓	✓	✓	✓	✓	✓	7
86	✓	—	—	—	—	—	—	1
88	✓	—	—	—	—	—	—	1
96	✓	—	✓	—	—	—	—	2
99	—	✓	—	—	—	—	—	1
101	✓	—	✓	—	—	✓	—	3
108	—	✓	—	—	—	—	—	1
122	✓	✓	✓	✓	—	—	—	4
129	—	—	—	—	—	—	✓	1
131	✓	✓	✓	✓	—	—	✓	5
133	—	—	—	—	—	—	✓	1
144	✓	✓	✓	—	—	—	—	3
158	—	—	—	—	—	—	✓	1
164	—	—	—	—	—	—	✓	1
165	—	✓	—	—	—	—	—	1
175	—	—	—	—	—	—	✓	1
207	—	✓	—	✓	—	✓	✓	4
210	—	—	—	—	—	—	✓	1
216	—	✓	—	✓	—	—	—	2
218	✓	✓	✓	✓	✓	✓	✓	7
Total	9%	12%	7%	7%	2%	4%	12%	

Table 6.10: Different approaches of outlier detection with all participants, which were detected as outliers by at least one approach. *Grubbs All* and *Grubbs Groups* refer to recursive GRUBBS' test, see Tables 6.8 and 6.7. *Fences All* and *Fences Groups* refer to outer fences according to TUKEY [126], see Table 6.9. *Grubbs Sum All* and *Grubbs Sum Groups* refer to recursive GRUBBS' test on score sum. *Worst & Best* are the worst and best participants in each group according to the score sum. *Sum* contains the number of detections for each dataset. Participants 85 and 218 are considered to be outliers by all seven approaches. Outer fences in groups according to Tukey seem to yield the best trade-off between a small total outlier number and group-specific outliers, except for participant 216, which is not considered an outlier by almost all other approaches.

Therefore, we applied implementations of several tests for normality from the R package *nortest* to the score variables: SHAPIRO-WILK test [117], KOLMOGOROV-SMIRNOV test [84], LILLIEFORS test [84], ANDERSON-DARLING test [7, 8], CRAMÉR-VON MISES test [124], PEARSON's chi-squared test [97], and SHAPIRO-FRANCIA test [107]. The results are shown in Table 6.11. For $\alpha = 0.05$, the hypothesis of the data being normally distributed cannot be rejected for most groups and rounds by a majority of the applied tests for normality. Note that for *all* normality tests, the alternative hypothesis is that the data is *not* normally distributed.

STUDENT's t-test—in contrast to WELCH's t-test—also requires homogeneity of variances between the groups. This has been tested using LEVENE's test [82], BROWN-FORSYTHE test [31] (both as implemented in R package *lawstat*), and BARTLETT's test [13]. Table 6.12 contains the results: LEVENE's and BROWN-FORSYTHE tests show qualitatively similar results, whereas BARTLETT's test yields quite different results. At least for rounds 1 and 4, with $\alpha = 0.05$ we cannot assume homogeneous variances between feedback groups. Thus, for the sake of comparability, WELCH's t-test will be used for comparison of score means for *all* rounds.

Normality and variance homogeneity for other variables are discussed in the correspondent sections.

6.2.4 Score Means

Table 6.13 lists quartiles, means, and standard deviations for all groups and rounds. Comparing rounds 1 and 3, which had the same initial values, we can conclude that all groups improved drastically (20% at least) except of *value* group. For the other five groups, this shows that participants acquired knowledge on how to control the model in the first rounds. *Value* group remained static (-4%) at a higher level than the other groups. A reason for this may be that participants profited so strong from the *value* feedback during the feedback rounds that their performance without feedback slightly decreased. However, the group's mean is on a high level, so there was not much space for improvement anyhow. The score shift from round 1 to round 3 can also be observed in the score histograms in Figure 6.5.

The data also show that *value* group was the best by far in all the rounds, *trend* group comes second. In contrast to Hypothesis (D) from Table 6.5, *control* group is not the worst, neither overall nor exclusively in the performance rounds. In fact, *control* group, *highscore* group, and *indicate* group are on a similar level, whereas *chart* group performs even worse at least in the feedback rounds. This changes in performance rounds, so one can suppose that the feedback consternated the participants. A possible reason could lie in a misinterpretation of the sensitivity information participants were given by this feedback. All other optimization-based feedback groups received direct information on the optimal solution.

The score boxplot in Figure 6.6 emphasizes these presumptions. Again, it seems quite obvious that *value* group and—except for round 3—*trend* group are better than the others. *Chart* group on the contrary rather suffered from the feedback and slightly improved in performance rounds. *Indicate* group could profit from the feedback only in round 1.

The results of WELCH's t-test in Table 6.14 confirm all these observations. For $\alpha = 0.05$, *value* group is significantly better than *control* group in all the rounds. *Trend* group misses significance only in round 3 by narrow margin, but exhibits significant differences in the other rounds. *Indicate* group is significantly better only in round 1. The remaining groups are not significantly different than *control* group.

A comparison between optimization-based feedback groups and the other two groups in Table 6.15 shows that participants who received optimization-based feedback performed significantly better in each round and in total. The difference between *value* group and all other groups is also signif-

Group	R	SW	KS	LF	AD	CVM	PEAR	SF	Sum
Control	1	0.1354	0.3426	**0.0343**	0.0907	0.0644	0.0961	0.1131	6
	2	0.0839	0.6261	0.2012	0.0873	0.0921	**0.0415**	**0.0389**	5
	3	**0.0307**	0.5794	0.1603	0.0926	0.2399	0.5745	**0.0295**	5
	4	0.1080	0.6182	0.1938	0.0904	0.0832	0.4244	0.2055	7
Highscore	1	0.8601	0.8621	0.5223	0.5180	0.3786	0.0770	0.6386	7
	2	0.4915	0.8945	0.5974	0.4431	0.5312	0.5259	0.2272	7
	3	**0.0130**	0.5894	0.1518	**0.0396**	0.0973	0.1718	**0.0095**	4
	4	**0.0126**	0.1512	**0.0020**	**0.0046**	**0.0047**	**0.0040**	**0.0248**	1
Indicate	1	0.3756	0.8380	0.4462	0.3032	0.3369	0.2636	0.1940	7
	2	0.4968	0.9670	0.7994	0.5963	0.6525	0.8007	0.6822	7
	3	0.7615	0.8883	0.5539	0.7595	0.7031	0.4594	0.8496	7
	4	0.3874	0.9839	0.8794	0.5187	0.5976	0.8007	0.5909	7
Trend	1	0.3551	0.7829	0.3931	0.4076	0.4910	0.1991	0.5054	7
	2	**0.0004**	0.1174	**0.0012**	**0.0005**	**0.0023**	**0.0054**	**0.0006**	1
	3	**0.0472**	0.6762	0.2481	**0.0467**	0.0968	0.0700	0.0577	5
	4	**0.0000**	**0.0039**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0001**	0
Value	1	**0.0051**	0.5937	0.1483	**0.0131**	**0.0210**	**0.0442**	**0.0063**	2
	2	0.1799	0.8373	0.4625	0.2399	0.2726	0.6718	0.2640	7
	3	0.1849	0.7917	0.3803	0.2394	0.3341	0.4512	0.1664	7
	4	0.0869	0.6414	0.1888	0.1214	0.1690	0.2925	0.1470	7
Chart	1	**0.0016**	0.3500	**0.0277**	**0.0016**	**0.0024**	**0.0036**	**0.0044**	1
	2	0.9751	0.9647	0.8014	0.9155	0.8828	0.6718	0.9479	7
	3	0.1256	0.8155	0.4217	0.2063	0.3096	0.4512	0.0855	7
	4	0.4943	0.6495	0.1965	0.4397	0.4170	0.1856	0.5025	7
All	1	**0.0000**	0.1167	**0.0017**	**0.0002**	**0.0006**	**0.0248**	**0.0000**	1
	2	**0.0242**	0.6226	0.2009	0.0887	0.1665	0.0763	**0.0315**	5
	3	**0.0000**	0.4790	0.0903	**0.0005**	**0.0040**	**0.0089**	**0.0000**	2
	4	**0.0000**	**0.0052**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0

Table 6.11: p-values of different normality tests (R: round, SW: SHAPIRO-WILK test, KS: KOLMOGOROV-SMIRNOV test, LF: LILLIEFORS test, AD: ANDERSON-DARLING test, CVM: CRAMÉR-VON MISES test, PEAR: PEARSON's chi-squared test, SF: SHAPIRO-FRANCIA test) on scores of all complete datasets without 6 outliers ($N = 94$). For $\alpha = 0.05$, the hypothesis of the data being normally distributed cannot be rejected for most groups and rounds by a majority of the applied tests for normality. Note that for *all* normality tests, the alternative hypothesis is that the data is *not* normally distributed.

Round	Levene	BF	Bartlett
1	**0.0004**	**0.0406**	**0.0000**
2	0.2927	0.3675	**0.0082**
3	0.0666	0.1349	**0.0029**
4	**0.0007**	**0.0261**	0.0776

Table 6.12: p-values of different variance homogeneity tests (BF: BROWN-FORSYTHE test) on scores of all complete datasets without 6 outliers ($N = 94$). LEVENE's and BROWN-FORSYTHE tests show qualitatively similar results, whereas BARTLETT's test yields quite different results. As at least for rounds 1 and 4, with $\alpha = 0.05$ we cannot assume homogeneous variances between feedback groups, WELCH's t-test will be used for comparison of score means.

icant in all rounds for $\alpha = 0.05$ (not in the table).

We can summarize that Hypotheses (A), (B), (C), (E), and (H) were proved and Hypotheses (D), (F), and (G) were disproved. Hypothesis (I) can at most be considered as proved partly, as the differences between rounds 1 and 3 for *value* group are very small and not significant. Optimization-based feedback could significantly improve participants' performance in the *IWR Tailorshop* microworld if the presentation was chosen appropriately. In our study, *value* group performed significantly better than all other groups. All WELCH's t-tests in this section have been confirmed qualitatively by WILCOXON rank sum tests.

6.2.5 General Properties

Histograms of the general properties collected with the first survey can be found in Figures 6.7 and 6.8. The distribution of gender reflects the distribution of students in mathematics and natural science lectures, where participants were mainly recruited from. The recruitment of participants from first and third term lectures is also reflected in the age distribution as almost all participants were in the groups 18-24 years and 25-29 years. The property *problem solving* reveals that almost all participants claim to solve problems systematically. This can be considered an *above average effect* [71].

Score boxplots for both *computer games* (Figure 6.10a) and *economics* (Figure 6.10b) do not show a clear trend for the performance of the corresponding *yes* and *no* groups. For *problem solving*, by means and medians the *yes* group outperforms the *no* group in each round. However, the differences are not significant, as a t-test reveals (Table 6.17). Mean score values can be found in Table 6.16. This eventually disproves Hypotheses (K), (L), and (M).

Because of the age distribution—some groups consist of one or two datasets only—, a comparison of all age groups does not make sense. Therefore we merged the data in three groups of age with 68, 20 and 6 datasets respectively: <25 (low), 25–29 (middle), and >29 (high). 25–29 group has the highest score means in all rounds, see Table 6.19, whereas >29 group has the lowest means in 3 of 4 rounds. Middle age group is significantly better than low age group except for round 2, where it scarcely misses significance. High age group is still too small to draw reliable conclusions. The boxplot in Figure 6.9 complements these results. We can only speculate on the reasons for middle age group's success. Regarding the participants' background, participants with middle age may have more experience with complex systems, which they can build on controlling a complex microworld. A hypothesis would be that these effects are based on the different development of *fluid* and *crystallized intelligence* during adulthood ([35], p. 272ff.).

One could wonder if all these general properties are correlated, e.g., a property like *computer games* with *gender* (although this is kind of a prejudice, see e.g., [1]). A correlation matrix, see Table 6.18, shows that correlation for all properties is very low and irrelevant in this study, though.

G	R	Min	25%	Median	75%	Max	Mean	SD
co	1	-268972.3	-50154.3	42066.4	90183.1	216017.6	24274.1	113871.4
	2	-255313.9	-82769.3	-68063.6	-18940.9	70124.4	-57289.0	65176.2
	3	-131941.7	76031.2	118906.3	199029.1	273425.1	128502.7	96555.0
	4	-239364.6	-33015.6	189881.7	361607.3	515249.8	166039.1	222732.1
	S	-307683.5	10123.9	272033.3	462332.5	1074817.0	261526.8	355735.9
hi	1	-167571.4	-12994.9	48132.9	71395.8	223102.9	29427.6	102902.4
	2	-227081.0	-91332.2	-69651.2	-23943.3	51796.2	-62394.1	68015.2
	3	-288026.4	56998.7	155530.9	213910.3	266810.8	114554.4	146052.4
	4	-128185.7	-231.9	13251.4	411936.9	490396.3	181818.6	235011.6
	S	-719763.9	53342.8	110737.4	546795.1	1032106.2	263406.5	444204.5
in	1	72854.7	105405.2	119647.3	133175.7	206425.5	125011.9	37418.4
	2	-119047.1	-97544.8	-60257.1	-3438.8	45760.5	-50069.6	59593.4
	3	8022.8	99019.0	173712.1	212900.8	266534.6	155561.8	85334.1
	4	-194799.7	-28726.9	246636.9	386168.8	505103.2	190322.4	264463.9
	S	-97790.9	-3706.4	420949.1	665375.5	976088.9	420826.5	402820.6
tr	1	-31777.7	75430.8	118862.6	196184.2	264094.7	123606.6	87157.5
	2	-155986.8	-4279.1	19997.2	50001.2	70603.8	10474.8	59322.9
	3	22537.6	138273.6	172730.1	220249.9	271909.4	167642.2	69113.9
	4	-188878.4	348854.9	368507.1	405435.0	442941.5	297328.6	182044.2
	S	-175989.9	553577.9	692629.8	825085.2	949601.3	599052.3	339750.2
va	1	98571.8	233237.0	262980.7	273080.7	288819.6	241696.5	54822.0
	2	40111.0	59777.6	80120.2	92578.5	95189.3	74541.3	19623.8
	3	134230.7	204746.9	232137.2	270527.7	281928.9	231867.6	45358.3
	4	226552.2	330559.5	403517.7	490299.6	517939.7	401167.1	109249.3
	S	610696.1	875420.6	924908.4	1095569.8	1157422.0	949272.5	169498.5
ch	1	-478441.6	-72877.2	49071.8	84558.3	127763.6	-50298.3	214314.1
	2	-139539.2	-78642.1	-49229.5	-31840.2	8063.8	-59124.2	41907.6
	3	-17715.5	116080.3	152277.6	185886.0	210841.4	137970.4	65091.6
	4	14819.6	225280.3	252473.9	347153.1	466244.6	262485.2	141433.0
	S	-458625.3	233027.0	302050.1	484527.4	534712.0	291033.2	283194.8
all	1	-478441.6	13487.8	86105.6	151144.0	288819.6	73539.7	138910.1
	2	-255313.9	-75645.0	-25405.5	37525.9	95189.3	-26952.9	73074.2
	3	-288026.4	98383.5	161247.4	215645.2	281928.9	151112.2	95348.8
	4	-239364.6	24199.9	311883.9	403088.7	517939.7	238678.2	211968.3
	S	-719763.9	105266.8	465693.5	756781.1	1157422.0	436377.2	409125.4

Table 6.13: Score quantiles (G: group —co: control, hs: highscore, in: indicate, tr: trend, va: value, ch: chart—, R: round, S: score sum, 25%: 25%-quartile, 75%: 75%-quartile, SD: standard deviation) for each round and feedback group for all complete datasets without 6 outliers ($N = 94$).

Round	Highscore	Indicate	Trend	Value	Chart
1	0.4429	**0.0001**	**0.0005**	**0.0000**	0.8531
2	0.5891	0.3804	**0.0002**	**0.0000**	0.5414
3	0.6216	0.2168	0.0507	**0.0000**	0.3622
4	0.4200	0.4037	**0.0133**	**0.0000**	0.0577
Sum	0.4947	0.1539	**0.0007**	**0.0000**	0.3935

Table 6.14: WELCH's t-test p-values of comparison of score means for each round to *control* group with all complete datasets without 6 outliers ($N = 94$). Alternative hypothesis was that mean of *control* group is lower. With $\alpha = 0.05$, only *value* group is significantly better than *control* group in all rounds. However, *trend* group misses significance only in round 3 by narrow margin.

Round	Means			t test	
	ch	control	of	ch < of	control < of
1	25869.2	24274.1	112042.8	**0.0009**	**0.0020**
2	-58869.2	-57289.0	-1174.4	**0.0000**	**0.0003**
3	124185.3	128502.7	172860.8	**0.0091**	**0.0182**
4	170923.2	166039.1	293403.4	**0.0029**	**0.0059**
Sum	262108.6	261526.8	577132.7	**0.0000**	**0.0002**

Table 6.15: WELCH's t-test p-values of comparison of score means for each round between *control* and *highscore* groups (ch) on the one side and groups with optimization-based feedback (of) on the other side with all complete datasets without 6 outliers ($N = 94$). With $\alpha = 0.05$, optimization-based feedback groups were significantly better than those without.

Round	Group	Gender	Age	Economics	Games	Problems
1	no/female/low	76190.5	57191.0	62749.4	80372.3	34947.1
	yes/male/middle	72471.5	120974.6	81191.0	65080.3	77626.0
2	no/female/low	-44839.1	-32025.7	-33809.9	-26799.0	-54595.8
	yes/male/middle	-19745.0	-4364.7	-22090.7	-27143.4	-24026.0
3	no/female/low	131822.7	144973.0	161470.0	144409.3	120682.2
	yes/male/middle	158885.6	183191.3	143767.6	159411.0	154334.2
4	no/female/low	154121.5	226455.9	229691.9	217952.0	160459.0
	yes/male/middle	272753.3	327212.8	245050.4	264339.3	246960.3

Table 6.16: Mean score values for general properties (Gender: female/male, Age: low/high, i.e., <25/25–29, Economics/Games/Problems: yes/no) with all complete datasets without 6 outliers ($N = 94$). For explanations of the groups, see also Table 6.2.

Round	Gender	Economics	Computer Games	Problem solving
1	0.5594	0.2684	0.7021	0.1763
2	**0.0423**	0.2254	0.5089	0.1598
3	0.0793	0.8206	0.2164	0.2309
4	**0.0119**	0.3665	0.1435	0.1920

Table 6.17: p-values of WELCH's t-test for general properties for all complete datasets without 6 outliers ($N = 94$). See Table 6.2 for an explanation of the headings with $\alpha = 0.05$. For *Economics*, *Games*, and *Problems*, alternative hypothesis was that *yes* group has a higher mean than *no* group. For *Gender*, alternative hypothesis was that *male* group has a higher mean than *female* group.

	Gender	Economics	Games	Problems
Gender	1.00	0.23	0.24	-0.05
Economics	0.23	1.00	0.02	0.17
Games	0.24	0.02	1.00	0.00
Problems	-0.05	0.17	0.00	1.00

Table 6.18: Correlation matrix of general properties for all complete datasets without 6 outliers ($N = 94$): no relevant correlation was observed.

Round	Low < 25	Middle $25 - 29$	High > 29
1	57191.0	120974.6	100709.1
2	-32025.7	-4364.7	-44754.8
3	144973.0	183191.3	113759.4
4	226455.9	327212.8	82083.3

(a) Mean score values

Round	Low < Middle	High < Middle	High < Low
1	**0.0148**	0.3127	0.8561
2	0.0509	0.1465	0.3605
3	**0.0210**	0.0668	0.2321
4	**0.0175**	**0.0449**	0.1361

(b) p-values of WELCH's t-test ($\alpha = 0.05$)

Table 6.19: Mean score values of age groups and pairwise comparison by WELCH's t-test with all complete datasets without 6 outliers ($N = 94$). 25–29 group is significantly better than those younger than 25. Comparisons with participants older than 29 do not make much sense because of the small group size.

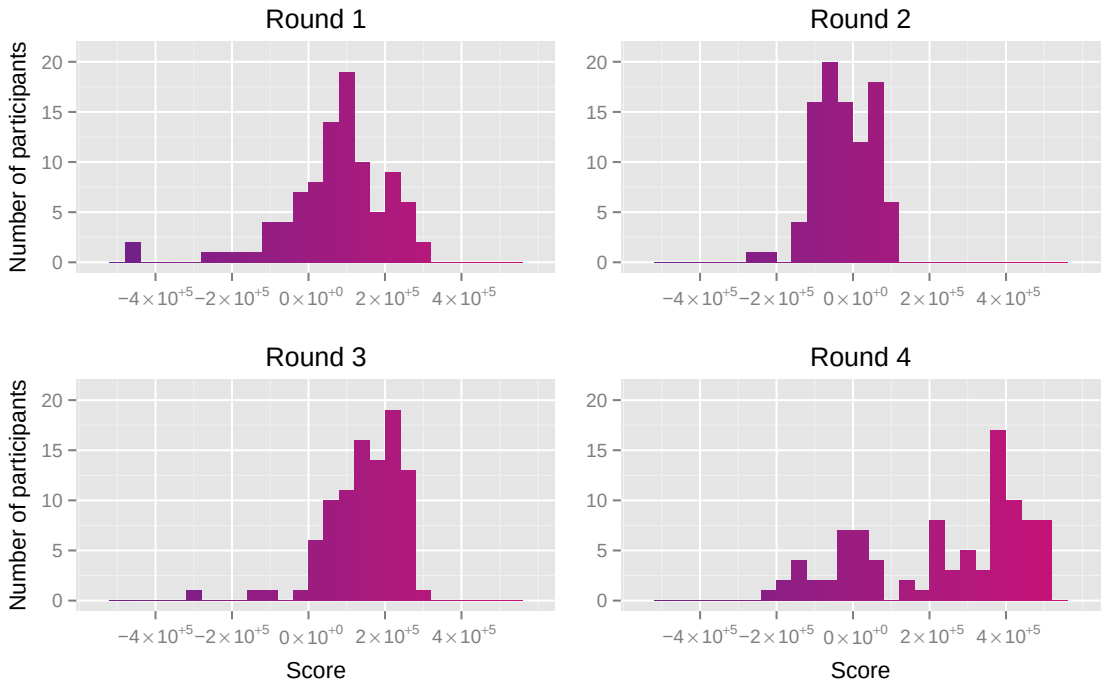


Figure 6.5: Score histogram for all four rounds for all complete datasets without 6 outliers ($N = 94$). Round 1 and 3 had the same initial values, whereas round 2 and 4 had pairwise different initial values. The distribution shifted slightly to the right, i.e., to higher scores from round 1 to round 3.

6.2.6 Gender and Feedback

Gender property deserves special attention. The t-test in Table 6.17 already indicates that there were gender-specific differences. A closer investigation targeted differences in the effects of feedback between the genders. For this, datasets were redivided into four groups: females in *control* or *highscore* group (*f/ch*), females in groups with optimization-based feedback (*f/of*), males in *control* or *highscore* group (*m/ch*), and males in groups with optimization-based feedback (*m/of*).

The boxplot in Figure 6.11 shows a quite surprising result. Obviously, in contrast to male participants, women could profit from the optimization-based feedback only in round 1. Here, both *f/of* and *m/of* were significantly better than *f/ch* and *m/ch*, as a t-test confirms (Table 6.20). There is no significant difference between *f/of* and *m/of* in the first round. However, this changes in the other rounds in which *f/of* is approximately at the same level as *f/ch* and *m/ch*. According to the t-test, *m/of* has a significantly higher mean than *all* the other groups, including *f/of*.

Mean score values for the four groups can be found in Table 6.21. A detailed overview of score means for each feedback group/gender combination including the gender distribution is given in Table 6.22. Note that for some female groups, values are based on one or two participants only and thus are not representative.

Given the design of the study and the data collected, it is not possible to determine a reason for this effect. The unbalanced distribution of female and male participants to feedback groups could also have influenced this observation. In order to investigate this aspect, one therefore would have to conduct a new study with, e.g., only one feedback and a control group and an approximately equal distribution of female and male participants.

Round	f/ch < m/ch	f/ch < f/of	f/ch < m/of	f/of < m/of	m/ch < f/of	m/ch < m/of
1	0.7352	**0.0024**	**0.0168**	0.6725	**0.0011**	**0.0074**
2	0.6332	0.1637	**0.0003**	**0.0339**	0.1116	**0.0001**
3	0.6801	0.5507	**0.0085**	**0.0254**	0.3817	**0.0099**
4	0.1794	0.2381	**0.0049**	**0.0323**	0.5331	**0.0054**

Table 6.20: p-values of WELCH's t-test for gender/feedback combinations (f/ch: females in *control* or *highscore* group, f/of: females in optimization-based feedback groups, m/ch: males in *control* or *highscore* group, m/of: males in optimization-based feedback groups) with all complete datasets without 6 outliers ($N = 94$). For $\alpha = 0.05$, except for round 1, men receiving optimization-based feedback were significantly better than all other groups. In contrast, women receiving optimization-based feedback could not achieve significant differences to both men and women in *control* and *highscore* groups. Thus, women could only profit from optimization-based feedback in round 1.

Round	f/ch	f/of	m/ch	m/of
1	38341.3	123502.0	18940.3	108605.1
2	-54680.6	-32537.3	-61196.2	8234.5
3	133548.5	129665.5	118983.6	185819.4
4	125463.6	189944.0	196178.6	324441.3

Table 6.21: Mean score values for each round corresponding to gender/feedback combinations (f/ch: females in *control* or *highscore* group, f/of: females in optimization-based feedback groups, m/ch: males in *control* or *highscore* group, m/of: males in optimization-based feedback groups) with all complete datasets without 6 outliers ($N = 94$).

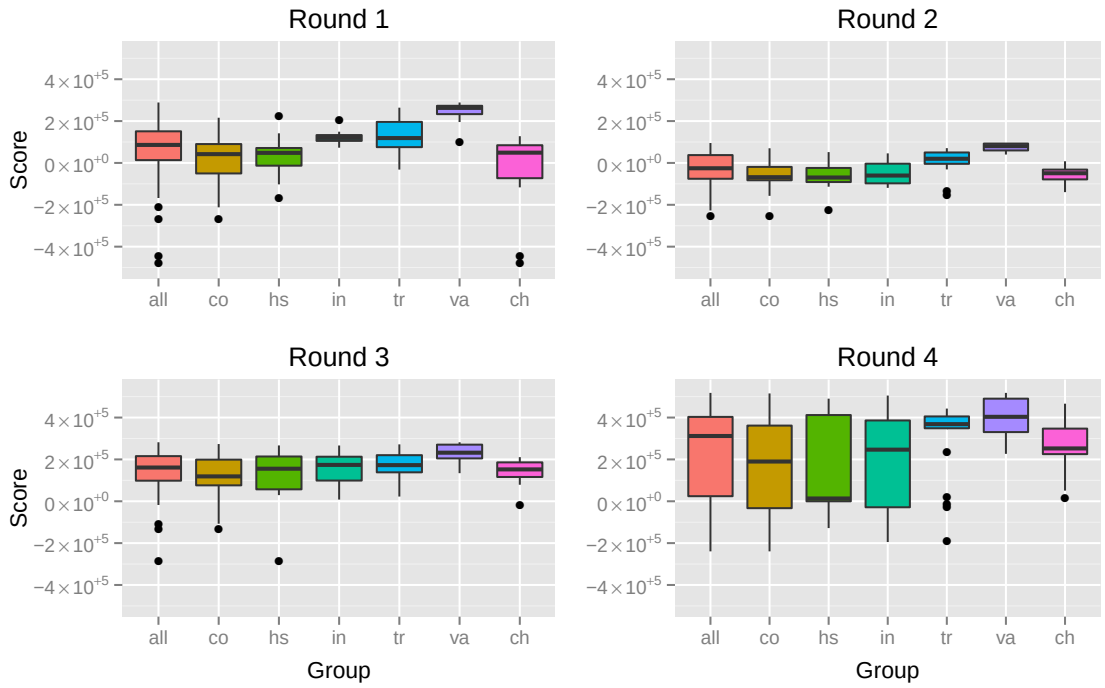


Figure 6.6: Score boxplot of all feedback groups (co: control, hs: highscore, in: indicate, tr: trend, va: value, ch: chart) for all rounds and all complete datasets without 6 outliers ($N = 94$). It seems quite obvious from the boxplot that *value* group and—except for round 3—*trend* group are better than the others, whereas *chart* group rather suffered from the feedback.

6.2.7 BFI-10

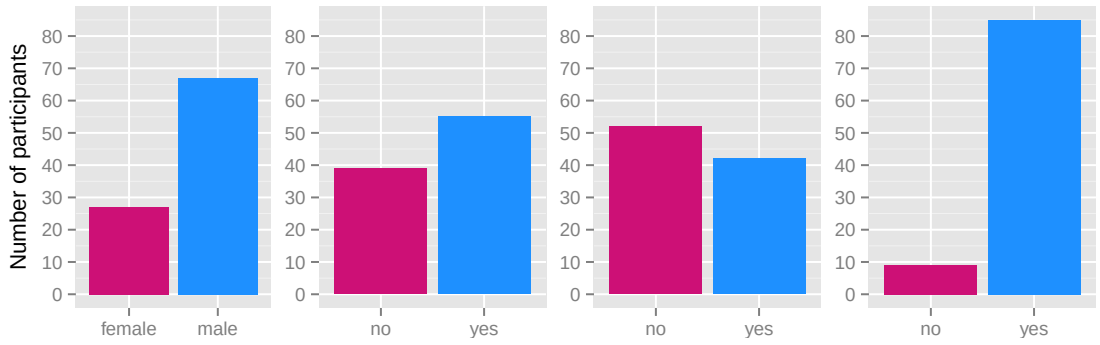
With the final survey, the *big five* dimensions of personality were determined via a 10-item short version of the Big Five Inventory proposed by Rammstedt and John [105]. Figure 6.12 contains histograms of all BFI-10 scales. All observed distributions fulfill the expectations: most values lie in the mid range, only *openness* exhibits a slight shift to higher values.

For an investigation of possible correlation of performance and scales, Figure 6.13 shows scatterplots for all five BFI-10 scales. No obvious correlation with score can be observed from the plots and indeed, correlation is close to 0 ($|\cdot| < 0.03$) for four scales, only for *agreeableness* it is about -0.19. Thus, there is no relevant correlation for any BFI-10 scale and Hypothesis (J) has to be discarded completely.

6.2.8 Processing Times

Within the statistical analysis, participants' processing times have also been analyzed. Because of the web-based data acquisition, the times determined can only serve as an approximation. Exact time measurement is extremely difficult in this case, because HTTP is a stateless protocol. But even if the time for which the user is logged in is measured correctly, it is almost impossible to determine when the user actually paid attention to the interface.

Nevertheless, we used the time information collected via explicit login or logout events together with reception times of participants' decisions to compute an approximate processing time per par-



(a) Gender **(b)** Economics **(c)** Computer games **(d)** Problem solving

Figure 6.7: Histograms of general properties collected via questionnaire at the beginning of the study task for all complete datasets without 6 outliers ($N = 94$). See Table 6.2 for details on the survey. Histogram (d) reveals an *above average* effect, as almost all participants claim to solve problems systematically. The distribution of gender in (a) reflects the distribution of students in mathematics and natural science lectures, where participants were mainly recruited from.

ticipant. In the following, we compare effective processing times only, i.e., time spent for computing optimal solutions for the optimization-based feedback is excluded.

Means of both processing and computing times per feedback group are shown in Figure 6.14, numerical values are given in Table 6.23c. Processing times decrease from round 1 to round 4 for all groups. However, the decrease is much higher for optimization-based feedback groups. For these groups, especially the processing times in feedback rounds are a lot higher than for *control* and *high-score* group.

Processing times have also been tested for normality with SHAPIRO-WILK test and KOLMOGOROV-SMIRNOV test, see Tables 6.23a and 6.23b. For almost all rounds and groups, the assumption of normality cannot be rejected. LEVENE's test and BROWN-FORSYTHE test in Table 6.23e can also not reject the hypothesis of variance homogeneity, thus STUDENT's t-test has been used to investigate statistical significance.

The results of STUDENT's t-test in Table 6.23d show that all optimization-based feedback groups had significantly higher processing times than *control* group in rounds 1 and 2. This is also reflected in total processing times. Hence, hypothesis (N) clearly is disproved, indeed the opposite is true. A boxplot of processing times (Figure 6.15) supports these results. An obvious assumption is that participants simply need more time to process the feedback information.

6.2.9 Model Knowledge and Uncertainty

A questionnaire (Table 6.3) was used at the end of the four rounds to determine the participants' knowledge about the *IWR Tailorshop* microworld. The overall ratio of correct answers varies a lot for the five claims. This shows that the questions had a varying difficulty, which was intended.

Correct answers are identified as *knowledge* about the model. Participants who chose *don't know* are considered to be *uncertain* about the corresponding claim. The focus of this section are the two variables *knowledge* and *uncertainty*.

With the analysis of these measures, we want to answer three different questions. The first one is if participants with a good performance, i.e., a high score, had a better knowledge or a lower uncertainty respectively about the model (Hypothesis (O)). Furthermore it will be examined if those with a

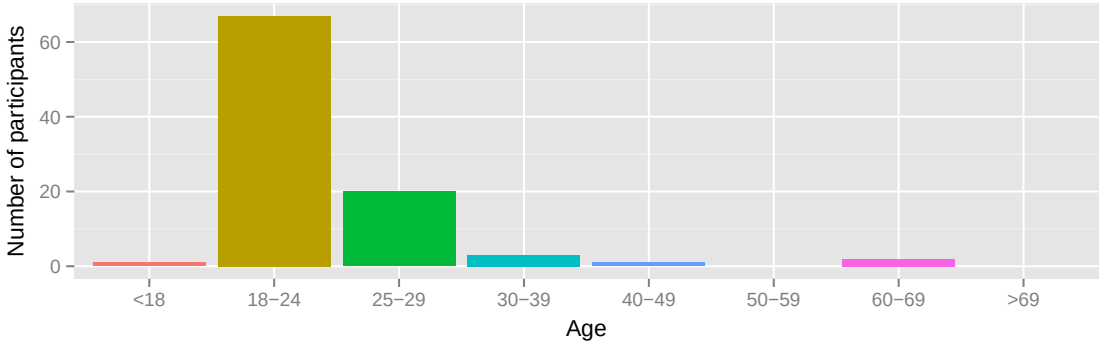


Figure 6.8: Histogram of participants' age collected via questionnaire at the beginning of the study task for all complete datasets without 6 outliers ($N = 94$). The distribution reflects the fact that participants were mainly recruited via 1st and 3rd term lectures. One participant claimed to be under the age of 18, although forbidden by terms of participation. There were no participants in the groups *50-59* and *above 69*.

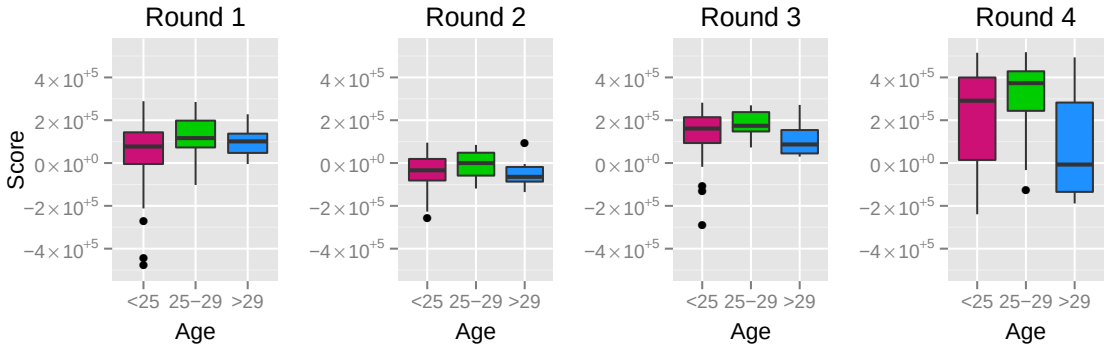


Figure 6.9: Boxplot of all four rounds for participants' age with all complete datasets without 6 outliers ($N = 94$). Participants older than 29 have been merged in one group, as there are only 6 such datasets in total. 25–29 group seems to outperform the others in all rounds, see also Table 6.19.

good knowledge or a low uncertainty about the model did manage vice versa to perform better (Hypothesis (P)). And finally, we want to analyze differences in model knowledge and uncertainty between the groups, i.e., if optimization-based feedback could enhance the participants' model knowledge (including Hypothesis (Q)).

To investigate the first aspect, quartiles have been used to build groups of participants with *high* (best 25%), *mid* (those between first and third quartile), and *low* (worst 25%) score for each round. Means of correspondent model knowledge and uncertainty scores can be found in Tables 6.24 and 6.25. *High* groups have the highest means which increase over the rounds. Except for round 1, *mid* groups are between *low* and *high* groups. In performance rounds, all differences are significant according to the WELCH's t-test, which verifies Hypothesis (O). Significance roughly increases over the rounds, which suggests that model knowledge is a crucial factor for successful control of the *IWR Tailorshop* microworld.

For the second question, participants have been merged in 3 (*low* (0/1), *mid* (2/3), and *high* (4/5)) and 2 (*low* (0/1) and *mid* (2/3)) groups respectively according to their *knowledge* and *uncertainty*

Round	Group	Control	Highscore	Indicate	Trend	Value	Chart
1	female	51455.1	-46898.5	107717.3	141283.8	266053.5	39341.3
	male	2189.5	43305.1	138847.6	118082.5	239260.8	-70218.2
2	female	-52784.4	-67005.6	-60936.5	-14490.4	56619.8	-65434.5
	male	-60949.1	-61555.7	-41376.1	18276.4	76333.5	-57721.9
3	female	122532.9	205150.1	100555.7	162528.2	199355.4	70883.7
	male	133353.1	98082.4	199566.7	169240.4	235118.9	152878.6
4	female	107902.8	239608.2	60698.4	264346.7	235081.7	239859.8
	male	213274.9	171311.4	294021.7	307635.5	417775.6	267513.1
#P	female	13	2	4	5	1	2
	male	16	11	5	16	10	9

Table 6.22: Mean score values separate for each gender and gender distribution corresponding to feedback groups (#P: number of participants) with all complete datasets without 6 outliers ($N = 94$). Note that for some female groups, values are based on one or two participants and thus are not representative. A reason why women could not profit from optimization-based feedback cannot be determined from the available data.

score, which both are between 0 and 5. No participant achieved an *uncertainty* score of 4 or 5, thus there are only two groups for *uncertainty*. Tables 6.26a and 6.26b contain the mean score values of all four rounds for these groups.

For *knowledge*, the *high* group has the highest score means by far. Except for round 1, *mid* group lies between *low* and *high* group. STUDENT's t-test in Table 6.26c shows that *high* group was almost always significantly better than the two other groups. Significance increases over the rounds, which means that model knowledge becomes a better predictor for participants success the more rounds the participants played. Comparing round 1 and 3, participants with low model knowledge could barely improve their performance, whereas the *high* group approximately doubled their score. Indeed, correlation between score and model knowledge increases from about 0.09 in round 1 to 0.48 in round 4. This proves Hypothesis (P).

For *uncertainty*, the *low* group has higher means in all rounds, but again the differences are much smaller than for *knowledge*. Hence, the differences between the groups are not significant. Correlation with score is about -0.2 for all rounds except the first.

Finally, for an analysis of differences between the groups, ratios of model knowledge and uncertainty levels and mean values are given in Table 6.27. *Trend* and *value* group have the highest knowledge, but only *highscore* and *trend* group are significantly better than *control* group. *Indicate* and *chart* group have a much lower knowledge, which together with these groups' performance suggests that participants were rather confused by the optimization-based feedback.

Trend group has by far the lowest uncertainty among the groups and is the only one which has significantly lower uncertainty than *control* group. All other groups are on a similar level. Thus, the second part of Hypothesis (Q) is proved and the first part disproved.

Round	Control	Highscore	Indicate	Trend	Value	Chart	All
1	**0.0089**	0.1595	0.5852	**0.0009**	**0.0209**	0.3442	**0.0000**
2	0.1443	0.0971	**0.0265**	**0.0007**	**0.0012**	**0.0114**	**0.0000**
3	0.3491	0.1613	**0.0010**	**0.0000**	**0.0000**	0.5525	**0.0000**
4	0.1905	0.3140	0.1119	**0.0000**	**0.0001**	**0.0347**	**0.0000**
Sum	0.0794	0.3366	0.4809	**0.0000**	**0.0004**	0.1850	**0.0000**

(a) Shapiro-Wilk test p-values ($\alpha = 0.05$)

Round	Control	Highscore	Indicate	Trend	Value	Chart	All
1	0.5251	0.5312	0.8690	0.2280	0.3688	0.9729	0.2909
2	0.7315	0.8037	0.5267	0.6913	0.2482	0.3527	**0.0283**
3	0.9203	0.6241	0.2020	**0.0141**	0.1319	0.7262	**0.0003**
4	0.6096	0.9071	0.9066	**0.0087**	0.1713	0.6441	**0.0039**
Sum	0.2131	0.7051	0.9070	0.2219	0.1189	0.8443	0.2910

(b) Kolmogorov-Smirnov test p-values ($\alpha = 0.05$)

Round	Control	Highscore	Indicate	Trend	Value	Chart	All
1	595.0	742.4	1200.5	1045.2	1071.2	1116.4	892.1
2	366.0	452.1	723.0	608.4	573.5	821.5	540.8
3	330.9	453.4	508.1	526.9	433.5	435.1	431.8
4	293.0	376.6	352.1	452.5	359.4	364.5	362.0
Sum	1579.7	1949.8	2916.3	2538.0	2437.6	2795.0	2405.7

(c) Mean effective times (i.e., total time without computing times) in seconds

R	Highscore	Indicate	Trend	Value	Chart
1	0.1193	**0.0004**	**0.0007**	**0.0018**	**0.0001**
2	0.1211	**0.0037**	**0.0008**	**0.0150**	**0.0001**
3	**0.0411**	0.0614	0.0576	0.1511	0.0580
4	0.0792	0.1609	0.0656	0.1650	0.1120
Sum	0.1000	**0.0010**	**0.0028**	**0.0170**	**0.0001**

(d) STUDENT's t-test p-values ($\alpha = 0.05$)

R	Levene	BF
1	0.3388	0.4871
2	0.0620	0.3558
3	0.3253	0.8733
4	0.5440	0.8615
Sum	0.5349	0.7490

(e) Variance homogeneity tests

Table 6.23: Analysis of processing times (R: round, BF: BROWN-FORSYTHE test) for all complete datasets without 6 outliers ($N = 94$). During feedback rounds, processing times for optimization-based feedback groups are significantly higher. STUDENT's t-test has been applied pairwise with alternative hypothesis of *control* group's mean being less.

Round	High Score	Mid Score	Low Score	High > Low	High > Mid	Mid > Low
1	3.17	2.50	2.79	0.1477	**0.0205**	0.8417
2	3.42	2.65	2.25	**0.0004**	**0.0063**	0.0770
3	3.46	2.74	2.04	**0.0000**	**0.0061**	**0.0068**
4	3.50	2.70	2.08	**0.0000**	**0.0023**	**0.0142**
Sum	3.33	2.80	2.04	**0.0001**	**0.0384**	**0.0035**

Table 6.24: Means of model knowledge for participants with high (i.e., best 25%), mid (between 1st and 3rd quartile), and low (i.e., worst 25%) score in the corresponding round with all complete datasets without 6 outliers ($N = 94$). Pairwise comparison of means by WELCH's t-test with $\alpha = 0.05$ shows, that high scorers know significantly more about the model than mid or low scorers.

Round	High Score	Mid Score	Low Score	High > Low	High > Mid	Mid > Low
1	0.75	1.07	0.79	0.4376	0.0909	0.8716
2	0.58	1.07	0.96	0.0711	**0.0181**	0.6733
3	0.67	0.87	1.25	**0.0097**	0.1667	0.0633
4	0.71	0.93	1.08	0.0820	0.1612	0.2740
Sum	0.67	0.98	1.04	0.0696	0.0930	0.3924

Table 6.25: Means of model uncertainty for participants with high (i.e., best 25%), mid (between 1st and 3rd quartile), and low (i.e., worst 25%) score in the corresponding round with all complete datasets without 6 outliers ($N = 94$). Uncertainty means are lower for high scorers. Pairwise comparison of means by WELCH's t-test with $\alpha = 0.05$ barely shows significance, however.

Round	Low (0/1)	Mid (2/3)	High (4/5)
1	101085.2	42407.9	110944.2
2	-51748.1	-40915.9	10319.9
3	108448.1	135943.2	200281.3
4	80163.6	214925.1	366269.3

(a) Mean score values for different levels of model knowledge

Round	Low (0/1)	Mid (2/3)
1	69516.5	86706.5
2	-22096.4	-42847.0
3	159269.1	124416.9
4	259346.6	171036.4

(b) Mean score values for different levels of model uncertainty

R	Low < High	Low < Mid	Mid < High
1	0.3740	0.9743	**0.0188**
2	**0.0005**	0.2657	**0.0010**
3	**0.0004**	0.1447	**0.0007**
4	**0.0001**	**0.0223**	**0.0001**

(c) STUDENT's t-test p-values for model knowledge

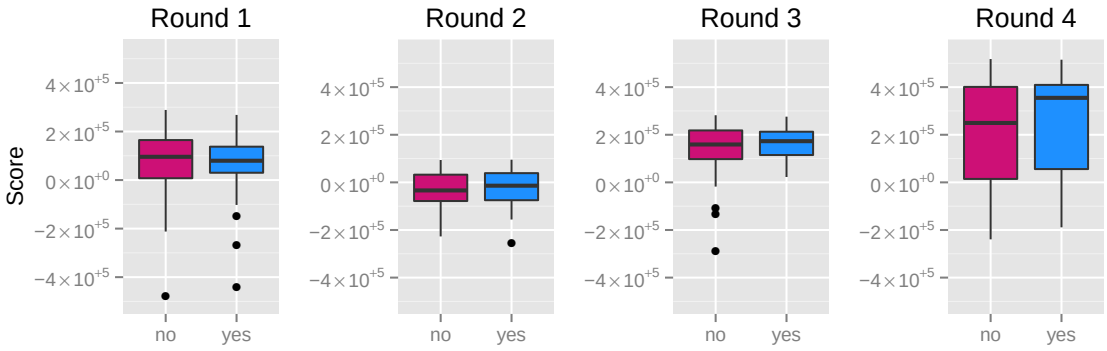
Round	Mid < Low
1	0.7335
2	0.1221
3	0.1020
4	0.0626

(d) STUDENT's t-test p-values for model uncertainty

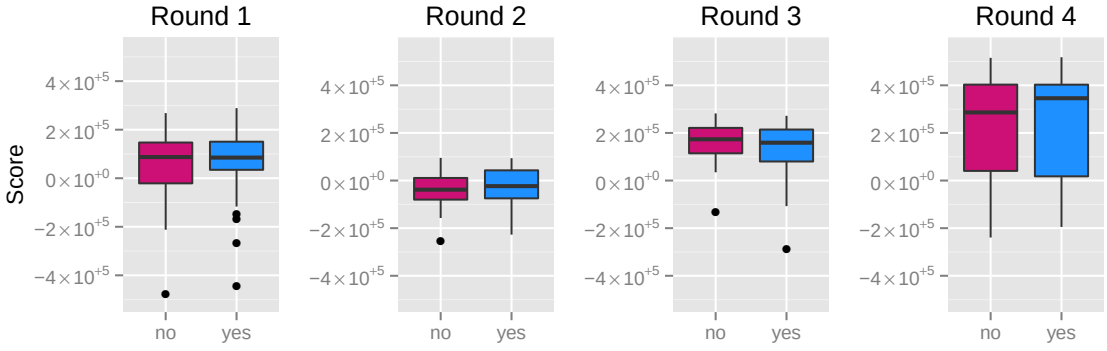
Table 6.26: Scores for different model knowledge and uncertainty levels (R: round) with all complete datasets without 6 outliers ($N = 94$). With $\alpha = 0.05$, participants with high model knowledge have achieved a significantly better score in almost all rounds. For model uncertainty, no significant score differences have been observed.

Property		co	hi	in	tr	va	ch	All
Knowledge	low	24%	8%	44%	5%	18%	9%	17%
	mid	59%	54%	33%	48%	36%	73%	52%
	high	17%	38%	22%	48%	45%	18%	31%
	mean	2.38	3.00	2.22	3.19	3.09	2.64	2.74
	t-test	—	**0.0451**	0.6241	**0.0113**	0.0824	0.2377	—
Uncertainty	low	72%	69%	67%	95%	82%	64%	77%
	high	28%	31%	33%	5%	18%	36%	23%
	mean	1.03	1.15	1.22	0.38	0.91	1.09	0.91
	t-test	—	0.6525	0.6630	**0.0017**	0.3545	0.5545	—

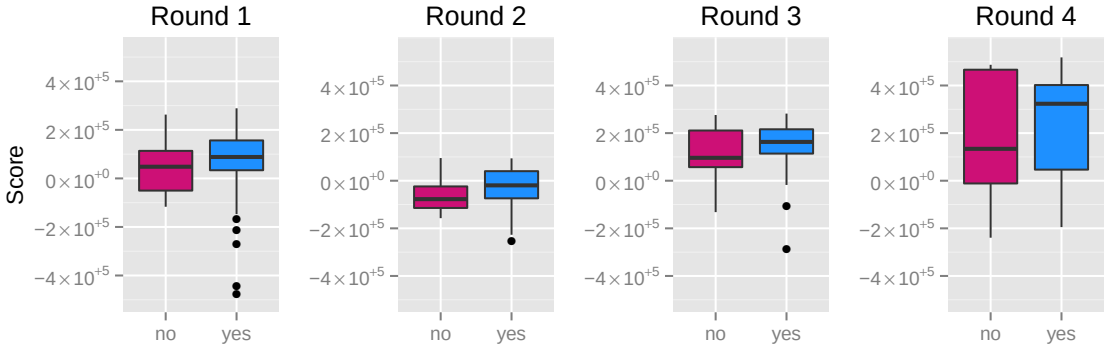
Table 6.27: Ratio of model knowledge and uncertainty levels for all feedback groups (co: control, hs: highscore, in: indicate, tr: trend, va: value, ch: chart) with all complete datasets without 6 outliers ($N = 94$). Mean refers to mean uncertainty and knowledge per group. Alternative hypothesis for WELCH's t-test was that mean of control group is lower (knowledge) or higher (uncertainty) respectively. For $\alpha = 0.05$, only *trend* group is significantly better in both knowledge and uncertainty. Differences to 100% are due to rounding.



(a) Computer games



(b) Economics



(c) Problem solving

Figure 6.10: Score boxplots of all rounds for general properties collected via questionnaire at the beginning of the study task with all complete datasets without 6 outliers ($N = 94$). See Table 6.2 for details on the survey. For (a) and (b), there is no clear trend. For (c), by means and medians the *yes* group outperforms the *no* group in all rounds. However, the differences are not significant, see also Table 6.17.

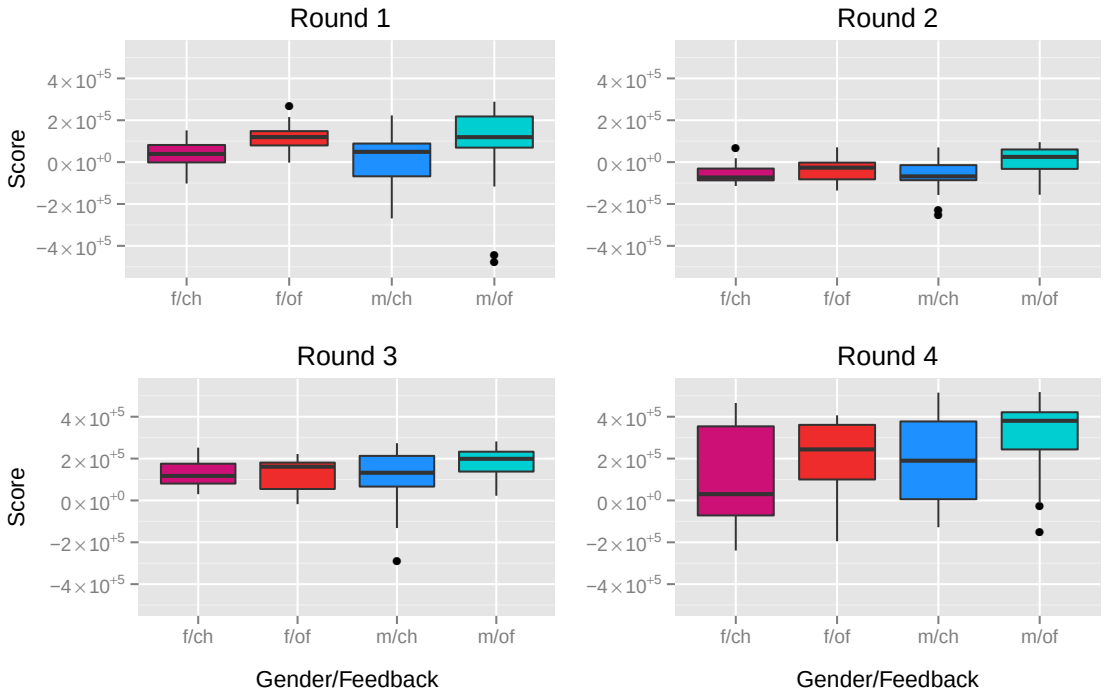


Figure 6.11: Boxplot of all four rounds for gender/feedback combinations (f/ch: females in *control* or *highscore* group, f/of: females in optimization-based feedback groups, m/ch: males in *control* or *highscore* group, m/of: males in optimization-based feedback groups) with complete datasets without 6 outliers ($N = 94$). Except for round 1, women could not significantly profit from optimization-based feedback. Whereas *m/of* group is significantly better than *f/ch* and *m/ch*, *f/of* exhibits no significant difference in round 2-4, see also Table 6.20.

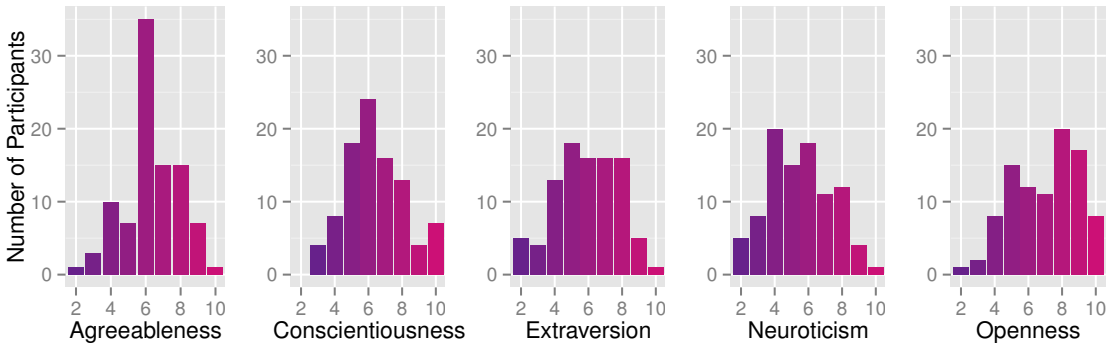
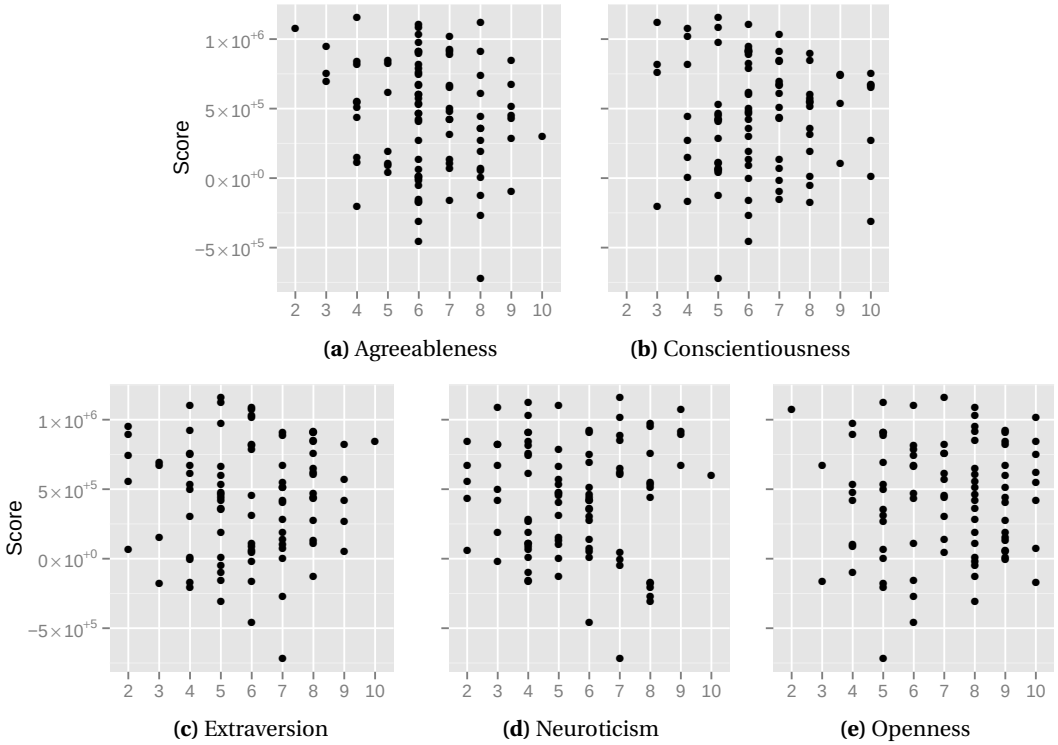


Figure 6.12: Histograms of all five BFI-10 scales for all complete datasets without 6 outliers ($N = 94$). All observed distributions fulfill the expectations.



(a) Agreeableness **(b)** Conscientiousness

(c) Extraversion **(d)** Neuroticism **(e)** Openness

Figure 6.13: Scatterplots of all five BFI-10 scales versus score sum for all complete datasets without 6 outliers ($N = 94$). No obvious correlation can be observed from the plots. Indeed, correlation is close to 0 ($|\cdot| < 0.03$) for four scales, only for agreeableness it is about -0.19.

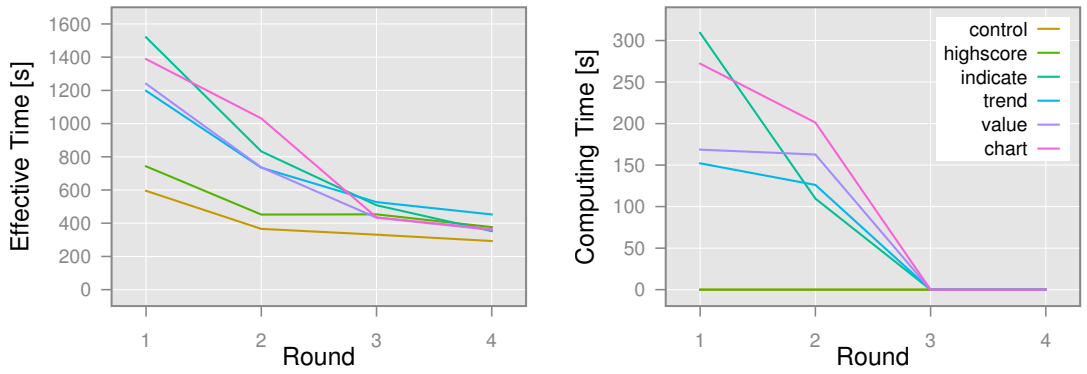


Figure 6.14: Mean effective time (i.e., total time minus computing time) consumed in each round by participants and mean computing time consumed per round by optimization according to the feedback groups for all complete datasets without 6 outliers ($N = 94$). Computing time is always 0 for control and highscore group as no optimal solution was computed online. In rounds 3 and 4, there was no optimization at all since no feedback was given.

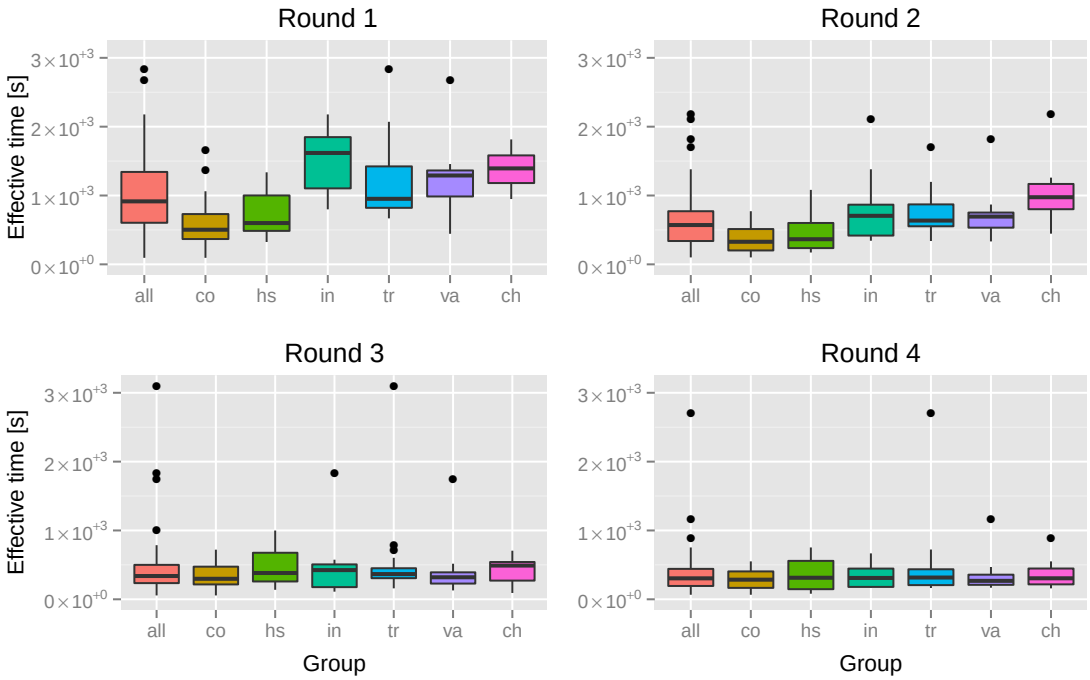


Figure 6.15: Boxplot of effective time (i.e., total time minus computing time) used in each round by participants according to the feedback groups (co: control, hs: highscore, in: indicate, tr: trend, va: value, ch: chart) with all complete datasets without 6 outliers ($N = 94$). In feedback rounds, optimization-based groups show significantly higher times. In performance rounds, all groups are on a similar level. For all groups, mean time decreases from round 1 to round 4.

6.3 Optimization-based Analysis

We applied the methods for an optimization-based analysis to the data as described in Section 5.1. These methods are implemented in the open-source software package *Antils* (*An*alysis *T*ool for *I*WR Tailorshop Resu*l*ts and *S*olutions). All computations were carried out on an *Intel Core i7 920 machine* with 12 GB RAM running *Ubuntu 14.04 64-bit*. For the solution of the arising optimization problems, *AMPL Version 20140331* together with *Bonmin 1.5* and *Ipopt 3.10* was used via *IWR Tailorshop*'s AMPL interface. For the optimization-based analysis, we considered the same 94 datasets as in the statistical analysis.

(1) participants learn to control the model	(6) participants who learn much perform well
(2) learning function is approximately logarithmic over all rounds	(7) participants who perform well learned much
(3) optimization-based feedback groups learn faster or more respectively	(8) participants with high model knowledge learned more than those with low knowledge
(4) *value* group does almost not learn in feedback rounds	(9) initial performance is not important for final performance
(5) *trend* group learns fastest or most respectively	(10) chart group got irritated by and suffered from feedback

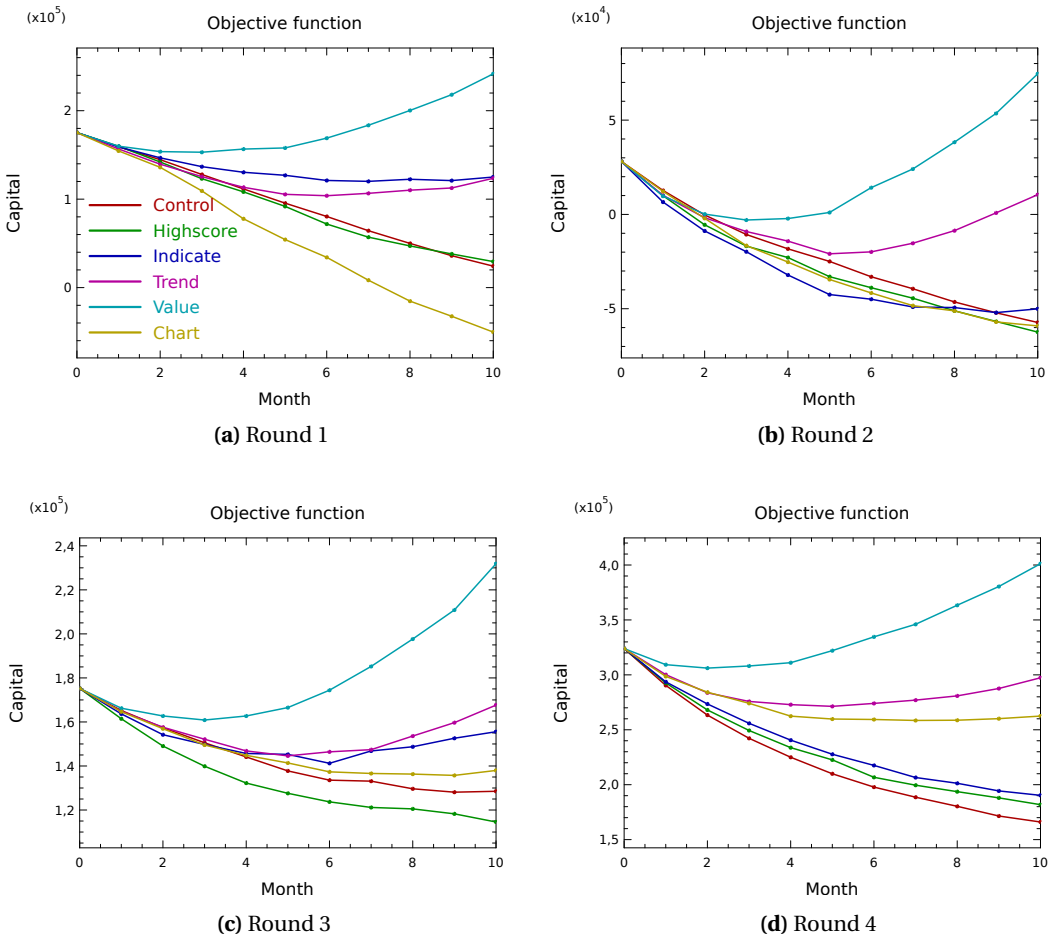
Table 6.28: Hypotheses on results of the optimization-based analysis

An overview of the average objective achieved by the participants in all six groups can be found in Figure 6.16. In all rounds *value* group is on top quite early. Other groups sometimes start similarly, but performance differs at a later time point, compare, e.g., control and trend group in round 2. Note that these plots do not yield insight into structural differences between the participants of the groups. If we were considering single participants, the performance of *trend* group in round 2 could be due to structural investments which pay off from month 5 on. For aggregated values an analysis of structural differences is not reasonable, however.

6.3.1 *How much is still possible*

As described in Section 5.1, we computed optimal solutions for each participant and month in all rounds starting from the model state derived by the participant's decisions. This means, we solved $94 \cdot 4 \cdot 10 = 3760$ optimization problems for this analysis. The resulting solutions, each evaluated at the final month, yield the *How much is still possible* function, which hence is a monotonically decreasing function. Average values of *How much is still possible* for all groups in all rounds are displayed in Figure 6.17.

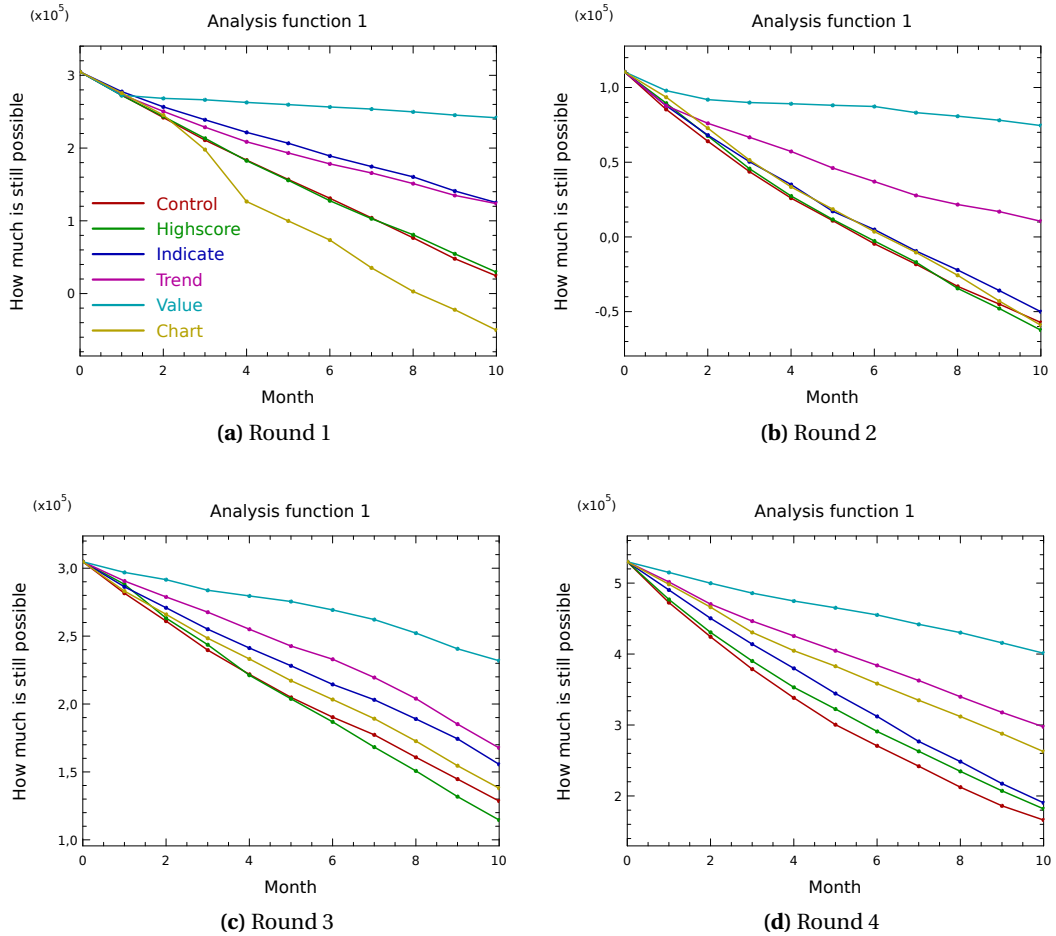
Again, *value* group is clearly the best from the beginning in all rounds. During feedback rounds, *value* group is almost constant, i.e., almost optimal, starting from month 1. This changes in performance rounds, when participants do not have access to the exact optimal values. In round 1, *chart* group falls apart after month 3, when participants possibly got confused by the feedback. *Chart*



(a) Round 1 **(b)** Round 2

(c) Round 3 **(d)** Round 4

Figure 6.16: Average objective according to feedback groups for all complete datasets without 6 outliers ($N = 94$). In all rounds *value* group is on top quite early. However, for the other groups, e.g., *control* and *trend* group in round 2, differences only arise at a later point of time.



(a) Round 1

(b) Round 2

(c) Round 3

(d) Round 4

Figure 6.17: Average *How much is still possible* according to feedback groups for all complete datasets without 6 outliers ($N = 94$). Again, *value* group is clearly the best in all rounds. In round 1, *chart* group falls apart after month 3. *trend* group and *value* group are almost parallel at the end of round 2, and *value* group is almost constant, i.e., almost optimal, in feedback rounds.

group continuously improves in the other rounds. Besides this, there are no abrupt changes in *How much is still possible*.

Note that *trend* and *value* group are almost parallel at the end of round 2. Thus *trend* group is possibly able to control the model almost as well as *value* group under feedback at this time point. However, without feedback in performance rounds, *trend* group does not achieve the level of *value* group.

6.3.2 *Use of potential*

For more insight into the performance of the participants, we determined the *Use of potential*, which is kind of a derivative of *How much is still possible* (see Section 5.1). *Use of potential* for all participants over all rounds is shown in Figure 6.18. Especially round 1 shows some severe outliers. One can also determine from the plot that performance is generally increasing during rounds 1–3 and that there is a big spread in round 4.

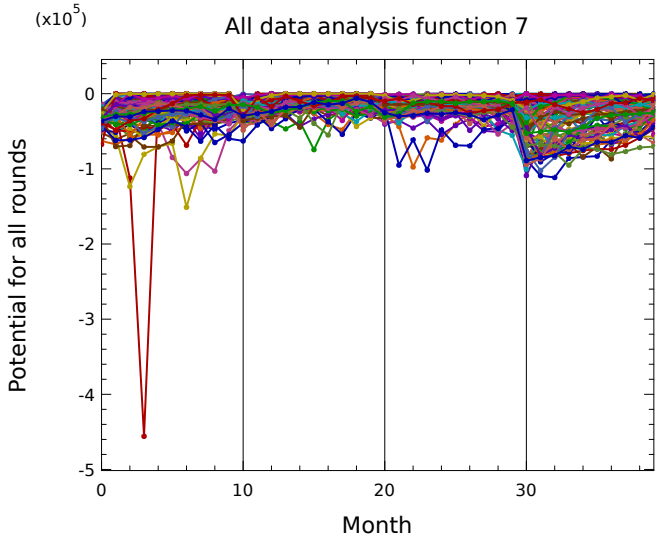


Figure 6.18: *Use of potential* for all complete datasets without 6 outliers ($N = 94$) over all rounds (one round consists of 10 months). Especially round 1 shows some severe outliers.

Figure 6.19 contains the average *Use of potential* for each feedback group over all rounds. This plot reveals much more detail on the performance of the different groups. *Value* group is always on top as expected and almost constant in feedback rounds, but decreases slightly in performance rounds. This means that the performance of participants in this group is on a very high level from the beginning and hardly improves, in fact rather impairs.

All other groups show a more or less severe decline at the beginning of round 4 with *control* and *highscore* group at the one end and *trend* group at the other. However, all groups except *value* group seem to improve their performance during the first three rounds. In contrast to all other groups, *chart* group oscillates in the first round with huge amplitude, which again suggests that participants in this group were confused by the feedback. Figure 6.20 gives an impression of this oscillation in comparison to the overall average. Although there is no direct evidence, it seems quite likely that Hypothesis (10) is true.

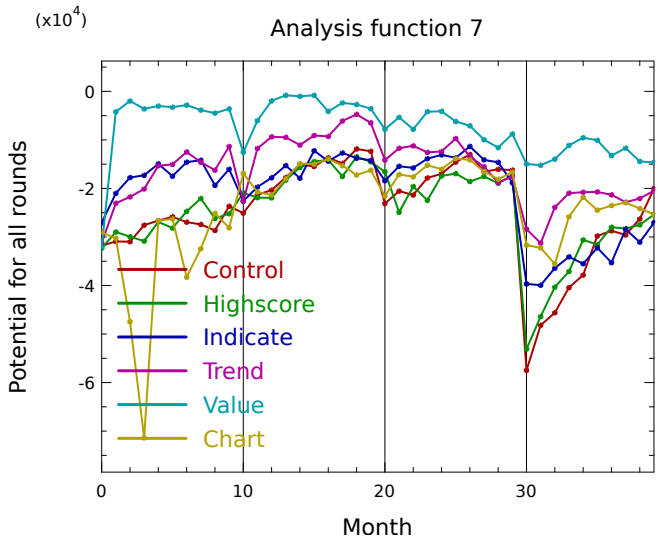
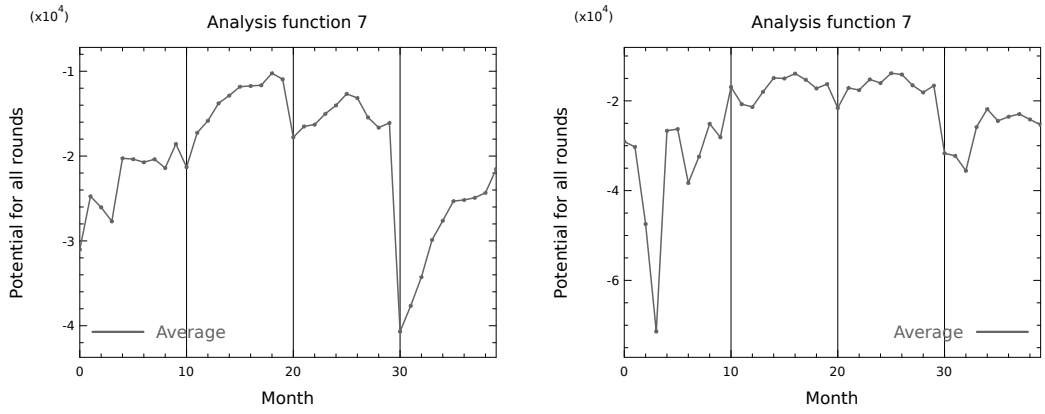
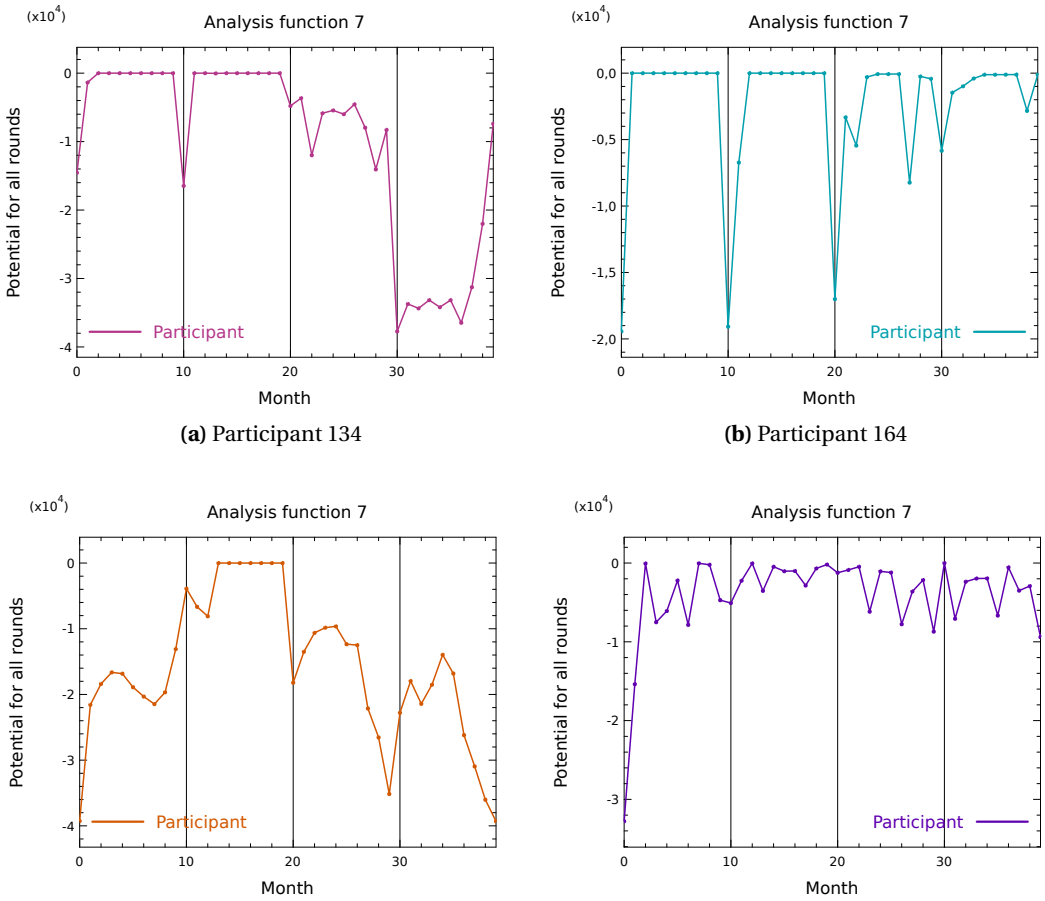


Figure 6.19: *Use of potential* according to feedback groups for all complete datasets without 6 outliers ($N = 94$) over all rounds (one round consists of 10 months). *value* group is always on top and almost constant in feedback rounds, but decreases slightly in performance rounds. All other groups show a more (*control* and *highscore* group) or less (*trend* group) severe decline at the beginning of round 4.



(a) All **(b)** Chart

Figure 6.20: *Use of potential* for all complete datasets without 6 outliers ($N = 94$) and those in *chart* group over all rounds (one round consists of 10 months). In the first round, *chart* group oscillates with huge amplitude. During feedback rounds, the group average increases and shows a sharp decline at the beginning of round 4.



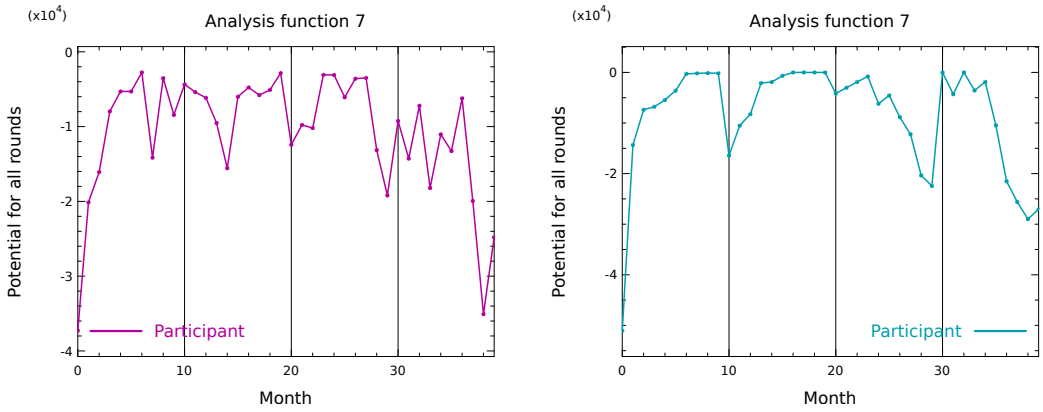
(a) Participant 134

(b) Participant 164

(c) Participant 165

(d) Participant 208

Figure 6.21: *Use of potential* for four single participants from *value* group over all rounds (one round consists of 10 months). These four participants exhibit different patterns: participants 134 and 164 seem to more or less copy the optimal solution in the feedback rounds (remember that the feedback for these participants consisted of the numeric values of the optimal solution), whereas especially participant 208 seems not to copy the solution. The success in performance rounds also varies a lot: participant 164 seems to remember the solution (round 1 and round 3 have the same initial values), but participant 134 obviously does not. Participant 165, who changes strategy during feedback rounds, decreases in round 3, too. Participant 208, however, stays on the same level throughout all rounds.



(a) Participant 110 **(b)** Participant 115

Figure 6.22: *Use of potential* for two single participants from *trend* group over all rounds (one round consists of 10 months). Both participants reach a quite high level of *Use of potential* and seem to learn how to control the model. Participant 115 shows monotonically increasing curves during the first two rounds and comes close to optimality at the ends.

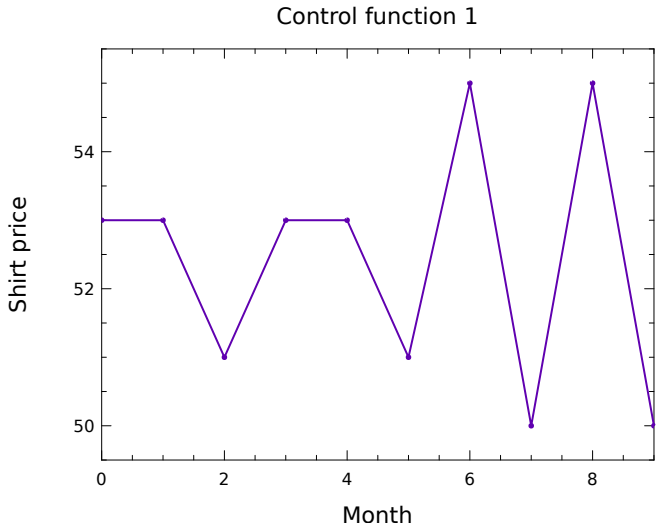


Figure 6.23: *Shirt price* decision of participant 188 (*chart* group) in round 3. Although already in a performance round, the participant seems quite unsure about the right strategy and changes the control a lot. This participant achieved a model knowledge of 2.

	control	highscore	indicate	trend	value	chart
Mean	-31807.3	-32308.6	-27065.5	-31202.2	-32194.4	-29073.8
KS test	0.2192	0.6468	0.5051	1.0000	0.6880	0.9652
***t*-test**	—	0.8988	0.1455	0.8231	0.9335	0.4110

Table 6.29: Comparison of *Use of potential* by feedback groups in first month for all complete datasets without 6 outliers ($N = 94$): no significant differences between groups. Values can be considered to be normally distributed.

A more detailed look on some single participants in *value* group reveals different decision patterns, although sample numbers of the group are too small to derive trustworthy results. Figure 6.21 shows *Use of potential* for participants 134, 164, 165, and 208. Participants 134 and 164 seem to more or less copy the optimal solution in the feedback rounds. Remember that feedback for these participants consisted of the numeric values of the optimal solution. Participant 208, in contrast, seems to pursue a different strategy which is less solution-oriented.

The success in performance rounds also varies a lot: participant 164 seems to remember the solution, which is especially useful in round 3 as it started with the same value as round 1, but participant 134 obviously does not and lacks knowledge how to control the model. Participant 165, who seems to change strategy during feedback rounds from exploration to solution-oriented, decreases in round 3, too. Participant 208, who possibly has found an own strategy, stays on the same level throughout all rounds.

For comparison, we also investigate two participants from *trend* group, participants 110 and 115, see Figure 6.22. Both participants reach a quite high level of *Use of potential* and learn how to control the model. Participant 115 shows monotonically increasing curves during the first two rounds converging to 0, i.e., comes close to optimality at the end of each round. Not surprisingly, a solution-oriented pattern like among the participants from *value* group in Figure 6.21, cannot be observed due to the different type of feedback.

Finally, Figure 6.23 shows the *shirt price* decision of a participant 188 from *chart* group. Although already in a performance round, the participant seems quite unsure about the right strategy and changes the control a lot. Such a pattern at that time point can particularly be found among the datasets from *chart* group. This again supports Hypothesis (10).

6.3.3 Participants' Prerequisite

One could argue that, given the low number of samples for some groups, participants in different groups had different prerequisites, e.g., one group simply consisted of better problem solvers at the beginning of the study. This would obviously have biased the groups' performance. The optimization-based analysis gives us the possibility to check this by comparing the first *Use of potential* value. At this point, all participants had received the same information, as feedback only started after the first decision, so there should be no significant difference in the performance.

Table 6.29 contains mean values, KOLMOGOROV-SMIRNOV test results, and WELCH's t-test results (in comparison to control group). The KOLMOGOROV-SMIRNOV test shows that the first *Use of potential* values can be considered to be normally distributed for all groups. No significant differences to *control* group can be observed by the WELCH's t-test for all groups, so we can suppose that there were no systematic differences among the participants of the six groups. Correlation between first *Use of potential* and score in performance rounds is 0.067. This also proves Hypothesis (9).

Round	control	highscore	indicate	trend	value	chart
1	-30622.4	-30836.2	-18760.4	-23089.9	-2969.4	-48049.1
2	-21556.0	-21706.3	-18897.0	-12217.8	-2708.2	-20016.8
3	-21286.8	-20595.5	-15387.0	-10933.9	-5267.3	-18001.7
4	-53127.9	-47590.5	-39841.6	-27086.2	-13237.8	-31695.5

(a) Means

Round	control	highscore	indicate	trend	value	chart
1	0.3517	0.2169	0.8968	0.4475	0.1231	0.1954
2	0.8416	0.7342	0.4724	0.0730	0.8747	0.9185
3	0.6993	0.6605	0.9975	0.7083	0.9003	0.7388
4	0.5704	0.6928	0.7805	0.4309	0.4863	0.6659

(b) KOLMOGOROV-SMIRNOV test

Table 6.30: Regression c by feedback groups for all complete datasets without 6 outliers ($N = 94$): means and KOLMOGOROV-SMIRNOV test. The values of all groups can be considered to be normally distributed in all rounds.

6.3.4 Learning

To enable conclusions on learning effects, we used R's `lm` to fit a *linear model* for *Use of potential* for each participant and each round,

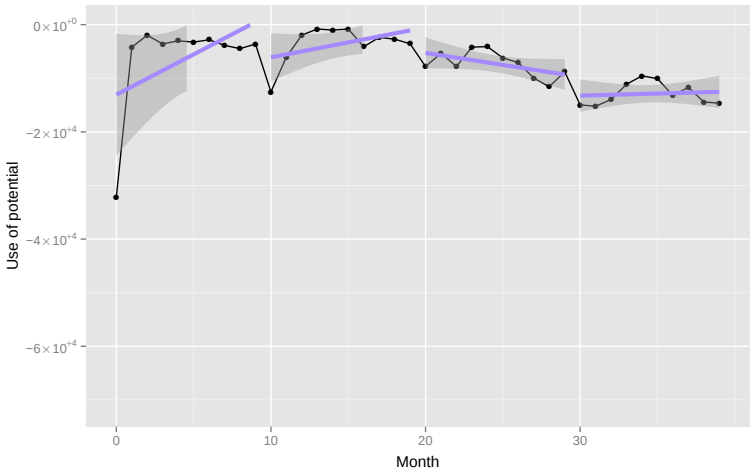
$$y = m \cdot x + c, \tag{6.1}$$

with the regression parameters m and c, which estimate the gradient and the intercept of *Use of potential*. An estimate for the gradient, vice versa, characterizes how much more potential the participant was able to use over time, i.e., how much the participant *learned*. Therefore regression m is used as a measure for *learning* in the remainder and *Use of potential* may also be considered as the *learning curve*.

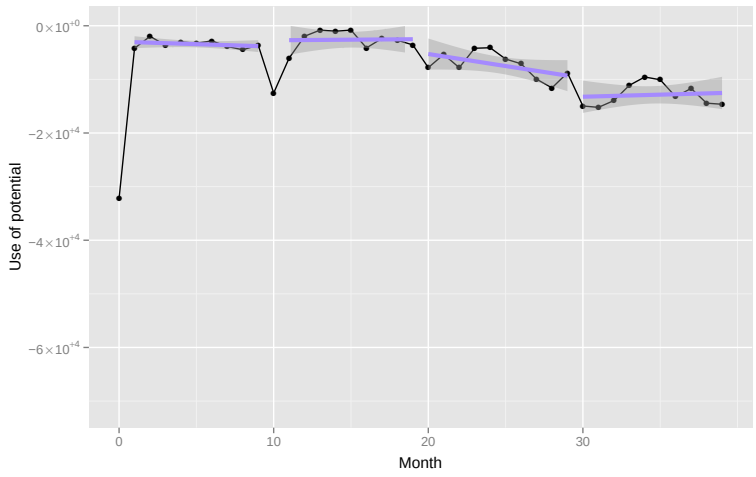
With this definition of learning, it is clear from the plot in Figures 6.19 and 6.20a that it makes no sense to fit a logarithmic model to the learning curve, neither per group nor globally. The single participant examples from Section 6.3.2 suggest that this is also not reasonable for most single participants. One could argue that an appropriate scaling of the rounds due to their varying difficulty is necessary, which could lead to different shape of learning curves. There is no objective measure how to choose such a scaling though, making scaling totally arbitrary. Hypothesis (2) is thus disproved.

An important aspect is the choice of range for the fit in each round. Figure 6.24 illustrates the problem for the *value* group. No feedback is given before the *first* decision and thus *Use of potential* changes drastically from month 0 to month 1. As we use a linear model, including month 0 leads to a very bad fit, which may not represent the gradient of *Use of potential* at all (Figure 6.24a). Without first month (Figure 6.24b), the fit is good for both *control* and *highscore* group and optimization-based feedback groups (see also Figure 6.25). In performance rounds, this effect does not occur, so all months are considered.

Mean values and KOLMOGOROV-SMIRNOV test results for regression parameter c can be found in Table 6.30. The values can be considered to be normally distributed for all groups in all rounds.



(a) All months



(b) Without first month in feedback rounds

Figure 6.24: Regression lines for *Use of potential* for *value* group over all rounds (one round consists of 10 months). In feedback rounds, participants received feedback only from month 1 on, i.e., after their first decision, due to the way feedback was computed for *chart* group. (a) shows a regression with all months of each round, for (b) the first month of feedback rounds has been excluded. It is necessary to exclude the first month of each feedback round, as the regression result with all months does not represent the actual gradient of the *Use of potential* function. See Figure 6.25 for a comparison with other groups.

However, for the remaining analysis, we concentrate on parameter m.

Table 6.31 contains mean values, KOLMOGOROV-SMIRNOV test results, and WELCH's t-test results for regression parameter m by feedback groups. The values can also be considered to be normally distributed in all rounds except for *chart* group in round 1. *Trend* group is the only group with a significant learning effect in both rounds 1 and 2. For *control* group, the learning effects get significant from round 2 on, and for *highscore* group they are significant in rounds 2 and 4. The mean values in performance rounds for *control* and *highscore* group are drastically higher than for the optimization-based feedback groups. *Value* group is the only one with a significantly decreasing performance in round 3 and also the only one with an overall mean below 0. Overall, participants show significant learning effects in all rounds except for round 3, as Table 6.32 shows. This proves that participants learn to control the model, which is hypothesis (1).

Figures 6.26–6.30 contain the fits for *control*, *highscore*, *indicate*, *trend*, and *chart* group. *Control* and *highscore* group increase during all rounds, but exhibit a decline at the beginning of round 3 and the most severe decline at the beginning of round 4. *Indicate* group shows a slight increase throughout all rounds, but stays approximately on the same level during the first three rounds. After an increase in the feedback rounds on a rather high level, *trend* group decreases slightly without feedback in round 3, but increases again in the last round. As already mentioned, *chart* group shows a lot of oscillation in round 1, so the fit for this round is almost meaningless. This changes for rounds 2–4.

A plot with connected regression lines for *Use of potential* for all groups in Figure 6.31 shows that in feedback rounds, all groups except *value* group seem to learn. The decrease at the beginning of round 4 is much smaller for groups with optimization-based feedback. Together with the results from Table 6.31, Hypotheses (4) and (5) can thus be considered proved.

6.3.5 Optimization-based Feedback

In Table 6.34, *control* and *highscore* group are compared with optimization-based feedback groups. The mean for parameter m for the latter is higher in round 1 and lower in all other rounds. This suggests that, given the performance of these groups, optimization-based feedback groups learned faster, namely mainly in the first round. However, WELCH's t-test only shows significance for rounds 2–4. Thus there is only indication that Hypothesis (3) might be true, but it cannot be fully proved with our data.

6.3.6 Performance

Another essential question is if there is a connection between performance and learning. To check if high performers also learned much, the datasets have been divided into three groups based on quartiles. *High* group contained all datasets above the higher quartile, *mid* group all between lower and higher quartile, and *low* group all below the lower quartile. Table 6.33 shows the mean values for parameter m and WELCH's t-test results can be found in Table 6.35. High performers have the highest mean for m in the first round and the lowest in all other rounds. The mean of *high* group, however, is not significantly higher in round 1, but significantly lower in rounds 2–4.

The effects of learning on the performance on the other hand, are documented in Tables 6.36, 6.37, and 6.38. There are big differences, depending on whether only learning in round 1, learning in feedback rounds, or learning in all rounds is considered. Remembering our findings up to this point, this is not surprising.

Considering all rounds (Table 6.37), *low* group is significantly better than the other two in almost all cases. This matches the means from Table 6.33, where low performers have by far the highest

Round	control	highscore	indicate	trend	value	chart
1	599.1	767.5	359.9	1286.5	-91.3	2366.5
2	1140.9	965.4	714.2	725.0	22.2	610.7
3	814.4	350.8	104.6	-616.5	-448.5	294.7
4	3717.4	2837.5	1304.5	847.9	78.0	1097.9
Feedback rounds sum	1740.0	1732.9	1074.1	2011.5	-69.1	2977.2
Performance rounds sum	4531.8	3188.3	1409.2	231.4	-370.5	1392.6
Total sum	6271.8	4921.2	2483.2	2242.9	-439.6	4369.9

(a) Means

Round	control	highscore	indicate	trend	value	chart
1	0.1551	0.2901	0.7662	0.4528	0.0748	**0.0493**
2	0.5016	0.9603	0.9348	0.4203	0.6070	0.6826
3	0.8186	0.9434	0.7300	0.7786	0.4601	0.9627
4	0.9961	0.8713	0.8615	0.9498	0.9832	0.6299

(b) KOLMOGOROV-SMIRNOV test

Round	control	highscore	indicate	trend	value	chart
1	0.1051	0.1820	0.2194	**0.0036**	0.6708	0.0960
2	**0.0002**	**0.0248**	0.1263	**0.0045**	0.4718	0.0787
3	**0.0002**	0.1528	0.3853	0.9399	0.9646	0.2284
4	**0.0000**	**0.0016**	**0.0435**	0.1053	0.4542	0.0858

(c) WELCH's t-test for μ>0

Round	control	highscore	indicate	trend	value	chart
1	0.8949	0.8180	0.7806	0.9999	0.3292	0.9040
2	0.9998	0.9752	0.8737	1.0000	0.5282	0.9213
3	0.9998	0.8472	0.6147	0.0601	**0.0354**	0.7716
4	1.0000	0.9984	0.9565	0.8947	0.5458	0.9142

(d) WELCH's t-test for μ<0

Table 6.31: Parameter m by feedback groups for all complete datasets without 6 outliers (N = 94): means, WELCH's t-test, and KOLMOGOROV-SMIRNOV test. The values of all groups can be considered to be normally distributed in all rounds except for *chart* group in round 1. *trend* group is the only group with a significant learning effect in the first two rounds, *value* group the only one with a significantly decreasing performance in round 3.

Round	Mean	t-Test $\mu > 0$
1	879.1	**0.0016**
2	789.9	**0.0000**
3	154.0	0.1365
4	1991.2	**0.0000**

Table 6.32: Regression m for all complete datasets without 6 outliers ($N = 94$): means and WELCH's t-test results ($\alpha = 0.05$). Participants show significant learning effects in all rounds except for round 3, in which especially *value* group is significantly < 0.

Round	Low	Mid	High
1	305.2	924.5	1365.9
2	1439.3	637.2	433.2
3	549.4	148.2	-230.2
4	4409.5	1888.7	-230.4
Feedback	1744.4	1561.7	1799.2
Performance	4958.9	2036.9	-460.7
Sum	6703.3	3598.6	1338.5

Table 6.33: Means for regression m according to performance in performance rounds (*low*: below lower quartile, *mid*: between lower and higher quartile, *high*: above higher quartile) for all complete datasets without 6 outliers ($N = 94$): high performers have the highest mean for m in the first round and the lowest in all other rounds.

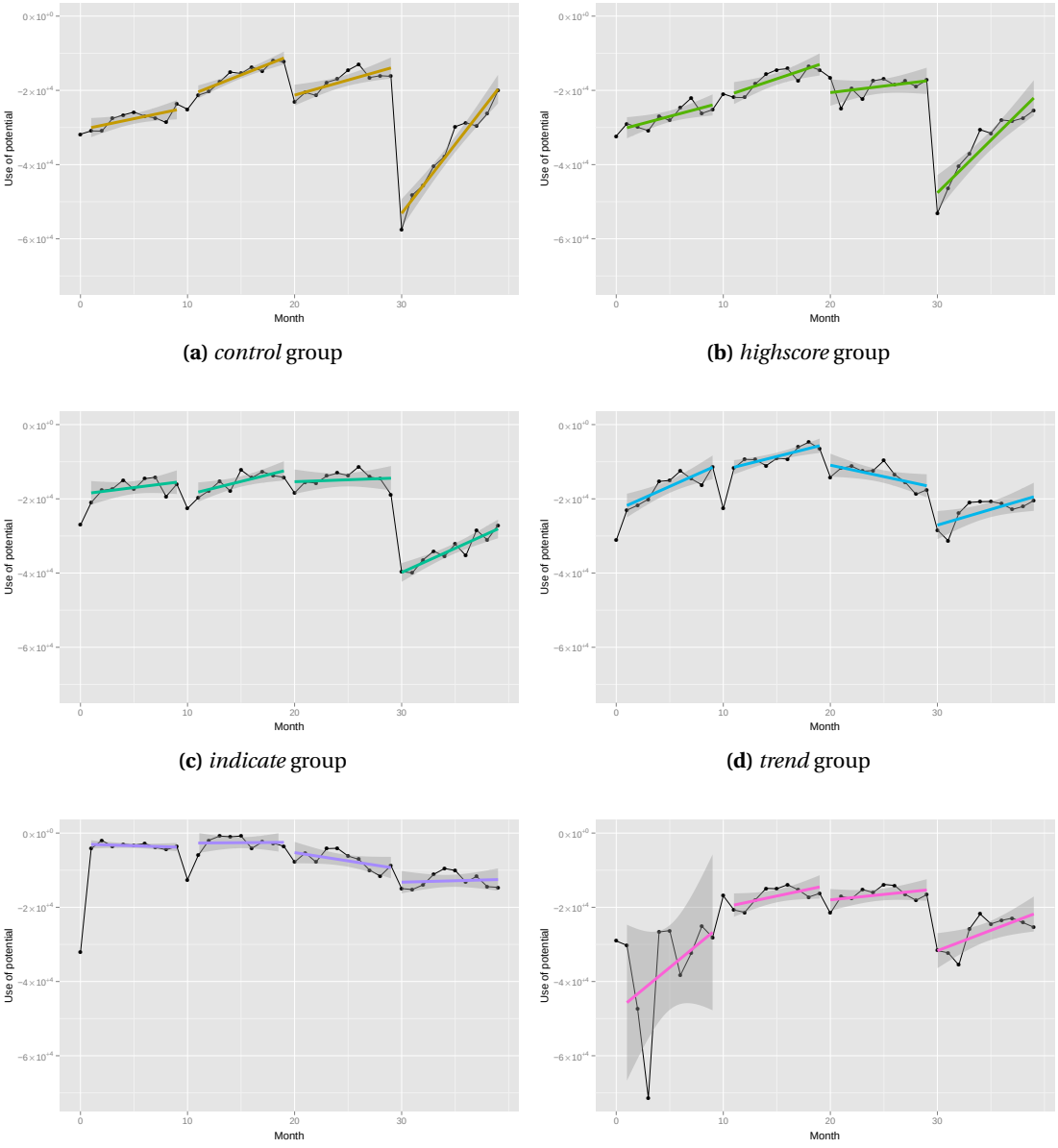
Round	Control & Highscore (ch)	Opt.-based Feedback (of)
1	651.2	1063.1
2	1086.6	550.3
3	670.9	-263.4
4	3445.1	817.0

(a) Regression m means

ch < of	of < ch
0.2384	0.7616
0.9642	**0.0358**
0.9997	**0.0003**
1.0000	**0.0000**

(b) WELCH's t-test

Table 6.34: Regression m comparison between *control* and *highscore* group on one side and groups with *optimization-based feedback* on the other side for all complete datasets without 6 outliers ($N = 94$): those with optimization-based feedback learned more in round 1, *control* and *highscore* group learned more in rounds 2–4. These differences are significant except for round 1.



(a) *control* group

(b) *highscore* group

(c) *indicate* group

(d) *trend* group

(e) *value* group

(f) *chart* group

Figure 6.25: Comparison of regression lines for *Use of potential* by feedback groups for all complete datasets without 6 outliers ($N = 94$) over all rounds (one round consists of 10 months). *chart* group shows a lot of oscillation in round 1. The decrease at the beginning of round 4 is much smaller for groups with optimization-based feedback. See Figures 6.24b, 6.26–6.30 for large plots of each group.

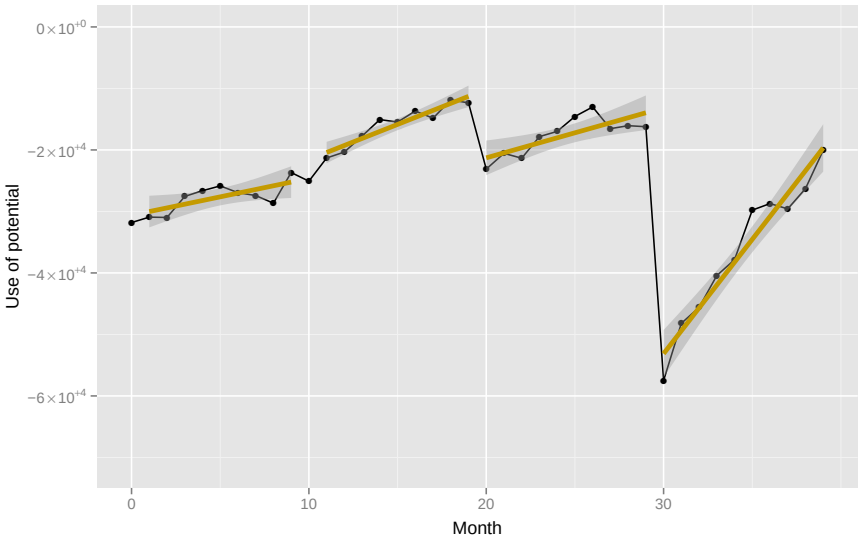


Figure 6.26: Regression lines for *Use of potential* for *control* group over all rounds (one round consists of 10 months). *Use of potential* increases during all rounds, but exhibits a decline at the beginning of round 3 and a severe decline at the beginning of round 4. See Figure 6.25 for a comparison with other groups.

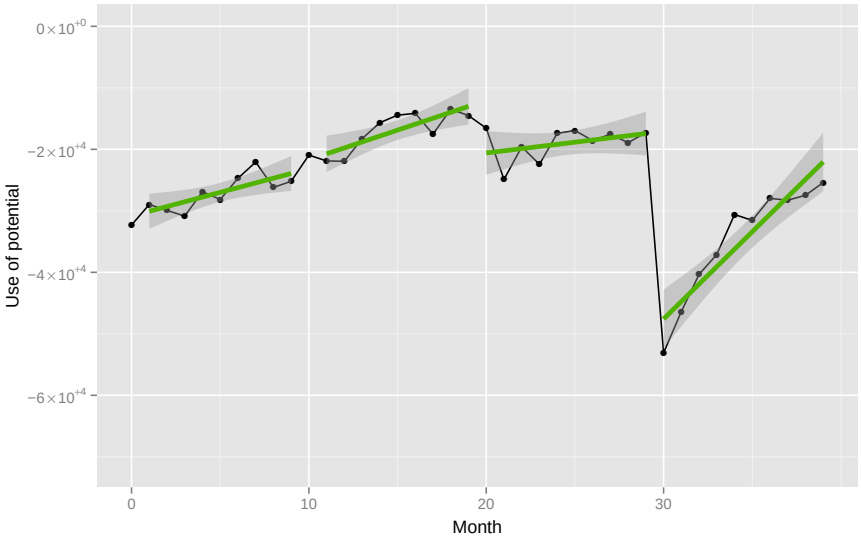


Figure 6.27: Regression lines for *Use of potential* for *highscore* group over all rounds (one round consists of 10 months). *Use of potential* increases during all rounds, but exhibits a decline at the beginning of round 3 and a severe decline at the beginning of round 4. See Figure 6.25 for a comparison with other groups.

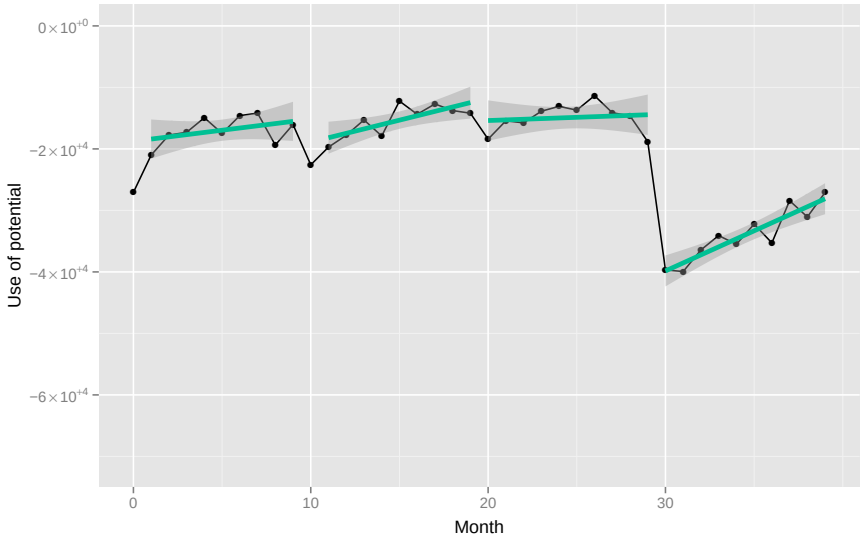


Figure 6.28: Regression lines for *Use of potential* for *indicate* group over all rounds (one round consists of 10 months): the group shows a slight increase throughout all rounds, but stays approximately on the same level during the first three rounds. See Figure 6.25 for a comparison with other groups.

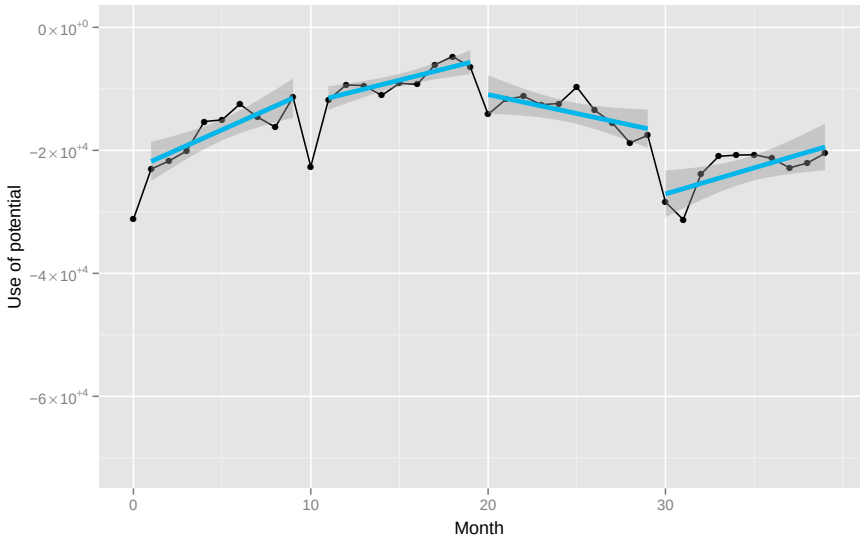


Figure 6.29: Regression lines for *Use of potential* for *trend* group over all rounds (one round consists of 10 months). After an increase in the feedback rounds on a high level, *Use of potential* decreases in round 3 without feedback, but increases again in the last round. See Figure 6.25 for a comparison with other groups.

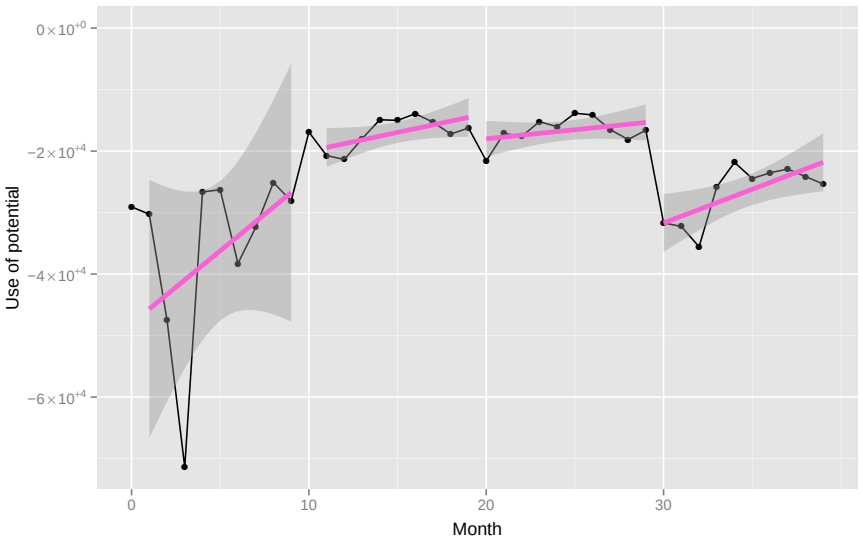


Figure 6.30: Regression lines for *Use of potential* for *chart* group over all rounds (one round consists of 10 months). This group exhibits lots of oscillation with high amplitude in round 1, which may be an indicator that participants in this group have been confused by the feedback. Oscillation almost vanishes in rounds 2–4. See Figure 6.25 for a comparison with other groups.

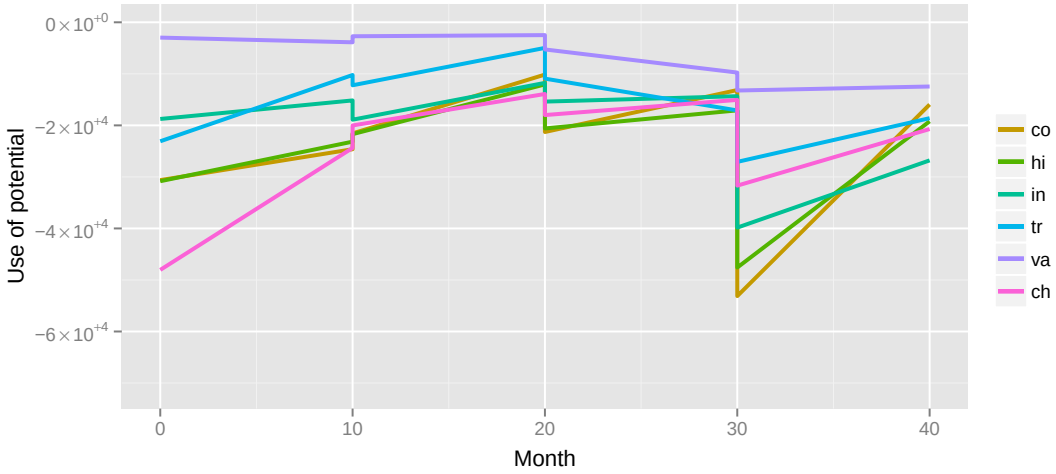


Figure 6.31: Connected regression lines for *Use of potential* (co: control, hs: highscore, in: indicate, tr: trend, va: value, ch: chart) for all complete datasets without 6 outliers ($N = 94$) over all rounds (one round consists of 10 months). In feedback rounds, all groups except *value* group seem to learn. The decrease at the beginning of round 4 is much smaller for groups with optimization-based feedback.

Round	Low < High	High < Low	Low < Mid	Mid < Low	Mid < High	High < Mid
1	0.1238	0.8762	0.0953	0.9047	0.3181	0.6819
2	0.9933	**0.0067**	0.9852	**0.0148**	0.7312	0.2688
3	0.9796	**0.0204**	0.8511	0.1489	0.9127	0.0873
4	1.0000	**0.0000**	0.9999	**0.0001**	0.9999	**0.0001**

Table 6.35: WELCH's t-test for regression m according to performance in performance rounds (*low*: below lower quartile, *mid*: between lower and higher quartile, *high*: above higher quartile) for all complete datasets without 6 outliers ($N = 94$): the mean of *high* group, see Table 6.33, is not significantly higher in round 1, but significantly lower in rounds 2–4.

Round	Low	Mid	High
1	101174.3	100803.0	-6349.5
2	-24182.7	-17629.0	-47593.9
3	148725.0	148247.0	158990.9
4	300693.4	208536.8	234434.3
3 & 4	449418.4	356783.8	393425.2
Sum	526410.0	439957.8	339481.8

(a) Score means

Low < High	Mid < High	Low < Mid
0.9876	0.9955	0.5045
0.8493	0.9578	0.3741
0.3708	0.2696	0.5061
0.8734	0.3147	0.9620
0.7539	0.2970	0.8862
0.9437	0.8775	0.7696

(b) WELCH's t-test

Table 6.36: Performance according to learning, i.e., regression m, in feedback rounds (*low*: below lower quartile, *mid*: between lower and higher quartile, *high*: above higher quartile) for all complete datasets without 6 outliers ($N = 94$): *high* group is significantly lower than the other two in round 1, but almost all other differences are not significant. Low learning in feedback rounds results in good performance, especially in the last round, which is due to the *value* group's low learning in feedback rounds.

sum of learning over all rounds. For feedback rounds, *high* group has a significantly lower score than the other groups in round 1, but almost all other differences are not significant. Considering only learning in round 1, *mid* group has the highest score mean in all rounds, but is significantly better than *low* and *high* group only in two rounds. In this analysis, it might seem like learning was counterproductive at first, but note also that *value* group has a very low average learning, but high average performance values.

So we can neither prove nor disprove Hypotheses (7) and (6). Anyhow, there is some evidence that (moderate) learning in the first round was crucial for performance and that low performers learn later.

6.3.7 Knowledge and Uncertainty

Finally, we want to have a look at the connection between learning and model knowledge and uncertainty. Therefore, datasets have been divided into groups with *high* (4–5), *mid* (2–3), and *low* (0–1) model knowledge and uncertainty respectively. Note that there were no datasets with high uncertainty.

The average *Use of potential* for the three model knowledge groups can be found in Figure 6.32.

Round	Low	Mid	High
1	124159.3	82159.9	6398.1
2	5425.6	-23146.4	-66627.3
3	179437.8	154541.1	116214.4
4	361354.9	237016.9	119185.7
3 & 4	540792.8	391558.0	235400.2
Sum	670377.7	450571.5	175171.0

(a) Score means

Low > High	Mid > High	Low > Mid
0.0010	**0.0154**	0.0950
0.0002	**0.0075**	**0.0497**
0.0078	0.0587	0.1321
0.0000	**0.0149**	**0.0033**
0.0001	**0.0157**	**0.0098**
0.0000	**0.0009**	**0.0115**

(b) WELCH's t-test

Table 6.37: Performance according to learning, i.e., regression m, in all rounds (*low*: below lower quartile, *mid*: between lower and higher quartile, *high*: above higher quartile) for all complete datasets without 6 outliers ($N = 94$): in almost all cases, *low* group performed significantly better than *mid* group, and *mid* group vice versa performed significantly better than *high* group.

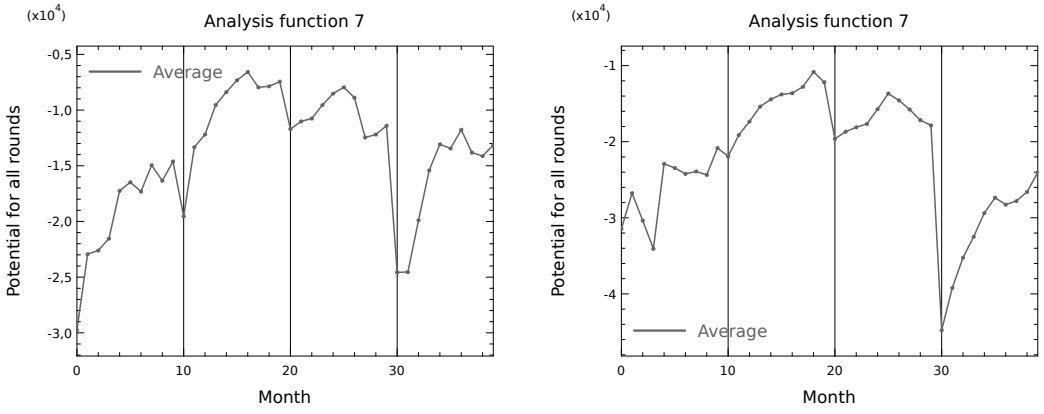
Round	Low	Mid	High
1	61209.2	124516.0	-11834.4
2	-51613.0	-12166.2	-30633.9
3	117780.8	172313.9	143807.1
4	183177.5	276613.1	221470.5
3 & 4	300958.3	448927.0	365277.6
Sum	310554.5	561276.7	322809.3

(a) Score means

Low < High	Mid > High	Low < Mid
0.9537	**0.0006**	**0.0166**
0.1470	0.1360	**0.0216**
0.1815	0.0625	**0.0314**
0.2873	0.1552	0.0524
0.2378	0.1165	**0.0327**
0.4594	**0.0052**	**0.0148**

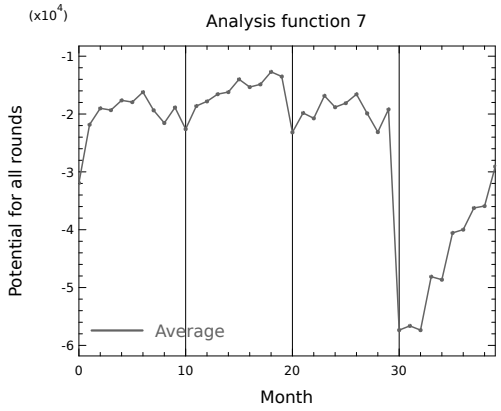
(b) WELCH's t-test

Table 6.38: Performance according to learning, i.e., regression m, in first round (*low*: below lower quartile, *mid*: between lower and higher quartile, *high*: above higher quartile) for all complete datasets without 6 outliers ($N = 94$): *mid* group performed best in all rounds, but is significantly better than *low* and *high* group only in some rounds.



(a) High

(b) Mid



(c) Low

Figure 6.32: *Use of potential* according to *high*, *mid*, and *low* model knowledge for all complete datasets without 6 outliers ($N = 94$) over all rounds (one round consists of 10 months). Participants with low knowledge show a severe decline at the beginning of round 4, whereas they stay on the same level in the rounds before. *High* and *mid* group show an increase in feedback rounds and *high* group also stays almost on the same level in round 4.

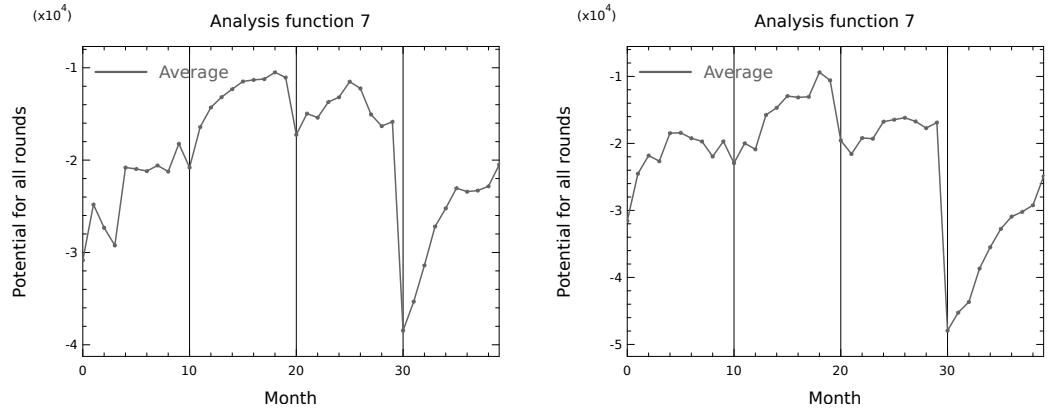
Round	Low	Mid	High
1	93.3	1013.0	1086.4
2	666.2	888.5	691.6
3	111.0	301.0	-70.5
4	3257.4	1979.4	1312.6

(a) Means for Regression m

Low < High	Mid < High	Low < Mid
0.0207	0.4518	0.0686
0.4788	0.7564	0.3262
0.6712	0.8680	0.3020
0.9828	0.8480	0.9291

(b) WELCH's t-test

Table 6.39: Regression m according to model knowledge (*low*: below lower quartile, *mid*: between lower and higher quartile, *high*: above higher quartile) for all complete datasets without 6 outliers ($N = 94$): those with low model knowledge learned less in the first round, and more in the last round. In comparison with *high* group, this is significant.



(a) Low **(b)** Mid

Figure 6.33: *Use of potential* according to *low* and *mid* model uncertainty for all complete datasets without 6 outliers ($N = 94$) over all rounds (one round consists of 10 months). Note that there were no datasets with high uncertainty. *low* group shows a smaller decrease at the beginning of round 4. Apart from this, no qualitative differences can be observed.

Participants with low knowledge show a severe decline at the beginning of round 4, whereas they stay on the same level in the rounds before. *High* and *mid* group show an increase in feedback rounds and *high* group also stays almost on the same level in round 4.

The values in Table 6.39 reveal that participants with low model knowledge learned significantly less in round 1 than those with high knowledge. Again, the situation reverses in round 4. Hypothesis (8) can thus be confirmed with a restriction to round 1.

For model uncertainty, see Figure 6.33, *low* group shows a smaller decrease at the beginning of round 4. Apart from this, no qualitative differences can be observed.

CHAPTER 7

Conclusion and Outlook

In this work, optimization methods were used in the context of *Complex Problem Solving* (CPS) at different levels: first, in the design stage of the complex problem scenario, second, as an analysis tool, and third, to provide feedback in real time for learning purposes. While first works on optimization-based analysis for CPS [111, 112] had a focus on understanding how external factors influence thinking, in the work at hand, we also investigated learning effects. The use of optimization as an analysis *and* feedback tool for psychological studies is completely new to our knowledge.

We explained what characteristics complex problems have and how the domain CPS emerged from the basic developments in the cognitive revolution. Among the coexisting models of human problem solving, functionalism which sees problem solving as information processing is the most important in the context of microworlds like *IWR Tailorshop*.

Different problem classes of optimization problems were formulated and we derived the discretized mixed-integer optimal control problem which can be considered a mixture of mixed-integer nonlinear programs and mixed-integer optimal control problems and can be used to describe both *Tailorshop* and *IWR Tailorshop*. For MINLPs, the generic optimization algorithms *branch and bound* and *outer approximation* were described and for the underlying NLPs, we considered *interior point* and SQP methods.

We presented a new microworld for complex problem solving, the *IWR Tailorshop*. This turn-based test-scenario yields a mixed-integer nonlinear program with nonconvex relaxation and consists of functional relations based on optimization results. With the *IWR Tailorshop*, we intend to start a new era beyond *trial-and-error* in the definition of microworlds for analyzing human decision making.

Compared to the *Tailorshop*, the variety of variables was shifted towards a more abstract level. The participants have to take care of the number of *production sites* of their company instead of buying or selling single *machines*, for instance. This abstract level was chosen for *IWR Tailorshop*, because it yields a more realistic position as a decision maker for the participants. The final model consists of 14 state variables and ten control variables including five integer controls.

For the optimization-based analysis, we solved a series of optimization problems for each participant. By starting an optimization in the microworld state a participant achieved for each month, we were able to determine how much could have been achieved if the participant's decisions would have been optimal. The *How much is still possible*-function which consists of these values gives insight *when* decision were bad or good respectively. The derived *Use of potential*-function indicates how much of the potential of optimal decisions was used by a participant. Our extension of this approach to the computation of an optimization-based feedback is quite similar. Starting from the state a participant derived until a certain month, we either computed optimal solutions for the next month or derived (pseudo-)sensitivities for the previous controls. Types of feedback presentation included a simple highlighting, arrows indicating the direction of the optimum, the exact optimal control value, and a bar chart with sensitivity information.

We investigated different reformulations for a minimum-expression in the *sales* equation. A linear combination using binary variables (and a variation thereof) was derived beneath a generalized disjunctive programming formulation and a simple inequality reformulation. Numerical results showed that a binary linear combination is the method of choice for *IWR Tailorshop*.

Hypothesis		**Proved**	**Hypothesis**		**Proved**
(A)	participants with opt.-based feedback perform better overall	✓	(O)	well-performers know more about the model	✓
(B)	participants with opt.-based feedback perform better in feedback rounds	✓	(P)	participants who know much about the model perform well	✓
(C)	participants with opt.-based feedback perform better in performance rounds	✓	(Q)	value group knows less, trend group knows most about the model	—/✓
(D)	control group performs worst	—	(1)	participants learn to control the model	✓
(E)	control group performs worse than opt.-based groups in performance rounds	✓	(2)	learning function is approximately logarithmic	—
(F)	trend group performs best overall	—	(3)	optimization-based feedback groups learn faster	(✓)
(G)	trend group performs best in performance rounds	—	(4)	*value* group does almost not learn in feedback rounds	✓
(H)	value group performs best in feedback rounds	✓	(5)	*trend* group learns fastest	✓
(I)	value group performs better in feedback rounds, worse in performance rounds	(✓)	(6)	participants who learn much perform well	?
(J)	participants with high BFI-10 values perform worse/better	—	(7)	participants who perform well learned much	?
(K)	participants who play computer games regularly perform better	—	(8)	participants with high model knowledge learned more	✓
(L)	participants interested in economics perform better	—	(9)	initial performance is not important for final performance	✓
(M)	participants who solve problems systematically perform better	—	(10)	chart group suffered from feedback	(✓)
(N)	control group needs more time than opt.-based feedback groups	—			

Table 7.1: Overview of results for hypotheses in this work.

To be able to get upper bounds for the resulting problems within reasonable times, we proposed a tailored decomposition approach, where the problem is divided into a master problem and several subproblems. This decomposition is built such that it yields a valid upper bound for the corresponding global solution of the original problem and thus can be used as an indicator for the quality of local solutions of the original problem.

We showed promising numerical results using this decomposition approach, which indicated a high potential. In a first (worst-case like) scenario with fixed variables, the gap between decomposition and original problem was between 4.0% and 16.3%, while the original problem could also be solved to global optimality. In a second scenario, it alternated between 4.0% and 8.0%. For this scenario, only with the decomposition it was possible to get a globally optimal solution for more than 2 turns. The computation times for the decomposition are below 2 min even for 10 turns with the global solver *Couenne*. Here, in future work, an approach could be to create microworlds which are actually defined as a decomposed model. The benefit of such a test-scenario would be twofold. On the one hand, if the model size and structure is comparable to the *IWR Tailorshop*, the decomposition gap would then be 0, of course, i.e. under certain circumstance one would be able to compute globally optimal solutions for the actual microworld. Furthermore, with an appropriate structure, these scenarios could also be used for investigation of group decision making while several participants control different parts of the model.

The parameter set used for the computations in this work has been set up manually to achieve a reasonable model behavior. Here we still see high potential for improvement. One could use derivative-free optimization methods to optimize the *parameter values* such that two (or even more) previously defined strategies (e.g., a high and a low price strategy) yield a similar objective value. By that, participants could follow different strategies and perform quite well in all of them if decisions are made appropriate.

In our web-based feedback study with 148 participants, we used the *IWR Tailorshop* microworld to investigate the effects of optimization-based feedback. An overview of the results for all hypotheses is given in Table 7.1. We could show that such a feedback can significantly improve participants' performance in a complex microworld and for some kinds of feedback, the difference to *control* group was huge. However, it also became apparent that the representation of feedback is important. Feedback based on a kind of sensitivity information seemed to rather confuse participants in this study, which was also suggested by our optimization-based analysis.

The best-performing group was the *value* group which received the most precise information about the optimal solution. Knowledge about the model was better amongst another well-performing group, the *trend* group. Since we could show that model knowledge is a predictor for performance, perhaps these participants would have outperformed the others on a longer timescale. More data is needed to verify this hypothesis, though.

There were significant differences between participants of different age. Because of the unbalanced distribution of age in this study, potential influence of fluid and crystallized intelligence should be investigated again, e.g., by a study with controlled age distribution and less feedback groups. In contrast, interest in economics, playing computer games regularly, and solving problems systematically in general (all as claimed by the participants) had no effects on the result.

Surprisingly, women could profit from optimization-based feedback only in the first round in this study. More research is necessary to determine if there is a systematic difference between women and men regarding an optimization-based feedback. This could be done, e.g., via a study with only one feedback and a control group with approximately equal gender ratios.

Optimization-based analysis could show that participants learn to control the model over time by an analysis of *Use of potential*. Different aspects of the analysis indicate that for a high performance, learning during the first round is crucial. It turned out that the best way to enforce learning at the be-

ginning was by *trend* feedback. Through the optimization-based analysis, we were also able to show that there were no systematic differences between the groups at the beginning and that initial performance was not relevant for performance at the end of the time scale. For some of the hypotheses, however, significance could not or only partly be shown. In these cases, more data and investigation will be necessary.

Another interesting aspect of future research could be if the widely spread assumption that positive feedback increases performance is true. In [12] it has been shown that negative feedback impairs performance. However, it is unclear if this is also true in the long run. From former studies we know that positive and negative feedback lead to different processing styles. Therefore one could expect that a quotient of positive and negative feedback (*carrot and stick*) impairs performance the most. 40% positive feedback and 60% negative feedback might lead to the best performance, for instance.

Finally, another exciting question which could be treated within the *IWR Tailorshop* framework is if humans unconsciously try to control a linearized version of the model. However, this question is only reasonable for participants who acquired enough model knowledge and know about most dependencies between the model variables. In our study, feedback was used to train participants to the nonlinear model, thus the study data cannot be used in this respect. Therefore, this aspect should possibly be treated in another study where participants are shown all dependencies at the beginning. Then one could compare their decisions with optimal solutions for a linearized *IWR Tailorshop* model.

APPENDIX A

Implementation Details of the Optimization Framework

This chapter describes the software developed for analysis and training of human decision making within this thesis. There are two parts of the software: a web front end for participants with support for optimization-based feedback and a back end for the analysis of datasets collected within the front end. Both front and back end are published as open-source software under *GPL version 3*.

A.1 Web Front End

Computer-based test scenarios like *IWR Tailorshop* emerged in CPS in a time when Internet access was only available for academic and military institutions if at all and the world wide web had not been invented yet. Therefore, researchers had to conduct studies with test scenarios running on local computers under their direct control. While this is beneficial for controlled experimental conditions, it is also a severe limitation for collecting data—it costs both participants and experimenter much time and parallelization is severely limited due to both equipment and staff.

In recent decades, Internet access became common for individuals and in recent years, mobile devices with Internet access like smartphones and tablets became widely spread. This yields a large potential for studies with microworlds as participants can in principle take part with their own device from nearly all over the world. Of course, experimental conditions then are less controlled and the experimenter has to judge how much control is needed from study to study.

A.1.1 AJAX-based Interface

At the beginning of the work on an *IWR Tailorshop* front end for the use by participants, we decided to implement a web-based interface for several reasons. First, a web-based front end is portable to a great extent as it basically only needs a device with a web browser (there always will be some requirements on the browser, though, as *Netscape 4.73* might have some problems with your AJAX driven interface) and thus can be run without additional software independently from the operating system on a large number of devices. To be able to run the *GW-BASIC* interface of the original *Tailorshop*, for instance, one nowadays needs *DOSBox*, a *DOS* emulator. Second, it drastically improves options for parallelization. In principle, datasets can be collected massively parallel from participants all over the world recruited via the Internet, if the corresponding web server can deal with it. On the other hand, it is still possible to conduct studies in a local network, if more control on the conditions is needed. And finally, in especially if AJAX is used for communication with the server, a web-based implementation is easily extendible, e.g., with additional platform-specific implementations of the front end.

A crucial requirement on the implementation of a microworld for CPS research is often to hide the complexity of the test scenario, i.e., the microworld's model with variable dependencies, from the participants. This requires especially that participants cannot determine the model from *HTML* and *JavaScript* code the web server delivers as clear text. Technically speaking, it should also not be possible to determine the model by reverse compiling of binary code, but this requires an incomparably higher effort than choosing "view source code" from some menu and can thus be neglected here.

There are three convenient options to avoid this problem: an *Adobe Flash* interface, a *Java Applet*, and an *AJAX-based XHTML and JavaScript* interface running the model on a web server. The latter can be combined, e.g., with a Flash interface, of course. Both Flash and Java Applets are regarded somewhat outdated nowadays (although they were not at the beginning of the *IWR Tailorshop* development) and are not available on mobile devices. Furthermore, although reverse compiling needs more work, for Java's and Flash's byte code it is relatively easy (compared, e.g., to an optimized C code).

The AJAX approach is to run the simulation on a server, which sends new values for states, feedback, and bounds on controls via XML upon request by the client. In general, this approach can be combined with other interfaces, but XHTML with JavaScript is the most portable and efficient way. To avoid manipulation by offensive users, it is important, however, that the server only accepts necessary information, i.e., new control values, and determines all other information from, e.g., a database. Server-side simulation requires more computing power from the web server, but enables usage with mobile devices with an appropriate interface. For our purpose—computing an optimization-based feedback online—it is particularly beneficial to run both simulation and optimization on the same machine, which also has to be under our control for the execution of optimization software and thus should be a server, not the client.

A.1.2 Front End and Back End Structure

Therefore, the *IWR Tailorshop* front end was implemented as an AJAX-based XHTML and JavaScript interface together with a server-side *PHP* application. The PHP application is object-oriented with problem-specific parts, e.g., the state progression function for simulation (see Chapter 4), encapsulated in a problem class. The problem-specific JavaScript code handles the communication via AJAX with the web server using the *jQuery* framework on the client. There are separate classes (and scripts) processing the *XML* requests for surveys, high score, and simulation on the server. Data is stored in a *MySQL* database on the server via the database abstraction layer *PEAR MDB2* and for optimization, optimizers *Bonmin* and *Ipopt* are called via the AMPL interface of *IWR Tailorshop*. Figure A.1 illustrates this setup. Configuration of the server-side software is done via a central configuration file.

Participants need to register and to successfully verify their e-mail address via an e-mail link to get access to the interface. During the registration, a *CAPTCHA* needs to be solved correctly. Together with the restriction to one account per address, this common method aims to both prevent bot and duplicate registrations (although this is not a guarantee, of course). After logging in, the communication with the server is managed via AJAX, which additionally prevents accidental control submission on a reload. Survey answers are sent to the server as a *POST request* and responded with either an error or a success message. For high score, on a *GET request*, the server sends an XML file with the high score, see Listing A.2 for an example. For simulation, a POST request with the control values (without states) is sent to the web server. The server determines previous states and the current month from the database, applies the controls u_k and sends an XML file with new state values x_{k+1}, including feedback and new bounds on controls. An example file is displayed in Listing A.1.

A.1.3 Optimization-based Feedback

Communication with AMPL for the solution of the optimization problems for optimization-based feedback is done via data file generation with a template class. A template snippet is shown in Listing A.4. The AMPL execution file writes solutions in the *tailor* file format, which will be explained below. Parallel execution of optimization tasks can be limited in the configuration file according to the available hardware. A limitation is necessary and cannot be left to the operating system's sched-

```
<?xml version="1.0" encoding="utf-8" ?>
<tailorshop>
        <newround>true</newround>
        <hint>Hint message...</hint>
        <month>
                <id>1</id>
                <state>
                        <name>EM</name>
                        <value>12</value>
                        <trend>1</trend>
                </state>
                ...
                <control>
                        <name>AD</name>
                        <value>1750</value>
                        <bounds>
                                <lower>1000</lower>
                                <upper>2000</upper>
                        </bounds>
                        <feedback>
                                <type>trend</type>
                                <value>-2</value>
                        </feedback>
                </control>
                ...
        </month>
</tailorshop>
```

Listing A.1: *IWR Tailorshop* XML for a simulation month with *trend* feedback.

```
<?xml version="1.0" encoding="utf-8" ?>
<tailorshop>
        <highscore>
                <item>
                        <rank>1</rank>
                        <name>Chuck</name>
                        <surname>Norris</surname>
                        <score>238192</score>
                </item>
        </highscore>
        ...
        <ownscore>-17329</ownscore>
</tailorshop>
```

Listing A.2: *IWR Tailorshop* high score XML.

```
## TAILOR V2
## MODEL IWR-Tailorshop-2013-1
#
# tailor data file, generated by Antils on 2014-06-13 17:06:29
# contains a user data set with 4 rounds and optimal solutions.
#
## BEGIN PROPERTIES
## feedback               2
## age                    2
## gender                 0
## economics              0
## problems               1
## games                  1
## bfi10_agreeableness    5
## bfi10_conscientiousness 6
## bfi10_extraversion     8
## bfi10_neuroticism      7
## bfi10_openness         10
## model_knowledge        3
## model_uncertainty      1
## END PROPERTIES

## BEGIN ROUND
# month          x_employees      x_prodSites      x_distSites      x_shirts...
0                14               1                1                319     ...
1                16               2                2                319     ...
2                15               2                2                319     ...
3                15               3                3                319     ...
4                15               4                5                172.5129...
5                16               5                6                36.31878...
6                18               6                6                36.31878...
7                18               6                6                36.31878...
8                18               6                6                36.31878...
9                18               6                6                36.31878...
10               18               6                6                36.31878...
## BEGIN SOLUTION
8                18               6                6                36.31878...
9                25               6                6                36.31878...
10               21               5                5                36.31878...
## END SOLUTION
...
## END ROUND
...
#
# End of file iwrtailorshop_u1337.tailor
#
```

Listing A.3: *IWR Tailorshop* tailor-File.

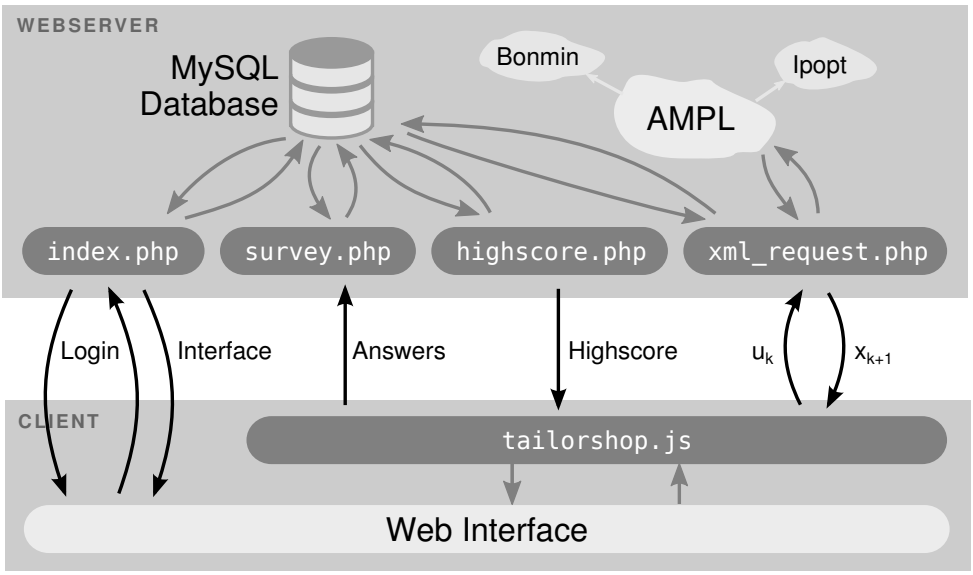


Figure A.1: Schematic representation of *IWR Tailorshop* web front end structure.

uler because idle time is a crucial factor for participants and may decide on complete or incomplete datasets. Parallel execution may also need to be limited because of the available main memory. Limitation is realized via a database table (see Figure A.2, table `schedule`) for running and waiting optimization tasks, which are queued, and can be supplemented by a hard time limit for the individual computations.

A.1.4 Database

A MySQL database is used to store all accumulating data. Sensitive information, e.g., user passwords, are saved as salted *SHA-256* hashes, which inhibits simple dictionary attacks and drastically increases the effort required to restore passwords in case of data theft. *SHA-256* is an *SHA-2* method and regarded as safe until the writing of this work, e.g., by the *National Institute for Standards and Technology* (NIST). To prevent SQL injection attacks, *prepared statements* are used throughout the server-side software.

In Figure A.2, a scheme of the database is shown. The relational database model separates data gained from the surveys and the problem solving from personal information about the users. This is necessary anyway for an anonymized analysis of the data. Keys in the database are `user_id`, an identification number for users referred to, e.g., in Chapter 6, and `initial_id`, an identification number for initial value sets. Data about users' login and logout is also saved in the database (table `log`) for analysis of processing times, but as HTTP is a stateless protocol, especially logout events cannot be tracked reliably.

A.1.5 Language Support

The interface supports multiple languages and receives all language-specific text elements from database table `language`. English and German were available for the web-based feedback study. Language support can easily be extended by translating the text fragments for all labels, which are re-

```
#########################################################################
#
# iwr-tailorshop_web.dat
#
# An AMPL version of the IWR Tailorshop model for execution from the
# web interface, data file
#
#########################################################################

#########################################################################
# Global Settings
#
param NS  := ^start_month/$;      # First month
param NX  := ^last_month/$;       # Last month + 1
param NTM := ^test_months/$;      # Number of test phase months

#########################################################################
# Parameters
#
...

#########################################################################
# Initial Values
#
let x_EM [NS] := ^EM/$; fix x_EM [NS];
let x_PS [NS] := ^PS/$; fix x_PS [NS];
let x_DS [NS] := ^DS/$; fix x_DS [NS];
let x_SH [NS] := ^SH/$; fix x_SH [NS];
let x_PR [NS] := ^PR/$; fix x_PR [NS];
let x_SA [NS] := ^SA/$; fix x_SA [NS];
let x_DE [NS] := ^DE/$; fix x_DE [NS];
let x_RE [NS] := ^RE/$; fix x_RE [NS];
let x_SQ [NS] := ^SQ/$; fix x_SQ [NS];
let x_MQ [NS] := ^MQ/$; fix x_MQ [NS];
let x_MO [NS] := ^MO/$; fix x_MO [NS];
let x_CA [NS] := ^CA/$; fix x_CA [NS];
...
```

Listing A.4: Snippet from *IWR Tailorshop* data file template: placeholders are enclosed by ^ and /$.

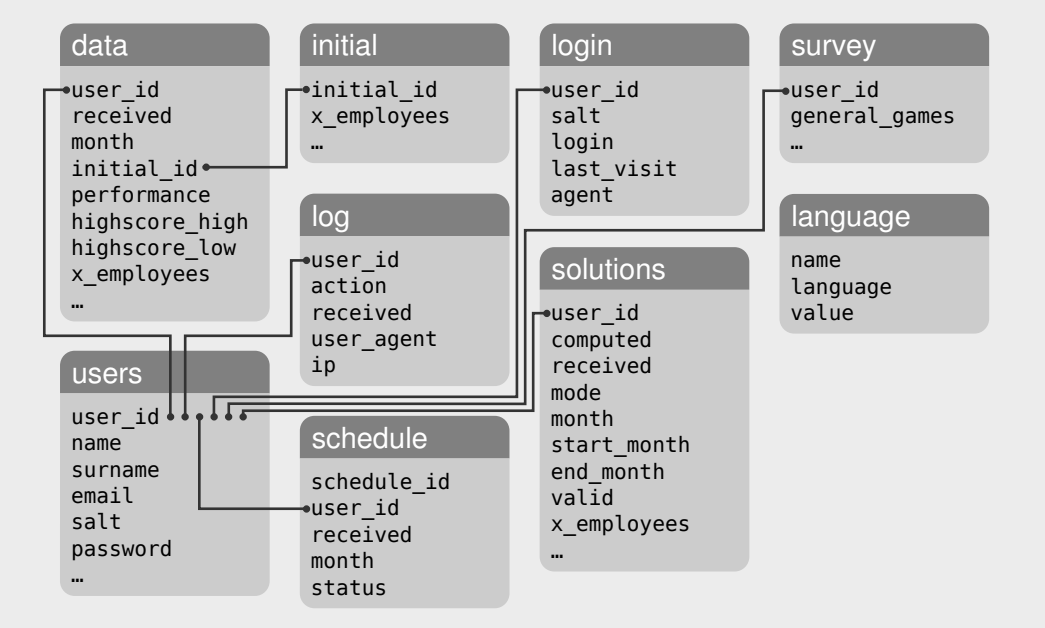


Figure A.2: Database scheme: tables and keys for *IWR Tailorshop* web interface.

placed in templates and XML responses on the fly, and using a corresponding language identifier (e.g., `de_DE` for German).

A.2 Data Formats

XML formats are easily extendible and methods for reading and writing are widely available. However, the disadvantages are that XML involves a big data overhead and from AMPL, e.g., it is easy to write a CSV-style file, but much more difficult to write an XML file. Therefore, there are two data formats available for data export from the database in the web interface: an XML format and the *tailor* format, which basically is a character-separated values format with tabulator as separator. An example for the *tailor* format is shown in Listing A.3, an XML example in Listing A.5. Datasets can be exported automatically from the database with an export tool in both formats.

The *tailor* format was already used in *Tobago*, an analysis software for the original *Tailorshop* microworld [112]. In *tailor* format version 2 (i.e., the version for *IWR Tailorshop*) at the top, there is information on format and model version,

```
## TAILOR V2
## MODEL IWR-Tailorshop-2013-1
```

while `V2` indicates version 2. The tabulator-separated data is enclosed by meta information initiated by a double number sign, `##`, and followed by a keyword like `BEGIN ROUND`. Lines starting with a single `#` are meant to be ignored. The number sign `#` has been chosen because it is also used for comments in the AMPL syntax. The keywords separate the different parts of data contained in a *tailor* file, like rounds and optimal solutions. In the head, meta data can be included with the keywords `BEGIN/END PROPERTIES`. This format is also used for AMPL optimization results within both the web interface and the analysis software.

```
<?xml version="1.0" encoding="UTF-8"?>
<tailorshop>
        <round>
                <month>
                        <id>0</id>
                        <state>
                                <name>EM</name>
                                <value>12</value>
                        </state>
                        ...
                        <control>
                                <name>SP</name>
                                <value>41</value>
                        </control>
                        ...
                </month>
                ...
                <solution>
                        <start>0</start>
                        <month>
                                <id>0</id>
                                <state>
                                        <name>EM</name>
                                        <value>12</value>
                                </state>
                                ...
                                <control>
                                        <name>SP</name>
                                        <value>55</value>
                                </control>
                                ...
                        </month>
                </solution>
                ...
        </round>
        ...
</tailorshop>
```

Listing A.5: *IWR Tailorshop* XML file.

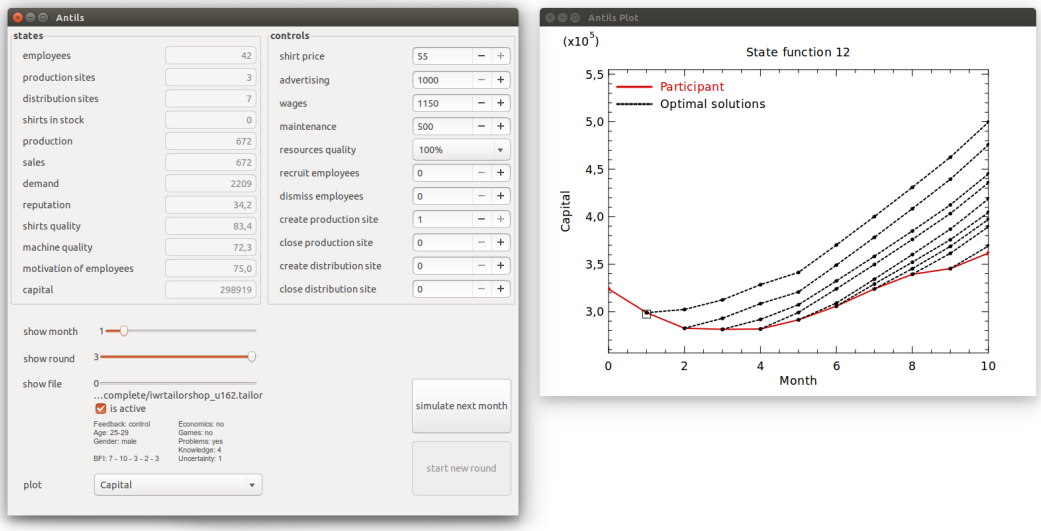


Figure A.3: Antils—the *IWR Tailorshop* analysis and optimization back end with all optimal solutions for variable *capital* for a single data file.

A.3 Analysis Back End

The analysis and optimization back end software *Antils, **An**alysis **T**ool for **I**WR Tailorshop Resu**l**ts and **S**olutions*, is based on *Tobago*, a software for the original *Tailorshop* microworld presented in [112]. It is an object-oriented C++ implementation using GTK+ for the user interface. Problem-specific parts are again encapsulated in a problem class and thus, adaption to other microworlds with similar properties should be possible without much effort. In general, compilation for *Windows* or *OS X* should be possible, but for the results in this work, *Antils* has only been used on a *Ubuntu* Linux distribution. A screenshot of *Antils* is shown in Figure A.3.

Antils reads and writes files in *tailor* format version 2 and is able to handle multiple datasets. It is possible to review the user's decisions and to simulate starting at an arbitrary time point. *Antils* offers various plot possibilities using the *PLplot* library including, e.g., separate averages for each feedback group and plots with all optimal solutions for one dataset. Plots can be exported as *PDF* and *PNG* files. The software uses *IWR Tailorshop*'s AMPL interface for automated optimization to implement the optimization-based analysis methods described in Chapter 3.

APPENDIX B

Statistical Hypothesis Tests

This appendix gives a short overview on statistical hypothesis test used for analysis of the web-based study data in this thesis. Extensive introductions into mathematical statistics and statistical hypothesis tests can be found, e.g., in [28, 98].

B.1 Hypothesis Tests

The aim of statistical hypothesis tests is to decide which one of two contrary assumptions about some characteristic of the probability distribution of a population, *hypotheses*, is true based on a given sample of that population and some statistical properties. Because of the data being realizations of random variables, there often is no definite result for such hypotheses. The approach is to make a decision with a controlled probability of choosing the wrong hypothesis, which is called the *significance level* and often denoted by α.

An analogy for a hypothesis test, which is commonly used, is a criminal trial. At the beginning of the trial, there are two hypotheses. The first one is that the defendant is innocent, which is called the *null hypothesis* H_0 and assumed to be true at first. The second is that the defendant is guilty, which is called the *alternative hypothesis* H_1. The aim is to proof the alternative hypothesis, but there are two types of possible errors. The defendant can be considered to be guilty based on the available information, but in fact is innocent, which is called a *type I error*. Vice versa, the defendant can be considered to be innocent, but is guilty. This is called a *type II error*. Usually, one strives for minimization of the probability for a type I error, i.e., one wants to avoid the conviction of an innocent. Unfortunately, it is not possible to control the probability of both types of errors at the same time in general, thus the probability for a type II error can be high in this setting.

The conventional approach of statistical hypothesis tests can be described as follows:

1. Formulate the null hypothesis H_0 and the alternative hypothesis H_1
2. Choose the appropriate test and test statistic T
3. Select the significance level α, the limit on the probability for a type I error, below which the null hypothesis will be rejected
4. Determine the critical region C in which the null hypothesis will be discarded for test statistic T and significance level α
5. Determine the sample's T_{obs} for the test statistic
6. Decide to keep or discard the null hypothesis depending on whether T_{obs} is within the critical region, i.e.,

$$T_{obs} \in C \Rightarrow \text{discard } H_0, H_1 \text{ true}$$
$$T_{obs} \notin C \Rightarrow \text{keep } H_0$$

However, in statistics software like *R* or *SPSS*, the approach slightly differs, as no significance level is set in advance in the software but a *p-value* is computed. The p-value is the probability to determine

a sample at least as extreme as the observed one under the assumption that the null hypothesis is true. It is common practice to reject the null hypothesis if the corresponding p-value is below a significance level α, which is often chosen to be 0.05 or 0.01. In this work, significance level $\alpha = 0.05$ was used in all hypothesis tests. Pestman [98] gives the following formal definition of a test.

Definition B.1 A *hypothesis test* is understood to be an ordered sequence

$$(X_1, \ldots, X_n; H_0, H_1; C), \tag{B.1}$$

where $X_1, \ldots, X_n$ is a sample, H_0 and H_1 are hypotheses concerning the probability distribution of the population, and $C \in \mathbb{R}^n$ a Borel set. The set C is called the *critical region* in the hypothesis test. If the outcome of $(X_1, \ldots, X_n)$ is an element of C, then H_1 is accepted; if not, then H_0 is accepted.

In the remainder of this Appendix, we give an overview of hypothesis tests applied to the data from the web-based feedback study in Chapter 6 in this work.

B.2 Tests for Mean Values

The most important tests in Chapter 6 are tests for comparing means of different samples. There are two well-known hypothesis tests for this purpose, depending on whether the two samples have equal or unequal variances. Both tests require normally distributed populations. For the following, let $X_1, \ldots, X_n$ and $Y_1, \ldots, Y_m$ be two statistically independent samples of sizes n and m from $N(\mu_x, \sigma_x)$- and $N(\mu_y, \sigma_y)$-distributed populations with unknown μ_x, μ_y, σ_x, and σ_y. We denote the samples' means by

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i, \qquad \overline{Y} = \frac{1}{m}\sum_{i=1}^{m} Y_i. \tag{B.2}$$

For a given Δ (in Chapter 6, we always have $\Delta = 0$), the hypotheses we want to test are

$$H_0\colon\ \mu_y - \mu_x = \Delta \qquad \text{against} \qquad H_1\colon\ \mu_y - \mu_x \neq \Delta. \tag{B.3}$$

With $\sigma_X = \sigma_Y$, i.e., equal variances, the test statistic is t-distributed with $m + n - 2$ degrees of freedom under the null hypothesis and the two-sided STUDENT's t-test can be applied with

$$T_{obs} = \frac{\overline{Y} - \overline{X} - \Delta}{S\sqrt{\frac{1}{m} + \frac{1}{n}}} = \sqrt{\frac{nm}{n+m}}\,\frac{\overline{Y} - \overline{X} - \Delta}{S} \tag{B.4a}$$

$$\text{with} \quad S = \sqrt{\frac{(n-1)S_X^2 + (m-1)S_Y^2}{m+n-2}}, \tag{B.4b}$$

$$S_X^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2, \tag{B.4c}$$

$$S_Y^2 = \frac{1}{m-1}\sum_{i=1}^{m}(Y_i - \overline{Y})^2, \tag{B.4d}$$

$$C = \left\{ t \,\middle|\, t > t_{1-\frac{\alpha}{2};n+m-2} \right\}, \tag{B.4e}$$

where S_X^2 and S_Y^2 are the sample variances.

If variances are not known to be equal, $\sigma_X \neq \sigma_Y$, the test statistic is not t-distributed and needs to be approximated by a t-distribution, which leads to the two-sided WELCH's t-test:

$$T_{obs} = \frac{\overline{Y} - \overline{X} - \Delta}{S} \approx t_\nu \tag{B.5a}$$

$$\text{with} \quad S = \sqrt{\frac{S_X^2}{n} + \frac{S_Y^2}{m}}, \tag{B.5b}$$

$$\nu = \frac{\left(\frac{S_X^2}{n} + \frac{S_Y^2}{m}\right)^2}{\frac{\left(\frac{S_X^2}{n}\right)^2}{n-1} + \frac{\left(\frac{S_X^2}{m}\right)^2}{m-1}}, \tag{B.5c}$$

$$C = \left\{ t \,\middle|\, t > t_{1-\frac{\alpha}{2};\nu} \right\}, \tag{B.5d}$$

with S_X^2 and S_Y^2 as above.

B.3 Tests for Normality

Normality, i.e., normally distributed populations, is a requirement for many other tests, especially for STUDENT's t-test and WELCH's t-test from the previous section. Therefore, it is often necessary to test if a normal distribution adequately describes the sample. Let now X be a random variable and $X_1, \ldots, X_n$ *independent and identically distributed* (iid) observations of that variable, which are sorted in ascending order, i.e., $X_1 > \ldots, X_n$. The hypotheses for normality test then are

$$H_0\colon\ F_X(x) = F_0(x) \qquad \text{against} \qquad H_1\colon\ F_X(x) \neq F_0(x), \tag{B.6}$$

with $F_X(x)$ the distribution function of X and $F_0(x) = N(\mu_X, \sigma_X)$. There are several hypothesis tests for distribution tests of which some are limited to normal distributions and some can be applied to arbitrary distributions. In the following, however, we will only consider the case of normal distributions.

The KOLMOGOROV-SMIRNOV test [84] uses the following KOLMOGOROV-SMIRNOV statistic,

$$D_n = \sup_x |F_n(x) - F_0(x)|, \tag{B.7}$$

with $F_n(x)$ the *empirical distribution function* and $I_{X_i \leq x}$ the *indicator function*,

$$F_n(x) = \frac{1}{n} \sum_{i=1}^{n} I_{X_i \leq x}. \tag{B.8}$$

According to the GLIVENKO-CANTELLI theorem, D_n converges to 0 for $n \to \infty$ if the sample comes from the distribution $F_0(x)$. The null hypothesis will be rejected if D_n is greater than the critical value $\frac{1}{\sqrt{n}} K_\alpha$, where K_α is determined from

$$\mathbb{P}(K \leq K_\alpha) = 1 - \alpha \tag{B.9}$$

with the KOLMOGOROV distribution K.

The LILLIEFORS test [84] is based on the KOLMOGOROV-SMIRNOV test, but uses the LILLIEFORS distribution instead. The ANDERSON-DARLING test [7, 8] and the CRAMÉR-VON MISES test [124] use

a test statistic which measures the distance between the empirical distribution function and the hypothesized distribution function. The test statistic in the SHAPIRO-WILK test [117] compares variance estimates,

$$T = \frac{b^2}{(n-1)s^2}, \tag{B.10}$$

with s^2 the estimate for the sample's variance and b^2 an estimate on how the variance should be if the sample was normally distributed. The SHAPIRO-FRANCIA test [107] is a modification thereof. Finally, PEARSON's chi-squared test [97] is a χ^2-test using the χ^2-distribution.

B.4 Tests for Variance Homogeneity

STUDENT's t-test requires equal variances for two samples, but WELCH's t-test does not. To determine which test could be applied, different tests for variance homogeneity have been used in this thesis. The hypotheses for tests on variance homogeneity are

$$H_0\colon \sigma_0^2 = \sigma_1^2 = \cdots = \sigma_k^2 \qquad \text{against} \qquad H_1\colon \exists(i,j) \text{ with } i \neq j\colon \sigma_i^2 \neq \sigma_j^2. \tag{B.11}$$

Let X_{ij} be the sample with $i = 1,\ldots,k$ indicating the group and $j = 1,\ldots,n_i$ the samples for each group, i.e., we have k groups and n_i samples in group i. With the group sample mean $\overline{X}_i$, the total sample number $n = \sum_{i=1}^{k} n_i$, and

$$Y_{ij} = \left| X_{ij} - \overline{X}_i \right|, \qquad \overline{Y}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} Y_{ij}, \qquad \overline{Y} = \frac{1}{k} \sum_{i=1}^{k} \overline{Y}_i, \tag{B.12}$$

the test statistic for LEVENE's test [82] is

$$T = \frac{(n-k)}{(k-1)} \frac{\sum_{i=1}^{k} n_i (\overline{Y_i} - \overline{Y})}{\sum_{i=1}^{k} \sum_{j=1}^{n_i} (Y_{ij} - \overline{Y_i})}. \tag{B.13}$$

The test statistic T can be approximated by $F_{k-1,n-k}$. BROWN-FORSYTHE test [31] is a modification of LEVENE's test with the usage of sample medians instead of sample means $\overline{X}_i$ for the computation of Y_{ij}, and thus, the test statistic is approximated by a χ^2 distribution. BARTLETT's test [13] is a slightly different approach, but known to be rather sensitive to non-normality and thus has only been applied for comparison.

B.5 Test for Outliers

GRUBBS' test [66, 67] is a test for the detection of outlying samples. The test detects one outlier at a time and needs normally distributed data. The hypotheses for the test are

$$\begin{aligned} & H_0\colon \text{no outliers in the data set} \\ \text{against} \quad & H_1\colon \text{at least one outlier in the data set.} \end{aligned} \tag{B.14}$$

Let $X_1,\ldots,X_n$ be a sample from a normally distributed population. Then, with the sample mean $\overline{X}$, the test statistic for GRUBBS' test is

$$T = \frac{\max\limits_{i=1,\ldots n} \left| X_i - \overline{X} \right|}{s}. \tag{B.15}$$

The null hypothesis is rejected if

$$T > \frac{n-1}{\sqrt{n}} \sqrt{\frac{t^2_{\alpha/(2n),n-2}}{n-2+t^2_{\alpha/(2n),n-2}}}. \tag{B.16}$$

The test can be applied recursively, but may detect most of the sample as outliers for small sample sizes. GRUBBS' test has been applied in this work together with other approaches for outlier detection like outer fences according to TUKEY [126]. For the actual selection of outliers, however, the results of GRUBBS' test have not been used.

APPENDIX C

Computation of M for the Big M Relaxation

For optimization, we need a numerical value for M and thus we need bounds on the three terms T^1_{k+1}, T^2_{k+1}, and T^3_{k+1}. This means we need bounds for the variables x^{EM}_{k+1}, x^{PS}_{k+1}, x^{DS}_{k+1}, x^{SH}_k, x^{PR}_{k+1}, and x^{DE}_{k+1}. Fortunately, most of these are quite easy to get. For production and distribution sites, for instance, we have

$$x^{PS}_{k+1} = x^{PS}_k - u^{dPS}_k + u^{DPS}_k, \qquad x^{DS}_{k+1} = x^{DS}_k - u^{dDS}_k + u^{DDS}_k, \tag{C.1a}$$

$$x^{PS}_k \geq 1, \qquad x^{DS}_k \geq 1, \tag{C.1b}$$

$$u^{DPS}_k \leq p^{DPS}, \qquad u^{DDS}_k \in [0, p^{DDS}], \tag{C.1c}$$

$$u^{dPS}_k + u^{dPS}_{k-1} \leq p^{dPS}, \qquad u^{dDS}_k \in [0, p^{dDS}], \tag{C.1d}$$

(C.1e)

and thus

$$x^{PS}_k \in [1, 1 + n_x], \qquad x^{DS}_k \in [1, 1 + 2 \cdot n_x], \tag{C.2}$$

with $n_x = t_f - t_0$. Furthermore, bounds for employees can be computed via

$$x^{EM}_k \geq 1 \qquad u^{EM}_k \in [-p^{dEM}, p^{DEM,0} \cdot x^{PS}_k + p^{DEM,1} \cdot x^{DS}_k] \tag{C.3a}$$

$$x^{EM}_{k+1} = x^{EM}_k + u^{EM}_k, \qquad = [-p^{dEM}, 15 + 25 \cdot n_x], \tag{C.3b}$$

which leads to

$$x^{EM}_k \in [1, x^{EM}_0 + (15 + 25 \cdot n_x) \cdot n_x] = [1, 10 + 15 \cdot n_x + 25 \cdot n_x^2]. \tag{C.4}$$

If we assume the stock capacity constraint for shirts which is not part of the final model, we have

$$x^{SH}_k \in [0, p^{SH,0} \cdot x^{DS}_k] = [0, 2000 \cdot [1, 1 + 2 \cdot n_x]] = [0, 2000 + 4000 \cdot n_x]. \tag{C.5}$$

Note that this assumption is valid for the computations in this section, as the constraint was far from being active in all test cases.

Then, we can compute bounds for T^1_{k+1} and x^{PR}_{k+1},

$$x^{PR}_{k+1} = p^{PR,0} \cdot x^{PS}_{k+1} \cdot \log\left(\frac{p^{PR,1} \cdot x^{EM}_{k+1}}{x^{PS}_{k+1} + x^{DS}_{k+1} + p^{PR,2}} + 1\right) \tag{C.6a}$$

$$= 99.9 \cdot [1, 1 + n_x] \cdot \log\left(\frac{2 \cdot [1, 10 + 15 \cdot n_x + 25 \cdot n_x^2]}{[1, 1 + n_x] + [1, 1 + 2 \cdot n_x] + 10^{-6}} + 1\right) \tag{C.6b}$$

$$= \left[99.9 \cdot \log\left(\frac{2}{2 + 3 \cdot n_x + 10^{-6}} + 1\right), 99.9 \cdot (1 + n_x) \cdot \log\left(\frac{20 + 30 \cdot n_x + 50 \cdot n_x^2}{2 + 10^{-6}} + 1\right)\right], \tag{C.6c}$$

$$T^1_{k+1} = p^{SA,0} \cdot x^{DS}_{k+1} \cdot \log\left(\frac{p^{SA,1} \cdot x^{EM}_{k+1}}{x^{PS}_{k+1} + x^{DS}_{k+1} + p^{SA,2}} + 1\right) \tag{C.7a}$$

$$= 99.9 \cdot [1, 1+2\cdot n_x] \cdot \log\left(\frac{2 \cdot [1, 10+15\cdot n_x + 25 \cdot n_x^2]}{[1, 1+n_x] + [1, 1+2\cdot n_x] + 10^{-6}} + 1\right) \tag{C.7b}$$

$$= \left[99.9 \cdot \log\left(\frac{2}{2+3\cdot n_x + 10^{-6}} + 1\right), 99.9 \cdot (1+2\cdot n_x) \cdot \log\left(\frac{20+30\cdot n_x + 50 \cdot n_x^2]}{2+10^{-6}} + 1\right)\right], \tag{C.7c}$$

and hence also for T^2_{k+1},

$$T^2_{k+1} = x^{SH}_k + x^{PR}_{k+1} \tag{C.8a}$$

$$= [0, 2000 \cdot (1+2\cdot n_x)] + \left[99.9 \cdot \log\left(\frac{2}{2+3\cdot n_x + 10^{-6}} + 1\right), 99.9 \cdot (1+n_x) \cdot \log\left(\frac{20+30\cdot n_x + 50\cdot n_x^2]}{2+10^{-6}} + 1\right)\right] \tag{C.8b}$$

$$= \left[99.9 \cdot \log\left(\frac{2}{2+3\cdot n_x + 10^{-6}} + 1\right), 2000 \cdot (1+2\cdot n_x) + 99.9 \cdot (1+n_x) \cdot \log\left(\frac{20+30\cdot n_x + 50\cdot n_x^2]}{2+10^{-6}} + 1\right)\right]. \tag{C.8c}$$

As a next step, for T^3_{k+1} we need to compute bounds for x^{DE}_{k+1},

$$x^{DE}_{k+1} = p^{DE,0} \cdot \exp\left(-p^{DE,1} \cdot u^{SP}_k\right) \cdot \log\left(p^{DE,2} \cdot u^{AD}_k + 1\right) \cdot \left(x^{RE}_k + p^{DE,3}\right) \tag{C.9a}$$

$$= 600.0 \cdot \exp([-1.1, -0.7]) \cdot \log([21, 41]) \cdot \left(x^{RE}_k + 0.5\right), \tag{C.9b}$$

and here, we also need bounds for x^{RE}_{k+1} which vice versa requires bounds on x^{SQ}_{k+1}, x^{MQ}_{k+1}, and x^{MQ}_{k+1}:

$$x^{RE}_{k+1} = p^{RE,0} \cdot x^{RE}_k + p^{RE,1} \log\left(\left(p^{RE,2} \cdot u^{AD}_k + p^{RE,3} \cdot u^{SP}_k \cdot (x^{SQ}_k)^2 + p^{RE,4} \cdot u^{WA}_k\right) + 1\right) \tag{C.10a}$$

$$= 0.5 \cdot x^{RE}_k + \log\left([1.085, 1.14] + [0.0035, 0.0055] \cdot (x^{SQ}_k)^2\right). \tag{C.10b}$$

For x^{MQ}_{k+1}, we have

$$x^{MQ}_{k+1} = p^{MQ,0} \cdot x^{MQ}_k \cdot \exp\left(-p^{MQ,1} \cdot \frac{x^{PR}_k}{x^{PS}_k + p^{MQ,2}}\right) + p^{MQ,3} \cdot \log\left(u^{MA}_k \cdot p^{MQ,4} + 1\right) \tag{C.11a}$$

$$= 0.8 \cdot x^{MQ}_k \cdot \exp\left(-0.6 \cdot 10^{-2} \cdot \frac{[6, 8667]}{[1, 11] + 10^{-6}}\right) + 0.13 \cdot \log([0, 5000] \cdot 0.2 + 1), \tag{C.11b}$$

and as for x^{MQ}_{k+1}, only the upper bound is important (the lower bound 0 is obvious), we proceed as follows:

$$x^{MQ}_{k+1} = 0.8 \cdot x^{MQ}_k \cdot \exp\left(-0.6 \cdot 10^{-2} \cdot \frac{6}{11 + 10^{-6}}\right) + 0.13 \cdot \log(1001) \tag{C.12a}$$

$$\leq 0.8 \cdot x^{MQ}_k + 0.9. \tag{C.12b}$$

This equation obviously has the fix point $x_k^{MQ} = 4.5$, so we have $x_k^{MQ} \in [0.0, 4.5]$ with an initial value within this interval. For the *shirt quality*, it holds

$$x_{k+1}^{SQ} = p^{SQ,0} \cdot x_k^{MO} + p^{SQ,1} \cdot x_k^{MQ} + p^{SQ,2} \cdot u_k^{RQ} \tag{C.13a}$$
$$= 0.2 \cdot x_k^{MO} + [0.25, 1.85], \tag{C.13b}$$

and for the *motivation of employees*, we can compute

$$\begin{aligned} x_{k+1}^{MO} = &\left(1 - p^{MO,0}\right) \cdot x_k^{MO} + p^{MO,0} \\ &\cdot \log\left(p^{MO,1} \cdot u_k^{DEM} + p^{MO,2} \cdot u_k^{DPS} + p^{MO,3} \cdot u_k^{DDS} + p^{MO,4} \cdot u_k^{WA} + p^{MO,5} \cdot x_k^{RE} + p^{MO,6}\right) \\ &\cdot \exp\left(-(p^{MO,7} \cdot u_k^{dEM} + p^{MO,8} \cdot u_k^{dPS} + p^{MO,9} \cdot u_k^{dDS}) + p^{MO,10}\right) \cdot p^{MO,11} \end{aligned} \tag{C.14a}$$
$$= 0.5 \cdot x_k^{MO} + 0.25 \cdot \log\left(\left[1.2, 2.3 + 4 \cdot 10^{-2} \cdot (15 + 25 \cdot n_x)\right] + 0.3 \cdot x_k^{RE}\right) \cdot \exp\left([-10.5, 1]\right). \tag{C.14b}$$

So, for x_{k+1}^{RE}, x_{k+1}^{SQ}, and x_{k+1}^{MO}, we have

$$x_{k+1}^{RE} = 0.5 \cdot x_k^{RE} + \log\left([1.085, 1.14] + [0.0035, 0.0055] \cdot (x_k^{SQ})^2\right), \tag{C.15a}$$
$$x_{k+1}^{SQ} = 0.2 \cdot x_k^{MO} + [0.25, 1.85], \tag{C.15b}$$
$$x_{k+1}^{MO} = 0.5 \cdot x_k^{MO} + 0.25 \cdot \log\left(\left[1.2, 2.3 + 4 \cdot 10^{-2} \cdot (15 + 25 \cdot n_x)\right] + 0.3 \cdot x_k^{RE}\right) \cdot \exp\left([-10.5, 1]\right). \tag{C.15c}$$

With the initial values $x_0^{RE} = 0.79$, $x_0^{SQ} = 0.75$, and $x_0^{MO} = 0.73$, we can compute upper bounds for these variables, see Table 5.1, and as the last column of the table is a fix point for our parameter set, we get

$$x_k^{RE} \in [0, 1.0], \qquad x_k^{SQ} \in [0, 2.6], \qquad x_k^{MO} \in [0, 3.5]. \tag{C.16}$$

Eventually, this means

$$x_{k+1}^{DE} = 600.0 \cdot \exp([-1.1, -0.7]) \cdot \log([21, 41]) \cdot ([0, 1] + 0.5) \qquad \in [304, 1660], \tag{C.17a}$$
$$T_{k+1}^{3} = p^{SA,3} \cdot x_{k+1}^{DE} = 1.0 \cdot x_{k+1}^{DE} \qquad \in [304, 1660]. \tag{C.17b}$$

Finally, for $n_x = t_f - t_0 = 10$, we get

$$T_{k+1}^{1} \in [6, 16545], \qquad T_{k+1}^{1} - T_{k+1}^{2} \in [-50661, 16539], \tag{C.18a}$$
$$T_{k+1}^{2} \in [6, 50667], \qquad T_{k+1}^{1} - T_{k+1}^{3} \in [-1654, 16241], \tag{C.18b}$$
$$T_{k+1}^{3} \in [304, 1660], \qquad T_{k+1}^{2} - T_{k+1}^{3} \in [-1654, 50363], \tag{C.18c}$$

and for the remaining constraints, with $x_k^{SA} \in [6, 1660]$,

$$T_{k+1}^{1} - x_{k+1}^{SA} \in [0, 16539], \tag{C.19a}$$
$$T_{k+1}^{2} - x_{k+1}^{SA} \in [0, 50661], \tag{C.19b}$$
$$T_{k+1}^{3} - x_{k+1}^{SA} \in [0, 1654], \tag{C.19c}$$

which means, that we can chose, e.g., $M = 50661$.

Bibliography

[1] 2013 sales, demographic and usage data: Essential facts about the computer and video game industry. Technical report, Entertainment Software Association, 2013.

[2] K. Abhishek, S. Leyffer, and J. T. Linderoth. FilMINT: An Outer-Approximation-Based Solver for Nonlinear Mixed Integer Programs. Preprint ANL/MCS-P1374-0906, Argonne National Laboratory, Mathematics and Computer Science Division, 9700 South Cass Avenue, Argonne, IL 60439, U.S.A., March 2008.

[3] T. Achterberg, T. Koch, and A. Martin. Branching rules revisited. *Operations Research Letters*, 33(1):42–54, 2005.

[4] D. N. Adams. *The Hitchhiker's Guide to the Galaxy*. Pan Books, 1979. ISBN 0330258648.

[5] J. R. Anderson. *How Can the Human Mind Occur in the Physical Universe?* Oxford Series on Cognitive Models and Architectures. Oxford University Press, 2007. ISBN 9780198043539.

[6] J. R. Anderson and C. Lebiere. *The Atomic Components of Thought*. Lawrence Erlbaum Associates, 1998. ISBN 9780805828177.

[7] T. W. Anderson and D. A. Darling. Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes. *Annals of Mathematical Statistics*, 23(2):193–212, 1952.

[8] T. W. Anderson and D. A. Darling. A test of goodness of fit. *Journal of the American Statistical Association*, 49(268):765–769, 1954.

[9] A. C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. SIAM, 2005.

[10] D. Applegate, R. Bixby, V. Chvátal, W. Cook, D. Espinoza, M. Goycoolea, and K. Helsgaun. Certification of an optimal TSP tour through 85, 900 cities. *Oper. Res. Lett.*, 37(1):11–15, 2009.

[11] C. M. Barth. *The Impact of Emotions on Complex Problem Solving Performance and Ways of Measuring this Performance*. PhD thesis, Ruprecht–Karls–Universität Heidelberg, 2010.

[12] C. M. Barth and J. Funke. Negative affective environments improve complex solving performance. *Cognition and Emotion*, 24:1259–1268, 2010.

[13] M. S. Bartlett. Properties of sufficiency and statistical tests. *Proceedings of the Royal Statistical Society Series A*, 160(901):268–282, 1937.

[14] R. E. Bellman. *Dynamic Programming*. University Press, Princeton, N.J., 6th edition, 1957. ISBN 0-486-42809-5 (paperback).

[15] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods and Software*, 24(4-5):597–634, 2009.

[16] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and A. Mahajan. Mixed-Integer Nonlinear Optimization. In A. Iserles, editor, *Acta Numerica*, volume 22, pages 1–131. Cambridge University Press, 2013.

[17] J. Benders. Partitioning Procedures for Solving Mixed-Variables Programming Problems. *Numerische Mathematik*, 4:238–252, 1962.

[18] P. Benner, V. Mehrmann, and D. C. Sorensen, editors. *Dimension Reduction of Large-Scale Systems: Proceedings of a Workshop held in Oberwolfach, Germany, October 19-25, 2003*, 2005. Springer, Berlin Heidelberg.

[19] L. T. Biegler. Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. *Computers & Chemical Engineering*, 8:243–248, 1984.

[20] R. Bixby, M. Fenelon, Z. Gu, E. Rothberg, and R. Wunderling. Mixed-Integer Programming: A Progress Report. In *The Sharpest Cut: The Impact of Manfred Padberg and His Work*. SIAM, 2004.

[21] H. Bock. *Numerische Berechnung zustandsbeschränkter optimaler Steuerungen mit der Mehrzielmethode*. Carl-Cranz-Gesellschaft, Heidelberg, 1978.

[22] H. Bock and R. Longman. Computation of optimal controls on disjoint control sets for minimum energy subway operation. In *Proceedings of the American Astronomical Society. Symposium on Engineering Science and Mechanics*, Taiwan, 1982.

[23] H. G. Bock. Numerical treatment of inverse problems in chemical reaction kinetics. In K. H. Ebert, P. Deuflhard, and W. Jäger, editors, *Modelling of Chemical Reaction Systems*, volume 18 of *Springer Series in Chemical Physics*, pages 102–125. Springer, Heidelberg, 1981.

[24] H. G. Bock and K. J. Plitt. A Multiple Shooting algorithm for direct solution of optimal control problems. In *Proceedings of the 9th IFAC World Congress*, pages 242–247, Budapest, 1984. Pergamon Press.

[25] P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5:186–204, 2008.

[26] P. Bonami, M. Kilinç, and J. Linderoth. Algorithms and software for convex mixed integer nonlinear programs. In J. Lee and S. Leyffer, editors, *Mixed Integer Nonlinear Programming*, volume 154 of *The IMA Volumes in Mathematics and its Applications*, pages 1–39. Springer New York, 2012. ISBN 978-1-4614-1926-6.

[27] P. Bonami, J. Lee, S. Leyffer, and A. Wächter. On branching rules for convex mixed-integer nonlinear optimization. *Journal of Experimental Algorithmics*, 18:6:1–6:31, 2013.

[28] J. Bortz and C. Schuster. *Statistik für Human- und Sozialwissenschaftler. Lehrbuch mit Online-Materialien*. Springer, 2010. ISBN 9783642127700.

[29] B. Brehmer. *Feedback delays in dynamic decision making*, pages 103–130. Complex problem solving: The European Perspective. Lawrence Erlbaum Associates, 1995.

[30] B. Brehmer and D. Dörner. Experiments with computer-stimulated microworlds: Escaping both the narrow straits of the laboratory and the deep blue sea of the field study. *Computers in Human Behavior*, 9:171–184, 1993.

[31] M. Brown and A. B. Forsythe. Robust tests for the equality of variances. *Journal of the American Statistical Association*, 69(346):364–367, 1974.

[32] R. Bulirsch. Die Mehrzielmethode zur numerischen Lösung von nichtlinearen Randwertproblemen und Aufgaben der optimalen Steuerung. Technical report, Carl-Cranz-Gesellschaft, Oberpfaffenhofen, 1971.

[33] J. Burgschweiger, B. Gnädig, and M. C. Steinbach. Optimization models for operative planning in drinking water networks. *Optimization and Engineering*, 10(1):43–73, 2008.

[34] J. L. Casti. *Connectivity, complexity, and catastrophe in large-scale systems*. International series on applied systems analysis. J. Wiley, 1979. ISBN 9780471276616.

[35] J. Cavanaugh and F. Blanchard-Fields. *Adult Development and Aging*. International student edition. Cengage Learning, 2005. ISBN 9780534520663.

[36] M. Conforti, G. P. Cornuéjols, and G. Zambelli. *Integer Programming*. Springer, 2014. ISBN 3319110071.

[37] M. Cronin, C. Gonzalez, and J. Sterman. Why don't well-educated adults understand accumulation? A challenge to researchers, educators, and citizens. *Organizational Behavior and Human Decision Processes*, 108:116–130, 2009.

[38] R. Dakin. A tree-search algorithm for mixed integer programming problems. *The Computer Journal*, 8:250–255, 1965.

[39] D. Danner, D. Hagemann, A. Schankin, M. Hager, and J. Funke. Beyond iq. a latent state-trait analysis of general intelligence, dynamic decision making, and implicit learning. *Intelligence*, 39:323–334, 2011.

[40] J. M. Digman. Personality structure: Emergence of the five-factor model. *Annual Review of Psychology*, 41:471–440, 1990.

[41] D. Dörner. *Problemlösen als Informationsverarbeitung*. Standards Psychologie. Kohlhammer, 1976. ISBN 9783170055179.

[42] D. Dörner. On the difficulties people have in dealing with complexity. *Simulation and Games*, 11:87–106, 1980.

[43] D. Dörner, H. W. Kreuzig, F. Reither, and T. Stäudel. *Lohhausen. Vom Umgang mit Unbestimmt und Komplexität*. Huber, 1983.

[44] D. Dörner, T. Stäudel, and S. Strohschneider. *Moro - Programmdokumentation (Memorandum Nr. 23)*. Lehrstuhl Psychologie II der Universität Bamberg, 1986.

[45] K. Duncker. *Zur Psychologie des Produktiven Denkens*. J. Springer, 1935.

[46] M. Duran and I. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36(3):307–339, 1986.

[47] M. Engelhart, J. Funke, and S. Sager. A decomposition approach for a new test-scenario in complex problem solving. *Journal of Computational Science*, 4(4):245–254, 2013.

[48] R. Fletcher and S. Leyffer. Solving Mixed Integer Nonlinear Programs by Outer Approximation. *Mathematical Programming*, 66:327–349, 1994.

[49] R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. *Mathematical Programming A*, 91:239–269, 2002.

[50] P. A. Frensch and J. Funke. *Complex Problem Solving: The European Perspective*. Taylor & Francis, 1995. ISBN 9781317781394.

[51] J. Funke. Einige Bemerkungen zu Problemen der Problemlöseforschung oder: Ist Testintelligenz doch ein Prädiktor? *Diagnostica*, 29:283–302, 1983.

[52] J. Funke. Using simulation to study complex problem solving: A review of studies in the frg. *Simulation & Games*, 19:277–303, 1988.

[53] J. Funke. Dynamic systems as tools for analysing human judgement. *Thinking and Reasoning*, 7:69–89, 2001.

[54] J. Funke. *Problemlösendes Denken*. Standards Psychologie. Kohlhammer, 2003. ISBN 3-17017-425-8.

[55] J. Funke. Complex problem solving: A case for complex cognition? *Cognitive Processing*, 11: 133–142, 2010.

[56] J. Funke and P. A. Frensch. *Complex problem solving: The European perspective – 10 years after*, pages 25–47. Learning to solve complex scientific problems. Lawrence Erlbaum, 2007.

[57] H. Gardner. *The Mind's New Science: A History of the Cognitive Revolution*. Basic Books, 1985. ISBN 9780786725144.

[58] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, New York, 1979.

[59] A. Geoffrion. Generalized Benders Decomposition. *Journal of Optimization Theory and Applications*, 10:237–260, 1972.

[60] A. M. Geoffrion. *Approaches to integer programming*, chapter Lagrangean Relaxation for Integer Programming, pages 82–114. North-Holland Pub Co, 1974.

[61] R. J. Gerrig. *Psychology and Life*. Pearson Education, 2012. ISBN 9780205948017.

[62] C. Gonzalez. Learning to make decisions in dynamic environments: Effects of time constraints and cognitive abilities. *Human Factors*, 46(3):449–460, 2004.

[63] I. Grossmann. Review of Nonlinear Mixed-Integer and Disjunctive Programming Techniques. *Optimization and Engineering*, 3:227–252, 2002.

[64] I. Grossmann and S. Lee. Generalized disjunctive programming: Nonlinear convex hull relaxation and algorithms. *Computational Optimization and Applications*, 26:83–100, 2003.

[65] I. Grossmann and J. Ruiz. Generalized Disjunctive Programming: A Framework for Formulation and Alternative Algorithms for MINLP Optimization. In J. Lee and S. Leyffer, editors, *Mixed Integer Nonlinear Programming*, volume 154 of *The IMA Volumes in Mathematics and its Applications*, chapter 4, pages 93–116. Springer, 2012.

[66] F. E. Grubbs. Sample criteria for testing outlying observations. *Annals of Mathematical Statistics*, 21(1):27–58, 1950.

[67] F. E. Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11: 1–21, 1969.

[68] S. P. Han. A globally convergent method for nonlinear programming. *Journal of Optimization Theory and Applications*, 22:297–310, 1977.

[69] M. Held and R. M. Karp. The traveling-salesman and minimum cost spanning trees. *Operations Research*, 18:1138–1162, 1970.

[70] M. Held and R. M. Karp. The traveling-salesman problem and minimum spanning trees: Part ii. *Mathematical Programming*, 1(1):6–25, 1970.

[71] V. Hoorens. Self-enhancement and superiority biases in social comparison. *European Review of Social Psychology*, 4(1):113–139, 1993. doi: 10.1080/14792779343000040.

[72] H. J. Hörmann and M. Thomas. Zum Zusammenhang zwischen Intelligenz und komplexem Problemlösen. *Sprache & Kognition*, 8:23–31, 1989.

[73] M. Hüfner, T. Tometzki, T. Kraja, and S. Engell. Learn2Control: Eine webbasierte Lernumgebung im Bio- und Chemieingenieurwesen. *Journal Hochschuldidaktik*, 22(1):20–23, 2011.

[74] N. Karmarkar. A New Polynomial Time Algorithm for Linear Programming. *Combinatorica*, 4 (4):373–395, 1984.

[75] C. Kirches, S. Sager, H. Bock, and J. Schlöder. Time-optimal control of automobile test drives with gear shifts. *Optimal Control Applications and Methods*, 31(2):137–153, March/April 2010.

[76] M. Kleinmann and B. Strauß. Validity and applications of computer simulated scenarios in personal assessment. *International Journal of Seclection and Assessment*, 6(2):97–106, 1998.

[77] R. H. Kluwe. *Knowledge and performance in complex problem solving*, pages 401–423. The cognitive psychology of knowledge. Elsevier Science Publishers, 1993.

[78] R. H. Kluwe, C. Misiak, and H. Haider. The control of complex systems and performance in intelligence tests. In H. Rowe, editor, *Intelligence: Reconceptualization and Measurement*, pages 227–244. Erlbaum, 1991.

[79] A. Land and A. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28:497–520, 1960.

[80] J. Lee and S. Leyffer. *Mixed Integer Nonlinear Programming*. The IMA Volumes in Mathematics and its Applications. Springer, 2012. ISBN 9781461419273.

[81] C. Lemarechal. Lagrangian relaxation. In M. Jünger and D. Naddef, editors, *Computational Combinatorial Optimization*, volume 2241 of *Lecture Notes in Computer Science*, chapter 4, pages 112–156. Springer, 2001.

[82] H. Levene. Robust tests for equality of variances. In I. Olkin, S. G. Ghurye, W. Hoeffding, W. G. Madow, and H. B. Mann, editors, *Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*, pages 278–292. Stanford University Press, 1960.

[83] L. Liberti. Writing global optimization software. In L. Liberti and N. Maculan, editors, *Global Optimization*, volume 84 of *Nonconvex Optimization and Its Applications*, pages 211–262. Springer, 2006. ISBN 978-0-387-28260-2.

[84] H. W. Lilliefors. On the Kolmogorov-Smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, 62(318):399–402, 1967.

[85] M. Matsumoto and T. Nishimura. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, 8:3–30, 1998. doi: 10.1145/272991.272995.

[86] G. McCormick. Computability of global solutions to factorable nonconvex programs. Part I. Convex underestimating problems. *Mathematical Programming*, 10:147–175, 1976.

[87] B. Meyer and W. Scholl. Complex problem solving after unstructured discussion. Effects of information distribution and experiece. *Group Process and Intergroup Relations*, 12:495–515, 2009.

[88] J. A. Muckstadt and S. A. Koenig. An application of lagrangian relaxation to scheduling in power-generation systems. *Operations Research*, 25(3):387–403, 1977.

[89] R. B. Myerson. *Game Theory*. Harvard University Press, 1997. ISBN 9780674341166.

[90] A. Newell and H. A. Simon. *Human problem solving*. Prentice-Hall, 1972.

[91] A. Newell, J. C. Shaw, and H. A. Simon. Elements of a theory of human problem solving. *Psychological Review*, 65(3):151–166, 1958.

[92] A. Newell, J. C. Shaw, and H. A. Simon. Report on a general problem-solving program. In *Proceedings of the International Conference on Information Processing*, pages 256–264, 1959.

[93] J. Nightingale. *Think Smart - Act Smart: Avoiding The Business Mistakes That Even Intelligent People Make*. Wiley, 2008. ISBN 9780470224366.

[94] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2006. ISBN 9780387400655.

[95] M. Osman. Observation can be as effective as action in problem solving. *Cognitive Science: A Multidisciplinary Journal*, 32(1):162–183, 2008.

[96] J. H. Otto and L. E.-D. Wahrgenommene Beeinflussbarkeit von negativen Emotionen, Stimmung und komplexes Problemlösen. *Zeitschrift für Differentielle und Diagnostische Psychologie*, 25:31–46, 2004.

[97] K. Pearson. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *Philosophical Magazine Series 5*, 50(302):157–175, 1900. doi: 10.1080/14786440009463897.

[98] W. Pestman. *Mathematical Statistics*. De Gruyter, 2009. ISBN 9783110208535.

[99] H. Pirkul and V. Jayaraman. A multi-commodity, multi-plant, capacitated facility location problem: formulation and efficient heuristic solution. *Computers & Operations Research*, 25 (10):869–878, 1998.

[100] M. J. D. Powell. A fast algorithm for nonlinearly constrained optimization calculations. In G. e. Watson, editor, *Numerical Analysis*, volume 3 of *Springer Verlag*, pages 144–157, Berlin, 1977.

[101] W. Putz-Osterloh. Über die Beziehung zwischen Testintelligenz und Problemlöseerfolg. *Zeitschrift für Psychologie*, 189:79–100, 1981.

[102] W. Putz-Osterloh, B. Bott, and K. Köster. Models of learning in problem solving – are they transferable to tutorial systems? *Computers in Human Behavior*, 6:83–96, 1990.

[103] I. Quesada and I. Grossmann. An LP/NLP Based Branch and Bound Algorithm for Convex MINLP Optimization Problems. *Computers & Chemical Engineering*, 16:937–947, 1992.

[104] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. ISBN 3-900051-07-0.

[105] B. Rammstedt and O. P. John. Measuring personality in one minute or less: A 10-item short version of the Big Five Inventory in English and German. *Journal of Research in Personality*, 41: 203–212, 2007.

[106] T. W. Robbins, E. J. Anderson, D. R. Barker, A. C. Bradley, C. Fearnyhough, R. Henson, S. R. Hudson, and A. Baddeley. Working memory in chess. *Memory & Cognition*, 24(1):83–93, 1996.

[107] P. Royston. A pocket-calculator algorithm for the Shapiro-Francia test for non-normality: an application to medicine. *Statistics in Medicine*, 12:181–184, 1993.

[108] S. Sager. *Numerical methods for mixed–integer optimal control problems*. Der andere Verlag, Tönning, Lübeck, Marburg, 2005. ISBN 3-89959-416-9.

[109] S. Sager. Reformulations and Algorithms for the Optimization of Switching Decisions in Nonlinear Optimal Control. *Journal of Process Control*, 19(8):1238–1247, 2009.

[110] S. Sager, G. Reinelt, and H. G. Bock. Direct Methods With Maximal Lower Bound for Mixed-Integer Optimal Control Problems. *Mathematical Programming*, 118(1):109–149, 2009.

[111] S. Sager, C. Barth, H. Diedam, M. Engelhart, and J. Funke. Optimization to measure performance in the Tailorshop test scenario — structured MINLPs and beyond. In *Proceedings EWMINLP10*, pages 261–269, CIRM, Marseille, April 12–16 2010.

[112] S. Sager, C. Barth, H. Diedam, M. Engelhart, and J. Funke. Optimization as an analysis tool for human complex problem solving. *SIAM Journal on Optimization*, 21(3):936–959, 2011.

[113] S. Sager, H. Bock, and M. Diehl. The Integer Approximation Error in Mixed-Integer Optimal Control. *Mathematical Programming A*, 133(1–2):1–23, 2012.

[114] S. S. Sawilowsky and C. R. Blair. A more realistic look at the robustness and Type II error properties of the t test to departures from population normality. *Psychological Bulletin*, 111(2): 352–360, 1992.

[115] W. H. Schilders, H. A. van der Vorst, and J. Rommes. *Model Order Reduction: Theory, Research Aspects and Applications*. Springer, Berlin Heidelberg, 2008.

[116] R. Selten, S. Pittnauer, and M. Hohnisch. Dealing with dynamic decision problems when knowledge of the environment is limited: An approach based on goal systems. *Journal of Behavioral Decision Making*, 25:443–457, 2012.

[117] S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3–4):591–611, 1965.

[118] G. Sierksma. *Linear and Integer Programming: Theory and Practice, Second Edition*. Advances in Applied Mathematics. Taylor & Francis, 2001. ISBN 9780824706739.

[119] W. F. Sierpiński. Sur une courbe dont tout point est un point de ramification. *C.R. Acad. Sci. Paris*, 160:302–305, 1915.

[120] Y. Song, S. Ray, and S. Li. Structural properties of buyback contracts for price-setting newsvendors. *Manufacturing & Service Operations Management*, 10(1):1–18, 2008.

[121] L. S.Pontryagin, V. G. Boltayanskii, R. V. Gamkrelidze, and E. F. Mishchenko. *Mathematical Theory of Optimal Processes*. Wiley, 1962.

[122] H.-M. Süß, K. Oberauer, and M. Kersting. Intellektuelle Fähigkeiten und die Steuerung komplexer Systeme. *Sprache & Kognition*, 12:83–97, 1993.

[123] M. Tawarmalani and N. Sahinidis. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*. Kluwer Academic Publishers, 2002.

[124] H. C. Thode. *Testing For Normality*. Statistics, textbooks and monographs. Taylor & Francis, 2002. ISBN 9780824796136.

[125] T. H. Tsang, D. M. Himmelblau, and T. F. Edgar. Optimal control via collocation and non-linear programming. *International Journal on Control*, 21:763–768, 1975.

[126] J. W. Tukey. *Exploratory Data Analysis*. Addison-Wesley Series in Behavioral Science: Quantitative Methods. Addison-Wesley, 1977.

[127] A. M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42:230–265, 1937.

[128] A. Wächter and L. T. Biegler. Line search filter methods for nonlinear programming: Motivation and global convergence. *SIAM Journal on Computing*, 16(1):1–31, 2005.

[129] G. Wallas. *The Art of Thought*. Harcourt, Brace & Company, 1926.

[130] J. B. Watson. Psychology as the behaviorist views it. *Psychological Review*, 20:158–177, 1913.

[131] D. Wenke and P. A. Frensch. *Is success or failure at solving complex problems related to intellectual ability?* The psychology of problem solving. Cambridge University Press, 2003.

[132] T. Westerlund and F. Pettersson. An Extended Cutting Plane Method for Solving Convex MINLP Problems. *Computers & Chemical Engineering*, 19:131–136, 1995.

[133] A. N. Whitehead and B. Russell. *Principia Mathematica, 3 Vols*. Cambridge University Press, 1927. ISBN 9780521067911.

[134] N. Wiener. *Cybernetics or Control and Communication in the Animal and the Machine*. M.I.T. Press, 1961. ISBN 9780262730099.

[135] R. B. Wilson. *A simplicial algorithm for concave programming*. PhD thesis, Harvard University, 1963.

[136] W. W. Wittmann and K. Hattrup. The relationship between performance in dynamic systems and intelligence. *Systems Research and Behavioral Science*, 21:393–409, 2004.

Nomenclature

Throughout this thesis, the variables i, j, and k are used as counting indices. The time is identified with t and x usually refers to state or dependent variables, u denotes control, decision, or free variables and p stands for parameters.

Meanings of Decorations

x^{XY}, x_k^{XY}	Microworld variable XY (in month k)
x^*	Locally/globally optimal values
x^T	Transposed
Δx	Difference, step
$x^{(i)}$	Linearization points in outer approximation, filter points
x^{P}	Values from participant data
$\hat{x}$	Free variables in decomposition master problem
$\hat{x}_k^{(j)}$	Fixed value for variable (j) in microworld round k
$\overline{x}$	Sample mean
x_0	Initial value

Roman Symbols

a_i	Coefficients in decoupled costs model
ch, {f,m}/ch	*Control* and *highscore* groups (female, male)
C	critical region
e	All-ones vector
f^1, f^2	Auxiliary functions in *Tailorshop* model
f_1, f_2	cost functions in decomposition
F	Objective function
F_P	Objective function value of problem P
G, G_i	Equality constraints, state progression law in dMIOCPs
$h(x)$	constraint violation of x, objective value for grid points
H, H_i	Inequality constraints
$\hat{H}_k$	Hessian approximation in iteration k
H_0, H_1	Competing hypotheses in tests
I	Identity matrix
J_G, J_H	Jacobian of G, H
L	List of problems/nodes
LB, lb	lower bound
m	Slope in learning curve fit
M	parameter in Big M relaxation, diagonal matrix with entries μ
n_e, n_i	Number of equality and inequality constraints
n_x, n_u, n_p	Number of states, controls, and parameters

n_x	Time horizon for microworlds
n_y, n_v	Number of integer variables, integer controls
N	Sample size
of,	Optimization-based feedback groups (female, male)
{f,m}/of	
p	Parameters, probability value in hypothesis tests
P, P_i	Nodes in branch and bound
Q_1, Q_3	Lower and upper quartile
s	Slack variables
S	Solution (in algorithms), diagonal matrix with entries s
S_X	Sample variance of sample X
t	Time
t_s	Start time for optimization
t_0, t_f	Start time, end time
T, T_{obs}	Test statistics (value for observation)
T_k^i	Terms in minimum-expression for *sales*
$u(t), u_k$	Control functions/variables
U	Neighborhood
UB, ub	upper bound
$v(t)$	Integer control functions
x	(continuous) Optimization variables
$x(t), x_k$	State functions/variables
X	Feasible region for optimization variables
X_i	Random sample variables
y	Integer optimization variables
y_i	Binary variables in *sales* reformulation
Y	Feasible region for integer optimization variables
Y_i	Boolean variables in GDP reformulation

Greek and Other Symbols

∇	Derivative with respect to x
∇^2_{xx}	Second derivative with respect to x
α	Step size, significance level
α_i	Coefficients of parameter optimization objective
β	Barrier parameter, homotopy parameter
Γ	Boolean function in GDP reformulation
ϵ	Convergence tolerance in outer approximation, equality constraint offset
η	Linearization error in outer approximation
λ	LAGRANGE multipliers (for equality constraints)
μ	LAGRANGE multipliers (for inequality constraints), mean value
Π	Feasible region for parameters in model parameter optimization
ρ	Equality constraint tolerance in decomposition
ϕ_i	Lower level objective function value in multilevel problem
ϕ_H	*How much is still possible*-function
ϕ_P	*Use of potential*-function
ψ	Parameter optimization objective function
Ω	Feasible region for controls

Calligraphic Symbols

$\mathcal{A}$	Active set
$\mathcal{F}$	Feasible set
$\mathcal{K}$	Set of linearization points in outer approximation
$\mathcal{L}$	Lagrangian function

Black Board Symbols

$\mathbb{R}$	Set of real numbers
$\mathbb{R}^n$	Space of n–vectors with elements from the set $\mathbb{R}$
$\mathbb{Z}, \mathbb{Z}_0^+$	Set of (positive) integer numbers (including 0)

List of Figures

List of Tables

List of Algorithms

List of Acronyms

ACT-R Adaptive Control of Thought—Rational (cognitive architecture)
AD ANDERSON-DARLING test
AJAX Asynchronous JavaScript and XML
AMPL A Mathematical Programming Language (algebraic modeling language)
BF BROWN-FORSYTHE test
CAPTCHA Completely Automated Public Turing test to tell Computers and Humans Apart
CPS Complex Problem Solving
CVM CRAMÉR-VON MISES test
DAE Differential Algebraic Equation
dMIOCP discretized mixed-integer optimal control problem
DOS Disk Operating System
DSS Decision Support Systems
ECP extended cutting planes
FMEA Failure Mode Effects Analysis
FTA Fault Tree Analysis
GBD generalized BENDERS decomposition
GDP generalized disjunctive programming
GPS General Problem Solver
GPL GNU General Public License
HTML Hypertext Markup Language
HTTP Hypertext Transfer Protocol
iid independent and identically distributed
IQR Interquartile Range
ITAP IWR Tailorshop analysis problem
ITFP IWR Tailorshop feedback problem
ITSFP IWR Tailorshop sensitivity feedback problem
ITOP IWR Tailorshop optimization problem
ITPOP IWR Tailorshop parameter optimization problem
IWR Interdisciplinary Center for Scientific Computing (German: Interdisziplinäres Zentrum für Wissenschaftliches Rechnen)
KKT KARUSH–KUHN–TUCKER
KS KOLMOGOROV-SMIRNOV test
LF LILLIEFORS test
LICQ Linear Independence Constraint Qualification
LT Logic Theorist
MCDA Multi-Criteria Decision Analysis
MILP mixed-integer linear program
MINLP mixed-integer nonlinear program
MIOCP mixed-integer optimal control problem
MU monetary unit
MySQL Relational database management system using Structured Query Language
NLP nonlinear program

OCP	Optimal Control Problem
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
PEAR	PEARSON's chi-squared test
PHP	PHP: Hypertext Processor (server-side scripting language)
sBB	spatial branch and bound
SD	standard deviation
SF	SHAPIRO-FRANCIA test
SQP	Sequential Quadratic Programming
SW	SHAPIRO-WILK test
XHTML	Extensible Hypertext Markup Language
XML	Extensible Markup Language