OPTIMIZATION AS AN ANALYSIS TOOL FOR HUMAN COMPLEX PROBLEM SOLVING*

SEBASTIAN SAGER†, CAROLA M. BARTH‡, HOLGER DIEDAM§
, MICHAEL ENGELHART§, AND JOACHIM FUNKE‡

Abstract. We present a problem class of mixed-integer nonlinear programs (MINLPs) with nonconvex continuous relaxations which stem from economic test scenarios that are used in the analysis of human complex problem solving. In a round-based scenario participants hold an executive function. A posteriori a performance indicator is calculated and correlated to personal measures such as intelligence, working memory, or emotion regulation. Altogether, we investigate 2088 optimization problems that differ in size and initial conditions, based on real-world experimental data from 12 rounds of 174 participants. The goals are twofold. First, from the optimal solutions we gain additional insight into a complex system, which facilitates the analysis of a participant's performance in the test. Second, we propose a methodology to automatize this process by providing a new criterion based on the solution of a series of optimization problems. By providing a mathematical optimization model and this methodology, we disprove the assumption that the "fruit fly of complex problem solving," the *Tailorshop* scenario that has been used for dozens of published studies, is not mathematically accessible—although it turns out to be extremely challenging even for advanced state-of-the-art global optimization algorithms and we were not able to solve all instances to global optimality in reasonable time in this study. The publicly available computational tool *Tobago* [TOBAGO web site https://sourceforge.net/projects/tobago] can be used to automatically generate problem instances of various complexity, contains interfaces to *AMPL* and *GAMS*, and is hence ideally suited as a testbed for different kinds of algorithms and solvers. Computational practice is reported with respect to the influence of integer variables, problem dimension, and local versus global optimization with different optimization codes.

1. Introduction. 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. 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 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.

Complex problem solving is defined as a *high-order cognitive process*. The complexity may result from one or several different characteristics, such as a coupling of subsystems, nonlinearities, dynamic changes, intransparency, or others [16]. The main

---

† Corresponding author. Interdisciplinary Center for Scientific Computing (IWR), Heidelberg University, 69120 Heidelberg, Germany (sebastian.sager@iwr.uni-heidelberg.de).

‡Department of Psychology, Heidelberg University, 69117 Heidelberg, Germany (carola.barth@psychologie.uni-heidelberg.de, joachim.funke@psychologie.uni-heidelberg.de).

§Interdisciplinary Center for Scientific Computing (IWR), Heidelberg University, 69120 Heidelberg, Germany (hdiedam@ix.urz.uni-heidelberg.de, michael.engelhart@iwr.uni-heidelberg.de).

intention of the research field *complex problem solving* of human beings is the desire to understand how certain *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 [53]. Other interesting personal variables are *working memory* [43], *amount of knowledge* [34], and *emotion regulation* [40]. Situational variables like the impact of *goal specificity and observation* [39], *feedback* [12], and *time constraints* [26] attracted less attention.

Psychologists have been working in the research fields of problem solving for approximately 80 years. One of the groundbreaking works by Ewert and Lambert in 1932 [17] was based on the disk problem, more commonly known as the *Tower of Hanoi*. Since the 1970s and 1980s computer-based test scenarios have also been used, e.g., LEARN [28], Moro [46], FSYS 2.0 [51], and *Tailorshop*, which is the basis for this study. *Tailorshop* is sometimes referred to as the "Drosophila" for problem solving researchers [24] and thus a prominent example for a computer-based test scenario. All mentioned scenarios try to reflect the characteristics of real-life problems by simulating a microworld [27].

The overall idea, compared to early works in problem solving, is still the same: one evaluates the performance of a participant by calculating an *indicator function* and either correlates it to personal attributes, such as the intelligence quotient [32], or analyzes the influence of different experimental conditions for groups of participants [4]. The main difference is that for the early test scenarios the correct solution is known at every stage. For more complex scenarios the performance evaluation is not so straightforward.

In this paper we address the question of how to get a reliable performance indicator for the *Tailorshop* scenario. *Tailorshop* has been used in a large number of studies, e.g., [42], [35], [33], [37], [4], [5]. Also comprehensive reviews on studies and results in connection with *Tailorshop* have been published; see [20], [22], [23], [25], [24], in which more information on the psychological background can be found.

In *Tailorshop* participants make economic decisions to maximize the overall balance of a small company specialized in the production and sales of shirts. To measure performance within the *Tailorshop* scenario, different indicator functions have been proposed in the literature. To use a comparison of accumulated capital at the final month 12 between all participants was proposed in [30]. This criterion seems natural, as this is what the participants 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 participants achieved. Analyzing the temporal evolution of state variables has also been proposed. In [41], [47] the evolution of profit, equivalent to the evolution of capital after interest x_k^{CA}, was proposed. In [21], [5] the evolution of the overall worth of the tailorshop x_k^{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 in infrastructure at the beginning (which is actually decreasing the overall capital as infrastructure loses value over time) to have a higher pay-off toward 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. To overcome this problem, we

propose to compare the decisions to mathematically optimal solutions. For a recent review on *Tailorshop* success criteria, see [15].

Because all previously used indicators have unknown reliability and validity, we propose to compare the decisions to mathematically optimal solutions. Hussy [31, p. 62] writes in 1985[1]

"Only when it will be possible, e.g., by means of mathematical optimization methods, to determine the objectively optimal solution process to compare the process chosen by the proband with it, will this severe problem be overcome."

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 [21], [47], [41]), then the conclusions frequently differ"
as stated by Wenke and Frensch [52, p. 95]. Based on a mathematical model of the *Tailorshop*, an optimization is performed for every round of the participant's data, starting with exactly the same conditions as the participant. By comparing these optimal values that indicate *how much is still possible* if all future decisions were made perfectly, an analysis of at what rounds potential has been lost by decisions can be obtained. Based on optimization theory, even further insight into what decisions were decisive for bad or good performance can be obtained by analyzing Lagrange multipliers.

To our knowledge, numerical optimization methods have only scarcely been used for the analysis of participants' decisions in complex environments like *Tailorshop*. Cognitive psychologists and economists have been using simulation methods for finding optimal solutions for simple tasks within strongly constrained environments. Also, in the context of *experimental economics* studies have been performed, however, to our knowledge not with explicit mathematical representations of the scenarios, including nonlinearities and integer variables. The general approach to compare performance to optimal solutions has been discussed by [36]. However, the authors do not provide a mathematical model for their test scenario *EPEX*. Hence, they need to use the software as a black box for brute-force simulation or derivative free strategies, such as Nelder–Mead [38], or genetic algorithms. Such strategies result in significantly higher computational runtimes, give less insight, and have poor theoretical convergence properties. Our approach formulates the simulation task as equality constraints of the optimization problem and thus allows us to apply modern optimization techniques, including simultaneous strategies that solve simulation and optimization tasks at the same time. They have shown to be superior in many cases; compare, e.g., [8], [7], [2].

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 our case they can be interpreted as discretized optimal control problems. See [44] 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 focus on combinatorial methods.

---

[1]Author's translation from the German original: "Erst wenn es gelingt, z.B. durch mathematische Optimierungsverfahren, den objektiv besten Lösungsweg zu bestimmen, um daran den tatsächlich gewählten Lösungsweg der Pbn messen zu können, wird dieses ernste Problem [die objektive Bestimmung der Problemlösegüte] aus dem Weg zu raümen sein."

Progress in mixed-integer linear programming (MILP) started with the fundamental work of Dantzig and coworkers on the Traveling Salesman problem in the 1950s. Since then, enormous progress has 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 [3]. Also, generic problems can often be solved very efficiently in practice, despite the known exponential complexity from a theoretical point of view [9].

The situation is different in the field of mixed-integer nonlinear programming (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., *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., [1], [10]. Most of them, however, still require the underlying functions to be convex. Comprehensive surveys on algorithms and software for convex MINLPs are given in [29], [11]. 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., [18], [48] for references on general under- and overestimation of functions and sets.

Our intention is to foster interdisciplinary research between psychologists and applied mathematicians. We provide the research community in the field of complex problem solving with the open source software tool *Tobago* [45]. This data generation and analysis tool can be hooked to a variety of optimization solvers. Currently the software supports *AMPL* [19] and *GAMS* [14] interfaces. This allows for the usage of solvers from the *COIN-OR* initiative, which are also available under a public license. In this study we used the global solvers *Couenne* [6] and the local solvers *Bonmin* [10] and *Ipopt* [50]. In addition, we ran the global solver *BARON* [49] on the NEOS server.

It turns out, however, that the size and complexity of the problems presented in this paper lead to extremely long runtimes of the global solvers and can only be used on a small subset of the problems. We present a problem-specific lower bound to avoid bad local maxima and guarantee monotonicity of the analysis function that builds on the locally optimal objective function values. However, additional future work in several mathematical areas will be needed to address all demands of researchers in *complex problem solving*.

The paper is organized as follows. In section 2 we explain the test scenario and derive a mathematical model for the *Tailorshop*. In section 3 details concerning the software implementation and solution of the series of optimization problems are given, together with numerical results. The implications for a psychological study we performed are mentioned in section 4. We give a summary and an outlook to future work in section 5.

2. Tailorshop MINLP model. The *Tailorshop* was developed and implemented as a test scenario in the 1980s by Dörner [16]. It has been used in numerous studies, e.g., [42], [35], [33], [37], [4], [5]. Also, comprehensive reviews have been published (see [20],

[22], [23], [25], [24]), in which more information on the psychological background can also be found.

A participant has to take economic decisions to maximize the overall balance of a small company specialized in the production and sales of shirts. The scenario comprises 12 rounds (months), in which the participant can modify infrastructure (employees, machines, distribution vans), financial settings (wages, maintenance, prices), and logistical decisions (shop location, buying raw material). As feedback he gets some key indicators in the next round, such as the current number of sold shirts, machines, employees, and the like. Arrows next to the indicators show if the value increased or decreased with respect to the previous round.

There are two different kinds of machines to produce either 50 or 100 shirts per month. Workers need to specialize for work on either one of them. The machines need to be maintained 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. Vans influence the demand in a positive way. Furthermore, advertisement, location of the sales shop, and shirt pricing decisions can be used to maximize profit.

We derive a mathematical formulation as an optimization problem. The basic idea is that for different initial values (the current state in round n_s of a participant's test run) the optimal solution for the remaining $N - n_s$ rounds can be calculated. The optimal solution can then either be used for a manual comparison and analysis of the participant's decisions, section 3, or for an automated indicator function, as discussed in section 4.

The *Tailorshop* was developed as a computer-based test scenario in `GW-Basic` code in the early 1980s. This implementation was the starting point for the mathematical modeling process. Figure A.1 in the appendix shows a short extract of this file. The scenario as it is implemented in `GW-Basic` has several shortcomings and assumptions one might disagree with. However, this implementation and similar ones have been used over years, and at the point where interdisciplinary cooperation started, most of the data of the 174 participants had already been evaluated in a cumbersome procedure. Hence the formulation of test scenarios that have better mathematical properties has been postponed to future work, and the mathematical model which we derive from the `GW-Basic` code can be considered as given, even if it is not in all aspects close to reality.

On the basis of the `GW-Basic` code we derived a mathematical optimization problem for a participant and month $0 \leq n_s < N$ as

$$\max_{x,u,s} F(x_N)$$

$$\text{s.t. } x_{k+1} = G(x_k, u_k, s_k, p), \qquad k = n_s \dots N - 1,$$

$$0 \leq H(x_k, x_{k+1}, u_k, s_k, p), \qquad k = n_s \dots N - 1,$$

$$u_k \in \Omega, \qquad k = n_s \dots N - 1,$$

(2.1) $$x_{n_s} = x_{n_s}^p.$$

The model is dynamic with a discrete time $k = 0 \dots N$, where $N = 12$ is the number of rounds. The control vector $u_k = u(k)$ has 15 (or 13 when van purchase is fixed; compare section A.2.2) entries for each $k = 0 \dots N - 1$ corresponding to the decisions the participant can make in the test. The vector of dependent state variables $x_k = x(k)$ comprises 16 entries. Both are listed in Table 2.1 (note that units of control and state variables are only given implicitly depending on how they enter the model equations and constraints). The vector s_k denotes slack variables we introduced to reformulate

TABLE 2.1
Controls and states in the Tailorshop *optimization problem with $k \in \{0, \ldots, 11\}$ for controls, respectively, $k \in \{0, \ldots, 12\}$ for states. Note that units are only given implicitly in the test scenario.*

Decision	u_k	unit[1]	State	x_k	unit[1]
advertisement	u_k^{AD}	MU	machines 50	$x_k^{M_{50}}$	machines
shirt price	u_k^{SP}	MU	machines 100	$x_k^{M_{100}}$	machines
buy raw material	$u_k^{\Delta\mathrm{MS}}$	shirts	workers 50	$x_k^{W_{50}}$	workers
workers 50	$u_k^{\Delta W_{50}}$	workers	workers 100	$x_k^{W_{100}}$	workers
workers 100	$u_k^{\Delta W_{100}}$	workers	demand	x_k^{DE}	shirts
buy machines 50	$u_k^{\Delta M_{50}}$	machines	vans	x_k^{VA}	vans
buy machines 100	$u_k^{\Delta M_{100}}$	machines	shirts sales	x_k^{SS}	shirts
sell machines 50	$u_k^{\delta M_{50}}$	machines	shirts stock	x_k^{ST}	shirts
sell machines 100	$u_k^{\delta M_{100}}$	machines	possible production	x_k^{PP}	shirts
maintenance	u_k^{MA}	MU	actual production	x_k^{AP}	shirts
wages	u_k^{WA}	MU	material stock	x_k^{MS}	shirts
social expenses	u_k^{SC}	MU	satisfaction	x_k^{SA}	—
buy vans	$u_k^{\Delta\mathrm{VA}}$	vans	machine capacity	x_k^{MC}	shirts
sell vans	$u_k^{\delta\mathrm{VA}}$	vans	base capital	x_k^{BC}	MU
choose site	u_k^{CS}	—	capital after interest	x_k^{CA}	MU
			overall balance	x_k^{OB}	MU

[1]MU means money units.

min–max expressions by standard techniques using the constraints (2.27)–(2.31). For details on these and further reformulations, see section A.2.2. We define

$$\left(x^{\mathrm{p}}, u^{\mathrm{p}}\right) = \left(x_0^{\mathrm{p}}, \ldots, x_N^{\mathrm{p}}, u_0^{\mathrm{p}}, \ldots, u_{N-1}^{\mathrm{p}}\right)$$

to be the vector of decisions and state variables for all months of a participant. Certain entries $x_{n_s}^p$ enter (2.1) as fixed initial values. Participant independent initial values $x_0^{\mathrm{p}} = p^{x_0}$ are given alongside fixed parameters p in Table A.1 in the appendix. Random values ξ appear in the equations, e.g., line 2810 in Figure A.1. However, a detailed analysis of the compiled code revealed that the random values are only dependent on an initialization (seed) within the GW-Basic code; hence they are identical for all participants and can be fixed in the optimization problem. See Table A.2 in the appendix.

The goal is to find decisions u_k that maximize the overall balance at the end of the time horizon. The objective function is given by

$$F(x_N) = x_N^{\mathrm{OB}}.$$

Whenever we use the expression *relaxed optimization problem* this will refer to the case in which the sets of points in (2.21)–(2.24) are replaced by their convex hulls. The state propagation law $G(\cdot)$ is determined by the following set of equations for all $k \in \{0, \ldots, 11\}$. For the sake of readability we omit the implicitly given units in the equations.

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:

$$(2.2) \qquad x_{k+1}^{M_{50}} = x_k^{M_{50}} + u_k^{\Delta M_{50}} - u_k^{\delta M_{50}},$$

$$(2.3) \qquad x_{k+1}^{M_{100}} = x_k^{M_{100}} + u_k^{\Delta M_{100}} - u_k^{\delta M_{100}}.$$

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 of workers, but we stick to the hiring control for historical reasons):

$$(2.4) \qquad x_{k+1}^{W_{50}} = x_k^{W_{50}} + u_k^{\Delta W_{50}},$$

$$(2.5) \qquad x_{k+1}^{W_{100}} = x_k^{W_{100}} + u_k^{\Delta W_{100}}.$$

Demand depends on a time dependent parameter p_k^{DE} as well as on the advertisement expenses and the number of vans multiplied by a factor depending on the site, $f^1(u_k^{CS})$ (see section A.2.1),

$$(2.6) \qquad x_{k+1}^{DE} = 100 p_k^{DE} - 50 + \left(\frac{u_k^{AD}}{5} + 100(x_k^{VA} + u_k^{\Delta VA} - u_k^{\delta VA}) \right) f^1(u_k^{CS}).$$

While the influence of advertisement is bounded (see below and section A.2.2) the effect of vans is unbounded. This leads to unboundedness of the whole problem. In section A.2.2 our approach to generate reasonable solutions anyway is described.

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

$$(2.7) \qquad x_{k+1}^{VA} = x_k^{VA} + u_k^{\Delta VA} - u_k^{\delta VA},$$

$$(2.8) \qquad x_{k+1}^{SS} = s_k^{SS},$$

$$(2.9) \qquad x_{k+1}^{ST} = x_k^{ST} + s_k^{PP} - s_k^{SS}.$$

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.

$$x_{k+1}^{PP} = (s_k^{M_{50}}(x_k^{MC} + 4p_k^{P_{50}} - 2) + s_k^{M_{100}}(2x_k^{MC} + 6p_k^{P_{100}} - 3))$$
$$(2.10) \qquad \cdot \left(\frac{1}{2} + \frac{u_k^{WA} - 850}{550} + \frac{u_k^{SC}}{800} \right)^{\frac{1}{2}},$$

$$(2.11) \qquad x_{k+1}^{AP} = s_k^{PP}.$$

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:

$$(2.12) \qquad x_{k+1}^{MS} = x_k^{MS} + u_k^{\Delta MS} - s_k^{PP},$$

$$(2.13) \qquad x_{k+1}^{\mathrm{SA}} = \frac{1}{2} + \frac{u_k^{\mathrm{WA}} - 850}{550} + \frac{u_k^{\mathrm{SC}}}{800},$$

$$(2.14) \qquad x_{k+1}^{\mathrm{MC}} = s_k^{\mathrm{MC}}.$$

The equation for base capital,

$$
\begin{aligned}
x_{k+1}^{\mathrm{BC}} = {} & x_k^{\mathrm{CA}} + s_k^{\mathrm{SS}} \cdot u_k^{\mathrm{SP}} - p_k^{\mathrm{PR}} \cdot u_k^{\mathrm{\Delta MS}} - 10000 u_k^{\Delta M_{50}} - f^2(u_k^{\mathrm{CS}}) \\
& + 8000 \frac{x_k^{\mathrm{MC}}}{p^{\mathrm{MM}}} u_k^{\delta M_{50}} - 20000 u_k^{\Delta M_{100}} + 16000 \frac{x_k^{\mathrm{MC}}}{p^{\mathrm{MM}}} u_k^{\delta M_{100}} \\
& - u_k^{\mathrm{AD}} - u_k^{\mathrm{MA}} - (x_k^{W_{50}} + u_k^{\Delta W_{50}} + x_k^{W_{100}} + u_k^{\Delta W_{100}}) \cdot (u_k^{\mathrm{WA}} + u_k^{\mathrm{SC}}) \\
& - 2 s_k^{\mathrm{PP}} - \frac{1}{2}(x_k^{\mathrm{MS}} + u_k^{\mathrm{\Delta MS}} - s_k^{\mathrm{PP}}) - x_k^{\mathrm{ST}} - 10000 \cdot u_k^{\Delta \mathrm{VA}} \\
& + (8000 - 100k) \cdot u_k^{\delta \mathrm{VA}} - 500(x_k^{\mathrm{VA}} + u_k^{\Delta \mathrm{VA}} - u_k^{\delta \mathrm{VA}}),
\end{aligned}
\tag{2.15}
$$

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 $s_k^{\mathrm{SS}} \cdot u_k^{\mathrm{SP}}$, the sale of machines $8000(x_k^{\mathrm{MC}}/p^{\mathrm{MM}})u_k^{\delta M_{50}}$ and $16000(x_k^{\mathrm{MC}}/p^{\mathrm{MM}})u_k^{\delta M_{100}}$ (depending on the current machine capacity), and the sale of vans $(8000 - 100k) \cdot u_k^{\delta \mathrm{VA}}$.

Money is spent for the raw material bought times the price of a raw material unit $-p_k^{\mathrm{PR}} \cdot u_k^{\mathrm{\Delta MS}}$, the purchase of machines $-10000 u_k^{\Delta M_{50}}$ and $-20000 u_k^{\Delta M_{100}}$, the purchase of vans $-10000 u_k^{\Delta \mathrm{VA}}$, advertisement and maintenance $-u_k^{\mathrm{AD}} - u_k^{\mathrm{MA}}$, and the number of workers times wages plus social expenses $-(x_k^{W_{50}} + u_k^{\Delta W_{50}} + x_k^{W_{100}} + u_k^{\Delta W_{100}}) \cdot (u_k^{\mathrm{WA}} + u_k^{\mathrm{SC}})$. Additionally, each unit of material in stock at the end of a round costs half a money unit (MU) $-1/2(x_k^{\mathrm{MS}} + u_k^{\mathrm{\Delta MS}} - s_k^{\mathrm{PP}})$, the production of a shirt costs two MU $-2s_k^{\mathrm{PP}}$, each shirt in stock costs one MU, and each van costs 500 MU per round $-500(x_k^{\mathrm{VA}} + u_k^{\Delta \mathrm{VA}} - u_k^{\delta \mathrm{VA}})$. There is another amount of money to be paid, which depends on the site, $-f^2(u_k^{\mathrm{CS}})$ (see section A.2.2).

From the base capital, the capital after interest is computed by multiplying it with an interest rate factor $(1 + p^{\mathrm{IR}})$. Overall balance, the objective function, beside 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:

$$(2.16) \qquad x_{k+1}^{\mathrm{CA}} = x_{k+1}^{\mathrm{BC}}(1 + p^{\mathrm{IR}}),$$

$$
\begin{aligned}
x_{k+1}^{\mathrm{OB}} = {} & \frac{x_k^{\mathrm{MC}}}{p^{\mathrm{MM}}} \left(8000(x_k^{M_{50}} + u_k^{\Delta M_{50}} - u_k^{\delta M_{50}}) + 16000(x_k^{M_{100}} + u_k^{\Delta M_{100}} - u_k^{\delta M_{100}}) \right) \\
& + (8000 - 100k) \cdot (x_k^{\mathrm{VA}} + u_k^{\Delta \mathrm{VA}} - u_k^{\delta \mathrm{VA}}) \\
& + 2(x_k^{\mathrm{MS}} + u_k^{\mathrm{\Delta MS}} - s_k^{\mathrm{PP}}) + 20(x_k^{\mathrm{ST}} + s_k^{\mathrm{PP}} - s_k^{\mathrm{SS}}) + x_k^{\mathrm{CA}}.
\end{aligned}
\tag{2.17}
$$

The feasible set of controls is defined by the following properties for all $k \in \{0, \ldots, 11\}$:

$$(2.18) \qquad u_k^{\mathrm{AD}} \in [0, 10000], \qquad u_k^{\mathrm{SP}} \in [10, 100],$$

$$(2.19) \qquad u_k^{\mathrm{\Delta MS}} \in [0, 50000], \qquad u_k^{\mathrm{MA}} \in [0.1, 100000],$$

$$(2.20) \qquad u_k^{\text{WA}} \in [850, 5000], \qquad u_k^{\text{SC}} \in [0, 10000],$$

$$(2.21) \qquad u_k^{\Delta W_{50}} \in \{-200, -199, \ldots, 200\}, \qquad u_k^{\Delta W_{100}} \in \{-200, -199, \ldots, 200\},$$

$$(2.22) \qquad u_k^{\Delta M_{50}} \in \{0, 1, \ldots, 200\}, \qquad u_k^{\Delta M_{100}} \in \{0, 1, \ldots, 200\},$$

$$(2.23) \qquad u_k^{\delta M_{50}} \in \{0, 1, \ldots, 200\}, \qquad u_k^{\delta M_{100}} \in \{0, 1, \ldots, 200\},$$

$$(2.24) \qquad u_k^{\text{CS}} \in \{0, 1, 2\}.$$

Furthermore, for all $k \in \{0, \ldots, 11\}$ the constraints

$$(2.25) \qquad u_k^{\Delta W_{50}} \geq -x_k^{W_{50}}, \qquad u_k^{\Delta W_{100}} \geq -x_k^{W_{100}},$$

$$(2.26) \qquad u_k^{\delta M_{50}} \leq x_k^{M_{50}}, \qquad u_k^{\delta M_{100}} \leq x_k^{M_{100}}$$

need to hold. Slack variables are used to reformulate min expressions (see also section A.2.2), and the bounds on the slack variables read as

$$(2.27) \qquad s_k^{\text{PP}} \leq x_k^{\text{MS}} + u_k^{\Delta\text{MS}}, \qquad s_k^{\text{PP}} \leq x_{k+1}^{\text{PP}},$$

$$(2.28) \qquad s_k^{\text{MC}} \leq p^{\text{MM}}, \qquad s_k^{\text{MC}} \leq 0.9 x_k^{\text{MC}} + 0.017 \frac{u_k^{\text{MA}}}{x_{k+1}^{M_{50}} + 10^{-8} x_{k+1}^{M_{100}} + 10^{-8}},$$

$$(2.29) \qquad s_k^{\text{SS}} \leq x_k^{\text{ST}} + x_{k+1}^{\text{AP}}, \qquad s_k^{\text{SS}} \leq \frac{5}{4}\left(\frac{x_k^{\text{DE}}}{2} + 280\right) \cdot 2.7181^{-\frac{u_k^{\text{SP}^2}}{4250}},$$

$$(2.30) \qquad s_k^{M_{50}} \leq x_{k+1}^{W_{50}}, \qquad s_k^{M_{50}} \leq x_{k+1}^{M_{50}},$$

$$(2.31) \qquad s_k^{M_{100}} \leq x_{k+1}^{W_{100}}, \qquad s_k^{M_{100}} \leq x_{k+1}^{M_{100}}$$

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. Here, the first 10^{-8} in the denominator comes from a modeling bug; see section A.2.2. Finally, s_k^{SS} is used to reformulate the minimum of shirts available for sale $x_k^{\text{ST}} + x_{k+1}^{\text{AP}}$ and a nonlinear term depending on the demand and the shirt price. Note that 2.7181 has been used in the GW-Basic code instead of exp.

To sum up, every single optimization problem is of the general form (2.1), where 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.

3. Optimization and numerical results. We want to solve a series of optimization problems of the form (2.1) for different participant data that have been obtained experimentally. In section 3.1 we describe the algorithms and software we used to achieve this goal. In section 3.2 examples of optimal solutions are displayed and discussed for illustration. The important issues of integrality and nonconvexity that arise in our problems are discussed in section 3.3. We close by discussing the use of Lagrange multipliers of artificial constraints as a means to further investigate good and bad decisions of a participant in section 3.4.

3.1. Implementation. To be able to analyze and visualize the data in a convenient way, to have a simulation environment for own studies, and to be able to automatize the optimization of all $2088 = 174 \cdot 12$ problems, we implemented the software framework *Tobago* [45]. It is publicly available under an open source license, includes a GUI, and may as well be used for experimental setups. In this study, however, we exclusively used the `GW-Basic` implementation for tests to have consistent data and *Tobago* only for optimization and analysis.

We interface the data with optimization solvers via an automated call of *AMPL* [19] to be able to easily exchange optimization solvers that have an *AMPL* interface. In this study we compare three different optimization solvers: *Ipopt* [50], *Bonmin* [10], and *Couenne* [6]. The first one is a local nonlinear programming solver based on an interior point method. *Bonmin* is a solver for MINLPs whose continuous relaxation is convex (*convex MINLPs*) and uses *Ipopt* for the solution of relaxed problems. *Couenne* is a global solver for MINLPs whose continuous relaxation is nonconvex (*nonconvex MINLPs*). All three are available within the *COIN-OR* open source initiative. We used the most current stable version, 0.2.2, of *Couenne*, and for better comparability the versions 1.1.1 of *Bonmin* and 3.6.1 of *Ipopt* it is interfaced with. For all solvers we used the default settings exclusively and the MA27 sparse solver for numerical linear algebra.

All computational times refer to a two core Intel CPU with 3GHz and 8GB RAM run under Ubuntu 9.10.

3.2. Optimal solutions. In total, 2088 optimization problems have been solved. Depending on the value of n_s in (2.1), each consists of $13(N - n_\mathrm{s})$ control, $16(N - n_\mathrm{s})$ state, and $5(N - n_\mathrm{s})$ slack variables. The total number of optimized variables for all 174 participants sums up to

$$n_\mathrm{var} = 174 \sum_{n_\mathrm{s}=0}^{N-1} 34(N - n_\mathrm{s}) = 174 \cdot 2652 = 461448.$$

This many variables are obviously difficult to discuss and visualize comprehensively. From this large set of results we chose a few which illustrate how optimal solutions relate
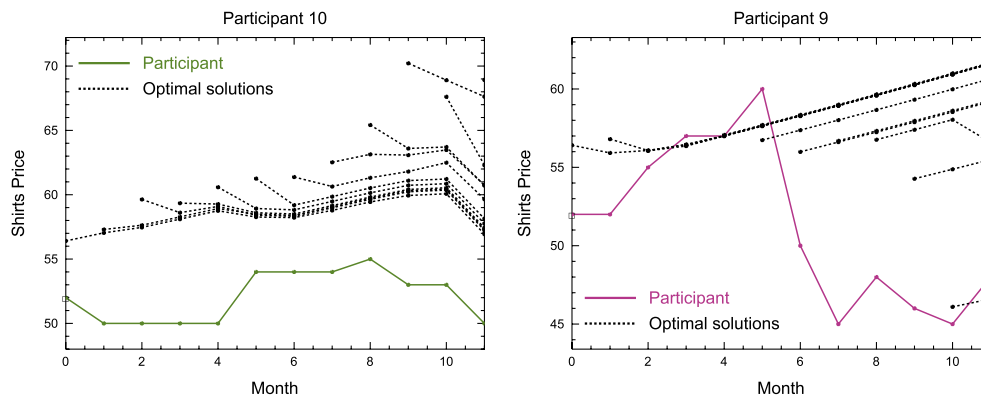

FIG. 3.1. *Shirt price u_k^{SP} of two different participants in solid lines. The dotted lines show optimal solutions of (2.1) starting at different months n_s. Both participants should have chosen higher prices most of the time. Depending on their other choices, the optimal solutions evolve differently over time. On the left-hand side the participant's tailorshop is developing towards high demand and little stock of shirts; hence the optimal shirt price to maximize profit is increasing. On the right-hand side the demand is declining and the stock of shirts increasing; hence the optimal price is falling with time.*
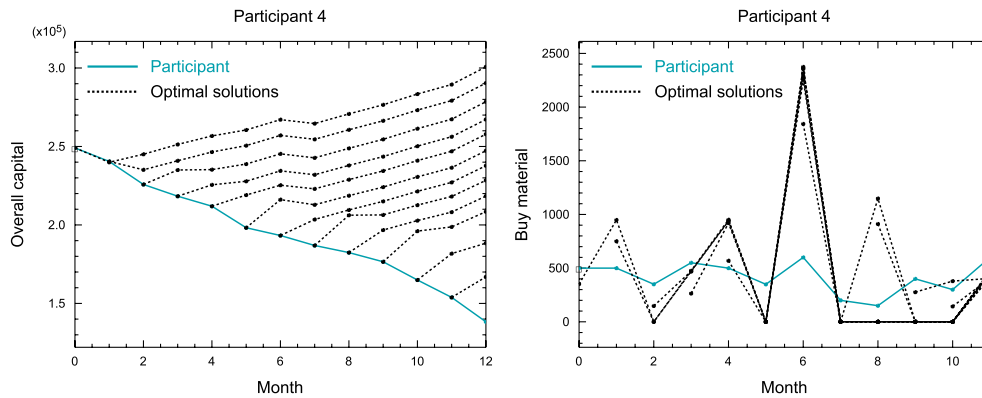
FIG. 3.2. *Left: State variable overall capital balance x_k^{OB}. The participant's trajectory in solid; the optimal solutions of (2.1) starting at different months n_s in dotted lines. The function is almost monotonically increasing, which is due to the number of vans being fixed to the participant's decision. The purchase of raw material is the main reason for the kink at month 6. Right: Purchase of raw material. Because of the comparatively low price in month 6 (compare Table A.2) a large part of the material that is needed for the months 7–12 is bought. Because the participant herself/himself did not do this, an additional kink at month 8 occurs for optimal solutions with $n_s = 7, 8$. This is qualitatively similar for almost all data sets.*

to the choices made by the participant, compared to solutions for different values of n_s, and compared to optimal solutions of other participants. These solutions have been obtained with the local optimization code *Ipopt* and an outer loop with random start values for the optimization. Hence it needs to be stressed that the interpretations are always under the assumption that the obtained results are close to the global optima.

In Figure 3.1 the shirt price control function u_k^{SP} of two participants is displayed. In addition to the values chosen by the participants, all optimal solutions are also depicted, giving an idea of what the participants could have done to improve their performance. It is interesting to observe that the optimal solutions corresponding to the two participants show different behavior, depending on the start values $x_{n_s}^p$ in (2.1).

In Figure 3.2 the important state variable x_k^{OB} is depicted for one representative participant. As the value of this function at the end time $k = N$ is the objective function that is to be maximized, the function shows how much better the optimal solution performs in comparison to the participant. There are only minor deviations from a monotonic increase that result mainly from the investment into raw material which is not profitable within the overall balance, but as a resource for future profit.

3.3. Local maxima and integer solutions. The optimization problems (2.1) are nonconvex. Depending on initial values for the optimization variables different local maxima can be found. Hence one has to use a global optimization solver, such as *Couenne* or one of the solvers listed on [13]. As mentioned above, we used different solvers to obtain solutions. Table 3.1 shows an overview of average computational times and objective function values that were obtained with *Ipopt* and *Bonmin*.

We ran the global solvers *Couenne* and *BARON* only on single optimization instances, as the computational demand was too high. On typical instances, *Couenne* was able to solve (2.1) for $n_s = 11$ in approximately 3 seconds. For the next larger problem, $n_s = 10$, however, the branch-and-bound tree grew too fast. The solver terminated after processing 600000 nodes in 7 hours because the computer ran out of memory. The stack comprised about 2 million open nodes at that time. The best solution at that time was 500497 with the upper bound of 506610 still leaving a certain gap. For comparison,

TABLE 3.1
Average computational times in seconds and average objective function values for the solutions of problems (2.1) *per participant calculated from the* 174 *data sets. The rows show the start month* n_s, *the columns results for* Ipopt *for the relaxation of* (2.1), *and* Bonmin, *respecting the integrality conditions.*

	CPU [sec]		Objective function		
n_s	*Ipopt*	*Bonmin*	*Ipopt*	*Bonmin*	Gap
0	0.65	2289	397613	340163	4.5 %
1	0.62	1545	379243	319133	6.9 %
2	0.48	908	359958	296037	6.4 %
3	0.39	556	341110	282423	10.3 %
4	0.33	366	323728	274949	9.6 %
5	0.31	163	307665	263333	12.8 %
6	0.25	66	292389	254858	14.4 %
7	0.20	14	277730	251187	15.1 %
8	0.15	5.48	262800	235850	17.2 %
9	0.11	2.34	249186	233318	17.8 %
10	0.07	0.54	236290	220031	15.9 %
11	0.03	0.10	220717	210760	14.4 %

the objective function values found by *Bonmin* and *Ipopt* are 490385 and 500779, respectively. When heuristic nonconvexity options `num_resolve_at_root` and `num_resolve_at_node` are used with a value of 1 (or 2) for *Bonmin*, an integer solution with value 500188 (500438) is found after 142 (317) seconds, which is considerably higher than the 0.2 seconds with the standard settings. With tight bounds on all state, control, and slack variables (some of them even fixed) and the newer version *Couenne* 0.3.2 a solution could be obtained in 30 minutes, but even so $n_s = 9$ was not solvable on our machine.

A similar behavior occurred when we used *BARON* with our *GAMS* interface on the NEOS server. Although computational times are not comparable due to the different servers and the different preprocessing steps of *AMPL* and *GAMS*, the runtime for *BARON* also increased drastically when the number of variables doubled from $n_s = 11$ to $n_s = 10$. While instances for $n_s = 11$ could be solved within 3 seconds, the ones for $n_s = 10$ could only be solved in the time limit of 8 hours when bounds were tightened to small intervals. An exact investigation of the reasons for this drastic increase in computational demand is future work.

Obviously for one participant data set the computational times are prohibitive for global approaches. For the analysis of all 174 participants we therefore solved 2088 NLP relaxations and MINLPs with the local optimizers *Ipopt* and *Bonmin*.

A crucial feature of our method is that the *How much is still possible*–function (see section 4.1) decreases monotonically with n_s increasing. To take this into account, we exploit this knowledge in our a posteriori analysis. We define

$$(x^*, u^*, s^*) = (x_{n_s}^*, \ldots, x_N^*, u_{n_s}^*, \ldots, u_{N-1}^*, s_{n_s}^*, \ldots, s_{N-1}^*)$$

as a locally optimal solution obtained by solving problem (2.1) for month n_s. We initialize the variables for problem (2.1) for month $n_s - 1$ according to

$$x_{n_s-1} = x^p_{n_s-1},$$
$$u_{n_s-1} = u^p_{n_s-1},$$
$$x_k = x^*_k, \qquad k = n_s \dots N,$$
$$u_k = u^*_k, \qquad k = n_s \dots N-1,$$
(3.1)
$$s_k = s^*_k, \qquad k = n_s \dots N-1,$$

and s_{n_s-1} according to (A.6–A.10). This is a feasible solution because of $x_{n_s} = x^*_{n_s} = x^p_{n_s} = G(x^p_{n_s-1}, u^p_{n_s-1}, s_{n_s-1}, p)$ with objective function value $x^{*,\mathrm{OB}}_N$. To avoid local maxima with a worse performance, we require that the inequality

(3.2)
$$x^{\mathrm{OB}}_N \geq x^{*,\mathrm{OB}}_N$$

holds. This inequality can either be added to (2.1) when relaxed problems are solved with local optimization algorithms or be used as a *cutoff* value in a branch-and-bound setting to reduce the search tree. Computational experience shows that the primal-dual interior point solver we are using cannot exploit the initialization to its full extent, and in many cases *Ipopt* converged to locally infeasible points although it started from a primally feasible one. Future studies should therefore include active-set based solvers. For this study we iterated in an inner loop with random initializations until for all problems inequality (3.2) was fulfilled, i.e., *Ipopt* returned a feasible solution.

Within our analysis approach, local maxima can lead to a violation of the goal to have an objective measurement for participant performance. Whenever possible, global solvers with a guaranteed, deterministic global maximum should be used. If the size of the problem is still too large for current algorithms and computational platforms, we propose to use relaxations and include (3.2) as a heuristic compromise.

Several of the control variables are restricted to integer values; compare (2.18)–(2.24). A comparison of (locally) optimal relaxed and integer solutions shows that some of the variables show typical (qualitatively similar throughout all solutions, e.g., variables
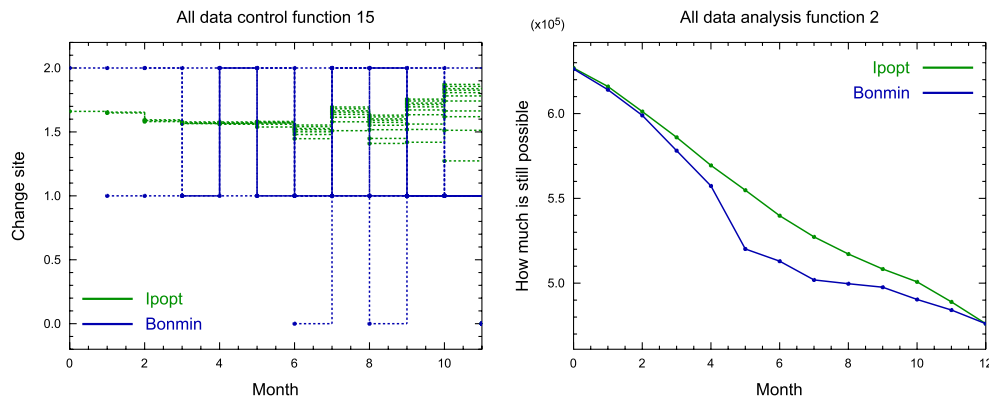

FIG. 3.3. *Left: Optimal choices of site for one participant and all start months* n_s, *calculated with* Ipopt *(green, relaxed values between* 1.1 *and* 1.9*) and* Bonmin *(blue, integer values of* 0, 1, *and* 2*). Right:* How much is still possible–*function for one participant, calculated with* Ipopt *(green, upper curve) and* Bonmin *(blue, lower curve). As in this figure, the integer gap seems to be largest for intermediate values of* n_s *for most instances; compare also the average values in Table* 3.1*. However, this interpretation is subject to the fact that all solutions are only locally optimal.*

are at their upper bounds) behavior for most $x_{n_{\mathrm{s}}}^p$, such as the maintenance u_k^{MA} or the purchase of raw material $u_k^{\Delta\mathrm{MS}}$. Others, in particular the numbers of machines and workers, the shirt price u_k^{SP}, and the choice of the site u_k^{CS}, are more sensitive to local optima and/or the fixation of some of the variables to integer values. Figure 3.3 shows an example.

3.4. Analyzing Lagrange multipliers. Using optimization as an analysis tool yields insight on several levels. A priori unknown structural properties of the problem, e.g., the unboundedness due to the van bug, can be detected. Also the performance of a participant can be compared to the optimal solution, and the *How much is still possible–* function to be discussed in section 4 delivers a temporal resolution of this performance.

But even a more detailed analysis is possible. While an analysis of the *How much is still possible–*function indicates at what rounds the participant made particularly good or bad decisions, the question of what of the decisions contributed significantly to the success or failure remains and might be of importance in a given test scenario. A global approach[2] would be to fix exactly one entry of $u_{n_{\mathrm{s}}}$ to the value chosen by the participant and compare the result of the optimization to the one without this constraint. The difference between the two objective function values then indicates exactly how much impact this particular decision had. The obvious drawback is that the number of optimization problems that need to be solved increases by a factor of $N \cdot n_u$, where n_u is the number of controls per month.

As a compromise we propose to combine two concepts. First, the comparison of the participant's decisions at month n_{s} with the optimal solution, $u_{n_{\mathrm{s}}}^{\mathrm{p}} - u_{n_{\mathrm{s}}}^*$, gives a global indication of differences in the controls. However, it is unclear from this comparison how significant a single deviation is. Therefore we use, second, Lagrange multipliers for the participant's decisions to measure the effect on the objective function. We augment problem (2.1) with the additional constraint

$$(3.3) \qquad u_{n_{\mathrm{s}}} = u_{n_{\mathrm{s}}}^{\mathrm{p}}$$

to obtain the optimization problem

$$\max_{x,u,s} F(x_N)$$
$$\text{s.t. } x_{k+1} = G(x_k, u_k, s_k, p), \qquad k = n_{\mathrm{s}} \dots N - 1,$$
$$0 \le H(x_k, x_{k+1}, u_k, s_k, p), \qquad k = n_{\mathrm{s}} \dots N - 1,$$
$$u_k \in \Omega, \qquad k = n_{\mathrm{s}} + 1 \dots N - 1,$$
$$x_{n_{\mathrm{s}}} = x_{n_{\mathrm{s}}}^p,$$
$$(3.4) \qquad u_{n_{\mathrm{s}}} = u_{n_{\mathrm{s}}}^{\mathrm{p}}.$$

Note that necessarily $x_{n_{\mathrm{s}}+1}^* = x_{n_{\mathrm{s}}+1}^{\mathrm{p}}$; hence problem (3.4) for month n_{s} has the same solution as problem (2.1) for $n_{\mathrm{s}} + 1$. The replacement of (2.1) by (3.4) will yield the same results for the series of all n_{s} and does not imply the need for additional optimization problems to be solved.

The advantage of formulation (3.4) is that an optimization code will also calculate the dual variables or *Lagrange multipliers* $\lambda_{n_{\mathrm{s}}}$ for the constraints (3.3). 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, it needs to be

---

[2]We assume that we solve all optimization problems to global optimality in this section.

TABLE 3.2

Lagrange multipliers for the specific case of one participant and the final month $n_s = 11$. The columns show different entries of the control vector $u(\cdot)$; compare Table 2.1. The rows show three things: first, the decisions $u_{n_s}^p$ taken by the participant; second, the optimal (relaxed) solution $u_{n_s}^*$ calculated with Ipopt; third, the Lagrange multipliers λ_{n_s} for the constraints (3.3).

	u_k^{AD}	u_k^{SP}	$u_k^{\Delta MS}$	u_k^{MA}	u_k^{WA}	u_k^{SC}	u_k^{CS}	
$u_{n_s}^p$	3700	53	999	1400	1130	100	1	
$u_{n_s}^*$	4e-07	64.9	3e-07	34.3	1510	4e-07	0	
λ_{n_s}	-1.003	473	-1.2	-1.003	10.7	4.9	-752	
	$u_k^{\Delta W_{50}}$	$u_k^{\Delta W_{100}}$	$u_k^{\Delta M_{50}}$	$u_k^{\Delta M_{100}}$	$u_k^{\delta M_{50}}$	$u_k^{\delta M_{100}}$	$u_k^{\Delta VA}$	$u_k^{\delta VA}$
$u_{n_s}^p$	0	0	0	0	0	0	1	0
$u_{n_s}^*$	-4.9	0	0	0	2.9	0	1	0
λ_{n_s}	-1233	3990	547.1	-4050	2552	40	-3726	619

stressed that this information is a local one for the point $(x_{n_s}^p, \ldots, x_N^p, u_{n_s}^p, \ldots, u_{N-1}^p)$ and assumes that the active-set of inequality constraints does not change. As an example the Lagrange multiplier for the shirt price $\lambda_{n_s}^{SP}$ will denote the deviation of the objective function for $u_{n_s}^{SP} + \epsilon$. Table 3.2 shows an example. The control vector of a participant, the optimal choice of controls, and the Lagrange multipliers are listed.

The analysis of a participant's decisions hence needs to take into account both the global information of the difference $u_{n_s}^p - u_{n_s}^*$ and the local quantification from the Lagrange multipliers λ_{n_s}. A good estimate can be obtained from the entries of the componentwise product $\lambda_{n_s} \cdot (u_{n_s}^p - u_{n_s}^*)$.

4. A correct indicator function for Tailorshop. We propose to use the solutions of (2.1) for all n_s as an indicator function for the performance of a participant. The approach described in section 4.1 is generic and should also be used for other test scenarios in complex problem solving in the future. In section 4.2 we describe the results we obtained by using this indicator function for a psychological study.

4.1. How much is still possible. On an individual basis, the performance of every participant can be better understood by a comparison with optimal solutions as illustrated in section 3. For an evaluation of large data sets that shall be related to characteristics of participants or experimental setup, an automatization and a reduction to an indicator function are necessary. Once the performance of all participants has been determined, an aggregation and further statistical analysis can be performed.

To measure performance within the *Tailorshop* scenario, different indicator functions have been proposed in the literature. As discussed in the introduction, they have usually unknown reliability and validity.

We solve an optimization problem (2.1) for every round of the participant's data, starting with exactly the same conditions as the participant. We compare these optimal values that indicate *How much is still possible* if all future decisions were optimal. Thus, we can analyze at what rounds potential for a higher end time objective function value has not been used.

A comparison of the end time capital with the one of the optimal solution from start month $n_s = 0$ (which is identical for all participants if we neglect the van purchase decision; compare section A.2.2) also gives an objective indicator. However, what we propose is a far more powerful analysis approach: we want to also say *when* (within the 12

rounds) significant performance deviations occurred, and we want to specify details on 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 $n_\mathrm{s} = 0$ would not yield a good indicator function, as there might be multiple ways to perform well. E.g., if, due to his previous actions, a participant has many shirts in his stock, good decisions may differ drastically from the optimal solution for starting month $n_\mathrm{s} = 0$ in which in every instance all shirts have been sold.

In a certain analogy to the cost-to-go-function in dynamic programming, the optimal objective function values for *all* rounds yield the monotonically decreasing *How much is still possible*–function. We look at the series of optimal objective function values $F^*(x_N; n_\mathrm{s})$ for $n_\mathrm{s} = 0, \ldots, N - 1$. By comparing $F^*(x_N; n_\mathrm{s} = k)$ with $F^*(x_N; n_\mathrm{s} = k + 1)$ we obtain the exact value of how much less the participant is still able to obtain, assuming he would take the best solutions from now on—in other words, whether the tailorshop is in a worse situation than it could be after the participant's decisions. We define the nonpositive (for global optima) *Use of potential*–function

$$(4.1) \qquad \Delta P_k =: F^*(x_N; n_\mathrm{s} = k + 1) - F^*(x_N; n_\mathrm{s} = k).$$

Note that in general a relative loss given as a percentage can also be used; however, this does not make sense when the function $F^*(\cdot)$ is not bounded as in our case.

As indicated in Figure 4.1 different ways to analyze the complex solving process may yield different results. Also, the important issue of selling all shirts and material in the last round is only insufficiently captured by the previous indicator functions. For most of the participants' data the previous indicator and the new, optimization-based one coincide; compare Figure 4.2(right). This is mainly due to the fact that two of the main effects to make nonintuitive investments into the future were almost never found by the participants: first, the purchase of a high number of vans to stimulate demand (compare
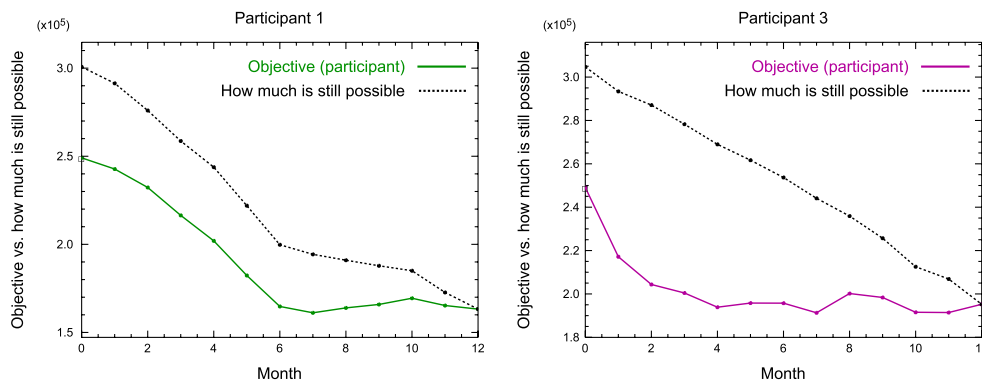

FIG. 4.1. *Different ways of determining good and bad participant–performance over time. The solid lines show the evolution of the objective function. The dotted lines show the* How much is still possible–*function which is composed of objective function values of separate optimization problems* (2.1). *The traditional way is to compare the changes in the objective function value. In our approach we compare the slopes of the* How much is still possible-*function. Left participant: The two variants would qualitatively coincide: not so good from* 0–6, *good performance from* 7–10, *not so good again from* 11–12. *In the right scenario the objective function values seem to correspond to alternations in the quality of the performance, which can not be verified by analyzing the* How much is still possible–*function which has an almost constant negative slope.*
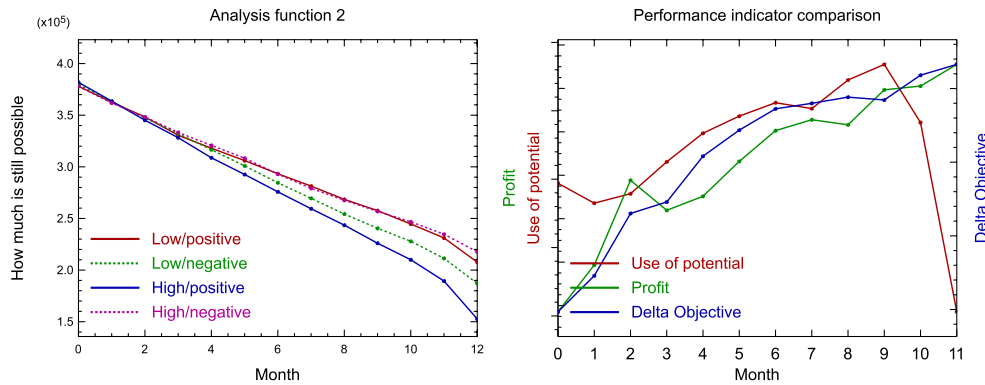
FIG. 4.2. *Left: Average values of the* How much is still possible–*function over all participants with emotion regulation properties/feedback a) low/positive, b) low/negative, c) high/positive, d) high/negative. Participants with a low ability of emotion regulation performed better with positive feedback; those with high ability of emotion regulation performed better with negative feedback. Right: Average values over all 174 participants for different indicator functions. The Use of potential–function ΔP_k is given by (4.1). Profit indicates $\Delta x_k^{CA} = x_{k+1}^{CA} - x_k^{CA}$, Delta objective indicates $\Delta x_k^{OB} = x_{k+1}^{OB} - x_k^{OB}$. The trajectories have been rescaled for better comparability. The potential in month $n_s = 11$ (selling all of the material on stock) has not been used by the majority of participants.*

section A.2.2) and second, the knowledge about the lowest price of the material in round 6.

We conclude that the newly proposed methodology is more reliable and generally applicable to test scenarios in complex problem solving. Non-optimization-based indicator functions give good estimates as long as the aforementioned effects are not exploited, which is to be expected, e.g., in studies of learning behavior when participants would be tested several times.

4.2. Impact of emotion regulation. In the study 174 data sets were used, every one from a different participant who had but one try. For 42 of them *positive feedback* was used in the sense that in every round, regardless of the decisions the participant took, a sum of 20000 money units (MUs) was added to the capital. For 42 participants *negative feedback* in the form of a reduction of 8000 MUs was implemented. These modifications are implemented in the model and readjusted in the a posteriori analysis, of course.

In a previous study [5] it was shown that participants who receive negative feedback perform better than those who receive positive feedback. In our new study we additionally considered the ability to regulate emotion. The psychological results of this study are explained in [4] in which details on the experimental setup can also be found. As a main result, an interaction between feedback and emotion regulation could be shown: participants with a high ability of emotion regulation perform better when they get negative feedback, while those with a low ability to regulate their emotions perform badly for negative and well for positive feedback. This is illustrated in Figure 4.2(left).

In a second study, films and music were used to induce happy, neutral, and sad effects. Additionally we measured emotion regulation. The study was based on data from 90 participants, 30 in each effect condition. Again, emotion regulation had a great impact on complex problem solving. A high ability to regulate emotion improved complex problem solving and reduced the number of mistakes.

5. Summary and outlook. We presented a challenging problem class of nonconvex MINLPs. They originate from economic test scenarios that are used in the analysis of human complex problem solving. Starting from GW-Basic source code of the test scenario we developed a mathematical optimization model to optimize performance starting from prespecified initial values. This model needed to be reformulated in several ways to avoid nondifferentiabilities, division by zero, and unboundedness.

The *Tailorshop* test scenario was invented more than 25 years ago, without any intention to set it up suited for mathematical optimization. Our study revealed several shortcomings of the model. This insight can be used for defining better test scenarios in the future. All characteristics such as nondifferentiabilities, random values, or unbounded decision variables should be left out, as they do not really contribute to the difficulty of the scenario itself but mainly to the difficulty of solving the problem mathematically to optimality.

We solved altogether 2088 optimization problems and discussed the role of integer variables and the nonconvexity by comparing different algorithms. The difficulties in doing so for a large number of medium-scale nonconvex MINLPs are challenging. We formulated and used a structure exploiting lower bound to exclude certain unwanted local maxima. The optimization results were used in two ways. First, to gain additional insight into individual performance by comparing it to the optimal solution which is often nonobvious. Second, to use the results in an automated way as a new analysis tool for process-dependent evaluation of the performance.

This novel methodology yields a valuable (and accessible [45]) analysis tool for psychologists to evaluate participants' performance. We discussed why there is no alternative to the *How much is still possible*–function, especially when participants have more insight, e.g., by repetition of tests. Furthermore, we proposed to add artificial constraints to the optimization problem and use the Lagrange multipliers of these constraints as an indication of what decisions contributed significantly to good or bad performance. By providing this mathematical technology to analyze participants' decisions in more detail, a whole set of interesting scenarios with a time- and decision-specific resolution can be included in future psychological investigations.

This paper provides a reference for researchers in complex problem solving. But we also hope for a stimulating effect on optimization. Future studies should concentrate on restarts for the MINLPs, on a comparison with active-set based solvers, problem-specific cuts, tight bounds also for nonlinear subexpressions, and on more efficient techniques to find global optima.

Appendix. A.

A.1. Details of the optimization model. We list several parameters and initial values that are of relevance for the optimization problem (2.1) in Tables A.1 and A.2. Figure A.1 shows an extract of the original source code.

A.2. Derivation of the optimization model. We discuss some properties of (2.1) in more detail.

A.2.1. Integer variables and bounds. For a carefully specified optimization problem the definition of the feasible set of all control variables is crucial. Within the test scenario, for several decisions there are no bounds and it is unclear whether variables are restricted to be from a finite set or not. Although the GW-Basic code does not specifically distinguish between integer and real variables, all participants restricted themselves to integer numbers for the choices they made. Hence we decided to define some of the variables, e.g., the number of workers to be hired, as integer variables.

TABLE A.1
Fixed initial values x_0 and parameters p. 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.

State [unit]	x_k	$x_0 =$
machines 50 [machines]	$x_k^{M_{50}}$	10
shirts stock [shirts]	x_k^{ST}	80.7164
machines 100 [machines]	$x_k^{M_{100}}$	0
vans [vans]	x_k^{VA}	1
workers 50 [workers]	$x_k^{W_{50}}$	8
material stock [shirts]	x_k^{MS}	16.06787
workers 100 [workers]	$x_k^{W_{100}}$	0
machine capacity [shirts]	x_k^{MC}	47.04
demand [shirts]	x_k^{DE}	766.636
capital after interest [MU[1]]	x_k^{CA}	165774.66
Parameter [unit]	p	$p =$
max. demand [shirts]	p^{MD}	900
interest rate [—]	p^{IR}	0.0025
max. machine capacity [shirts]	p^{MM}	50
debt rate [—]	p^{DR}	0.0066
max. satisfaction [—]	p^{MS}	1.7

[1]MU means money units.

TABLE A.2
Fixed, but time-dependent, parameters p. Note that only p_k^{PR} has an implicitly given unit. The other parameters are dimensionless.

k	p_k^{PR} [MU[1]]	p_k^{DE} [—]	$p_k^{P_{50}}$ [—]	$p_k^{P_{100}}$ [—]
0	4.00000	0.616192	0.583334	0.178080
1	4.09497	0.269502	0.080131	0.365665
2	8.26718	0.692422	0.599074	0.725099
3	4.87143	0.844487	0.177331	0.207369
4	4.85305	0.697927	0.075705	0.092567
5	5.90983	0.253290	0.669259	0.318009
6	5.18731	0.805071	0.587936	0.056364
7	7.09909	0.457335	0.107187	0.543777
8	6.77216	0.889342	0.788597	0.157994
9	7.61718	0.371173	0.370508	0.746488
10	8.02385	0.029353	0.908646	0.204585
11	2.68115	0.362480	0.166743	0.303585

[1]MU means money units.

The only clearly defined integer variable within the GW-Basic code is the choice of the showroom where the shirts are being sold. There are only three choices: city center, city, and suburbs, which we identify with 2, 1, and 0, respectively. We define

$$f^1(u_k^{\text{CS}}) := \begin{cases} 1.2 & \text{if } u_k^{\text{CS}} = 2 \\ 1.1 & \text{if } u_k^{\text{CS}} = 1, \\ 1.0 & \text{if } u_k^{\text{CS}} = 0 \end{cases} \qquad f^2(u_k^{\text{CS}}) := \begin{cases} 2000 & \text{if } u_k^{\text{CS}} = 2, \\ 1000 & \text{if } u_k^{\text{CS}} = 1, \\ 500 & \text{if } u_k^{\text{CS}} = 0. \end{cases}$$

```
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
```

FIG. A.1. *Extract of the original* GW-Basic *code of the* Tailorshop *example which is the basis of the mathematical optimization problem. Special care is necessary to separate already updated variables x_{k+1} from the values x_k; compare the role of $x_k^{\mathrm{MS}} \approx RL$ and $x_k^{\mathrm{PP}} \approx PM$ in lines* 2690 *to* 2710.

To be able to relax the feasible set of u_k^{CS}, we write these functions as

$$f^1(u_k^{\mathrm{CS}}) := 1 + \frac{u_k^{\mathrm{CS}}}{10}, \qquad f^2(u_k^{\mathrm{CS}}) := 500 + 250 u_k^{\mathrm{CS}} + 250 u_k^{\mathrm{CS}} \cdot u_k^{\mathrm{CS}}.$$

For optimization algorithms the existence of tight lower and upper bounds makes a huge difference in runtime. By a process of trial and error we found several bounds that were never violated by any optimal or participant control. We define the feasible set Ω of the control variables as given by the conditions (2.18)–(2.24).

A.2.2. Reformulations. Although there are some shortcomings in the economic model and the mathematical representation including nondifferentiabilities and no tight bounds on the variables is everything but favorable for a fast and reliable solution, we had to postpone the formulation of test scenarios with better properties to future work since most of the data of the 174 participants had already been evaluated when the interdisciplinary cooperation started. Hence the main issue was to reformulate the optimization problem to be able to solve it under the constraint to keep it compatible with the available data.

Concerning nondifferentiability we strived to formulate the problem as a smooth optimization problem to allow more solvers to be able to treat the problem instances, if possible without additional binary variables.

As a first example, consider the state progression equation for the machine capacity x_k^{MC}. A direct translation of the code would read as

$$(\mathrm{A.1}) \qquad x_{k+1}^{\mathrm{MC}} = \min\left(p^{\mathrm{MM}}, 0.9 x_k^{\mathrm{MC}} + 0.017 \frac{u_k^{\mathrm{MA}}}{x_{k+1}^{M_{50}} + 10^{-8} x_{k+1}^{M_{100}}} \right).$$

What was intended here was to include a safeguard to avoid division by zero by using $x_{k+1}^{M_{50}} + x_{k+1}^{M_{100}} + 10^{-8}$ as the denominator, but the `GW-Basic` implementation used for the evaluation includes the erroneous first version. In our model we add 10^{-8} to the denominator in (A.1) to avoid division by zero but get comparable values for x_{k+1}^{MC}.

Intuitively the fact that we are dealing with a nonconvex model and that there are no bounds on the variables probably means that the problem is unbounded. Indeed, the analysis of optimization results confirmed that due to a combination of a modeling error and the unboundedness of the controls it is possible to drive the overall profit to infinity. In the equation that describes the overall demand

$$x_{k+1}^{\mathrm{DE}} = a + \left(\min\left(\frac{u_k^{\mathrm{AD}}}{5}, p^{\mathrm{MD}} \right) + 100 x_{k+1}^{\mathrm{VA}} \right) \cdot b$$

there is an upper bound on the effect of the advertisement u_k^{AD} by means of a min expression but not on the impact of vans x_k^{VA}. In other words, by buying more and more vans you can create an arbitrarily high demand. Demand itself enters into the number of shirts sold

$$x_{k+1}^{\mathrm{SS}} = \min\left(x_k^{\mathrm{ST}}, \frac{5}{4}\left(\frac{x_k^{\mathrm{DE}}}{2} + 280 \right) \cdot 2.7181^{-\frac{u_k^{\mathrm{SP}^2}}{4250}} \right).$$

Therefore you can sell an arbitrarily high number of shirts, if only you buy enough vans. However, none of the participants detected this error in the model—this only happened in a related study where participants got several repetitions. We discussed several ways to remove this unboundedness from the problem, e.g., setting a lower bound on the capital to avoid unrealistic infinite debts, possibly by fixing this lower bound to the lowest value over all data sets to keep things consistent. However, the effect of the vans was still too strong. Eventually we decided to fix the number of vans in the optimization problem to exactly that of the respective participant and to focus on the other decisions that need to be taken.

The two expressions

$$(\mathrm{A.2}) \qquad x_{k+1}^{\mathrm{SA}} = \min\left(p^{\mathrm{MS}}, \frac{1}{2} + \frac{u_k^{\mathrm{WA}} - 850}{550} + \frac{u_k^{\mathrm{SC}}}{800} \right),$$

$$(\mathrm{A.3}) \qquad x_{k+1}^{\mathrm{DE}} = \min\left(\frac{u_k^{\mathrm{AD}}}{5}, p^{\mathrm{MD}} \right)$$

can be directly replaced by

$$(\mathrm{A.4}) \qquad x_{k+1}^{\mathrm{SA}} = \frac{1}{2} + \frac{u_k^{\mathrm{WA}} - 850}{550} + \frac{u_k^{\mathrm{SC}}}{800}, \qquad \frac{1}{2} + \frac{u_k^{\mathrm{WA}} - 850}{550} + \frac{u_k^{\mathrm{SC}}}{800} \leq p^{\mathrm{MS}},$$

(A.5)
$$x_{k+1}^{\mathrm{DE}} = \frac{u_k^{\mathrm{AD}}}{5}, \qquad \frac{u_k^{\mathrm{AD}}}{5} \leq p^{\mathrm{MD}}.$$

We replace the remaining min–max expressions by introducing

(A.6)
$$s_k^{\mathrm{PP}} \approx \min(x_{k+1}^{\mathrm{PP}}, x_k^{\mathrm{MS}} + u_k^{\Delta \mathrm{MS}}),$$

(A.7)
$$s_k^{\mathrm{MC}} \approx \min\left(p^{\mathrm{MM}}, 0.9 x_k^{\mathrm{MC}} + 0.017 \frac{u_k^{\mathrm{MA}}}{x_{k+1}^{M_{50}} + 10^{-8} x_{k+1}^{M_{100}} + 10^{-8}} \right),$$

(A.8)
$$s_k^{\mathrm{SS}} \approx \min\left(x_k^{\mathrm{ST}} + x_{k+1}^{\mathrm{AP}}, \frac{5}{4}\left(\frac{x_k^{\mathrm{DE}}}{2} + 280 \right) \cdot 2.7181^{-\frac{u_k^{\mathrm{SP2}}}{4250}} \right),$$

(A.9)
$$s_k^{M_{50}} \approx \min(x_{k+1}^{W_{50}}, x_{k+1}^{M_{50}}),$$

(A.10)
$$s_k^{M_{100}} \approx \min(x_{k+1}^{W_{100}}, x_{k+1}^{M_{100}}),$$

and adding the corresponding constraints (2.27)–(2.31).

A constraint that states that new machines may only be bought when the machine capacity x_k^{MC} has at least the value of 35, or in other form

(A.11)
$$0 \leq u_k^{\Delta M_{100}} \leq \begin{cases} 0 & \text{if } x_k^{\mathrm{MC}} < 35, \\ \infty & \text{if } x_k^{\mathrm{MC}} \geq 35 \end{cases}$$

would be a little bit more tricky to reformulate in a way that is suited for a derivative-based optimization algorithm. Fortunately, due to the model bug in (A.1), x_k^{MC} will often be at its upper bound p^{MM} in optimal solutions. The model error whenever a participant should have $x_k^{\mathrm{MC}} < 35$ seems thus acceptable. Thus we simply ignore constraint (A.11).

Another issue is the interest rates, which have a constant value but a different one for positive or negative capital x_{k+1}^{BC}. This nondifferentiability in the right-hand side could be smoothed out easily by defining an appropriate function piecewise with the constant value p^{IR} for $x_{k+1}^{\mathrm{BC}} \geq \delta$, the constant value p^{DR} for $x_{k+1}^{\mathrm{BC}} \leq -\delta$, and a smoothing function for the interval $[-\delta, \delta]$, e.g., based on an arcus tangens. However, to facilitate implementation, we chose to use only the positive interest rate p^{IR}. Whenever the optimal solution does not require lending money (hence no $x_k^{\mathrm{BC}} < 0$ for any month k), obviously without loss of generality, this solution is also optimal for the case with the higher interest rate. This requires another postprocessing that we needed to automatize.

The absolute value that occurs in the right-hand side of the state x_{k+1}^{PP} can be neglected because of the lower bound of 850 for the wages u_k^{WA}.

REFERENCES

[1] 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, 2006.

[2] J. Albersmeyer and M. Diehl, *The lifted Newton method and its application in optimization*, SIAM J. Optim., 20 (2010), pp. 1655–1684.

[3] D. Applegate, R. E. 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 (2009), pp. 11–15.

[4] 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.

[5] C. M. Barth and J. Funke, *Negative effective environments improve complex solving performance*, Cognition and Emotion, 24 (2010), pp. 1259–1268.

[6] P. Belotti, *Couenne: A User's Manual*, Technical report, Lehigh University, 2009; also available online from   https://projects.coin-or.org/Couenne/browser/trunk/Couenne/doc/couenne-user-manual.pdf?format=raw.

[7] L. T. Biegler, *An overview of simultaneous strategies for dynamic optimization*, Chemical Engrg. and Processing, 46 (2007), pp. 1043–1053.

[8] T. Binder, L. Blank, H. G. Bock, R. Bulirsch, W. Dahmen, M. Diehl, T. Kronseder, W. Marquardt, J. P. Schlöder, and O. V. Stryk, *Introduction to model based optimization of chemical processes on moving horizons*, in Online Optimization of Large Scale Systems: State of the Art, M. Grötschel, S. O. Krumke, and J. Rambau, eds., Springer, 2001, pp. 295–340.

[9] R. E. 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, Philadelphia, 2004, pp. 309–325.

[10] 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 Optim., 5 (2009), pp. 186–204.

[11] P. Bonami, M. Kilinc, and J. Linderoth, *Algorithms and software for convex mixed integer nonlinear programs*, IMA Volumes, to appear.

[12] B. Brehmer, *Feedback delays in dynamic decision making*, Complex Problem Solving: The European Perspective, Lawrence Erlbaum Associates, 1995, pp. 103–130.

[13] M. R. Bussieck, *GAMS Performance World*, http://www.gamsworld.org/performance.

[14] GAMS Development Corporation, GAMS homepage. http://www.gams.com/.

[15] 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 (2011) (in press).

[16] D. Dörner, *On the difficulties people have in dealing with complexity*, Simulation and Games, 11 (1980), pp. 87–106.

[17] P. H. Ewert and J. F. Lambert, *Part* II*: The effect of verbal instructions upon the formation of a concept*, J. of General Psychology, 6 (1932), pp. 400–411.

[18] C. A. Floudas, I. G. Akrotirianakis, S. Caratzoulas, C. A. Meyer, and J. Kallrath, *Global optimization in the* 21*st century: Advances and challenges*, Comput. Chem. Eng., 29 (2005), pp. 1185–1202.

[19] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*, Duxbury Press, 2002.

[20] P. A. Frensch and J. Funke, eds., *Complex Problem Solving: The European Perspective*, Lawrence Erlbaum Associates, 1995.

[21] J. Funke, *Einige Bemerkungen zu Problemen der Problemlöseforschung oder: Ist Testintelligenz doch ein Prädiktor?*, Diagnostica, 29 (1983), pp. 283–302.

[22] J. Funke, *Using simulation to study complex problem solving: A review of studies in the FRG*, Simulation and Games, 19 (1988), pp. 277–303.

[23] J. Funke, *Problemlösendes Denken*, Kohlhammer, 2003.

[24] J. Funke, *Complex problem solving: A case for complex cognition?*, Cognitive Processing, 11 (2010), pp. 133–142.

[25] J. Funke and P. A. Frensch, *Complex problem solving: The European perspective – 10 years after*, Learning to Solve Complex Scientific Problems, Lawrence Erlbaum Associates, 2007, pp. 25–47.

[26] C. Gonzalez, *Learning to make decisions in dynamic environments: Effects of time constraints and cognitive abilities*, Human Factors, 46 (2004), pp. 449–460.

[27] C. Gonzalez, P. Vanyukov, and M. K. Martin, *The use of microworlds to study dynamic decision making*, Computers in Human Behavior, 21 (2005), pp. 273–286.

[28] A. Grössler, F. H. Maier, and P. M. Milling, *Enhancing learning capabilities by providing transparency in business simulators*, Simulation & Gaming, 31 (2000), pp. 257–278.

[29] I. E. Grossmann, *Review of nonlinear mixed-integer and disjunctive programming techniques*, Optim. Eng., 3 (2002), pp. 227–252.

[30] H. J. Hörmann and M. Thomas, *Zum Zusammenhang zwischen Intelligenz und komplexem Problemlösen*, Sprache & Kognition, 8 (1989), pp. 23–31.

[31] W. Hussy, *Komplexes Problemlösen—Eine Sackgasse?*, Zeitschrift für Experimentelle und Angewandte Psychologie, 32 (1985), pp. 55–74.

[32] W. Hussy, *Komplexes Problemlösen und Verarbeitungskapazität*, Sprache & Kognition, 10 (1991), pp. 208–220.

[33] M. Kleinmann and B. Strauss, *Validity and applications of computer simulated scenarios in personal assessment*, International J. of Selection and Assessment, 6 (1998), pp. 97–106.

[34] R. H. Kluwe, *Knowledge and performance in complex problem solving*, The Cognitive Psychology of Knowledge, Elsevier Science Publishers, 1993, pp. 401–423.

[35] Z. H. Kluwe, C. Misiak, and H. Haider, *Systems and performance in intelligence tests*, Intelligence: Reconceptualization and Measurement, Lawrence Erlbaum Associates, 1991, pp. 227–244.

[36] S. Kolb, F. Petzing, and S. Stumpf, *Komplexes Problemlösen: Bestimmung der Problemlösegüte von Probanden mittels Verfahren des Operations Research—ein interdisziplinärer Ansatz*, Sprache & Kognition, 11 (1992), pp. 115–128.

[37] B. Meyer and W. Scholl, *Complex problem solving after unstructured discussion. Effects of information distribution and experience*, Group Process and Intergroup Relations, 12 (2009), pp. 495–515.

[38] J. A. Nelder and R. Mead, *A simplex method for function minimization*, Computer Journal, 7 (1965), pp. 308–313.

[39] M. Osman, *Observation can be as effective as action in problem solving*, Cognitive Science: A Multidisciplinary Journal, 32 (2008), pp. 162–183.

[40] J. H. Otto and E.-D. Lantermann, *Wahrgenommene Beeinflussbarkeit von negativen Emotionen, Stimmung und komplexes Problemlösen*, Zeitschrift für Differentielle und Diagnostische Psychologie, 25 (2004), pp. 31–46.

[41] W. Putz-Osterloh, *Über die Beziehung zwischen Testintelligenz und Problemlöseerfolg*, Zeitschrift für Psychologie, 189 (1981), pp. 79–100.

[42] 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 (1990), pp. 83–96.

[43] 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*, Mem. Cognit., 24 (1996), pp. 83–93.

[44] S. Sager, *Reformulations and algorithms for the optimization of switching decisions in nonlinear optimal control*, J. of Process Control, 19 (2009), pp. 1238–1247.

[45] S. Sager, H. Diedam, and M. Engelhart, *Tailorshop: Optimization Based Analysis and Data Generation Tool*, TOBAGO web site https://sourceforge.net/projects/tobago.

[46] S. Strohschneider and D. Güss, *The fate of the moros: A cross-cultural exploration of strategies in complex and dynamic decision making*, International J. of Psychology, 34 (1999), pp. 235–252.

[47] H.-M. Süss, K. Oberauer, and M. Kersting, *Intellektuelle Fähigkeiten und die Steuerung komplexer Systeme*, Sprache & Kognition, 12 (1993), pp. 83–97.

[48] M. Tawarmalani and N. Sahinidis, *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002.

[49] M. Tawarmalani and N. V. Sahinidis, *A polyhedral branch-and-cut approach to global optimization*, Math. Program., 103 (2005), pp. 225–249.

[50] A. Wächter and L. T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Math. Program., 106 (2006), pp. 25–57.

[51] D. Wagener, *Personalauswahl und -Entwicklung mit komplexen Szenarios*, Wirtschaftspsychologie, 3 (2001), pp. 69–76.

[52] 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, Cambridge, U.K., 2003, pp. 87–126.

[53] W. W. Wittmann and K. Hattrup, *The relationship between performance in dynamic systems and intelligence*, Systems Research and Behavioral Science, 21 (2004), pp. 393–409.